UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA DE INGENIERÍA ELÉCTRICA

DISEÑO Y CONSTRUCCIÓN DE SISTEMA DE MEDICIÓN DE ENERGÍA ELÉCTRICA PARA APLICACIONES EN BANCOS DE PRUEBA DE TRANSFORMADORES PARA USO EN EL LABORATORIO DE POTENCIA DE LA ESCUELA DE INGENIERÍA ELÉCTRICA, FIA-UES

PRESENTADO POR:

JORGE EDUARDO MÉNDEZ FLORES

ENRIQUE DOLORES ROSALES ORTIZ

PARA OPTAR AL TITULO DE:

INGENIERO ELECTRICISTA

CIUDAD UNIVERSITARIA, FEBRERO 2020.

UNIVERSIDAD DE EL SALVADOR

RECTOR :

MSC. ROGER ARMANDO ARIAS ALVARADO

SECRETARIA GENERAL :

ING. FRANCISCO ANTONIO ALARCON SANDOVAL

FACULTAD DE INGENIERIA Y ARQUITECTURA

DECANO :

DOCTOR EDGAR ARMANDO PEÑA FIGUEROA

SECRETARIO :

ING. JULIO ALBERTO PORTILLO

ESCUELA DE INGENIERIA ELÉCTRICA

DIRECTOR :

ING. ARMANDO MARTÍNEZ CALDERÓN

UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

INGENIERO ELECTRICISTA

Título :

DISEÑO Y CONSTRUCCIÓN DE SISTEMA DE MEDICIÓN DE ENERGÍA ELÉCTRICA PARA APLICACIONES EN BANCOS DE PRUEBA DE TRANSFORMADORES PARA USO EN EL LABORATORIO DE POTENCIA DE LA ESCUELA DE INGENIERÍA ELÉCTRICA, FIA-UES

Presentado por :

JORGE EDUARDO MÉNDEZ FLORES

ENRIQUE DOLORES ROSALES ORTIZ

Trabajo de Graduación Aprobado por:

Docente Asesor :

MSC. LUIS ROBERTO CHÉVEZ PAZ

SAN SALVADOR, FEBRERO 2020.

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSC. LUIS ROBERTO CHÉVEZ PAZ

NOTA Y DEFENSA FINAL

En esta fecha, lunes 3 de febrero 2020, en la Sala de Lectura de la Escuela de Ingeniería Eléctrica, a las 5:00 p.m. horas, en presencia de las siguientes autoridades de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

1. Ing. Armando Martínez Calderón
   Director

   Firma

2. MSc. José Wilber Calderón Urrutia
   Secretario

   Firma

Y, con el Honorable Jurado de Evaluación integrado por las personas siguientes:

- MSC. LUIS ROBERTO CHEVEZ PAZ
  (Docente Asesor)

  Firma

- ING. ARMANDO MARTINEZ CALDERÓN

  Firma

- MSc. HUGO MIGUEL COLATO RODRIGUEZ

  Firma

Se efectuó la defensa final reglamentaria del Trabajo de Graduación:

DISEÑO Y CONSTRUCCIÓN DE SISTEMA DE MEDICIÓN DE ENERGÍA ELÉCTRICA PARA APLICACIONES EN BANCOS DE PRUEBA DE TRANSFORMADORES PARA USO EN EL LABORATORIO DE POTENCIA DE LA ESCUELA DE INGENIERÍA ELÉCTRICA, FIA-UES

A cargo de los Bachilleres:

- MÉNDEZ FLORES JORGE EDUARDO

- ROSALES ORTIZ ENRIQUE DOLORES

Habiendo obtenido en el presente Trabajo una nota promedio de la defensa final: 8.9

(Ocho punto nueve)

AGRADECIMIENTOS.

Le agradezco a Dios por permitirme terminar mi carrera universitaria por mantenerme con buena salud a lo largo de mis estudios y por darme la sabiduría necesaria para concluir esta etapa de la mejor manera.

A mi mama Mercedes Ortiz por el apoyo que me han brindado a lo largo de mis estudios universitarios ya que ellos han sido los que me han inspirado a poner el mayor de mis esfuerzos para completar de la mejor manera mis estudios estaré eternamente agradecido.

También le agradezco a mi esposa Iris Palacios por ayudarme a lo largo de la carrera y durante el proceso de elaboración de la tesis, me ha ayudado a superar cada reto que hemos tenido, por tenerme mucha paciencia y por siempre creer en mí.

También le agradezco a mis hermanas Jessy y Zuleyma por ser parte importante de mi vida y que con sus consejos me han ayudado a afrontar los retos que se me han presentado.

Le agradezco especialmente a nuestro asesor de tesis Msc. Luis Roberto Chevez Paz por el gran apoyo que nos ha brindado en cada etapa de nuestro trabajo de graduación, por aportar sus conocimientos para orientarnos de la mejor manera y por darnos los consejos necesarios para llegar a nuestra meta de la mejor manera posible.

También agradezco mi compañero de tesis Jorge Méndez, porque cada una con sus valiosas aportaciones hicieron posible este proyecto de tesis y así haber logrado culminar nuestro gran objetivo con perseverancia y dedicación.

Finalmente, gracias a todas las personas que de manera directa o indirecta nos brindaron su apoyo en la realización de este proyecto.

Enrique Dolores Rosales Ortiz.

AGRADECIMIENTOS.

Le agradezco a Dios en primer lugar por darme la sabiduría, deseo, fortaleza y la perseverancia para finalizar mi carrera universitaria, superando así las dificultades que a lo largo de mi vida se presentaron.

Dedico este logro a mi familia por todo los expresado hacia mí, especialmente a mis padres Jorge Alberto Méndez y Marta Sonia Flores por todos los consejos, paciencia y la motivación a alcanzar mi meta que siempre me brindaron, por su apoyo a lo largo de mi vida. A mi abuela Cleotilde Sánchez, por todo su cariño, y palabras de aliento que fueron siempre lo que necesité para seguir con mis estudios. A mis hijos Jorge Méndez y Jonathan Méndez que han sido gran parte de mi inspiración y motivo de lucha en esta etapa que hoy culmino.

Un agradecimiento especial a nuestro asesor de tesis Msc. e Ing. Luis Roberto Chevez Paz y por su aporte en cada etapa de nuestro trabajo de graduación, y darnos así los consejos necesarios para finalizar este trabajo de la mejor manera posible.

También agradezco mi compañero de tesis Enrique Ortiz, por la perseverancia para superar todas las adversidades que tuvimos en el desarrollo de esta etapa y dedicación para finalizar con éxito nuestro trabajo final.

Finalmente, gracias a todas las personas que de manera directa o indirecta me brindaron su apoyo en mi formación académica y profesional.

Jorge Eduardo Méndez Flores.

TABLA DE CONTENIDO.

LISTA DE FIGURAS.

LISTA DE TABLAS.

INTRODUCCION.

En lo formación profesional de los estudiantes de la carrera de Ingeniería Eléctrica, es de gran relevancia contar con equipos de instrumentación para la medición de variables eléctricas, como, por ejemplo: potencia, corriente, voltaje, etc. Los cuales permiten corroborar de manera práctica los conocimientos desarrollados en las exposiciones teóricas de las asignaturas relacionadas a los sistemas eléctricos de potencia.

Es por ello por lo que surge la necesidad de utilizar equipos para la medición de energía eléctrica, en bancos de pruebas de transformadores ubicados en el edificio de Potencia de la Universidad de El Salvador.

El objetivo principal del presente documento es la elaboración de un medidor de variables eléctricas basado en el dispositivo de hardware libre muy popular en la fecha denominado proyecto Arduino, el cual consiste en una placa electrónica que contiene un microcontrolador de placa simple, el cual puede ser utilizado para una diversidad de creativas soluciones.

El hecho que conlleva a la elección del uso del dispositivo Arduino, es la ventaja de que este cuenta con un estándar de comunicaciones denominado SPI[1], lo cual nos genera el punto de partida inicial para optar a la decisión de utilizar dicho dispositivo, ya que existen muchos circuitos integrados de propósito específico que utilizan dicho estándar de comunicación, además de la versatilidad para adquirir de una opción entre diversas pantallas LCD[2] para la presentación de nuestros datos lo hace aún más atractivo para nuestro interés, o para nuestra caso la fácil adquisición de datos utilizando su software libre , el cual está disponible tanto para plataforma Linux como Windows.

El desarrollo de este tipo de hardware libre; ha hecho hincapié en la comunidad tecnológica, llegando a desarrollar diversidad de sensores lo cual está creando un gran auge en el área de sistemas de automatización y control, así como también en la adquisición de datos del cual nos será de gran utilidad en el desarrollo de nuestro equipo para medición de energía.

[1] Serial Peripheral Interface
[2] Liquid Cristal Display.

OBJETIVOS.

Objetivo General:

Construir un sistema para la medición de energía eléctrica de múltiples bancos de transformadores, utilizando hardware y software libre para reducir los costes de desarrollo.

Objetivo específico:

1. Elaborar circuitos electrónicos que nos brinde soporte para el uso de transformadores de núcleo abierto y divisores de voltaje o la fabricación de bobinas Rogowski, que nos facilite los requerimientos solicitados en las especificaciones técnicas de entrada de señales para el dispositivo ADE7758.

2. Implementar con el uso de circuitos integrados medidores de energía, para el cual utilizaremos el ADE7758 [3] del fabricante Analog Devices[4].

3. Obtener con el uso de **Arduino** la adquisición de datos del circuito integrado medidor de energía ADE7758, utilizando el estándar de comunicación SPI.

4. Elegir el dispositivo de visualización y el entorno gráfico para la presentación de datos acorde a nuestro requerimiento de desarrollo, Raspbian será nuestro sistema operativo y el diseño de la interfaz gráfica de usuario se efectuará utilizando la librería TkInter del interprete Python.

5. Desarrollar un sistema de multiplexado que nos ayude a disminuir cantidad unitaria el uso del circuito integrado ADE7758, el cual nos permitirá agregar 6 mediciones trifásicas utilizando un dispositivo.

[3] El ADE7758 es un CI de medición de energía eléctrica trifásica de alta precisión con una interfaz en serie y dos salidas de pulso.

[4] Analog Devices es una multinacional estadounidense productora de dispositivos semiconductores. Analog Devices es especialista en ADC y DAC, MEMS y DSP.

1 MARCO TEÓRICO.

1.1 MEDICIÓN DE POTENCIA ELÉCTRICA.

La medición de la potencia eléctrica es la esencia del presente documento, para iniciarnos en el recorrido de la elaboración de un dispositivo para medir potencia eléctrica, es importante tener claros los conceptos de potencia en los sistemas de corriente alterna, así como también cabe resaltar que los comportamientos de las cargas conectadas a nuestra red eléctrica no interactúan de la misma manera, ya que está restringido a la naturaleza de la carga las cuales se mencionan a continuación:

- Carga Resistiva
- Cargas Reactivas
- Cargas No Lineales.

1.1.1 CARGAS RESISITIVAS

Las cargas resistivas utilizan toda la energía brindada por la red eléctrica, su consumo de corriente es igual al voltaje divido por su resistencia (Ley de Ohm), estos tipos de carga son aquellas que producen calor, además, la corriente y el voltaje de suministro no presentas desfase al ser medidos en la carga.
Ejemplos de cargas resistivas son las siguientes: Bombillos incandescentes, calentadores de agua, cocinas eléctricas, etc.
Las formas de ondas de voltaje, corriente y potencia de una carga puramente resistiva la podemos apreciar en la Figura 1.

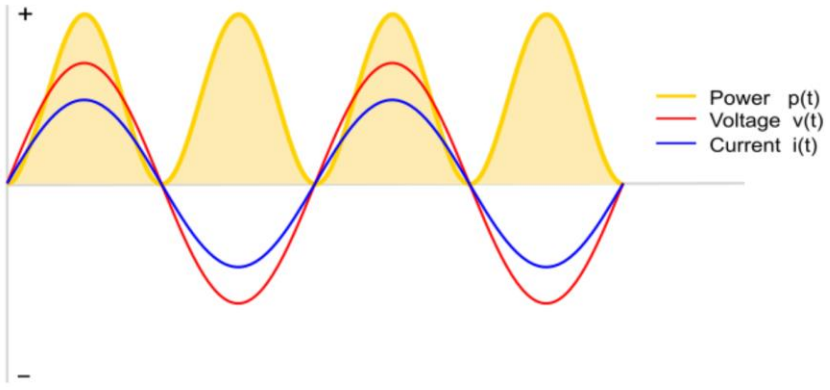


Figura 1. Formas de ondas de voltaje, corriente, potencia para una carga resistiva en una red de corriente alterna.

Podemos observar de la Figura 1, la forma de onda en color amarillo la cual describe la potencia en un instante dado, generalmente esta potencia se conoce como **potencia instantánea**, la cual en una carga resistiva es igual al producto de el voltaje y la corriente en tiempo determinado. Se verifica a partir de la forma de onda de potencia que es positiva en cualquier instante de tiempo por lo que podemos determinar que el flujo de energía es de la fuente a la carga.

1.1.2 CARGAS PARCIALMENTE REACTIVAS.

Existen además cargas que contienen como impedancia parte resistiva y parte reactiva, la cual es el tipo más común en nuestro medio como, por ejemplo: refrigeradores, lavadoras, taladros, compresores de aire, etc., estos aparatos consumen una cierta cantidad de energía, luego liberan parte de dicha energía en la red. Estos tienen componentes inductivos (por ejemplo, motores) o capacitivos (por ejemplo, soldadores de arco[5]) además del componente resistivo. Una carga parcialmente inductiva proporciona una salida de forma de onda de voltaje y corriente similar a la que se muestra en la Figura 2.

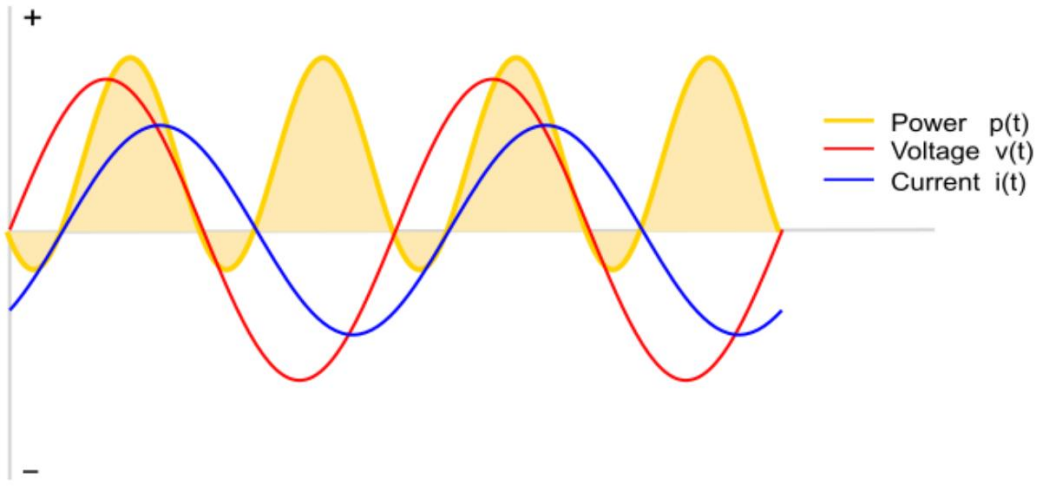


Figura 2. Formas de ondas de voltaje, corriente, potencia para una carga parcialmente reactiva.

Es importante notar que en la Figura 2 la forma de onda de potencia instantánea representada por la curva color amarillo, tiene una región negativa en un período de

[5] A veces llamada soldadura electrógena

tiempo, esto nos indica que hay una parte de energía que fluye de la carga a la fuente de energía, además podemos observar que la curva también tiene una región positiva en un periodo de tiempo, y nos indica que la energía fluye a la carga para su consumo.

También podemos observar que las formas de onda de corriente y voltaje se han desplazado (las formas de ondas no se encuentran en fase como en el caso de una carga resistiva).

Consideremos la carga un condensador de una alta capacitancia (μF) con una resistencia en serie (esto con el objetivo que el capacitor no pueda cargarse instantáneamente): para empezar, el condensador se descarga. El voltaje de suministro aumenta, y es más alto que el voltaje en el condensador, por lo que la corriente fluye al condensador (la dirección positiva en el gráfico), lo que hace que el voltaje del capacitor se eleve. El voltaje de suministro cae. Ahora, el voltaje a través del condensador cargado es más alto que el voltaje de suministro. La corriente comienza a fluir hacia atrás en la dirección del suministro (la dirección negativa en el gráfico). Esto hace que la forma de onda actual parezca desplazada, como se muestra en el gráfico. (Esto se conoce como cambio de fase).

1.1.3 POTENCIA ELÉCTRICA.

Un punto importante para comprender sobre la medición de consumo de energía es el de conocer cómo se clasifica la potencia en los sistemas eléctricos, esta describe a continuación.

- **Potencia instantánea.**

La potencia instantánea absorbida por una carga es el producto de el voltaje instantáneo *v(t)* en las terminales de la carga y la corriente instantánea *i(t)* a través de la carga esta se mide en **watts**, básicamente la potencia instantánea es la potencia absorbida por la carga en un instante de tiempo específico, de la Figura 1 y Figura 2 la potencia instantánea está representada por un punto de la curva de potencia (marcada en color amarillo).

$\boldsymbol{p(t) = v(t)i(t)}$**,** La potencia instancia es la potencia en cualquier instante de tiempo.

- **Potencia Real.**

Si observamos los gráficos de voltaje, corriente y potencia de las cargas presentadas, tenemos que, en la frecuencia de la red, el consumo de energía fluctúa 50/60 [6] veces por segundo. No podemos seguir el ritmo del cambio a esta velocidad, por lo que tenemos un valor más útil para la potencia y este es el promedio de la potencia instantánea, que

[6] En El Salvador, es a 60 Hz.

llamamos **potencia real o activa,** esta también se mide en **watts**. La potencia real se define como la potencia utilizada por un dispositivo para producir un trabajo útil.

- **Potencia reactiva.**

La potencia reactiva es una medida de la potencia que se suministra y retorna entre la carga y la fuente la cual no sirve para nada.

La potencia reactiva (y la energía reactiva) no es una potencia (energía) realmente consumida en la instalación, ya que no produce trabajo útil debido a que su valor medio es nulo. Aparece en una instalación eléctrica en la que existen bobinas y/o capacitores, y es necesaria para crear campos magnéticos y eléctricos en dichos componentes. Se representa por Q y se mide en voltiamperios reactivos (VAr).

La compañía eléctrica mide la energía reactiva con el contador (kVArh) y si se superan ciertos valores, incluye un término de penalización por reactiva en la factura eléctrica.

1.2 SENSORES DE CORRIENTE.

Son 4 los tipos de sensores de corriente. Estos se mencionan a continuación y se detallan en las siguientes secciones:

1. Transformador de corriente
2. Bobina Rogowski
3. Shunt de corriente de baja resistencia
4. Sensor de efecto Hall.

1.2.1 TRANSFORMADOR DE CORRIENTE.

Un transformador es un dispositivo electromagnético, que permite disminuir o aumentar la tensión de un circuito eléctrico, o pasar de corriente alterna a corriente continua Los transformadores son dispositivos basados en el fenómeno de la inducción electromagnética y están constituidos, en su forma más simple, por dos bobinas devanadas sobre un núcleo cerrado. En un transformador ideal, la potencia que ingresa al equipo es igual a la de salida. Pero en la realidad, siempre existirá una pequeña pérdida, dependiendo del tamaño o diseño, etc.

Entre 1884 y 1885, los ingenieros húngaros Zipernowsky, Bláthy y Deri de la compañía Ganz crearon en Budapest el modelo "ZBD" de transformador de corriente alterna, basado en un diseño de Gaulard y Gibbs (Gaulard y Gibbs sólo diseñaron un modelo de núcleo abierto). Descubrieron la fórmula matemática de los transformadores.[7]

Los transformadores de corriente cumplen un rol fundamental en la vida cotidiana de las personas en general y de las empresas que hacen de la utilización de estos dispositivos una parte fundamental del funcionamiento de la amplia gama de industrias que existen en nuestro país y en el mundo.

1.2.1.1 CARACTERÍSTICAS.

Los transformadores tienen la capacidad de transformar el voltaje y la corriente a niveles más altos o bajos. Está construido superponiendo numerosas chapas de aleación acero–silicio, con el fin de reducir las pérdidas por histéresis magnética y aumentar la resistividad del acero.

[7] https://es.wikipedia.org/wiki/Transformador

No crean la energía a partir de la nada; por lo tanto, si un transformador aumenta el voltaje de una señal, reduce su corriente; y si reduce el voltaje de la señal, eleva la corriente. En otras palabras, la energía que fluye a través de un transformador no puede ser superior a la energía que haya entrado en él.

1.2.1.2 APLICACIONES.

Para transportar corriente, los transformadores permiten llevar tensiones elevadas, con baja intensidad para minimizar la pérdida de potencia ocasionada por la resistencia de los conductores. Para circuitos electrónicos como los de radio, podemos utilizar transformadores de impedancia.

En el uso cotidiano, la mayoría de los artefactos eléctricos que utilizamos, requieren de un *transformador* para ajustar la corriente de la red a los requerimientos de funcionamiento. Por ejemplo, los monitores de las computadoras, los aparatos reproductores de CD, los teléfonos celulares, etc.

1.2.1.3 COMPONENTES.

Si bien es cierta la cantidad de transformadores de corriente disponibles hoy en día en el mercado son de una gran variedad y aplicaciones, todos presentan una columna vertebral que es invariante. Puede que cambien en su forma de construcción del transformador en sí, pero todos cuentan con un núcleo magnético elaborados preferentemente con una aleación de hierro – silicio (hierro preferentemente), el cual constituye el circuito magnético cuya función principal es la de conducir el flujo magnético. Luego nos vamos a encontrar con bobinados, los cuales están ensamblados alrededor del núcleo magnético y sujetado a una estructura de soporte. Se encuentran dos bobinas, la primaria la cual tiene por función crear un campo magnético con una pérdida de energía muy pequeña y una bobina secundaria que aprovecha el flujo magnético para producir una fuerza electromotriz. Estas bobinas pueden ser de alambre delgado, grueso o barra y los materiales comúnmente utilizados son el cobre y aluminio. Un punto importante de mencionar es la razón de transformación entre la bobina primaria y secundaria, que depende del número de vueltas que tenga cada uno, por lo que puede ser elevador o reductor dependiendo del número de espiras de cada bobina. Todo esto se instala en una caja la cual puede estar fabricada en aluminio para la corrosión o en plástico para el aislamiento.

Figura 3. Transformadores de corriente.

1.2.2 BOBINA ROGOWSKI.

La bobina Rogowski, llamada así en honor a su inventor Walter Rogowski, es un dispositivo electrónico, usado como transductor para medir corriente alterna (AC) o pulsos rápidos de corriente. Esta consiste en una bobina uniformemente arrollada en un núcleo de material no magnético de sección transversal constante, distribuido en forma de lazo cerrado. La forma más simple es la de un toroide circular cerrado y rígido, o abierto y flexible para que pueda cerrarse sobre sí mismo y así facilitar su montaje alrededor de un conductor por el que circula la corriente a evaluar. La función de esta bobina es aplicar la Ley de Ampere, la cual dice que la corriente que circula por un conductor es proporcional a la integral de circulación de la intensidad de campo magnético alrededor de un camino cerrado que rodea a dicho conductor.

La idea anterior puede expresarse también diciendo que la bobina Rogowski, cuyo principio de funcionamiento se conoce desde 1912, se basa en medir los cambios del campo magnético que se produce alrededor de un hilo portador de corriente para producir una señal de voltaje, la cual es proporcional a la derivada de la corriente (di/dt), para lo que se necesita un integrador que convierta apropiadamente la señal.

La tarea de crear un integrador que fuera estable y exacto durante la larga vida de un medidor había desalentado a los estudiosos de este rubro. Sin embargo, la reciente implementación digital del integrador tiene la promesa de convertir esta tecnología en una realidad para los medidores eléctricos (Ver Figura 4).

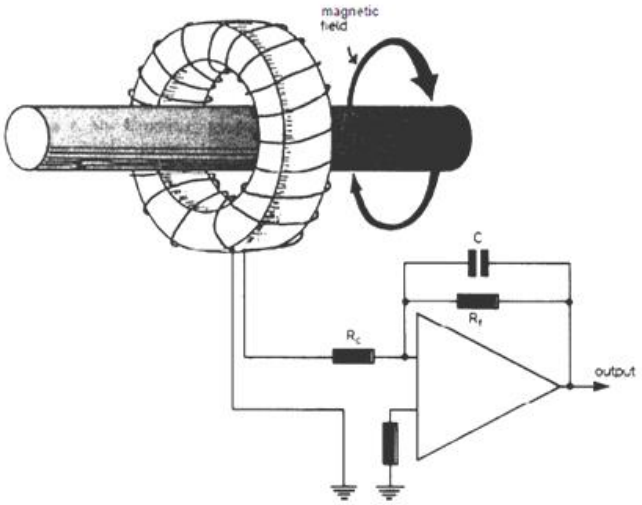


Figura 4. Esquema de un Bobina Rogowski con integrador.

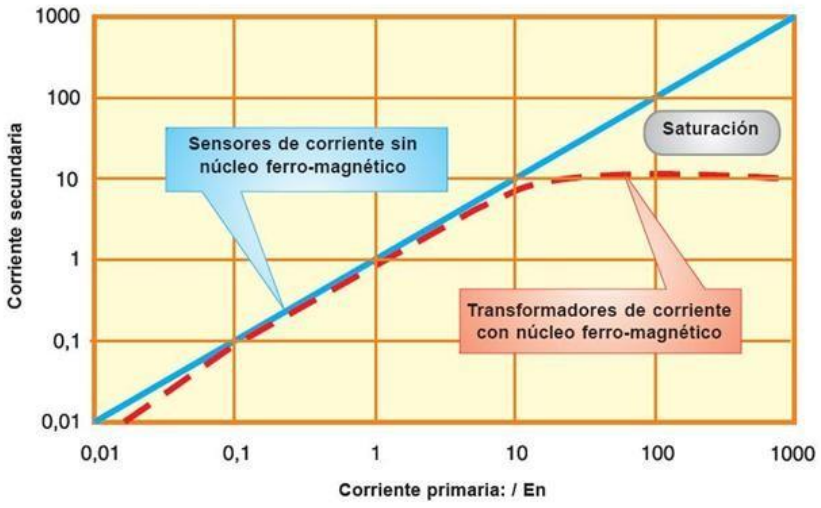


Figura 5. Comportamiento de linealidad, efecto Rogowski.

Figura 6. Bobina Rogowski Comercial, para medidores de calidad de energía Dranetz Explorer.

Dentro de las grandes ventajas de utilizar la bobina Rogowski es que al poseer núcleo de aire no presenta histéresis, saturación o problemas de no linealidad (ver Figura 5). Además, posee una elevada capacidad para manejar altas corrientes, siendo el límite superior teórico de la bobina el voltaje de ruptura del mismo aire. Por último, se destaca que posee un bajo costo con respecto a otras tecnologías de sensores de corriente.

1.2.3 SHUNT DE CORRIENTE DE BAJA RESISTENCIA.

De los distintos tipos de sensores, los Shunt de corriente son los de menor valor, además de ofrecer una lectura sencilla y con una buena precisión. Este tipo de sensor es la solución más utilizada (popular) para la medición de corriente.

Cuando se realizan las mediciones de corriente con alta precisión, es necesario tener en cuenta la inductancia del Shunt (relación entre el flujo magnético y la intensidad de corriente eléctrica, generalmente positiva) y aunque sólo a frecuencias relativamente altas afecta la magnitud de la impedancia (medida que establece la relación (cociente) entre la tensión y la intensidad de corriente), ya que su efecto es suficiente para provocar un error a un bajo factor de potencia. Por lo tanto, es importante seleccionar una resistencia “Shunt” apropiada de sensor de corriente que debe tener un valor muy bajo

de resistencia para minimizar la disipación de potencia, un valor bajo de inductancia y una tolerancia razonablemente pequeña para mantener una precisión global en el circuito.

Por ejemplo, los Shunt que se utilizan en instrumentos portátiles o de laboratorio están preparados para una caída de tensión de 60 mV. Para instrumentos de tablero se emplean los Shunt con caídas normalizadas de: 30, 45, 60, 100, 120, 150, 300 mV. Los Shunt se clasifican, según su exactitud, en cinco clases: 0.05, 0.1, 0.2, 0.5 y 1%. Cuando el instrumento (mili voltímetro) se conecta con el Shunt mediante cables, su calibración se efectúa en conjunto con los cables. Los métodos de medición se dividen en cuatro grupos principales: métodos voltiamperimétricos (técnicos), métodos de cero (puentes), métodos de deflexión y métodos de compensación. Cada uno de estos métodos tiene su campo de aplicación que se rige por la precisión requerida, por el alcance de la magnitud medida y por la disponibilidad del equipo.

Aunque es posible adquirir resistencias "Shunt" de sensor de corriente a fabricantes como: IRC, Dale, Ultronix, Isotek, y K-tronics, que son sólo algunos de los proveedores que fabrican resistencias apropiadas para aplicaciones de sensor de corriente, es también posible hacer una resistencia sensor utilizando diversos materiales interesantes como el cobre o la manganina.

En la Figura 6 se muestran varios tipos de Shunt, que por lo general poseen cuatro bornes, que es el nombre dado en electricidad a cada uno de los terminales de metal en que suelen terminar algunas máquinas y aparatos eléctricos, y que se emplean para su conexión a los hilos conductores, como también lo muestra la figura 5, que presenta una resistencia Shunt sin su cuerpo protector. Esta geometría permite evitar errores causados por la resistencia de contactos.

Figura 7. Distintos tipos de Shunt.

1.2.4 EL SENSOR DE EFECTO HALL.

Para ilustrarse este tipo de sensores es necesario conocer primero el Efecto Hall, el cual es una consecuencia de la fuerza que se ejerce sobre una carga eléctrica en movimiento cuando se encuentra sometida a la acción de un campo eléctrico y un campo magnético. En 1879 el físico E. Hall descubrió que cuando un conductor sobre el que circulaba corriente era colocado en un campo magnético de dirección perpendicular a la misma, podía medirse una pequeña diferencia de potencial en la dirección perpendicular a la corriente y al campo.

El sensor de efecto Hall sirve para la medición de campos magnéticos o corrientes, o para la determinación de la posición.

Su función sigue unos pasos en donde si fluye corriente por un sensor Hall y se aproxima a un campo magnético que fluye en dirección vertical al sensor, entonces el sensor crea un voltaje saliente proporcional al producto de la fuerza del campo magnético y de la corriente. Si se conoce el valor de la corriente, entonces se puede calcular la fuerza del campo magnético; si se crea el campo magnético por medio de corriente que circula por una bobina o un conductor, entonces se puede medir el valor de la corriente en el conductor o bobina.

Si tanto la fuerza del campo magnético como la corriente son conocidas, entonces se puede usar el sensor Hall como detector de metales.

Existen dos tipos principales de sensores de Efecto Hall, anillo abierto (open-loor) y anillo cerrado (closed-loop). El segundo ofrece mejor precisión y rangos dinámicos más amplios, pero a un costo mayor, y la mayoría de los sensores de Efecto Hall que se encuentran en medidores de energía usan el diseño anillo abierto para lograr costos más bajos. El sensor de Efecto Hall tiene una excelente respuesta a la frecuencia y está capacitado para medir corrientes muy altas.

La señal obtenida del sensor Hall puede ser procesada para dar una señal digital o analógica. De modo que cuando desea obtener una salida los sensores se denominan interruptores Hall. Y cuando se requiere que la salida sea proporcional a la señal que se desea medir, se denominan sensores Hall de tipo lineal. Estos últimos son los empleados para la medida de corrientes y cubren un rango que se extiende desde pocos mA hasta cientos de mA En la Figura 8 se muestran algunos sensores de corriente que emplean el efecto Hall.

Algunas aplicaciones para los sensores Hall los podemos encontrar en la industria del automóvil el sensor Hall se utiliza de forma frecuente, ej. en sensores de posición del cigüeñal (CKP) en el cierre del cinturón de seguridad, en sistemas de cierres de puertas, para el reconocimiento de posición del pedal o del asiento, etc.

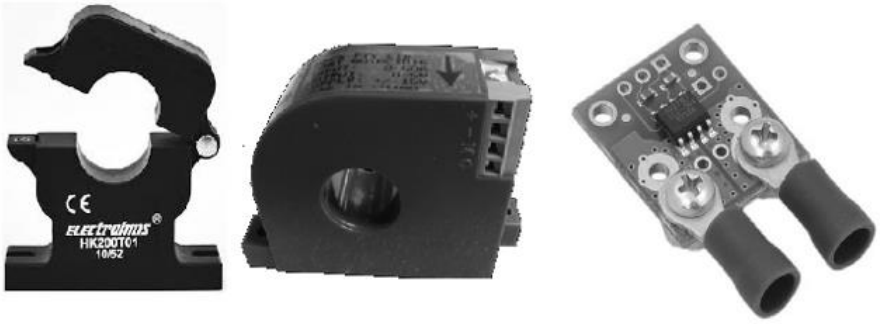


Figura 8. Algunos tipos de sensores de efecto Hall.

1.2.5 COMPARACIÓN DE LOS SENSORES.

A continuación, se muestran dos tablas comparativas el cual resume las características y los efectos para los cuatro tipos de sensores. Posteriormente una tabla cualitativa el cual resume como está el sensor en comparación con el resto de acuerdo con las distintas tecnologías utilizadas para cada uno.

Sensor de Corriente.	Características	Efectos
Shunt.	De 100 a 500μΩ. Bajo costo. Inmune al problema de la saturación DC. No está aislado. Buena linealidad. No útiles para grandes corrientes.	Alta disipación de potencia en forma de calor. Necesidad de aislamiento galvánico.
Transformador de Corriente.	Capaz de medir altas corrientes. Se necesita emparejar la fase. Baja disipación comparada con la de Shunt. Proporciona aislamiento. Tiene problemas de saturación DC.	Un desfase de 0.1^0 produce un error en la facturación.
Bobina Rogowski **(Sensor di/dt).**	Tiene todas las ventajas del transformador de corriente, pero es más barato. Necesita un integrador. Inmune a la saturación DC. Buena linealidad. Bajo consumo.	Es difícil tener un integrador que sea muy estable con el tiempo.
Efecto Hall.	Son más caros. Baja linealidad. Bueno para altas corrientes.	Consumo medio. Variación alta con la temperatura.

Tabla 1. Resumen de características y efectos para cada sensor.

La tabla anterior nos refleja que en caso de altas corrientes es aconsejable utilizar el sensor de efecto hall en vez del shunt, ya que este último disipa mucha potencia en forma de calor. Los sensores más económicos son los de efecto shunt en conjunto con el Rogowski, ambos no tienen problemas con la saturación de la corriente continua.

Tecnología del sensor	Shunt de corriente	Transformador de corriente	Sensor de Efecto Hall	Bobina Rogowski
Costo	Muy bajo	Medio	Alto	Bajo
Linealidad en el rango de la medición	Muy buena	Buena	Pobre	Muy buena
Capacidad de medición de alta corriente	Muy pobre	Buena	Buena	Muy buena
Consumo de potencia	Alto	Bajo	Medio	Bajo
Problema de saturación de corriente DC	No	Sí	Sí	No
Variación de la salida con respecto a la temperatura	Medio	Bajo	Alto	Muy bajo
Problema de offset de DC	Sí	No	Sí	No
Problema de saturación e histéresis	No	Sí	Sí	No

Tabla 2. Comparación de las diversas tecnologías de sensores de corriente.

1.3 CIRCUITO INTEGRADO PARA LA MEDICIÓN DE ENERGÍA ELÉCTRICA.

1.3.1 MEDIDORES DE ENERGIA.

Los medidores de energía, técnicamente son aquellos que miden la cantidad de energía eléctrica en una carga. Existen diferentes clases de medidores de energía, que son mostrados brevemente a continuación con la intención de nombrar las características generales de estos dispositivos.

1.3.1.1 MEDIDOR ELECTROMECANICO

Medidores electromecánicos o medidores de inducción; compuestos por un conversor electromecánico (básicamente un vatímetro con su sistema móvil de giro libre) que actúa sobre un disco, cuya velocidad de giro es proporcional a la potencia demandada, y provistos de un dispositivo integrador.

Figura 9. Medidor de Energía Electromecánico.

1.3.1.2 MEDIDORES TOTALMENTE ELECTRÓNICOS.

La medición de energía y el registro se realizan por medio de un proceso análogo-digital (sistema totalmente electrónico) utilizando un microprocesador, visualización digital y memorias, permitiendo así manejar datos digitales que puedan ser compartidos por

otros sistemas digitales.
Los medidores electrónicos ofrecen varios beneficios, además de medir la potencia instantánea, pueden medir otros parámetros como el factor de potencia y la potencia reactiva. Los datos pueden ser medidos y almacenados a intervalos específicos, lo que permite la utilidad de ofrecer planes de precios basados en la hora del día de uso. Esto permite a los consumidores inteligentes de ahorrar dinero mediante la ejecución de los electrodomésticos, como lavadoras y secadoras, durante los períodos de menor costo, de baja demanda, y las empresas de servicios públicos pueden evitar la construcción de nuevas plantas de energía, ya que se requiere menos capacidad durante los períodos pico. Los contadores electrónicos no están influenciados por imanes externos o la orientación del propio metro, por lo que son más a prueba de manipulaciones que los medidores electromecánicos. Los medidores electrónicos también son altamente confiables.

Analog Devices ha sido un actor clave en la transición de los contadores electromecánicos a electrónicos, enviando más de 225 millones de circuitos integrados 30 de medición de energía hasta la fecha. Según IMS Research4, el 75% de todos los contadores de energía enviados en 2007 fueron electrónicos en lugar de electromecánicos.

Figura 10. Medidor totalmente electrónico.

1.3.2 CIRCUITOS INTEGRADOS MEDIDORES DE ENERGIA ANALOG DEVICES.

Hemos utilizado un sistema de estado sólido basado en un circuito integrado medidor de energía de Analog Devices, en esta aplicación se ha tenido como premisa el diseñar un instrumento versátil y de costo reducido. El rango de aplicación de este dispositivo va desde el reemplazo de los actuales medidores electromecánicos, hasta su uso como

instrumento de precisión en tareas de supervisión de consumo de equipo y maquinaria semiindustrial e industrial.

La familia de dispositivos analógicos IC (ADE) combina tecnología de conversión de datos líder en la industria con un procesador de señal digital de función fija (DSP) para realizar los cálculos esenciales para una electrónica medidor de energía.

La compañía Analog Devices ofrece en el mercado, la familia ADE, con una gran variedad de integrados capaces de resolver la gran gama de variaciones de configuraciones de sistemas de medición en el mundo, incluyendo la de este trabajo. Mas aplicaciones con circuitos integrados de esta compañía están disponibles en su sitio WEB www.analog.com tal como se muestra en la Figura 11.

Permitiendo a diseñadores escoger componentes diferentes de la familia dependiendo del tipo de medidor que se desea construir. La familia de integrados ADE se divide en dos grupos, medición monofásica y medición polifásica, con cinco mediciones críticas disponibles: vatios, V rms, I rms, VA y VAR.

.

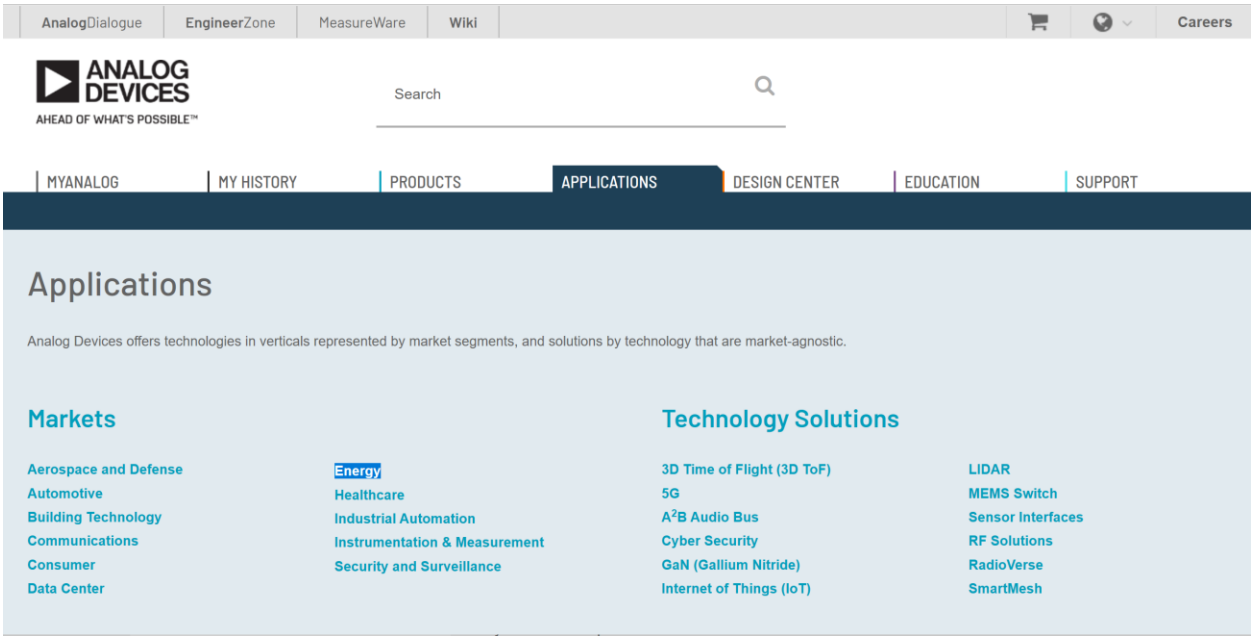


Figura 11. Soluciones disponibles con IC Analog Devices.

1.3.3 CIRCUITOS INTEGRADOS ADE77XX DE ANALOG DEVICES.

Los circuitos integrados de medición de energía de Analog Devices son referenciados por sus siglas en inglés ADE (Analog Devices Energy).
Estos dispositivos poseen el mejor rendimiento en su clase a costos atractivos, ofreciendo soluciones para sistemas polifásicos o monofásicos, este último en medición de energía o en comunicación de datos.

Los circuitos Integrados ADE77XX son el desarrollo de la tecnología de estado sólido para medición de energía eléctrica trifásica con interfaces serial con características de alta precisión los cuales son desarrollados por la empresa Analog Devices. Los dispositivos incorporan convertidores analógico - digital (ADC) todo el procesamiento de señal requerido para realizar de medición total (fundamental y armónico) de energía activa, reactiva, todo ello depende del modelo de Chipset a ser elegido entre la diversa gama con la que cuenta la compañía.

En detalle los circuitos integrados ADE digitalizan señales de corriente y voltaje por medio de ADC (convertidores análogo-a-digital) antes de hacer los cálculos de energía. El procesamiento de señal digital de las señales permite un cálculo estable y exacto por encima de las variaciones de tiempo y medio ambiente.

La compañía Analog Devices posee una amplia gama de circuitos integrados para la medición de energía polifásica con interfaz de comunicación chip to chip, para nuestro caso se utilizó un dispositivo con Interfaz de Periféricos Seriales (SPI).

El ADE7758 presenta un convertidor analogico digital de segundo orden sigma-delta, y está diseñado para medidores de energía trifásicos de rango medio.

Para cada fase, el chip mide energía activa, reactiva y aparente, así como voltaje RMS y corriente RMS. Se accede a estas mediciones a través de un SPI eso permite una calibración digital totalmente automatizada.
El ADE7758 interactúa con una variedad de sensores, incluyendo transformadores de corriente y corriente di /dt, como las bobinas de Rogowski.

Descripción del ADE7758.
Aplicaciones.
Sistemas eléctricos trifásicos Estrella/ Delta de 3 hilos.
Sistemas eléctricos trifásicos Estrella/ Delta de 4 hilos.
Sensores permitidos.
Transformadores de corrientes
Bobinas Rogowski
Mediciones disponibles.
Energía Activa
Energía Reactiva
Energía Aparente
Voltaje RMS
Corriente RMS

Tabla 3. Algunas características del IC ADE7758.

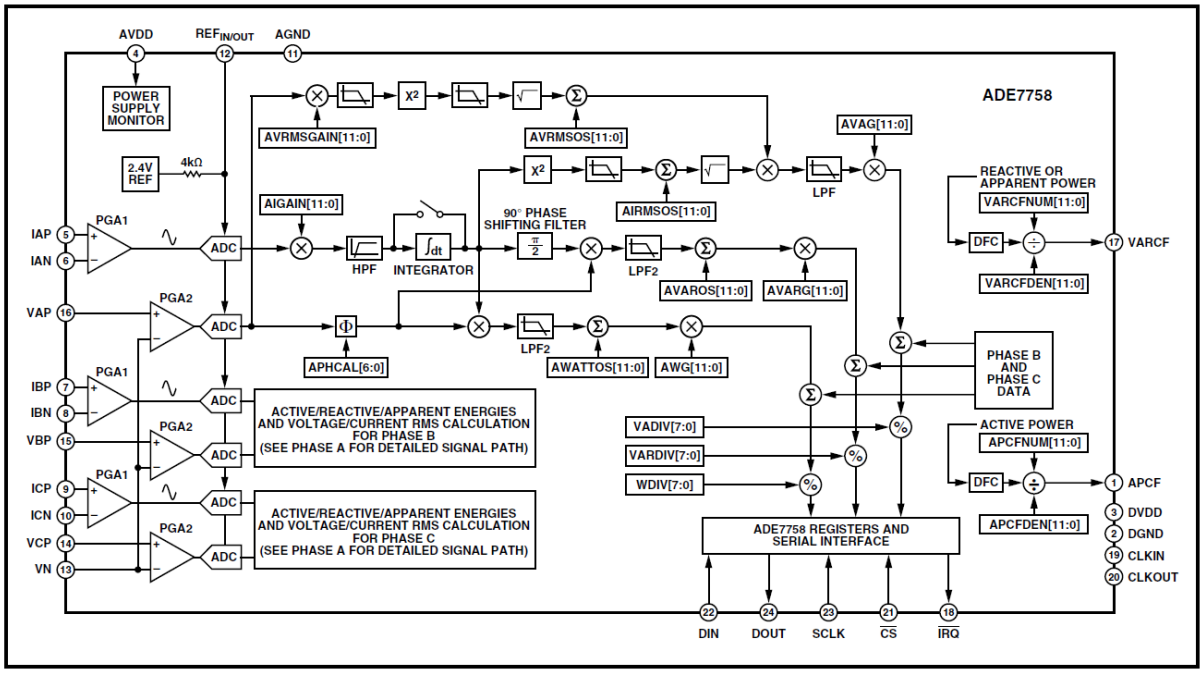


Figura 12. Diagrama de bloques del ADE7758.

1.4 INTERFAZ DE COMUNICACIÓN DE CIRCUITOS INTEGRADOS.

1.4.1 INTRODUCCIÓN.

La Interfaz Periférica Serial (SPI) es un bus de interfaz comúnmente utilizado para enviar datos entre microcontroladores y pequeños periféricos, como registros de desplazamiento, sensores y tarjetas SD. Utiliza líneas de reloj y datos separadas, junto con una línea de selección para elegir el dispositivo con el que desea comunicar.

Es importante para abordar el tema, conocer sobre la comunicación serial, la cual se describe a continuación.

Cabe destacar que la electrónica integrada se trata de circuitos de interconexión (procesadores u otros circuitos integrados) para crear un sistema simbiótico. Para que esos circuitos individuales intercambien su información, deben compartir un protocolo de comunicación común. Cientos de protocolos de comunicación se han definido para lograr este intercambio de datos, y, en general, cada uno se puede separar en una de dos categorías: paralelo o en serie.

1.4.2 COMUNICACIÓN PARALELA VERSUS COMUNICACIÓN SERIAL.

Es importante analizar desde sus inicios la comunicación que han presentado los dispositivos electrónicos y del porqué de sus cambios, previo a la comunicación serial era muy utilizada la comunicación paralela, las interfaces paralelas transfieren múltiples bits al mismo tiempo. Generalmente requieren buses de datos, que transmiten a través de ocho, dieciséis o más cables. Los datos se transfieren en enormes ondas de 1 y 0.

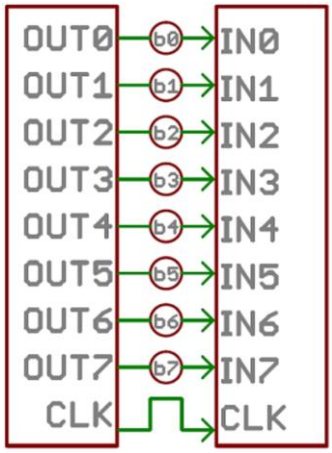


Figura 13. Comunicación paralela de un bus de datos de 8 bits, controlado por un reloj.

En la Figura 13 podemos observar un bus de datos de 8 bits, controlador por un reloj, que transmite un byte a cada pulso de reloj, se utilizan 9 cables como se puede observar para dicho propósito.

Para el caso de las interfaces en serie, estas transmiten sus datos, un bit por bit. Estas interfaces pueden operar con tan solo un cable, generalmente nunca más de cuatro.

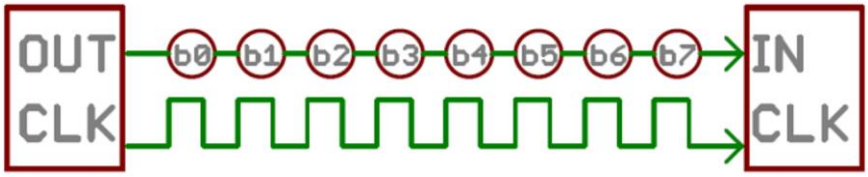


Figura 14. Ejemplo de una interfaz serial, transmitiendo un bit por cada pulso de reloj.

La comunicación paralela ciertamente tiene sus beneficios. Es rápida, sencilla y relativamente fácil de implementar. Pero requiere muchas más líneas de entrada / salida (E / S). En el desarrollo del este proyecto estamos utilizados la plataforma de Raspberry Pi B, se sabe que las líneas de E / S en un microprocesador pueden ser preciosas y pocas. Por lo tanto, a menudo optamos por la comunicación en serie, sacrificando la velocidad potencial en la comunicación entre dispositivos.

1.4.3 TIPOS DE COMUNICACIÓN SERIAL (SÍNCRONA Y ASÍNCRONA).

En la evolución en el tiempo que ha tenido los protocolos serie se han diseñado para satisfacer las necesidades particulares de los sistemas integrados. USB (bus serie universal) y Ethernet son algunas de las interfaces en serie de computación más conocidas. Otras interfaces series muy comunes incluyen SPI, I2C y el estándar de serie del que estamos aquí para hablar hoy. Cada una de estas interfaces en serie puede clasificarse en uno de dos grupos: síncrono o asíncrono.

Una **interfaz en serie síncrona** siempre empareja su (s) línea (s) de datos con una señal de reloj, de modo que todos los dispositivos en un bus en serie síncrono comparten un reloj común. Esto permite una transferencia en serie más directa, a menudo más rápida, pero también requiere al menos un cable adicional entre los dispositivos de comunicación. Los ejemplos de interfaces síncronas incluyen SPI e I2C.

Una **interfaz en serie asíncrona** significa que los datos se transfieren sin soporte de una señal de reloj externo. Este método de transmisión es perfecto para minimizar los cables necesarios y los pines de Entrada/Salida (E / S), pero significa que debe esforzarse un poco para transferir y recibir datos de manera confiable. Ejemplo de interfaz asíncrona podemos mencionar la UART (Universal Asynchronous Receiver / Transmitter).

1.4.4 ¿QUÉ ESTÁ MAL CON LOS PUERTOS SERIE ASÍNCRONOS?

Un puerto serie común, del tipo con líneas TX y RX, se denomina "asíncrono" (no síncrono) porque no hay control sobre cuándo se envían los datos o si se garantiza que ambos lados se ejecutan exactamente a la misma velocidad. Como las computadoras normalmente se basan en todo lo que se sincroniza con un único "reloj" (el cristal principal conectado a una computadora que lo maneja todo), esto puede ser un problema cuando dos sistemas con relojes ligeramente diferentes intentan comunicarse entre sí.

Para evitar este problema, las conexiones en serie asíncronas agregan bits adicionales de inicio y detención a cada byte para ayudar al receptor a sincronizar los datos a medida que llegan. Ambos lados también deben acordar la velocidad de transmisión (como 9600 bits por segundo) de antemano. Las ligeras diferencias en la velocidad de transmisión no son un problema porque el receptor se vuelve a sincronizar al comienzo de cada byte.

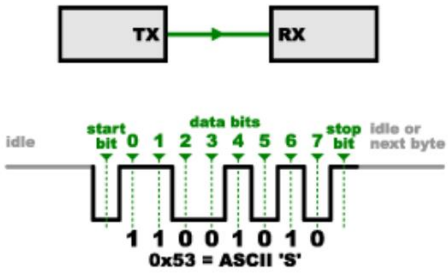


Figura 15. Ejemplo de comunicación serial asíncrona

La comunicación serial asíncrona funciona bien, pero tiene una gran sobrecarga tanto en los bits adicionales de inicio y finalización enviados con cada byte, como en el complejo hardware requerido para enviar y recibir datos. Además, cabe mencionar nuevamente que, si ambos lados no están configurados a la misma velocidad, los datos recibidos serán basura. Esto se debe a que el receptor está muestreando los bits en momentos muy específicos (las flechas en el diagrama anterior). Si el receptor mira los momentos incorrectos, verá los bits equivocados.

1.4.5 UNA SOLUCIÓN SINCRÓNICA

La comunicación SPI funciona de una manera ligeramente diferente. Es un bus de datos "síncrono", como ya se hizo mención, significa que utiliza líneas separadas para los datos y un "reloj" que mantiene a ambos lados en perfecta sincronización. El reloj es una señal oscilante que le dice al receptor exactamente cuándo debe muestrear los bits en la línea de datos. Este podría ser el flanco ascendente (de menor a mayor) o descendente (de mayor a menor) de la señal del reloj; la hoja de datos especificará cuál usar. Cuando el receptor detecta ese flanco, inmediatamente mirará la línea de datos para leer el siguiente bit (ver las flechas en el siguiente diagrama de la Figura 16). Debido a que el reloj se envía junto con los datos, especificar la velocidad no es importante, aunque los dispositivos tendrán una velocidad máxima a la que pueden operar.

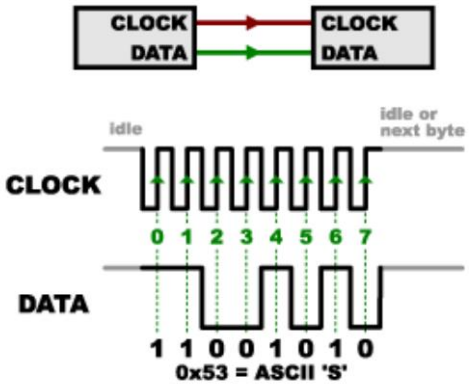


Figura 16. Esquema básico Comunicación SPI.

Una razón por la que SPI es tan popular es porque el hardware receptor puede ser un simple registro de desplazamiento. Este es un hardware mucho más simple (¡y más barato!) Que el UART (Receptor / Transmisor Asíncrono Universal) que requiere el serial asíncrono.

1.4.6 RECIBIENDO INFORMACIÓN.

Es de tomar en cuenta que según como se ha explicado en la sección anterior, solo hemos mostrado la comunicación en un solo sentido, sin embargo, la comunicación SPI es Full Dúplex, lo que implica que puede enviar y recibir información desde y hacia el dispositivo maestro.

En la comunicación SPI, solo un lado genera la señal de reloj (generalmente llamada CLK o SCK para Serial ClocK). El lado que genera el reloj se llama el "maestro", y el otro lado se llama el "esclavo". Siempre hay un solo maestro (que es casi siempre su microcontrolador), pero puede haber múltiples esclavos.

Cuando los datos se envían desde el maestro a un esclavo, se envían a una línea de datos llamada MOSI, para "Master Out / Slave In". Si el esclavo necesita enviar una respuesta al maestro, el maestro continuará generando un número de ciclos de reloj predispuestos, y el esclavo colocará los datos en una tercera línea de datos llamada MISO, para "Master In / Slave Out".

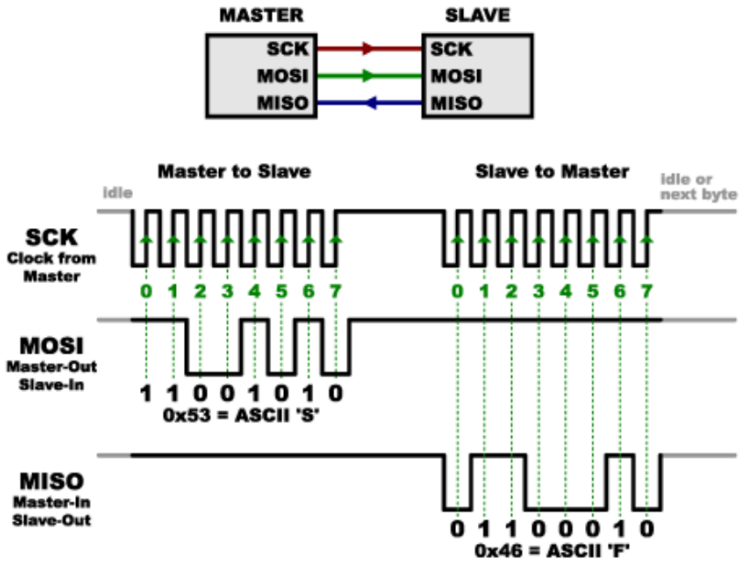


Figura 17. Esquema básico Comunicación SPI Full Duplex.

Observe que dijimos "predispuestos" en la descripción anterior. Debido a que el maestro siempre genera la señal del reloj, debe saber de antemano cuando un esclavo necesita devolver los datos y la cantidad de datos que se devolverán. Esto es muy diferente de la serie asincrónica, donde se pueden enviar cantidades aleatorias de datos en cualquier dirección en cualquier momento. En la práctica, esto no es un problema, ya que SPI generalmente se usa para hablar con sensores que tienen una estructura de comando muy específica. Por ejemplo, si envía el comando de "lectura de datos" a un dispositivo, sabrá que el dispositivo siempre le enviará, por ejemplo, dos bytes a cambio. (En los casos en que desee devolver una cantidad variable de datos, siempre puede devolver uno o dos bytes especificando la longitud de los datos y luego hacer que el maestro recupere la cantidad completa).

Tenga nuevamente en cuenta que SPI es "Full Duplex" (tiene líneas de envío y recepción separadas, Figura 17) y, por lo tanto, en ciertas situaciones, puede transmitir y recibir datos al mismo tiempo (por ejemplo, solicitando una nueva lectura del sensor mientras recupera los datos del el anterior). La hoja de datos de su dispositivo le dirá si esto es posible.

1.4.7 SELECCIÓN DE ESCLAVO (SS- SLAVE SELECT)

Hay una última línea que debe tener en cuenta, llamada **SS (Slave Select)** para seleccionar esclavos. Esto le dice al esclavo que debe despertarse y recibir / enviar datos y también se usa cuando hay varios esclavos presentes para seleccionar el que le gustaría hablar.

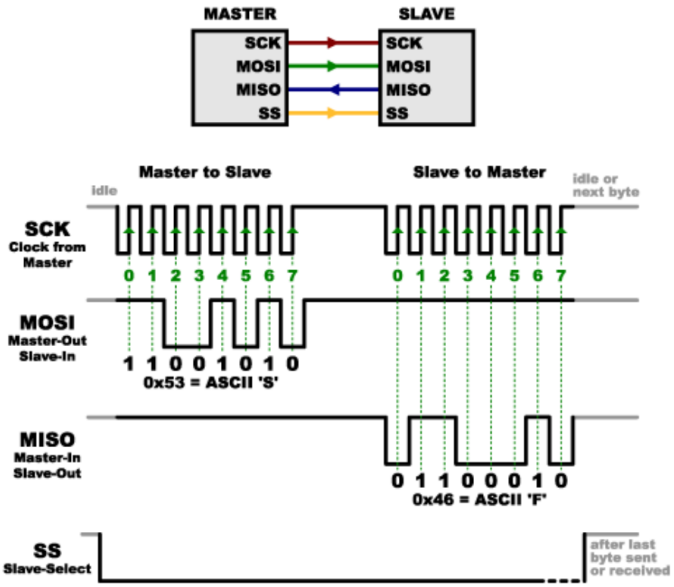


Figura 18. Esquema Comunicación SPI, utilizando el habilitador SS para seleccionar dispositivo.

La línea SS normalmente se mantiene alta, lo que desconecta al esclavo del bus SPI. (Este tipo de lógica se conoce como "activo bajo", y a menudo se verá utilizado para habilitar y restablecer líneas). Justo antes de que se envíen datos al esclavo, la línea baja, lo que activa el esclavo. Cuando termine de usar el esclavo, la línea se vuelve alta nuevamente. En un registro de desplazamiento, esto corresponde a la entrada "latch", que transfiere los datos recibidos a las líneas de salida.

1.4.8 MÚLTIPLES ESCLAVOS.

Hay dos formas de conectar múltiples esclavos a un bus SPI, las cuales se describen a continuación:

I. En general, cada esclavo necesitará una línea SS separada. Para hablar con un esclavo en particular, hará baja la línea SS de ese esclavo y mantendrá el resto de ellos en alto (no quiere que se activen dos esclavos al mismo tiempo, o ambos pueden tratar de hablar sobre la misma línea MISO que resulta en datos confusos). Muchos esclavos requerirán muchas líneas de SS; si te estás quedando sin salidas, hay chips decodificadores binarios que pueden multiplicar tus salidas SS.

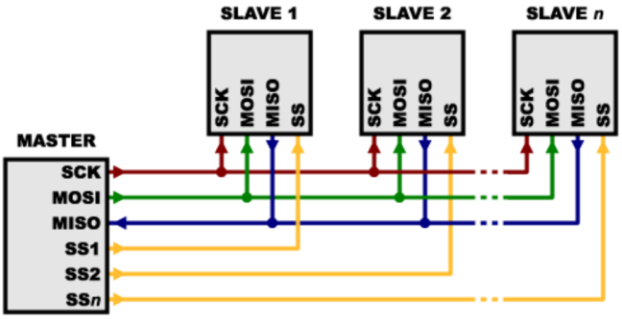


Figura 19. Esquema Comunicación SPI con múltiples esclavos, utilizando una línea de SS para cada dispositivo a comunicarse.

II. Por otro lado, se pueden conectar los dispositivos en cadena (Figura 20) en el cual el MISO (salida) de uno va al MOSI (entrada) del siguiente. En este caso, una sola línea SS va a todos los esclavos. Una vez que se envían todos los datos, se eleva la línea SS, lo que hace que todas las fichas se activen simultáneamente. Esto se usa a menudo para registros de desplazamiento en cadena y controladores de LED direccionables.

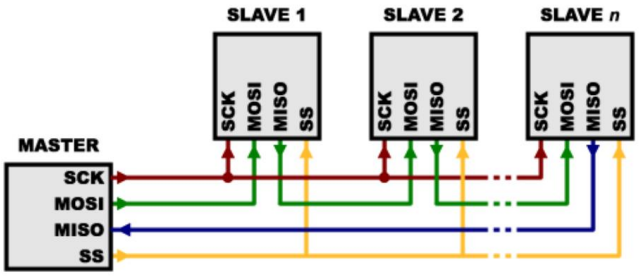


Figura 20. Esquema Comunicación SPI con múltiples esclavos, utilizando una línea de SS para todos los dispositivos a comunicarse.

Se tiene en cuenta que, para este diseño, los datos se desbordan de un esclavo al siguiente, por lo que, para enviar datos a cualquier esclavo, se tendrá que transmitir suficientes datos para alcanzarlos a todos. Además, se tiene en cuenta que la primera información que transmita terminará en el último esclavo.

Este tipo de diseño se usa generalmente en situaciones de solo salida, como conducir LED donde no necesita recibir datos. En estos casos, puede dejar la línea MISO del maestro desconectada. Sin embargo, si los datos necesitan ser devueltos al maestro, puede hacerlo cerrando el bucle de cadena margarita (cable azul en el diagrama de arriba). Tenga en cuenta que, si hace esto, los datos de retorno del esclavo 1 deberán pasar por todos los

esclavos antes de volver al maestro, así que asegúrese de enviar suficientes comandos de recepción para obtener los datos que necesita.

1.5 PLACA CONTROLADORA ARDUINO.

1.5.1 INTRODUCCION

Arduino se ha convertido en un referente del hardware libre que surgió como una herramienta para estudiantes pero que, poco a poco, ha conseguido romper barreras.
Debido a su filosofía, existe actualmente una gran comunidad de desarrolladores y desarrolladoras y a toda la tecnología y plataformas de las que disponemos, Arduino se ha convertido en una herramienta básica en el movimiento maker (crear objetos artesanales, pero utilizando la tecnología), la docencia (en las áreas de ciencia, ingeniería, tecnología, y matemáticas), el impulso del IoT (Internet of Things o «Internet de las cosas») y el prototipado.

La función de la plataforma Arduino es facilitar el uso de un microcontrolador (MCU).

1.5.2 ¿QUE ES ARDUINO?

Arduino es una placa controladora y su entorno de programación que permiten de manera rápida y sencilla realizar proyectos de electrónica, automatismo, control, domótica, etc. Arduino nació en el Instituto IVREA (Italia) para facilitar a sus alumnos las tareas de programación de microcontroladores. Es ese uno de los motivos de su éxito: cualquier persona con pocos conocimientos de informática y electrónica puede programarlo e implementarlo.

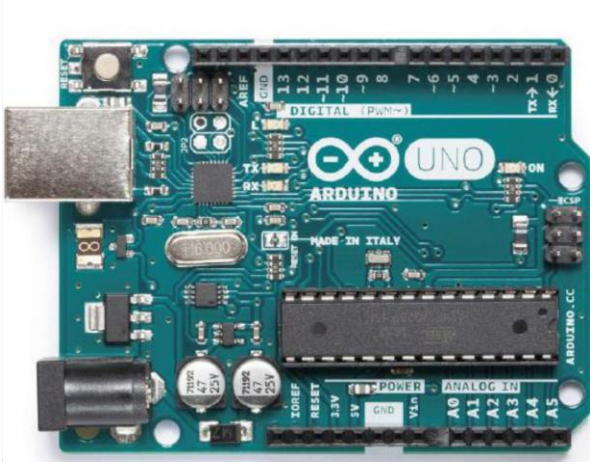


Figura 21. Placa Arduino UNO.

1.5.3 MICROCONTROLADORES (MCU).

Un Microcontrolador (MCU) es un circuito integrado que puede ser reprogramado y que está diseñado para el control de procesos mediante la lectura y generación de señales. Para entender mejor qué es Arduino y un MCU supongamos que tenemos que implementar una solución para el control de encendido de luces.
Si deseamos desarrollar la solución con un microcontrolador MCU. Primeramente, tenemos que escoger el más adecuado, considerando las características técnicas del proceso que vamos a controlar. En el mercado disponemos de varios fabricantes de MCU, todos ellos con un amplio catálogo. Tendremos que decantarnos por un fabricante y escoger un MCU de entre todos los que tiene; para analizar los MCU, necesitamos trabajar con sus manuales.

El manual de un MCU no está desarrollado para un usuario común, independientemente de su formación, pueda interpretarlo. Nos enfrentaremos a manuales de hasta 450 páginas, con un lenguaje muy técnico, por lo que no solo es necesario un conocimiento previo sobre MCU, sino sobre programación, muchos «mecanismos internos», como Timers, interrupciones, etc. Posiblemente, si nunca hemos programado, no entenderemos para qué sirven.
.
Además, debemos analizar sus limitaciones en cuanto a potencia eléctrica, este dispositivo debe brindarse la incorporación de las instrucciones por medio de su entorno de programación.

Como acabamos de ver, existe una gran cantidad de obstáculos para simplemente empezar a programar un Microcontrolador (MCU) y, con todo esto, una vez que los superemos, tenemos que ser capaces de llegar a una solución, desarrollar el programa adecuado e implementar físicamente este sistema (conectarlo a un entorno).

1.5.4 DESCRIPCION DE PINES PARA LA PLACA CONTROLADORA ARDUINO.

A continuación, veremos las distintas partes que conformar nuestro Arduino como son entradas, salidas, alimentación, comunicación.

- ✓ **Entradas:** son los pines de nuestra placa que podemos utilizar para hacer lecturas. En la placa Uno son los pines digitales (del 0 al 13) y los analógicos (del A0 al A5).

- ✓ **Salidas:** los pines de salidas se utilizan para el envío de señales. En este caso los pines de salida son sólo los digitales (0 a 13).

- ✓ **Otros pines:** también tenemos otros pines como los GND (tierra), 5V que proporciona 5 Voltios, 3.3V que proporciona 3.3 Voltios, los pines REF de referencia de voltaje, TX (transmisión) y RX (lectura) también usados para comunicación serial, RESET para resetear, Vin para alimentar la placa .

- ✓ **Alimentación:** Como hemos visto el pin Vin sirve para alimentar la placa, pero lo más normal es alimentarlo por el Jack de alimentación usando una tensión de 7 a 12 Voltios. También podemos alimentarlo por el puerto USB, pero en la mayoría de aplicaciones no lo tendremos conectado a un ordenador.

- ✓ **Comunicación:** En nuestro proyecto nos comunicaremos con Arduino mediante USB para cargar los programas o enviar/recibir datos. Sin embargo, no es la única forma que tiene Arduino de comunicarse. Cuando insertamos una shield ésta se comunica con nuestra placa utilizando los pines ICSP (comunicación ISP), los pines 10 a 13 (también usados para comunicación ISP), los pines TX/RX o cualquiera de los digitales ya que son capaces de configurarse como pines de entrada o salida y recibir o enviar pulsos digitales.

- ✓ **Shields:** traducido del inglés significa escudo. Se llama así a las placas que se insertan sobre Arduino a modo de escudo ampliando sus posibilidades de uso. En el mercado existen infinidad de shields para cada tipo de Arduino. Algunas de las más comunes son las de Ethernet, Wi-Fi, Ultrasonidos, Pantallas LCD, relés, matrices LED's, GPS...

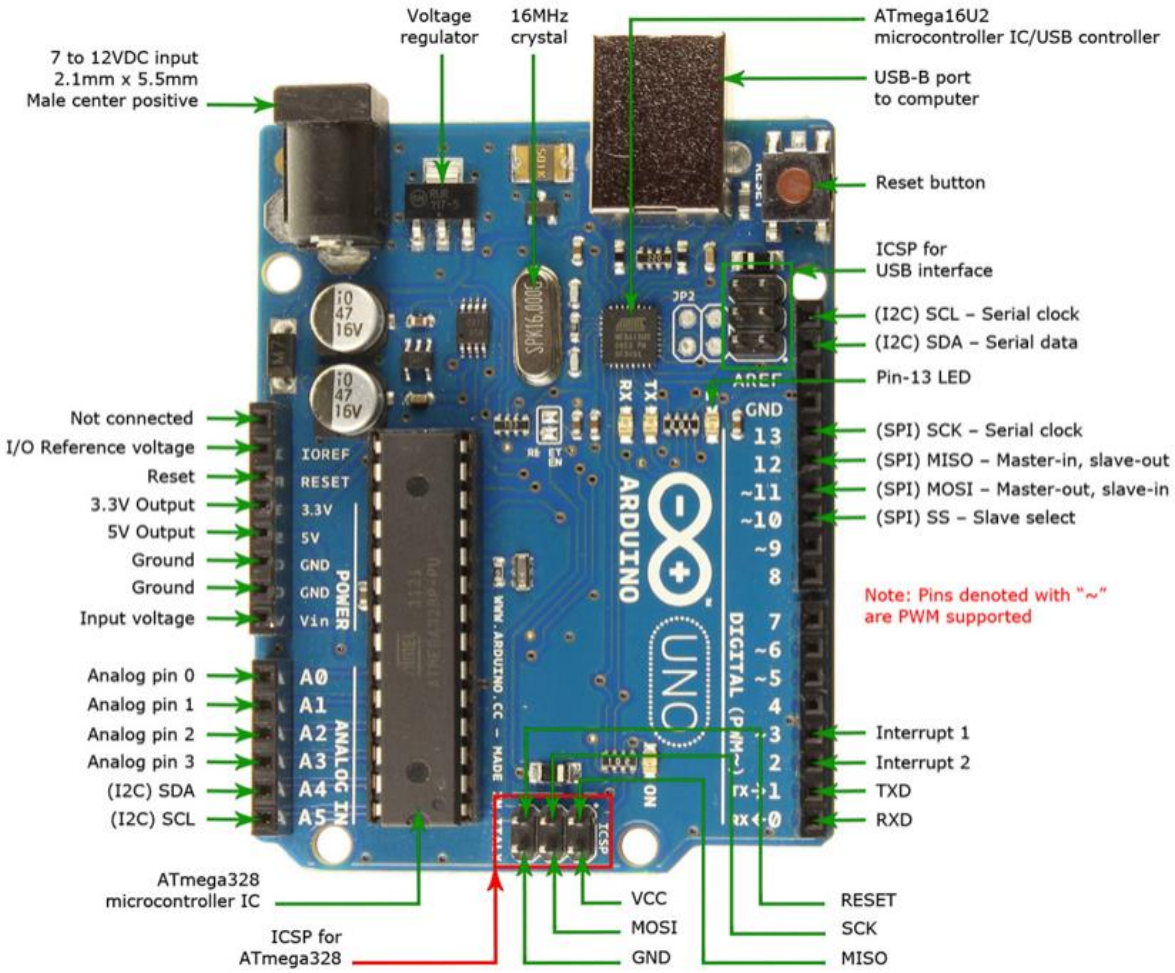


Figura 22. Pin- Out para el Arduino UNO.

1.5.5 SPI EN ARDUINO.

El propósito para el uso de la placa controladora Arduino, es la efectuar la de efectuar la adquisición de datos del circuito integrado medidor de energía ADE7758, por medio de la comunicación Chip to Chip SPI.

Arduino dispone de pines «dedicados» a comunicación SPI; se pueden configurar otros, pero la velocidad será menor. Los pines, según el modelo de Arduino, varían; en el caso del Arduino UNO:

- ✓ 10 -> SS
- ✓ 11 -> MOSI
- ✓ 12 -> MISO

- ✓ 13 -> SCK

El pin SS por hardware se emplea para establecer una comunicación con un Arduino que actúe como esclavo. En el caso de trabajar con el Arduino como maestro, se puede usar cualquier pin, como SS, y varios en el caso de tener varios esclavos.
Disponemos de una librería para trabajar con el protocolo SPI, que nos permite utilizar un Arduino como dispositivo maestro, el cual es nuestro caso, ya que el ADE7758 será muestro esclavo.

2 DISEÑO E IMPLEMENTACIÓN DE HARDWARE DEL EQUIPO PARA MEDICIÓN.

2.1 INTRODUCCIÓN.

En este capítulo se analiza con detalle el diseño del hardware, con él cual se ha trabajado para llevar a cabo el proceso de medición de energía. Se presenta cada etapa del diseño en el diagrama de bloque de la Figura 23.

A continuación, se presentan las etapas con las que se ha efectuado el diseño e implementación del equipo.

- Adquisición de señales para cada medición (El equipo puede obtener hasta 6 mediciones).
- Etapa de multiplexado de señales.
- Dispositivo para medición de energía y otras variables (ADE7758).
- Receptor SPI, utilizando Arduino.

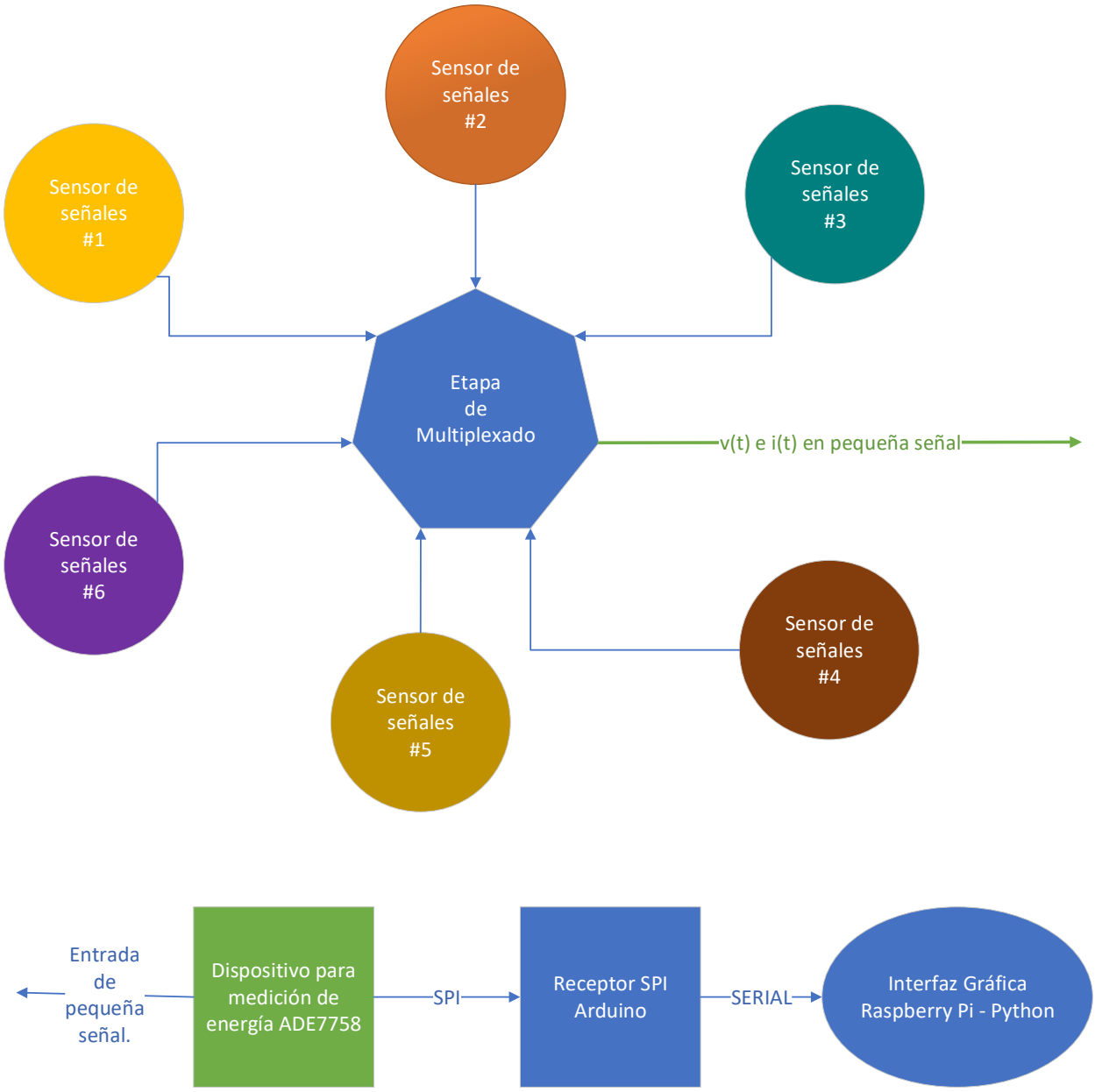


Figura 23. Diagrama de bloques para medidor de potencia.

2.2 ADQUISICIÓN DE SEÑALES.

Para la Adquisición de señales se utilizará, divisores de tensión para las lecturas de Voltaje y Transformadores de corriente, los dispositivos requeridos se describen a continuación.

2.2.1 TRANSFORMADORES DE CORRIENTE.

En su forma comúnmente citadas los transformadores de corriente (CT) son sensores que miden la corriente alterna (CA). El tipo de núcleo dividido, como el CT en la Figura 24, puede acoplarse al cable de Fase o neutro que alimenta una determinada carga, sin la necesidad de realizar ningún trabajo eléctrico de alto voltaje. Como cualquier otro transformador, un transformador de corriente tiene un devanado primario, un núcleo magnético y un devanado secundario.

Figura 24. Transformador de Corriente seleccionado en nuestro diseño.

En el caso de la medición de corriente eléctrica en una carga, el devanado primario es el cable vivo o neutro (¡NO ambos!) y pasa a través de la abertura en el TC. El devanado secundario está hecho de muchas vueltas de alambre fino alojado dentro de la caja del transformador.

La corriente alterna que fluye en el primario produce un campo magnético en el núcleo, que induce una corriente en el circuito secundario del devanado

La corriente en el devanado secundario es proporcional a la corriente que fluye en el devanado primario:

$$Isecundario = CT(relación, Transformación)\ x\ \ Iprimario$$

$$CT(relación, Transformación) = \frac{Número\ de\ Vueltas\ (Primario)}{Número\ de\ Vueltas\ (Secundario)}$$

El número de vueltas del secundario en la TC mostrado en la Figura 24 es de 2000, entonces la corriente en el secundario es una 1/2000 de la corriente en la primaria. Normalmente, esta relación se escribe en términos de corrientes en amperios, por ejemplo: 100: 5 (para un medidor de 5 A, escala de 0 a 100 A). La relación para el CT que hemos elegido se escribiría como 100: 0.05.

El circuito secundario está aislado galvánicamente del circuito primario. (es decir, no tiene contacto metálico).

2.2.1.1 RESISTOR DE CARGA (BURDEN RESISTOR)

Una "salida de corriente" del TC debe usarse con una resistencia de carga. La resistencia de carga completa o cierra el circuito secundario de CT. El valor de carga se elige para proporcionar un voltaje proporcional a la corriente secundaria. El valor de carga debe ser lo suficientemente bajo para evitar la saturación del núcleo de TC.

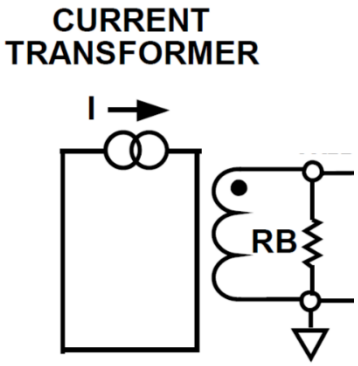


Figura 25. Resistencia Burden.

2.2.1.2 CONDICIONES DE SEGURIDAD.

Por regla general, un CT nunca debe estar en circuito abierto una vez que está conectado a un conductor portador de corriente. Un CT en estas condiciones es potencialmente peligroso si está en circuito abierto.

Si se abre el circuito con la corriente que fluye en el primario, el secundario del transformador intentará continuar impulsando la corriente hacia lo que efectivamente es una impedancia infinita. Esto producirá un voltaje alto y potencialmente peligroso a través del secundario.

Algunos CT tienen protección incorporada. Algunos tienen diodos Zener protectores, como es el caso con el SCT-013-000 el cual hemos utilizado en nuestro diseño para su uso en este proyecto.

2.2.2 DIVISORES DE TENSIÓN.

El método utilizado para obtener la medición de voltaje es menos complejo que la de Corriente, teniendo el requerimiento de entrada máxima de adquisición de señales requeridas por el IC ADE7758, la cual se muestra en la Figura 26, podemos apreciar que se necesita un máximo de 500 m V pico, para ellos utilizaremos un divisor de tensión como el mostrado en la Figura 27.

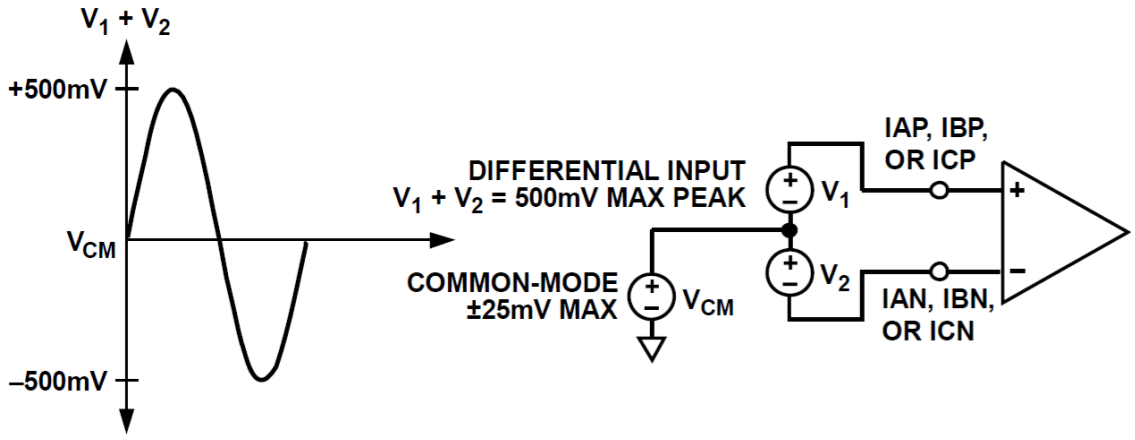


Figura 26. Voltaje máximo de entrada permitido para la adquisición de voltaje.

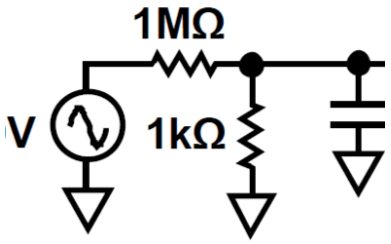


Figura 27. Voltaje máximo de entrada permitido para la adquisición de voltaje.

2.3 MULTIPLEXADO DE SEÑALES.

2.3.1 INTRODUCCION.

Se han adquirido las señales de voltaje y corriente a través de la Etapa 1, con ello obtendremos toda la información necesaria para que sea procesada por el Chipset medidor de energía ADE7758, y para lograr la medición desde 6 puntos diferentes utilizando únicamente uno de los Chipset mencionados, utilizaremos una etapa de Multiplexación de Señales, de la cual a partir de 36 señales se obtendrá 6 señales para su respectiva selección a medición.

2.3.2 DESCRIPCIÓN DE LOS CIRCUITOS MULTIPLEXORES.

Un Multiplexor o "**Selector de datos"** es un circuito lógico que acepta varias entradas de datos y permite que sólo una de ellas pase a un tiempo a la salida. El enrutamiento de la entrada de datos hacia la salida está controlado por las entradas de selección (a las que se hace referencia a veces como las entradas de dirección).

El multiplexor, también conocido como **MUX,** actúa como un conmutador multiposicional controlado digitalmente, donde el código digital aplicado a las entradas de selección controla cuáles entradas de datos serán conmutadas hacia la salida. Por ejemplo, la salida será igual a la entrada de datos, llamémosle D_0, para el código de entrada de selección que sea cero (ABC=000 en el diagrama de abajo); la salida será igual D_1 para cuando el código de selección sea uno y así sucesivamente. Establecido de otra manera, un multiplexor selecciona **1** de ***N*** fuentes de datos y transmite los datos seleccionados a un solo canal de salida. Esto se llama multiplexión o multiplexaje.

Los multiplexores son representados en diagramas de bloques como trapezoides isósceles. A continuación en la Figura 28 , muestro el esquemático de un multiplexor de dos entradas y una salida con su respectivo bit de selección:

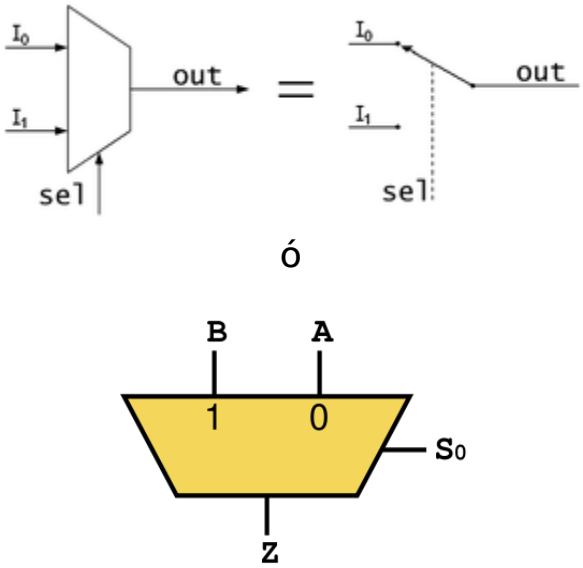


Figura 28. Representación de Circuitos Multiplexores.

Un ejemplo de multiplexores se ve en las líneas telefónicas. Éstas usan exactamente este principio. Transmiten varias llamadas telefónicas (señales de audio) a través de un único par cableado usando la técnica de "multiplexado" y cada señal de audio va únicamente al receptor al que está destinado, para nuestro caso el funcionamiento serán el de recibir las mediciones de los 6 puntos de medición los cuales llegarán a un único chip ADE7758 el cual nos entregará los valores medidos para cada punto de medición según sea el seleccionado en nuestra interfaz gráfica.

Otra aplicación común para los MUX es encontrada en las computadoras, en las cuales la memoria dinámica usa las mismas líneas de dirección para el direccionamiento tanto de las filas como de las columnas. Un grupo de multiplexores es usado para primero seleccionar las direcciones de la columna y luego cambiar para seleccionar la de la fila. Este esquema permite que grandes cantidades de memoria sean incorporadas dentro de una computadora mientras se limita a la vez la cantidad de conexiones de cobre requeridas para conectar la memoria al resto del circuito. Por eso es que también se les conoce a veces como "selectores de datos".

Ya se vio el símbolo esquemático del multiplexor de 2 entradas y una salida, pero los multiplexores no están limitados a 2 entradas. Si las líneas de selección son dos podemos alternar entre 4 datos de entrada, si son 3 entre 8 y así sucesivamente.

A continuación, en la Figura 29, se muestran los símbolos esquemáticos de los multiplexores de "4 a 1" (cuatro entradas y una salida), "8 a 1" (ocho entradas y una salida) y "16 a 1" (dieciséis entradas y una salida) con sus respectivas líneas de selección, respectivamente.

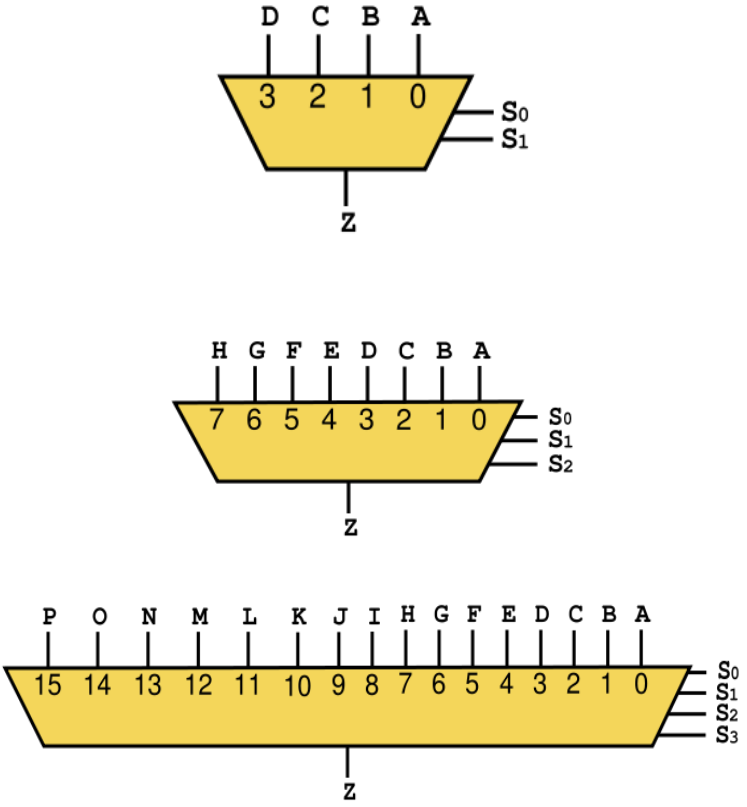


Figura 29. Representación gráfica de Circuitos Multiplexores de 4 a 1, 8 a 1 y 16 a 1.

En todos los casos la salida es Z, las entradas de selección S y el resto es la entrada que será multiplexada.

A veces pueden verse en forma rectangular asemejando el circuito integrado que representan, pero en este caso siempre deben ir bien identificados cada uno de los pines.

Por ejemplo:

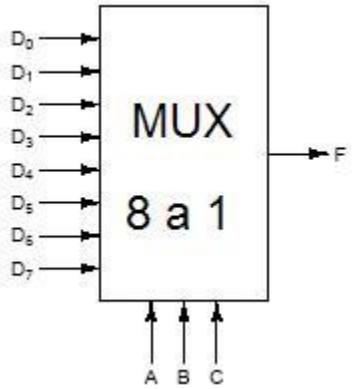


Figura 30. Representación como bloque rectangular de Multiplexor de 8 a 1.

El esquema anterior en la Figura 30 representa (como se ve indicado) un multiplexor a nivel MSI de 8 entradas (que implica las 3 líneas de selección) y una salida (F). Las entradas de selección, o sea, quienes indicarán cuál de las entradas será reflejada en la salida, vienen dadas por el código binario representado por ABC. ABC son las "entradas de direccionamiento" o de dirección o de selección, como usted lo quiera llamar, ya que estas serán quienes indican el dato a acceder. Este mismo concepto es el usado en las memorias.

2.4 CIRCUITO INTEGRADO PARA LA MEDICIÓN DE ENERGÍA ELÉCTRICA.

2.4.1 MEDIDORES DE ENERGIA.

Hemos utilizado un sistema de estado sólido basado en un circuito integrado medidor de energía de Analog Devices, en esta aplicación se ha tenido como premisa el diseñar un instrumento versátil y de costo reducido. El rango de aplicación de este dispositivo va desde el reemplazo de los actuales medidores electromecánicos, hasta su uso como instrumento de precisión en tareas de supervisión de consumo de equipo y maquinaria semiindustrial e industrial.

2.4.2 CIRCUITOS INTEGRADOS ADE77XX DE ANALOG DEVICES.

Los circuitos Integrados ADE77XX son el desarrollo de la tecnología de estado sólido para medición de energía eléctrica trifásica con interfaces serial con características de alta precisión los cuales son desarrollados por la empresa Analog Devices. Los dispositivos incorporan convertidores analógico - digital (ADC) todo el procesamiento de señal requerido para realizar de medición total (fundamental y armónico) de energía activa, reactiva, todo ello depende del modelo de Chipset a ser elegido entre la diversa gama con la que cuenta la compañía.

2.4.3 DESCRIPCION GENERAL DEL CIRCUITO INTEGRADO ADE7758.

El ADE7758 es un circuito integrado para medición de energía eléctrica trifásica con interfaces serial y tres salidas de pulso flexibles con características de alta precisión. Los dispositivos incorporan convertidores analógico - digital (ADC) de segundo orden, un integrador digital circuito de referencia y todo el procesamiento de señal requerido para realizar de medición total (fundamental y armónico) de energía activa, reactiva.

El ADE7758 puede medir la energía activa, reactiva y aparente en varias configuraciones trifásicas, como servicios en estrella o delta, con tres y cuatro cables. Aparte de las medidas rms regulares, que son actualizados cada 8 kHz, estos dispositivos miden valores de rms de rizado bajo, que se promedian internamente y se actualizan cada 1,024 segundos. El dispositivo proporciona funciones de calibración del sistema para cada fase, las cuales son, corrección de desplazamiento eficaz, calibración de fase y calibración de ganancia.

El AD7758A también incorpora medidas de calidad de energía, tales como detección de baja o alta tensión de corta duración, corta duración alta variación de corriente, medición de período de voltaje de línea y ángulos entre voltajes de fase y corrientes.

Interfaz periférica serial (SPI), puede comunicarse con los dispositivos. Una alta velocidad dedicada interfaz.

El dispositivo tiene dos pines de solicitud de interrupción, IRQ0 e IRQ1, para indicar que ha ocurrido un evento de interrupción habilitado.

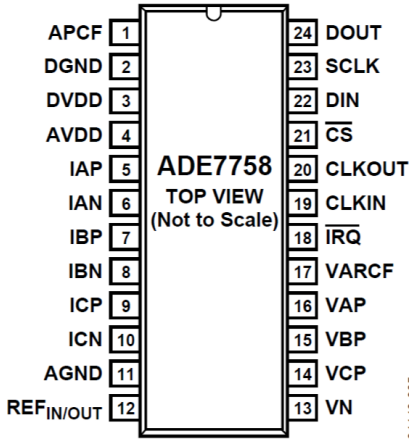


Figura 31. Distribución de Pines del ADE7758.

2.4.4 TEORÍA, FUNCIONAMIENTO DEL CIRCUITO INTEGRADO ADE7758.

A continuación, se brinda la descripción del funcionamiento del Circuito Integrado ADE7758, para ellos se describirá las etapas más importantes para su funcionamiento básico.

2.4.5 ENTRADAS ANALÓGICAS.

El ADE7758 tiene seis entradas analógicas, pero la corriente de neutro se elimina de estos dispositivos. Los canales de corrientes consisten en cuatro pares de entradas de voltajes completamente diferencial: IAP e IAN, IBP e IBN, ICP e ICN, y INP y INN. Estos pares de entrada de voltaje tienen un máximo señal diferencial de ± 500mV pico. Además, el máximo nivel de señal en las entradas analógicas para cada par IxP / IxN es de ± 500mV V pico con respecto a AGND. El modo común máximo la señal permitida en las entradas

es ± 25 mV. La Figura 32 muestra un esquemático de la entrada para los canales actuales y su relación con la tensión máxima en modo común.

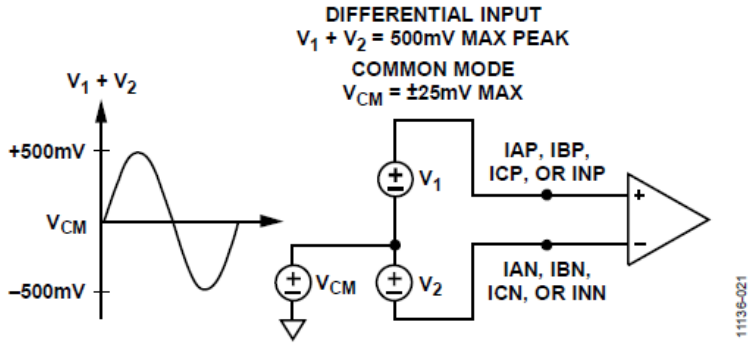


Figura 32. Distribución de Pines del ADE7758.

Todas las entradas tienen un amplificador de ganancia programable (PGA) con posible selección de ganancia de 1, 2, 4, 8 o 16. La ganancia de las entradas IAx, IBx e ICx se configuran en Bits [2: 0] (PGA1 [2: 0]) de la ganancia registro.
El canal de voltaje tiene tres entradas de voltaje de una única vez: VAP, VBP y VCP. Estas entradas de voltaje de terminación única tienen un máximo voltaje de entrada de ± 0.5 V con respecto a VN. además, el nivel máximo de señal en entradas analógicas para VxP y VN es ± 0.5 V con respecto a AGND. El máximo en modo común de señal permitida en las entradas es ± 25 mV.

En la Figura 33 se observa el esquema para un esquema de las entradas del canal de voltaje y su relación con la tensión máxima en modo común.

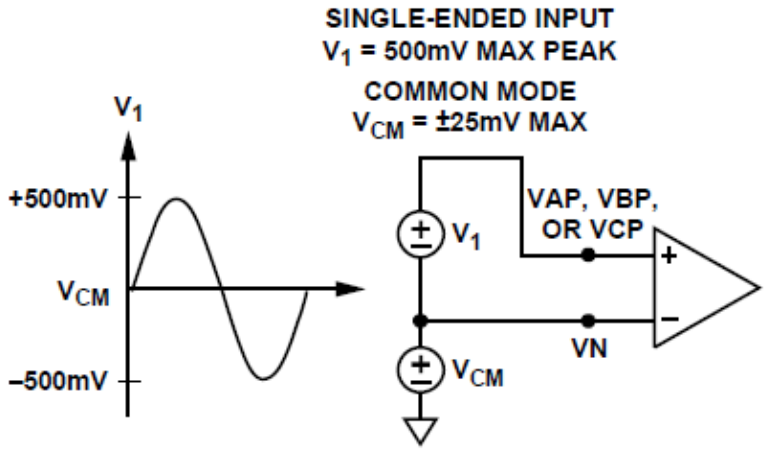


Figura 33. Máximo nivel de entrada, canales de voltajes, ganancia = 1.

Todas las entradas tienen una ganancia programable con una ganancia posible selección de 1, 2, 4, 8 o 16.

La Figura 34 muestra cómo la selección de ganancia del registro de ganancia funciona tanto en los canales de corriente como de voltaje.

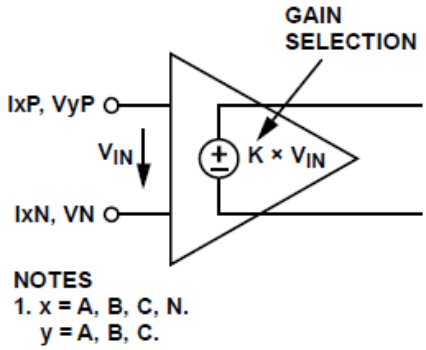


Figura 34. Amplificadores de ganancia programables en canales de voltaje y corriente.

2.4.6 CONVERSIÓN ANÁLOGO DIGITAL.

El ADE7758 tiene seis Σ-Δ ADCs. A continuación, se describen los modos de operación:

- En el modo PSM0, todos los ADC están activos.
- En el modo PSM1, solo los ADC que miden las corrientes de Fase A, de fase B y fase C están activas. Los ADC que miden el corriente neutro y los voltajes de fase A, B y C
  están apagados.
- En los modos PSM2 y PSM3, los ADC están apagados
  para minimizar el consumo de energía.

Para simplificar, el diagrama de bloques de la Figura 35 muestra Σ-Δ ADC de primer orden. El convertidor está compuesto por el modulador Σ-Δ y el filtro digital de paso bajo.

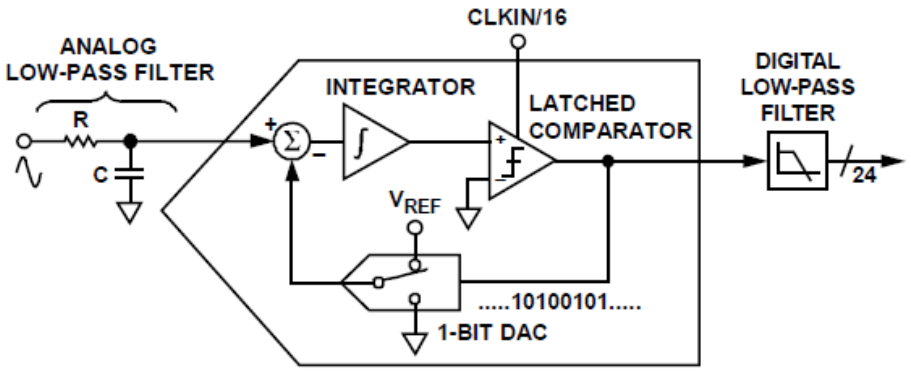


Figura 35. ADC Σ-Δ de primer orden.

El modulador Σ-Δ convierte la señal de entrada en una secuencia en serie de 1s y 0s a una velocidad determinada por el muestreo reloj. En el ADE7858A, el reloj de muestreo es igual a 1.024 MHz (CLKIN / 16). La bandera marcada 1-BIT DAC en el circuito de retroalimentación es manejado por los datos de transmisión serial. La salida DAC se resta de la señal de entrada. Cuando la ganancia del lazo es lo suficientemente alta, el valor promedio de la salida DAC (y, por lo tanto, el flujo de bits) puede acercarse a esa del nivel de señal de entrada. Para cualquier valor de entrada dado en un solo intervalo de muestreo, los datos del ADC de 1 bit son virtualmente sin sentido. Solo cuando hay una gran cantidad de muestras promediado es un resultado significativo obtenido. Este promedio ocurre en la segunda parte del ADC (el filtro digital de paso bajo). Por promedio de una gran cantidad de bits del modulador, el paso bajo filtro puede producir palabras de datos de 24 bits que son proporcionales al nivel de señal de entrada
El Σ-Δ ADC utiliza dos técnicas para lograr una alta resolución de lo que es esencialmente una técnica de conversión de 1 bit. El primero la técnica es sobremuestreo. El sobremuestreo significa que la señal se muestrea a una velocidad (frecuencia) que es muchas veces mayor que el ancho de banda de interés. Por ejemplo, la tasa de muestreo en el ADE7758 es 1.024 MHz, y el ancho de banda de interés es de 40 Hz a 2 kHz.

El sobremuestreo tiene el efecto de propagar el ruido de cuantificación (ruido debido al muestreo) en un ancho de banda más amplio. Con el ruido se extendió más finamente en un ancho de banda más amplio, el ruido de cuantificación en la banda de interés disminuye, como se muestra en la Figura 36. Sin embargo, el sobremuestreo solo no es lo

suficientemente eficiente como para mejorar la relación señal de ruido (SNR) en la banda de interés. Por ejemplo, un se requiere un factor de sobremuestreo de 4 solo para aumentar la SNR en un solo 6 dB (un bit). Para mantener la relación de sobremuestreo a un razonable nivel, es posible dar forma al ruido de cuantificación para que la mayoría del ruido se encuentre en las frecuencias más altas

En el modulador Σ-Δ, el integrador forma el ruido que tiene una respuesta de tipo de paso alto para el ruido de cuantificación. Esta es la segunda técnica utilizada para lograr una alta resolución. El resultado es que la mayor parte del ruido está en las frecuencias más altas donde el filtro digital de paso bajo lo elimina. Esta configuración del ruido se muestra en la Figura 36.

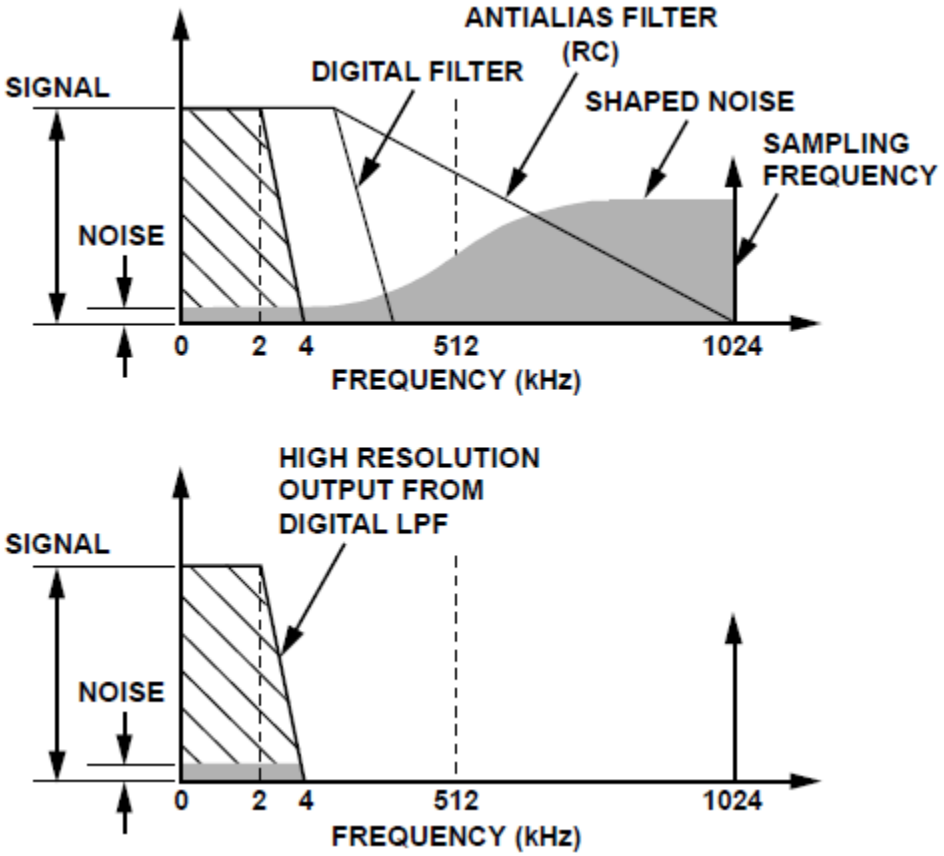


Figura 36. Reducción de ruido debido al sobre muestreo y Modelado de ruido en el modulador analógico

2.4.7 FILTRO ANTIALIASING.

La Figura 37 también muestra un filtro de paso bajo (RC) en la entrada al ADC. Este filtro se coloca fuera del ADE7758, y su función es evitar el aliasing. Aliasing es un fenómeno que afecta los sistemas muestreos como se muestra en la Figura 14. Aliasing significa que los componentes de frecuencia en la señal de entrada al ADC, que son más altos que la mitad de la frecuencia de muestreo del ADC, aparecen en la señal muestreada a una frecuencia inferior a la mitad de la tasa de muestreo. Los componentes de frecuencia que superan la mitad de la frecuencia de muestreo (también conocida como frecuencia Nyquist, es decir, 512 kHz) se visualizan o se vuelven a plegar por debajo de 512 kHz. Esto sucede con todos los ADC independientemente de la arquitectura. En el ejemplo que se muestra, solo las frecuencias cercanas a la frecuencia de muestreo, es decir, 1.024 MHz, se mueven a la banda de interés para la medición, es decir, 40 Hz a 2 kHz. Para atenuar el ruido de alta frecuencia (cerca de 1.024 MHz)

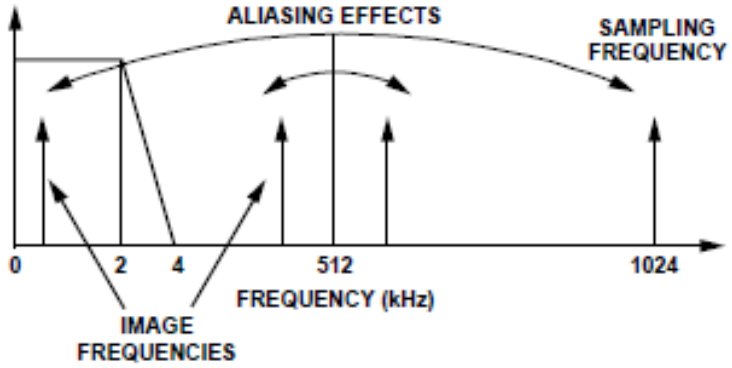


Figura 37. Efecto Aliasing.

y evitar la distorsión de la banda de interés, debe introducirse un filtro de paso bajo (LPF). Para los sensores de corriente convencionales, se recomienda usar un filtro RC con una frecuencia de esquina de 5 kHz para que la atenuación sea suficientemente alta a la frecuencia de muestreo de 1.024 MHz. La atenuación de 20 dB por década de este filtro suele ser suficiente para eliminar los efectos del Aliasing para los sensores de corriente

convencionales. Sin embargo, para un sensor di/dt tal como una bobina Rogowski, el sensor tiene una ganancia de 20 dB por década. Esto neutraliza la atenuación de 20 dB por década producida por LPF. Por lo tanto, al usar un sensor di/ dt, tenga cuidado de compensar la ganancia de 20 dB por década. Un enfoque simple es conectar en cascada un filtro RC adicional, produciendo así una atenuación de -40 dB por década.

2.4.8 CANAL DE CORRIENTE DEL ADC.

La Figura 15 muestra el ADC y la ruta de procesamiento de la señal para la Entrada corriente IA de los canales actuales (es lo mismo para los canales de corriente IB e IC). Las salidas de ADC se complementan dos veces y completan palabras de datos de 24 bits y están disponibles a una velocidad de 8 kSPS (mil muestras por segundo). Con la señal de entrada analógica a escala completa de ± 0.5 V, el ADC produce su valor máximo de código de salida. La Figura 38 muestra una señal de voltaje a escala real aplicada a las entradas diferenciales (IAP e IAN). La salida del ADC oscila entre -5,928,256 (0xA58AC0) y +5,928,256 (0x5A7540).

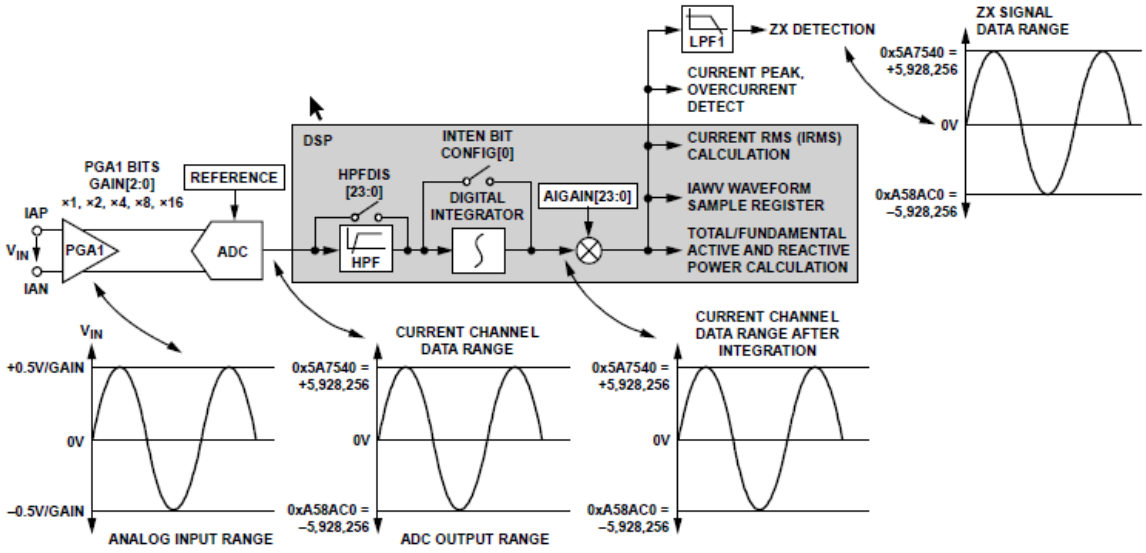


Figura 38. Ruta de la señal de corriente.

- **Registros de ganancia de la corriente.**

Hay un multiplicador en la ruta de señal de cada fase. La forma de onda de la corriente se puede cambiar en ± 100% escribiendo el número correspondiente de dos en dos en los registros de ganancia de la forma de onda de la corriente de 24 bits (AIGAIN, BIGAIN y

CIGAIN). Por ejemplo, si 0x400000 es escrito en esos registros, la producción de ADC se amplía en un 50%. Para escalar la entrada en -50%, escriba 0xC00000 en los registros. La ecuación describe matemáticamente la función de los registros de ganancia de la forma de onda actual.

$$Current\ Waveform = ADCOutput \times \left(1 + \frac{Content\ of\ Current\ Gain\ Register}{2^{23}}\right)$$

Cambiar el contenido de los registros AIGAIN, BIGAIN, CIGAIN o INGAIN afecta todos los cálculos; es decir, afecta la fase correspondiente activa, reactiva, energía aparente y cálculo de rms actual.

- **Canal de corriente HPF.**

El ADC pueden contener un desplazamiento de DC. Este desplazamiento puede crear errores en la potencia y cálculos de RMS. Filtros de paso alto (HPF) se colocan en la ruta de la señal de la fase y las corrientes neutras y de los voltajes de fase. Si está habilitado, el HPF elimina cualquier DC desplazamiento en el canal actual. Todos los filtros se implementan en DSP y, de forma predeterminada, están todos habilitados: el registro HPFDIS de 24 bits se borra a 0x00000000. Todos los filtros están deshabilitados configurando HPFDIS en un valor distinto de cero.

2.4.9 CANAL DE VOLTAJE DEL ADC.

La Figura 39 muestra el ADC y la cadena de procesamiento de señal para la Entrada VA en el canal de voltaje. Los canales VB y VC tienen cadenas de procesamiento similares. Las salidas de ADC se complementan con dos palabras de 24 bits y están disponibles a una velocidad de 8 kSPS. Con la señal de entrada analógica a escala completa de ± 0.5 V, el ADC produce su valor máximo de código de salida. La Figura 16 muestra una señal de voltaje de escala completa que se aplica a las entradas diferenciales (VA y VN). La salida del ADC oscila entre -5,928,256 (0xA58AC0) y +5,928,256 (0x5A7540).

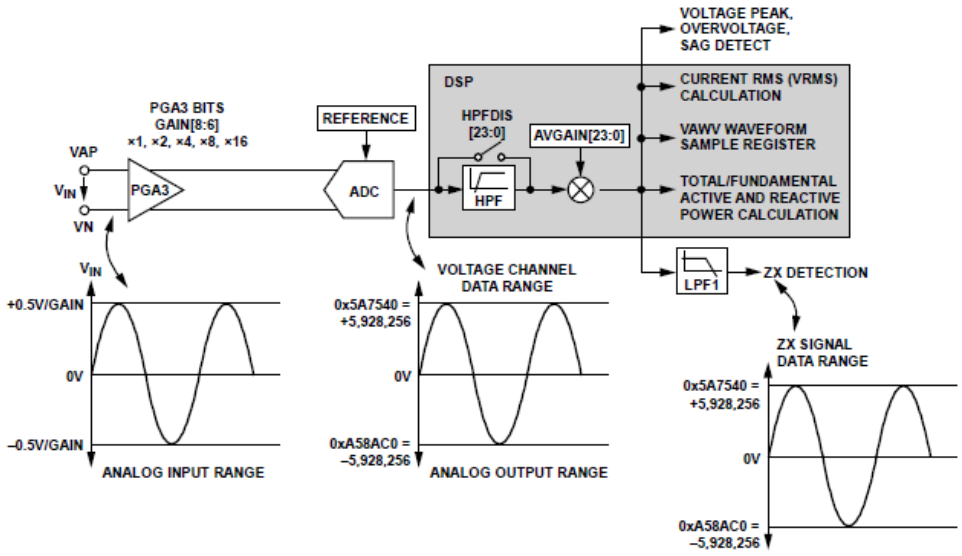


Figura 39. Ruta de la señal de voltaje.

2.5 PROTOCOLO DE COMUNICACIÓN ADE7758.

El ADE7758 tiene integrada una interfaz de comunicación serial, la cual está compuesta de 4 señales: SCLK, DIN, DOUT y $\overline{\mathbf{CS}}$.

***SCLK*:** Señal de reloj. Todas las operaciones de transferencia de datos son sincronizadas con esta señal de reloj.

***DOUT*:** Salida lógica que permite que los datos salgan en el flanco de subida de la señal de reloj.

***DIN*:** Entrada lógica en la cual entran los datos en el flanco de bajada de la señal SCLK.

$\overline{\boldsymbol{CS}}$**:** Entrada lógica que actúa como la selección del integrado. Esta entrada es usada cuando múltiples dispositivos comparten el bus serial. Un flanco de bajada en $\overline{\boldsymbol{CS}}$ coloca al **ADE7758** en modo de comunicación y debe dejarse esta entrada en activo bajo para toda la transferencia de los datos.

Todas las operaciones en el ADE 7758 deben empezar con una escritura al registro de comunicación. El registro de comunicación es un registro de solo escritura de 8 bits. El bit más significativo determina si la siguiente transferencia de datos es lectura o escritura. Si el bit más significativo es 1 indica que la operación que se quiere llevar a cabo es de escritura mientras que si es cero la operación es de lectura. Los siete bits menos significativos contienen la dirección del registro al cual se quiere acceder.

2.5.1 OPERACIÓN DE ESCRITURA SERIAL.

Con el ADE7758 en modo de comunicación y la entrada *chip select* ($\overline{\boldsymbol{CS}}$) en activo bajo, una escritura debe ser realizada al registro de comunicación. El bit más significativo de este byte enviado debe ser ajustado a 1, indicando que la próxima operación es de escritura de datos. El *ADE7758* empieza a leer los datos en el próximo flanco descendente del reloj serial *SCLK* y los bits restantes son leídos en el flanco descendente de los pulsos de reloj posteriores como se observa en la Figura 40. Si ocurre otra transferencia de datos, esta debe terminar como mínimo 900nS después de que haya ocurrido la transferencia del byte anterior.

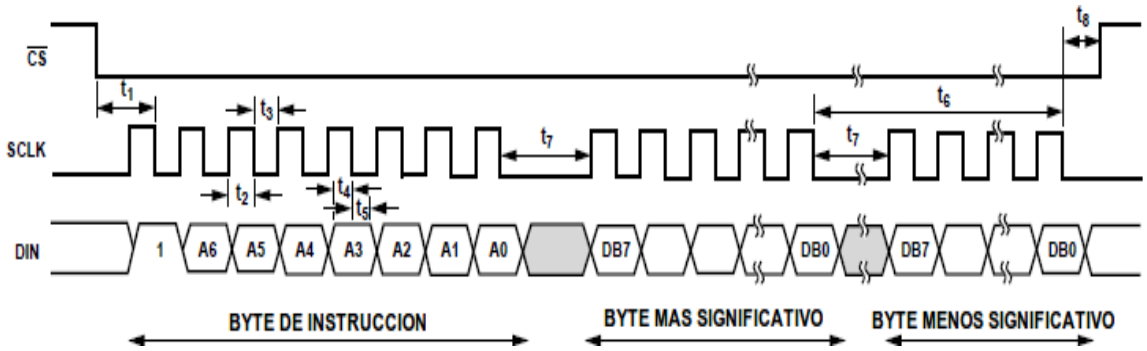


Figura 40. Operación de escritura serial ADE7758.

2.5.2 OPERACIÓN DE LECTURA SERIAL.

Durante una operación de lectura del **ADE7758** los datos son transferidos en el flanco ascendente del reloj serial. Como en el caso de la operación de escritura, una escritura debe ser realizada al registro de comunicación.

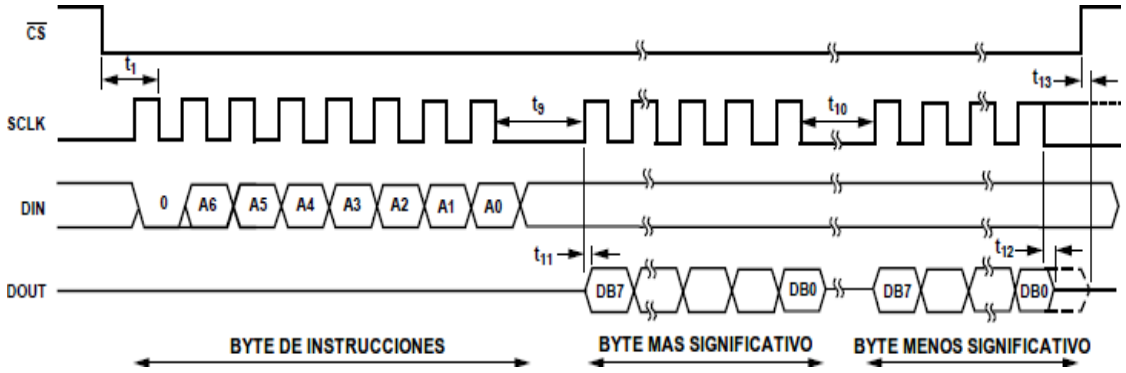


Figura 41. Operación de lectura serial ADE7758.

Con el **ADE7758** en modo de comunicación y **$\overline{\mathbf{CS}}$** en lógico bajo, se realiza una escritura al registro de comunicación con el bit más significativo de este ajustado a 0, indicando que se realizara una operación de lectura de datos. Los siete bits menos significativos contienen la dirección del registro que se quiere leer. En el momento en que el **ADE7758** recibe el último bit del primer byte la salida **DOUT** sale de un estado de alta impedancia y empieza a enviar datos por el bus serial. Cuando una operación de lectura se lleva a cabo, el comando de lectura no debería suceder al menos 1.1us después de la escritura en el registro de comunicación. Lo anterior se aprecia en la figura 41.

2.5.3 CÁLCULO DE POTENCIA ACTIVA ADE7758.

La potencia activa promedio sobre un número de ciclos de línea está dada por la siguiente expresión:

$$P = \frac{1}{nT}\int_0^{nT} p(t)dt = V_{rms} \times I_{rms} \times \cos\alpha$$

Donde α es la diferencia de fase entre la corriente y la tensión y T es el periodo del ciclo de línea.

La señal de potencia instantánea es generada multiplicando las señales de corriente y voltaje en cada fase. La componente DC de la potencia instantánea en cada fase es extraída por LPF2 (filtro pasa bajas).

La potencia activa de cada fase se acumula en el correspondiente registro de 16 bits (*AWATTHR, BWATTHR, o CWATTHR*), la señal de potencia activa obtenida a plena escala en el ADE7758 se representa en la Figura 42.

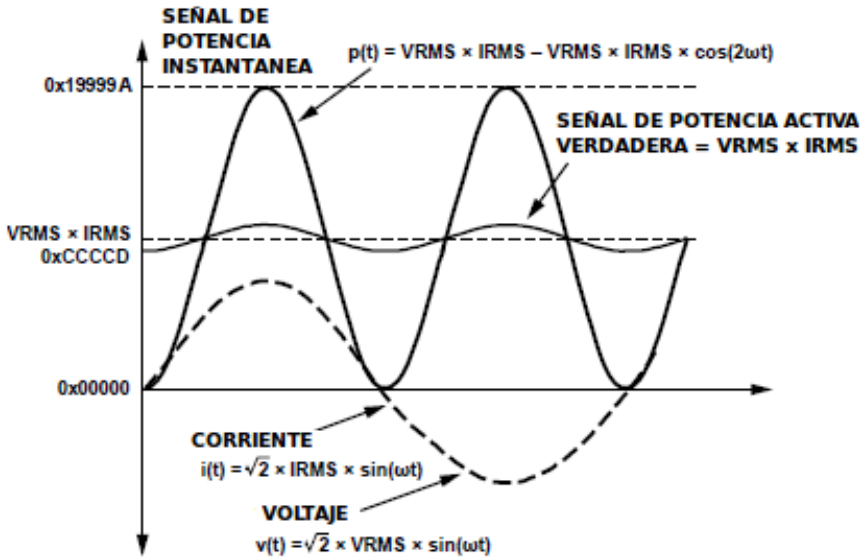


Figura 42. Señal de potencia activa a plena escala en el ADE7758.

El ADE7758 realiza la integración de la señal de potencia activa acumulando continuamente esta señal en registros internos de energía de 40 bits. Los registros *WATTHR (AWATTHR, BWATTHR, o CWATTHR)* representan los 16 bits más significativos de estos registros internos. La acumulación discreta en el tiempo es equivalente a la integración en tiempo continuo. La Figura 43 muestra el esquema del proceso empleado para el cálculo de la energía activa en al ADE7758.

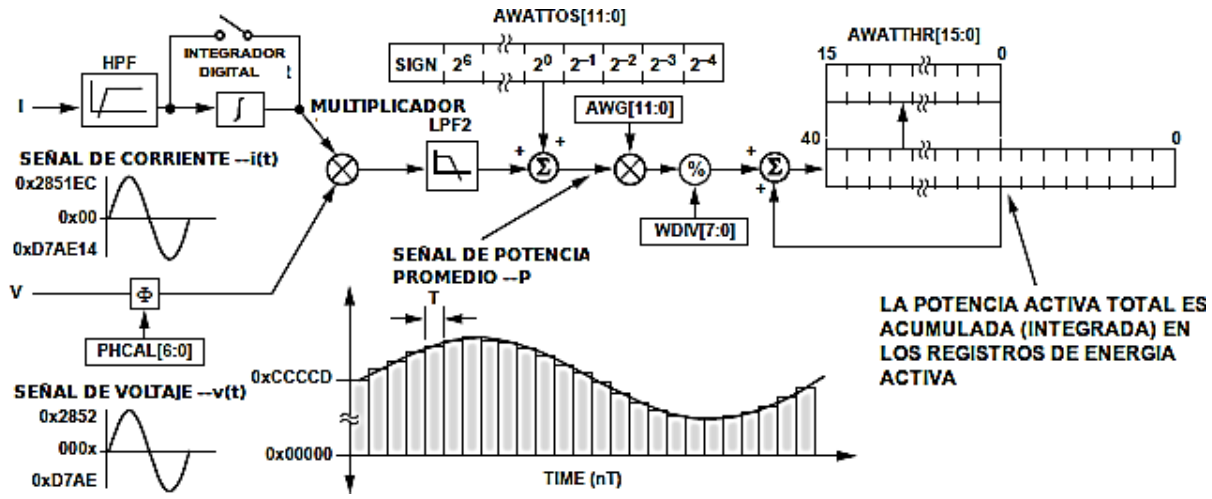


Figura 43. Cálculo de energía activa ADE7758

El promedio de la señal de potencia activa es continuamente agregado al registro interno de energía. Esta adición es una operación con signo. Si la energía es negativa entonces es substraída del registro de energía activa. Los valores que se aprecian en la figura son valores cuando las entradas de tensión y corriente son a plena escala o valores nominales.

La potencia activa promedio es dividida entre el contenido de un registro divisor antes de ser agregado al registro de acumulación correspondiente *WATTHR*. Cuando el valor en el registro *WDIV [7:0]* es 0 ó 1, la potencia activa es acumulada sin división. *WDIV* es un registro sin signo, de 8 bits que es útil para aumentar el tiempo que les toma a los registros de energía en desbordarse.

Con señales de entrada a plena escala el tiempo mínimo que le toma a estos registros en desbordarse depende del registro de ganancia *xWATT GAIN*. Cuando este registro presenta valores de *7FFh,000h y 800h* los tiempos mínimos son 0.13, 0.52 y 0.79 segundos respectivamente.

Se puede activar una interrupción en el *ADE7758* para que avise cuando un registro de acumulación de energía está lleno hasta la mitad, con el fin de leerlo antes que este registro se desborde y se pierdan los datos. Otra forma de leer estos registros es activar el bit RSTREAD con el fin de que estos registros vuelvan a cero después de una lectura.

El periodo de muestreo en tiempo discreto para la acumulación de energía es 0.4us (4/*CLKIN*), por lo tanto, se toman 2500 kSPS (kilo muestras por segundo). Si se supone una carga estable, el tiempo mínimo que transcurre antes que los registros de energía se desborden está dado por la siguiente ecuación:

$$Tiempo_{minimo} = \frac{0xFF,FFFF,FFFF}{0xCCCCD} \times 0.4\ \mu S = 0.524\ segundos$$

Donde 0x*FF,FFFF,FFFF* es el valor máximo que puede almacenar el registro interno de energía y 0x*CCCCD* es la máxima salida del filtro pasa bajas.

Para el cálculo de la potencia activa se divide el contenido del registro de energía entre el tiempo de acumulación de este registro, dicho tiempo se calcula de la siguiente forma:

$$Tiempo_{acumulacion} = Tiempo_{minimo} \times WDIV[7:0]$$

Donde es el tiempo de acumulación cuando el registro de ganancia es y que corresponde a 0.52 segundos, y *WDIV[7:0]* es el factor de escalamiento el cual puede ser de 255 veces máximo.

El *ADE7758* también provee un pin de salida (*APCF*) capaz de entregar información de la energía activa a través de pulsos de frecuencia, los cuales son proporcionales a la energía activa medida.

2.5.4 CÁLCULO DE POTENCIA REACTIVA ADE7758.

La potencia reactiva promedio sobre un número de ciclos de línea está dada por la siguiente expresión:

$$Q = \frac{1}{nT}\int_0^{nT} q(t)dt = V_{rms} \times I_{rms} \times \sin\alpha$$

Al igual que para la potencia activa, la componente DC de la señal de potencia reactiva instantánea en cada fase es extraída por un filtro pasa bajas. Este proceso es ilustrado en la figura 44. La potencia reactiva de cada fase es acumulada en el registro correspondiente de 16 bits *(AVARHR, BVARHR, CVARHR).*

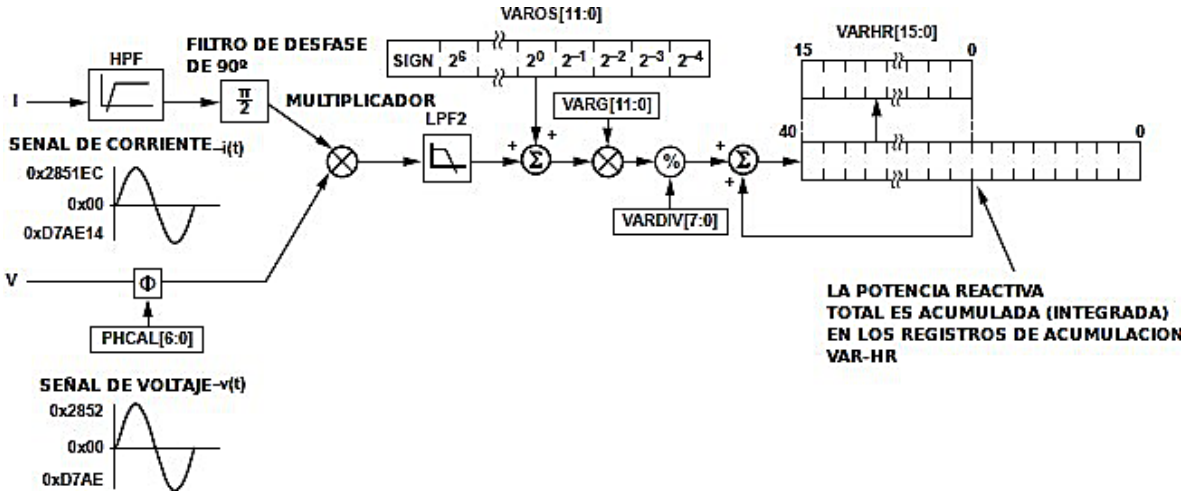


Figura 44. Cálculo de energía reactiva ADE7758.

La señal de potencia reactiva es acumulada constantemente en los registros internos de energía. Esta adición es una operación con signo por lo tanto energías negativas son substraídas. La potencia activa promedio es dividida entre el contenido de un registro divisor antes de que esta sea agregada al registro de energía con el fin de aumentar el tiempo en el cual se desborda el registro de energía.

El pin 17 (*VARCF*) de *ADE7758* es una salida que emite pulsos proporcionales a la energía reactiva total.

2.5.5 CÁLCULO DE POTENCIA APARENTE ADE7758.

ADE7758 usa el método de aproximación aritmética para calcular la potencia aparente.

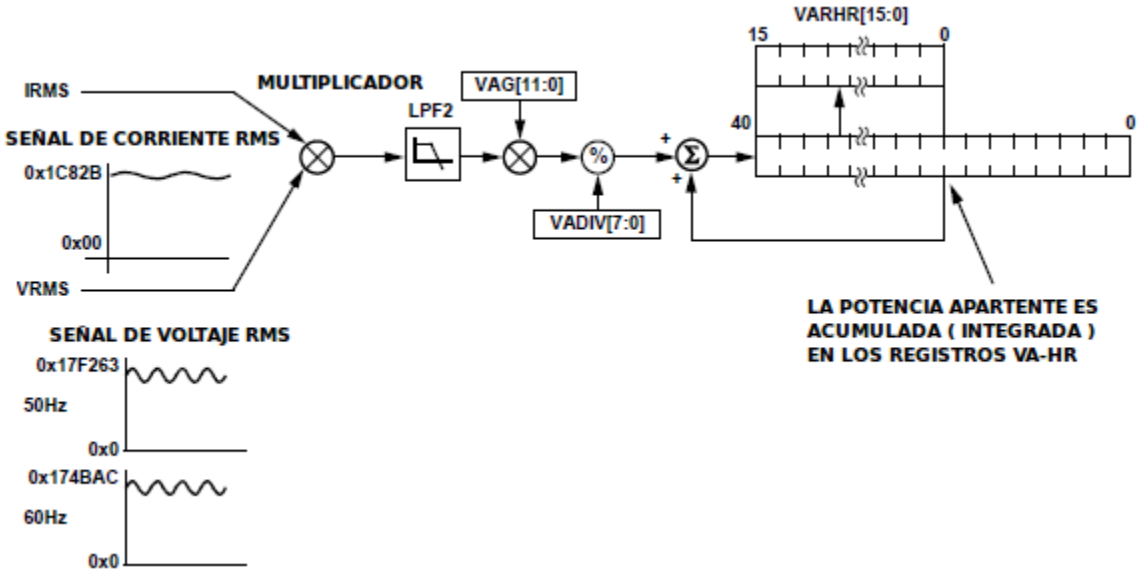


Figura 45. Cálculo de potencia aparente ADE7758.

Los valores eficaces de la tensión y la corriente son multiplicados en cada fase para producir la potencia aparente de la correspondiente fase. La salida del multiplicador pasa por un filtro pasa bajas para obtener la potencia aparente promedio.

A diferencia de la potencia activa y reactiva, no existen registros para compensar el *offset* existente en canal de potencia aparente. Esto se debe a que la compensación de *offset* que se le hace a los registros de tensión y corriente es suficiente.

El proceso para calcular la energía aparente es el mismo explicado anteriormente para la energía activa y reactiva. El único aspecto diferente es el tiempo mínimo de acumulación de energía, el cual es mayor en este caso y se muestra a continuación:

$$Tiempo_{mínimo} = \frac{0x1\text{FFFFFFFFF}}{0x\text{B9954}} * 0.4\mu s = 1.157\ segundos$$

Donde 0xB9954 es la máxima salida del filtro.

Utilizando el registro *VADIV* se puede aumentar el tiempo de acumulación de la energía aparente en los registros respectivos.

Ajustando el bit 7 en el registro *WAVMODE* se asegura que la salida de pulsos en el pin 17 muestra información acerca de la potencia aparente y no de la potencia reactiva.

2.5.6 CÁLCULO DE CORRIENTE ADE7758.

La figura 46 muestra el proceso para calcular el valor la tensión *rms* por parte del *ADE7758* para una fase. El mismo procedimiento es usado para las dos fases restantes.

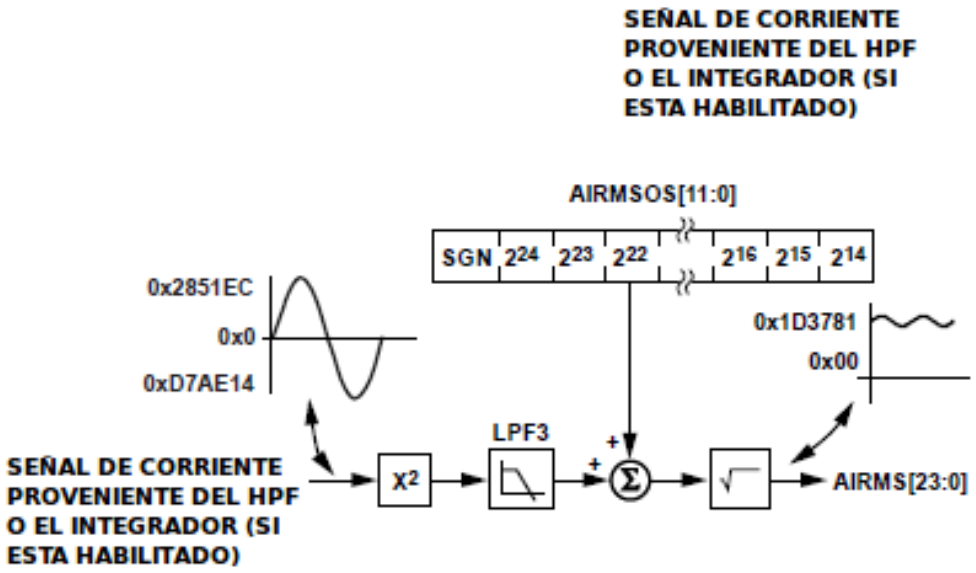


Figura 46. Cálculo de corriente ADE7758.

El filtro LPF3 extrae el promedio de la señal de corriente al cuadrado, la señal resultante se suma con un registro que corrige el *offset* (*AIRMSOS*), para después extraer la raíz cuadrada y guardar el resultado en el registro de 24 bits *AIRMS*.

Con las señales de entrada a plena escala, el conversor produce un código de salida aproximado de 1D3781h. El ADE7758 tiene registros capaces de remover el *offset* presente en cualquiera de las fases (*AIRMSOS, BIRMSOS, CIRMSOS*). Un *offset* puede existir debido a ruidos de entrada que son integrados en la componente DC cuando se eleva al cuadrado la corriente.

2.5.7 CÁLCULO DE TENSIÓN ADE7758.

Con las señales análogas de entrada a plena escala (0.5V), el filtro LPF1 produce un código de salida aproximado de +/-9,372d a 60 Hz. Posteriormente esta salida es elevada

al cuadrado y pasa nuevamente por un filtro pasa bajas LPF3 con el fin de extraer el valor promedio de la señal de tensión. Luego es extraída la raíz cuadrada y finalmente se suma la señal resultante con un registro que corrige el offset que se presente en el canal de tensión de cada fase. El registro xVRMSGAIN[8] es usado para escalar las salidas de los conversores A/D en +/- 50%.

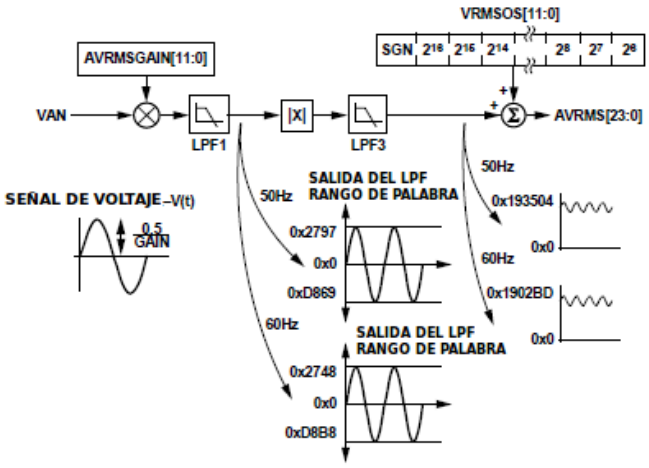


Figura 47. Cálculo de tensión ADE7758.

El error típico en la medición de la tensión RMS es de 0.5% y esta medición tiene un ancho de banda de 260Hz.

2.5.8 INTERRUPCIONES.

Las interrupciones en el *ADE7758* son manejadas a través del registro de estado de interrupción, (*STATUS [23:0],* dirección 19h) y del registro mascara de interrupción (*MASK [23:0],* dirección 18h). Cuando un evento de interrupción ocurre en el *ADE7758*, la bandera correspondiente en el registro estado de interrupción cambia a 1 lógico. Si el bit de mascara para esta interrupción es un 1 lógico, entonces la salida lógica pasa a activa bajo. Para determinar la fuente de la interrupción, el microcontrolador debe realizar una lectura del registro *RSTATUS*. Después de realizada la lectura del registro la salida vuelve a su estado normal activo alto.

[8] "x" hace referencia a la fase A, B o C.

2.6 ESQUEMAS ELECTRICOS

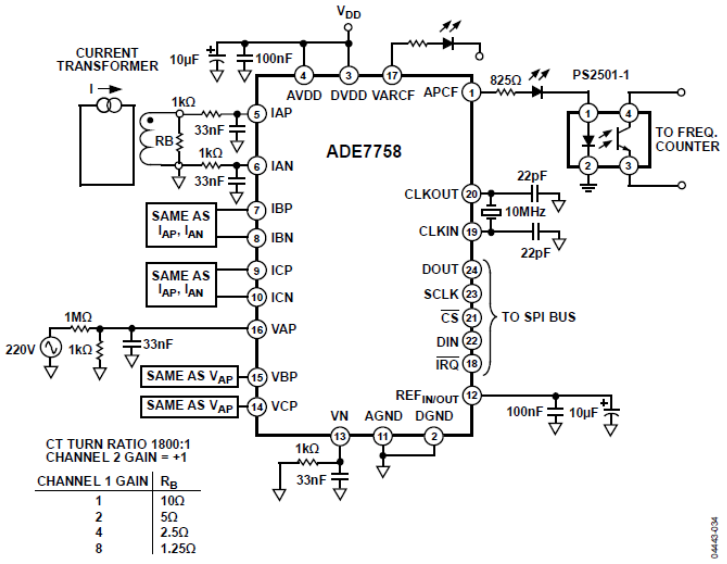


Figura 48 Esquema básico de conexión IC Medidor de Energía ADE7758.

Con lo ya expuestos del sistema de multiplexores en la Sección 2.3, y lo detallado en el esquema básico de conexión para el circuito medidor de energía del fabricante Analog Devices ADE7758 en la Sección 2.4 el cual podemos ver en la Figura 48 y lo mostrado de la comunicación SPI para el ADE7758 en la Sección 2.5, se presentan a continuación los esquemas eléctricos.

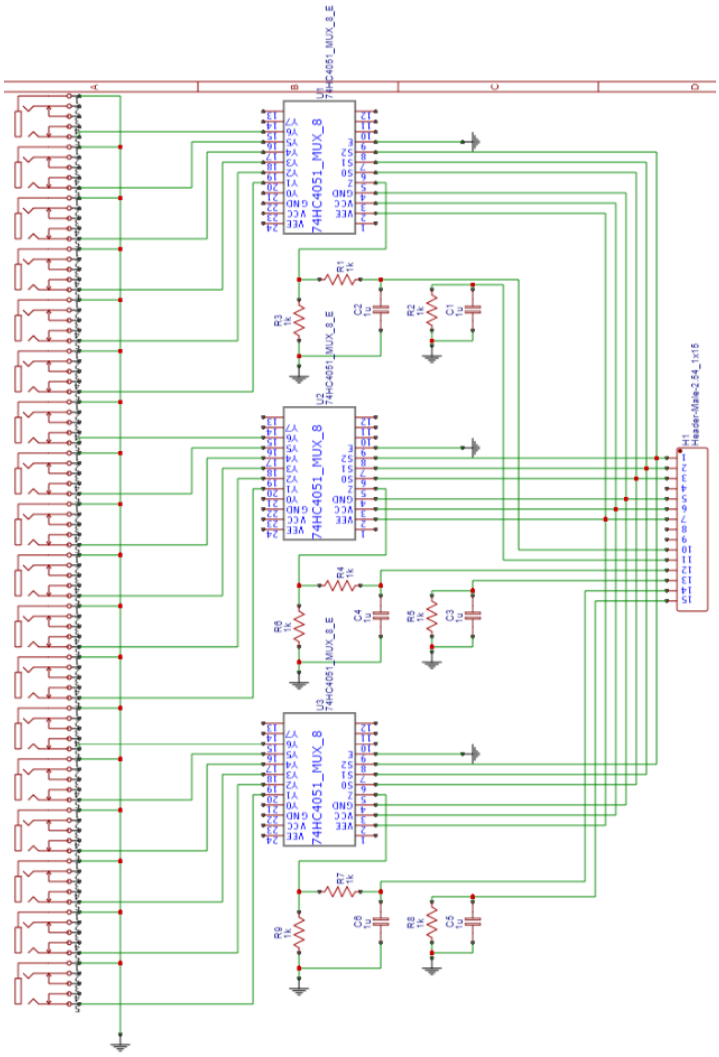


Figura 49. Etapa de Multiplexado de señales de corriente.

En la Figura 49 observamos la Etapa de multiplexado para las señales de corriente provenientes de los sensores de corrientes expuestos en la sección 2.1, podemos observar 3 circuitos multiplexores de los 6 disponibles, cada multiplexor representa una fase de corriente, para cada uno de ellos se ha elegido el circuito integrado **74HC4051,** el cual se eligió por su excelentes características de multiplexados de señales analógicas y digitales , este se muestra en la Figura 50.

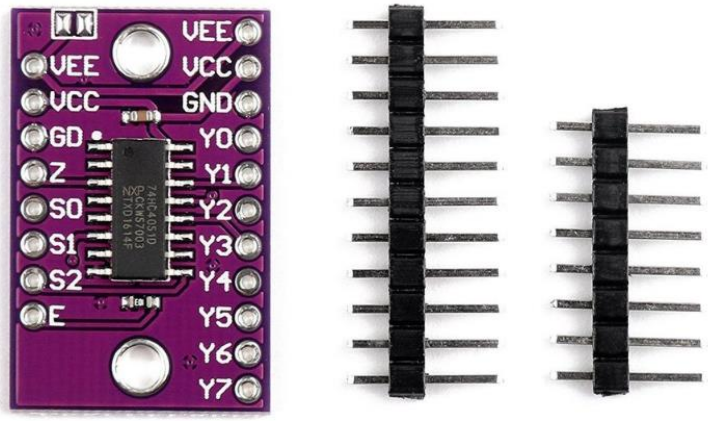


Figura 50. Circuito Integrado 74HC4051.

El sistema de multiplexado similar al de la Figura 49 , es utilizado para las señales de voltaje , dichas señales provienen de los divisores de tensión expuestos en la Sección 2.2.2, en el proceso de multiplexado de las 36 señales de entrada se obtienen 6 salidas de ellas por todo el circuito de multiplexado una por cada fase de voltaje y corriente, los cuales son enviadas a la entrada de señales del circuito integrado medidor de energía ADE7758, este procesa las señales y entrega los resultados a través de la comunicación SPI que es enviada al Arduino que el cual posee el mismo protocolo de comunicación denominado Chip to Chip, los datos son extraídos desde la plataforma Arduino IDE, instalado en Raspbian (sistema operativo basado en Linux , para hardware Raspberry Pi) utilizando el puerto USB , siendo presentados dichos datos en una interfaz gráfica desarrollada con el interprete Python. Lo descrito se presenta en el esquema eléctrico en la Figura 51.

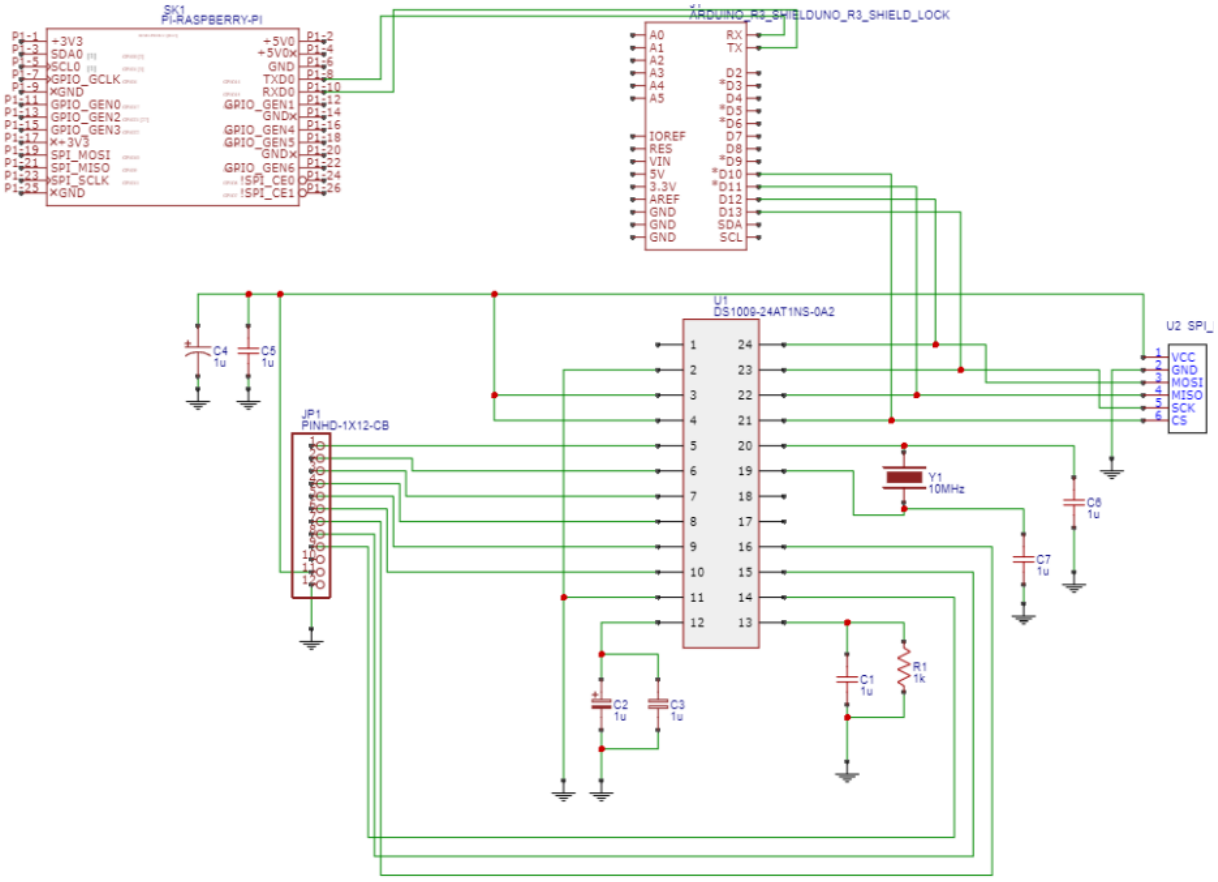


Figura 51. Esquema eléctrico de la etapa ADE7758 - Arduino- Raspberry Pi.

2.7 DISEÑO DE CIRCUITOS PCB CON EASY EDA.

Para el diseño de los circuitos PCB o comúnmente llamados Tarjetas de Circuitos impresos se utiliza la herramienta EasyEDA el cual brinda el servicio de fabricación de mayor calidad, con bajo costo de producción, pero con costos de envío a considerar, ya que estos son fabricados en China.
EasyEDA es una herramienta gratuita, que no requiere instalación y es una aplicación basada en la nube, diseñada para proporcionar a ingenieros electrónicos, educadores, estudiantes de ingeniería y aficionados a la electrónica una experiencia sencilla. Permite un sencillo diseño de circuitos, simulación y diseño de PCB desde su navegador.

A continuación, se describe las funcionalidades de la herramienta EasyEDA.

2.7.1 DISEÑO DE ESQUEMAS.

Se pueden dibujar esquemas rápidamente en el navegador usando las bibliotecas disponibles, este posee la ventaja que se va actualizando los dispositivos acorde a como la tecnología de fabricación de circuitos integrados va en crecimiento.

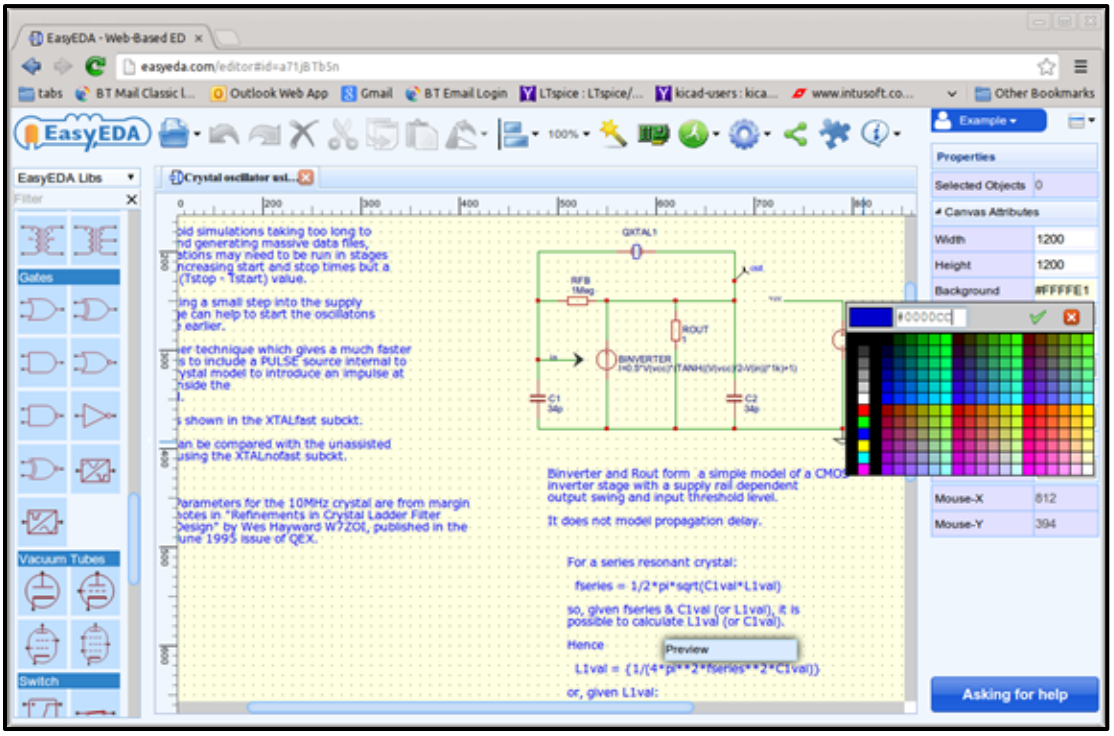


Figura 52. EasyEDA Layout.

2.7.2 SIMULADOR DE CIRCUITOS.

Podemos verificar mediante la simulación para circuitos analógicos, digitales y de señal mixta con subcircuitos y modelos spice, tal como podemos observar en la Figura 53.

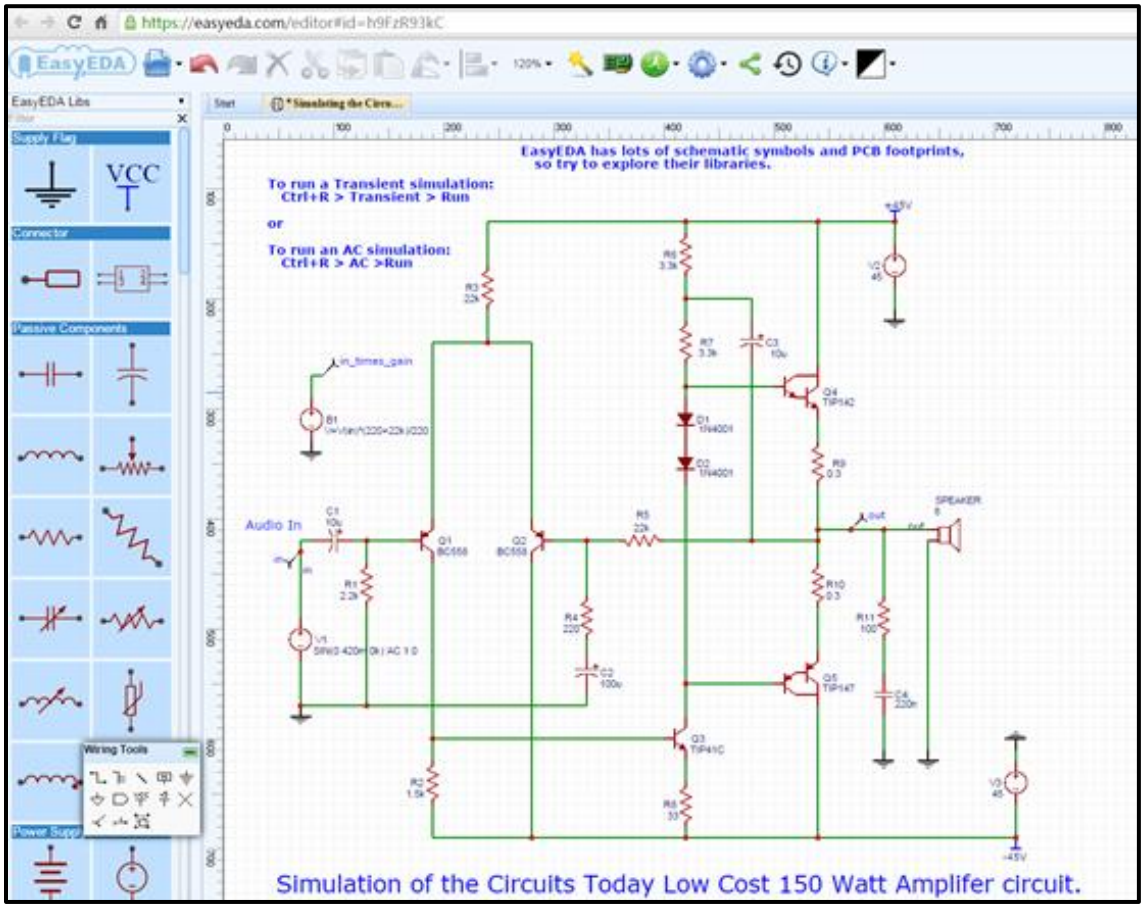


Figura 53. Desarrollo de circuitos esquemáticos para su simulación en línea.

2.7.3 DISEÑO DE TARJETAS DE CIRCUITOS IMPRESOS (PCB) EN LINEA.

Podemos diseñar fácilmente con esta herramienta, circuitos de una o dos capas. El sistema es muy estable, fiable y fácil de aprender. El interfaz de usuario es muy agradable y de respuesta ágil. EasyEDA tiene una amplia biblioteca de miles de componentes electrónicos (tanto de circuitos y circuitos impresos como para modelado), y decenas de miles de ejemplos de esquemas. Cualquiera puede usar esa biblioteca y ampliarla. También se pueden importar diseños existentes hechos en Altium, Eagle y KiCad y editarlos en EasyEDA.

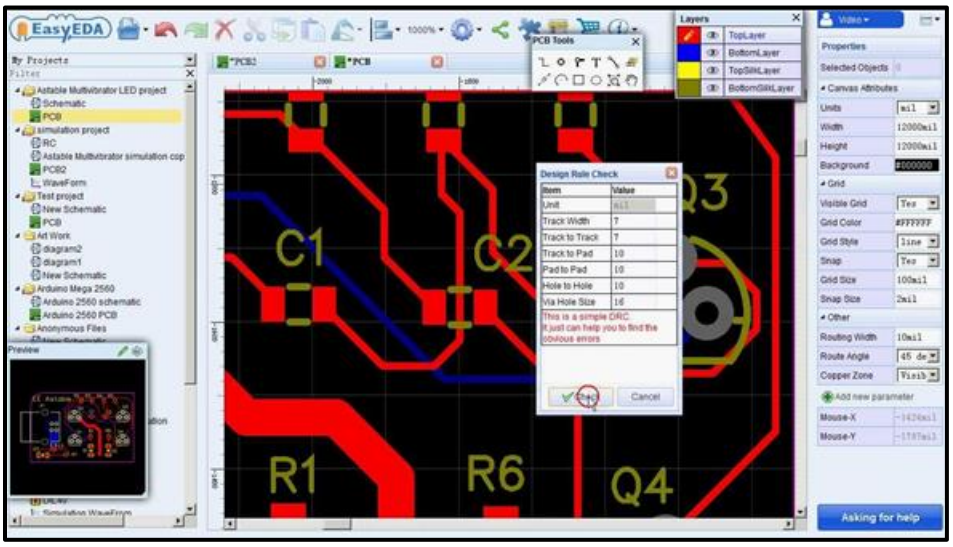


Figura 54. Diseño de PCB utilizando Easy EDA.

Además, otra característica destacada es que los usuarios tienen acceso a módulos Open Source desarrollados por miles de ingenieros electrónicos.

2.7.4 DESARROLLO DE PROYECTOS EN EASYEDA.

Ingresaremos a EasyEDA con el correo de Google dando clic en login, si eres un usuario nuevo daremos clic en Register y sigue los pasos que se te piden para crear la cuenta.

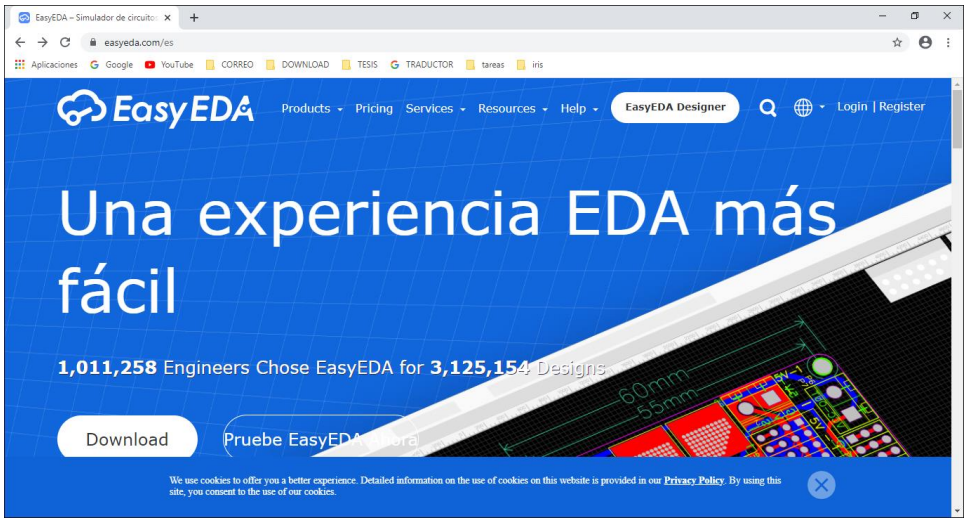


Figura 55. Página de inicio de EasyEDA.

Iniciaremos cesión con la cuenta de correo de Google.

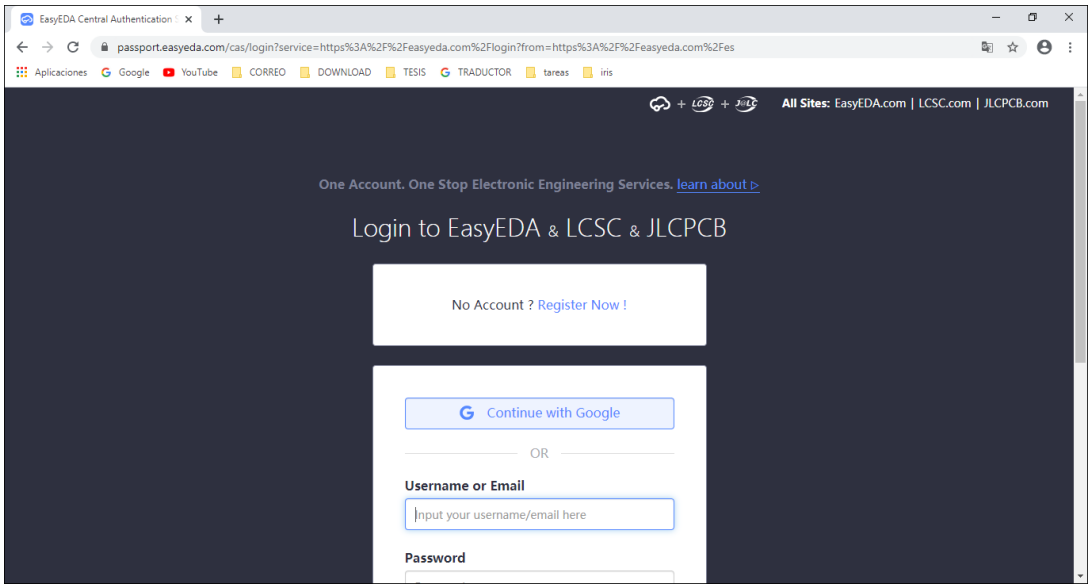


Figura 56. Registro de EasyEDA.

En la Figura se muestra los proyectos desarrollados o que se están desarrollando.

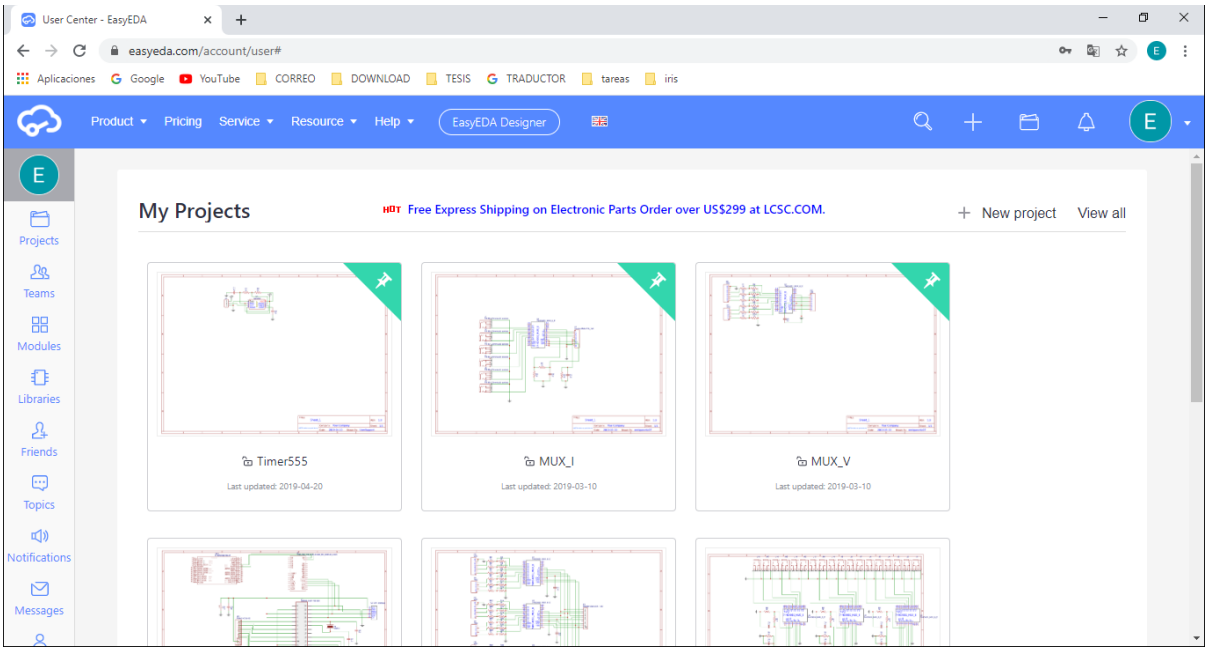


Figura 57. Proyectos en EasyEDA.

Iniciamos un nuevo proyecto llevaremos el puntero a Product se desplegará un menú, clic en online Editor, se nos abrirá una nueva pestaña en la que trabajaremos.

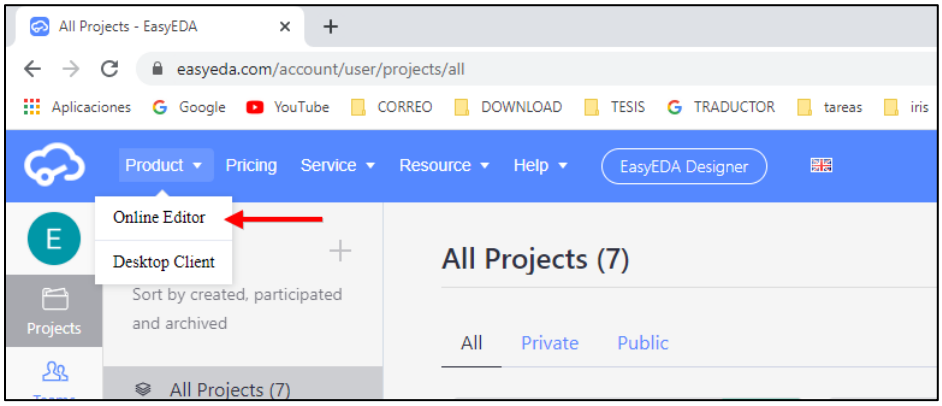


Figura 58. Proyecto EasyEda.

Comencemos a hacer nuestro diagrama esquemático con EasyEDA.
Para comenzar con el esquema en EasyEDA, necesitamos tener nuestro panel de esquema. Para hacer esto, haga clic en el icono "carpeta", haga clic en "Nuevo" y podrá ver que hay muchas opciones sobre qué proyecto desea hacer. Haga clic en "esquema" para continuar con el panel de esquema.

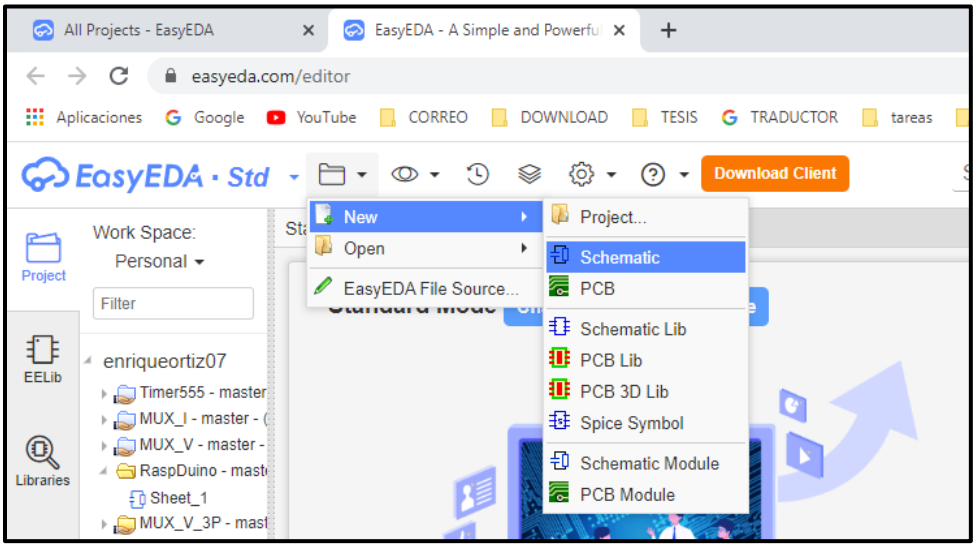


Figura 59. Inicio de nuevo proyecto en EasyEDA.

Para comenzar con la adición de componentes, haga clic en el botón "EElib" y obtendrá los componentes con su símbolo esquemático. Si no desea desplazarse uno por uno por el componente para ver qué busca, puede filtrar la categoría que busca en el cuadro de filtro. Si tiene dificultades para encontrar el componente, simplemente haga clic en el botón "partes" y busque todos los componentes disponibles.

Figura 60. Elementos fundamentales en la plataforma de EasyEDA.

Podemos organizar los componentes y reemplazar los valores de sus componentes. Para cambiar los valores, se hace doble clic en el valor e se ingresa nuevamente el valor deseado.
A continuación, un ejemplo para agregar un componente:
Hacer clic en EElib para mostrar los componentes. Hacer clic en la resistencia de EElib y hacer clic en el panel esquemático para agregar resistencia.

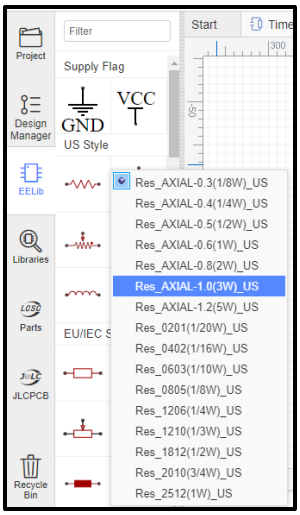


Figura 61. Adición de componentes.

Si no puedo encontrar el circuito integrado en la librería EElib. Se busca el componente en el botón "partes" y en la barra de búsqueda se coloca el elemento a buscar y verán los resultados disponibles. Se muestran muchas opciones, se elige una que corresponda al circuito deseado y se hace clic en el panel esquemático.

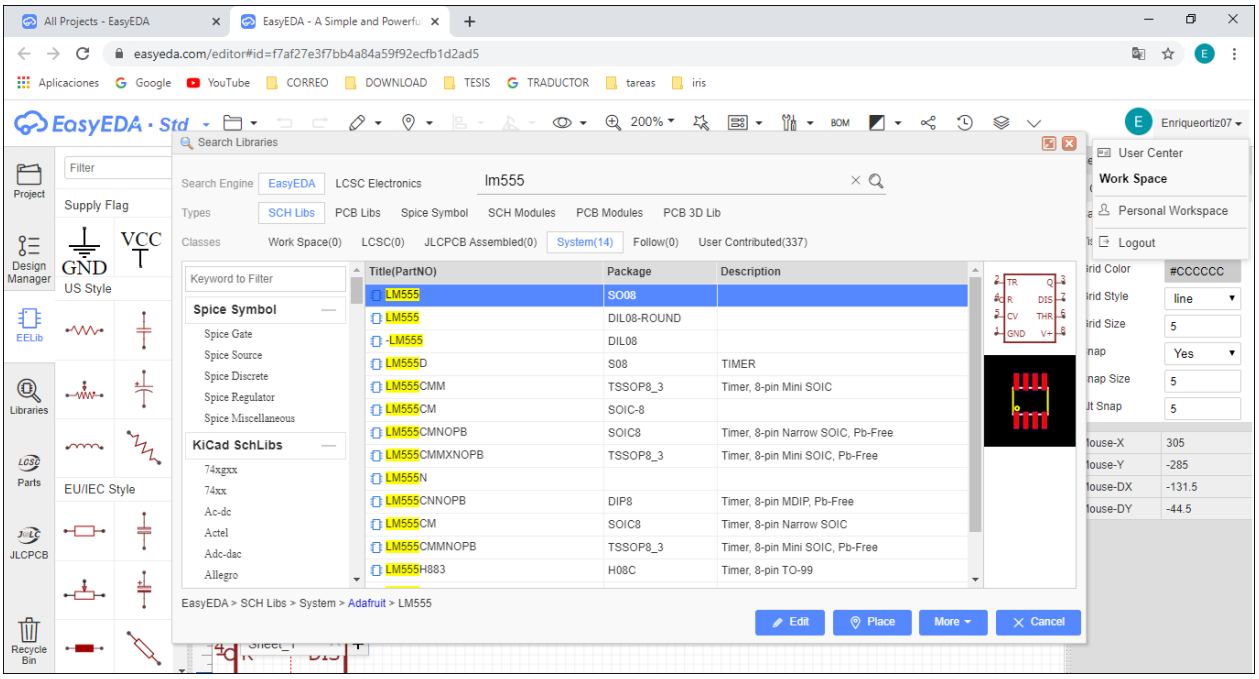


Figura 62. Buscando circuitos integrados.

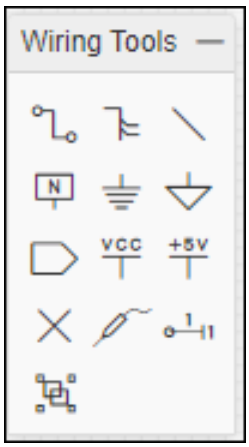


Figura 63. Herramientas de cableado.

Wire.
Hay tres formas de ingresar al modo de cable en EasyEDA. Haga clic en el botón Cablear de la paleta Herramientas de cableado. Presione la tecla de acceso rápido W. Haga clic en el extremo de un pin de componente (donde aparece el punto de pin gris si selecciona el componente):

Bus.
Cuando diseñe un esquema profesional, tal vez usará muchos cables. Si realiza el cableado uno por uno, se desperdiciará mucho tiempo y luego deberá usar el Bus

NetLabel.
Se utiliza usar para dar nombres a sus cables para ayudarlo a encontrarlos e identificar cualquier conexión incorrecta. Puede encontrar NetLabel en la paleta Herramientas de cableado o utilizando la tecla de acceso rápido N. Al seleccionar la etiqueta de red, encontrará sus atributos en el panel de propiedades de la derecha:

NetFlag.
NetFlag es lo mismo que NetLabel, puede encontrar el NetFlag en la paleta Herramientas de cableado o utilizando las teclas de acceso rápido Ctrl + G para GND o Ctrl + Q para VCC. También puede cambiar su nombre, por ejemplo, de VCC a VDD:

NetPort.
En EasyEDA, Net Port funciona como Net Label, no diferencia el puerto neto de entrada y salida. Cuando no desee enrutar demasiados cables.

No Connect Flag.

Puede encontrar el indicador NO Connect a través de la herramienta de cableado En el siguiente esquema, si no agrega un indicador NO Connect, hay un indicador de error en la colección de redes del administrador de diseño.

Voltaje Prueba.
EasyEDA proporciona una función de simulación para el esquema. Después de que se ejecute la simulación, verá la forma de onda donde colocó las sondas de voltaje en el circuito.

Pin.
Cuando crea un nuevo símbolo en lib esquemático y esquemático, debe usar Pin para crear pines para el nuevo símbolo; de lo contrario, su símbolo no se puede conectar con cables.

Group/Ungroup.

En la paleta Herramientas de cableado está el botón Agrupar o Desagrupar símbolo.

El esquemático de un ejemplo se muestra en la Figura 64 a continuación.

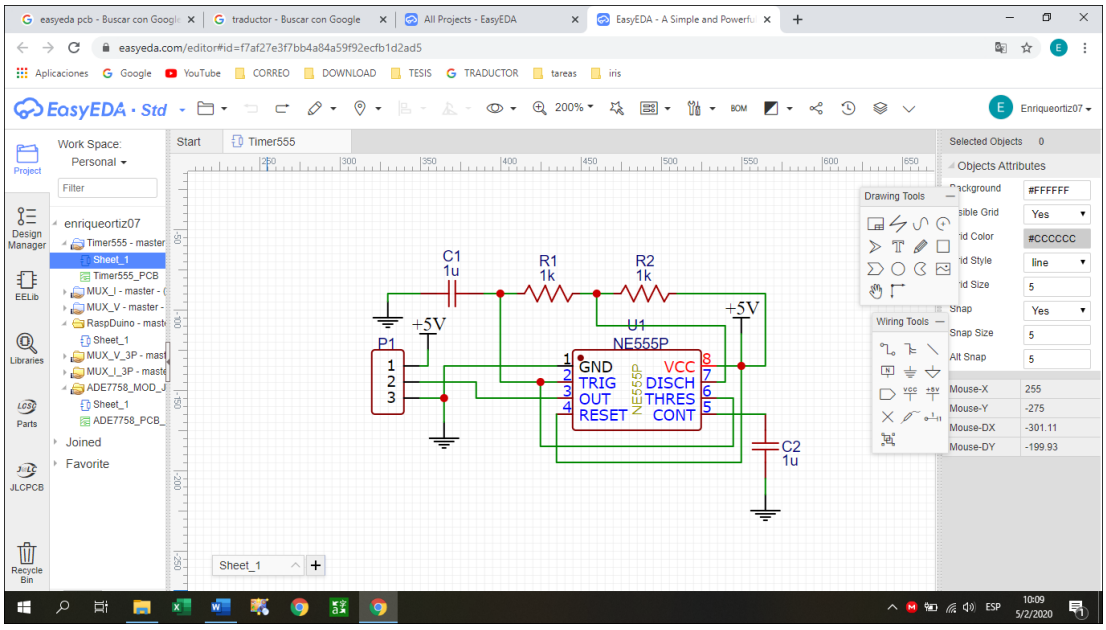


Figura 64. Ejemplo de un esquemático en EasyEDA.

Daremos un clic en convert to pcb.

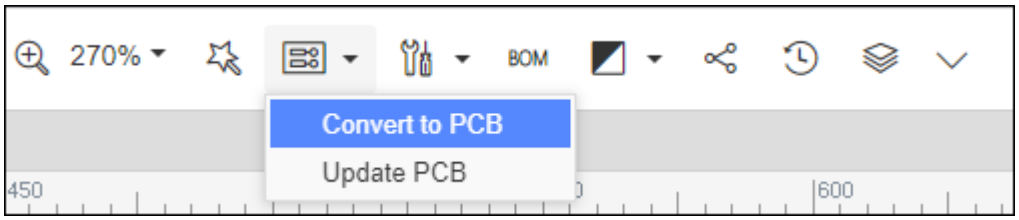


Figura 65. Diseño de esquematico con las herramientas de EasyEDA.

El diseño de PCB debe convertirse del esquema. Ahora, hay una nueva pestaña y un nuevo conjunto de herramientas para dibujar la PCB.

Hay dos dispositivos, uno son las herramientas de PCB y el otro es para capas de PCB. El ícono de lápiz al lado de la capa significa que esta es la capa utilizada actualmente para la herramienta utilizada activa (cable, texto, etc.). Al presionar el botón de engranaje, puede cambiar el color de la capa y habilitar o deshabilitar algunas capas adicionales.

Se nos mostrara la siguiente ventana, aquí trabajaremos dentro del cuadro de color morado ordenaremos los elementos del circuito.

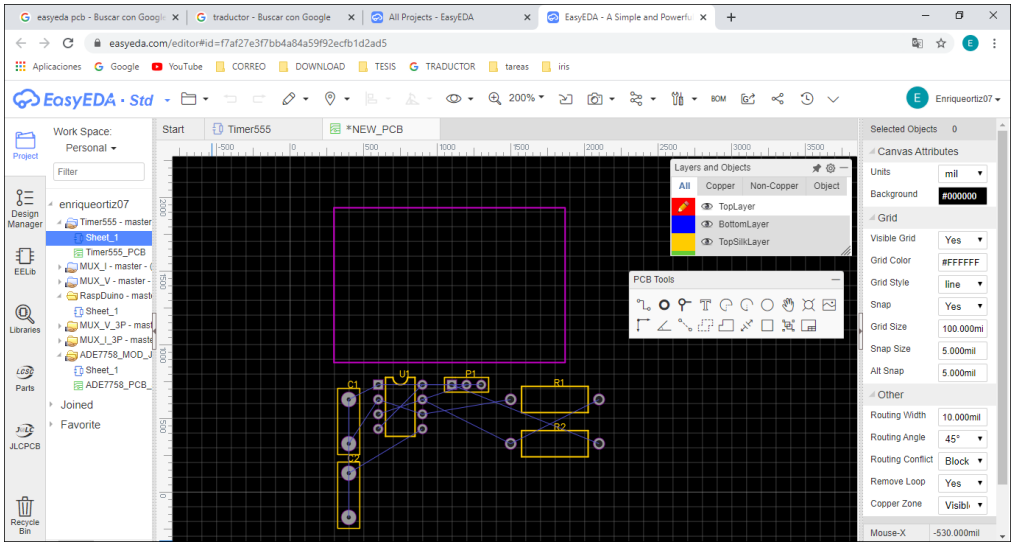


Figura 66. Layout de trabajo PCB en EasyEDA.

Lo primero que debe hacer después de crear la PCB es hacer la colocación correcta para los componentes. Para este propósito, una práctica herramienta llamada "Sonda cruzada" toma las mismas partes seleccionadas del esquema y las coloca de la misma manera que el esquema en el editor de PCB. Vea esta característica en acción en el registro adjunto. Después de la colocación de los componentes, comienza el paso de dibujar trazas.

Notas importantes:
Recuerde siempre después de realizar cualquier cambio en el esquema para actualizar la PCB. Las modificaciones en el esquema no se convierten a PCB sin la solicitud de actualización del usuario.
2- Como dijimos anteriormente, recuerde guardar su diseño. Los cambios no se guardan automáticamente.

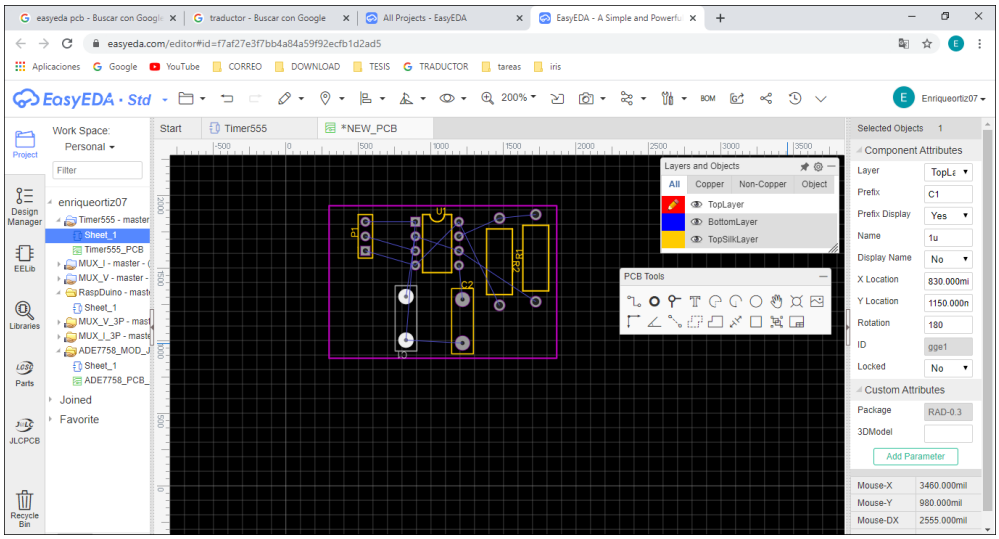


Figura 67. Herramienta Autoruter de EasyEDA.

Autorouter.

Al igual que cualquier otra herramienta de enrutador automático, debe establecer alguna configuración antes de la etapa de enrutamiento y luego ejecutarla.

Tiene dos opciones para usar el enrutador automático en línea o descargar la herramienta en su PC.

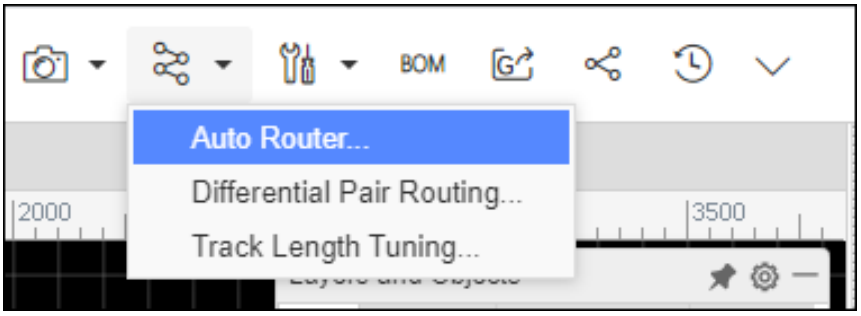


Figura 68. Menú AutoRouter.

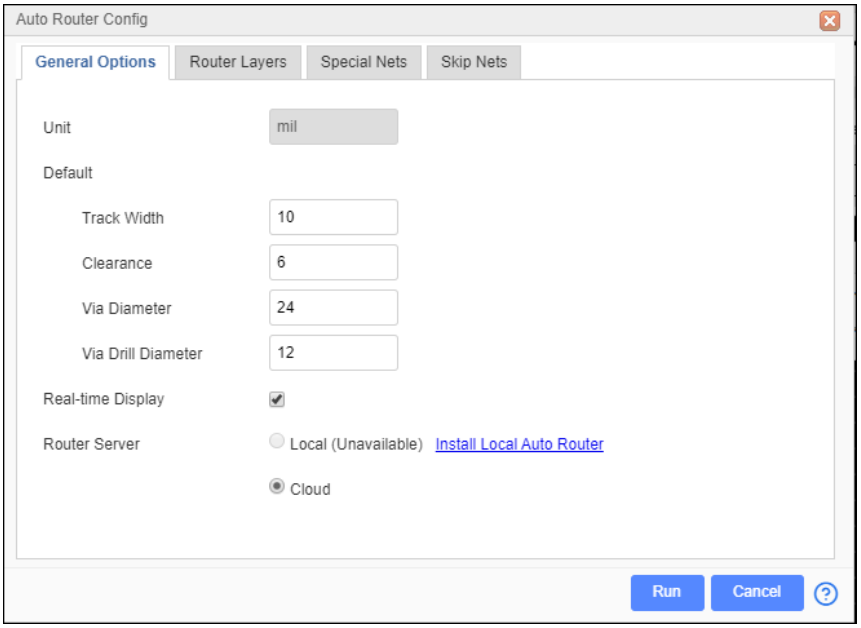


Figura 69. Menú selección de doble cara y dimensiones.

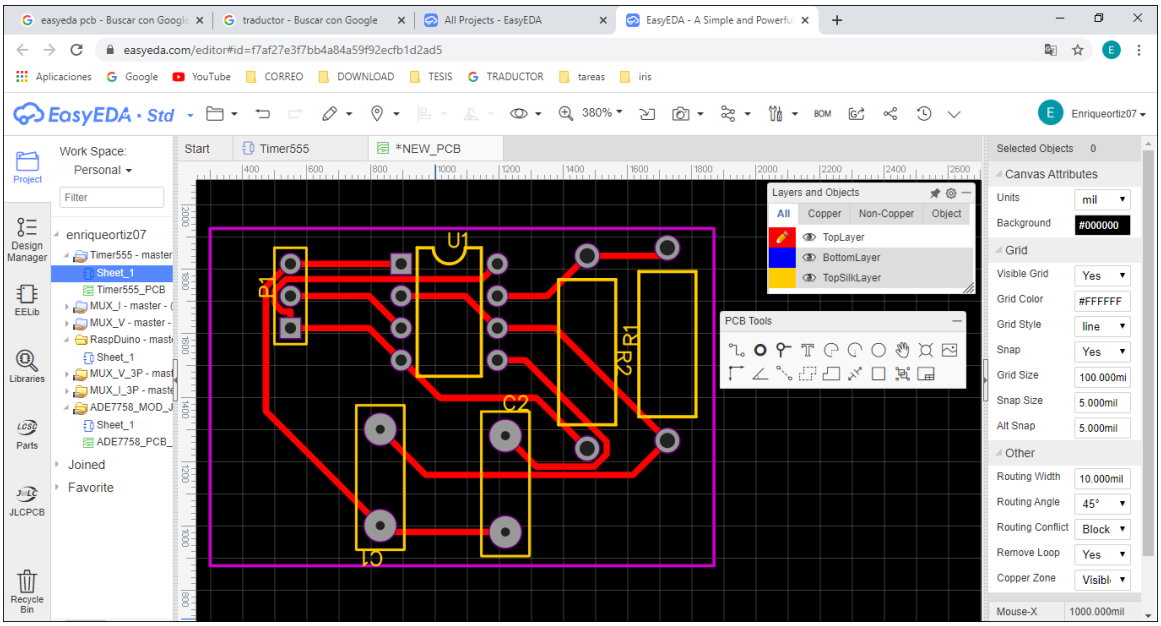


Figura 70. Finalización del PCB.

El mercado de la electrónica tiende a proporcionar soluciones integrales. Con esta herramienta se pueden solicitar la lista de materiales y fabricar el PCB por un socio o una compañía hermana. EasyEDA forma parte de otro grupo compuesto por un distribuidor de piezas electrónicas llamado LCSC y una fábrica de PCB llamada JLCPCB.

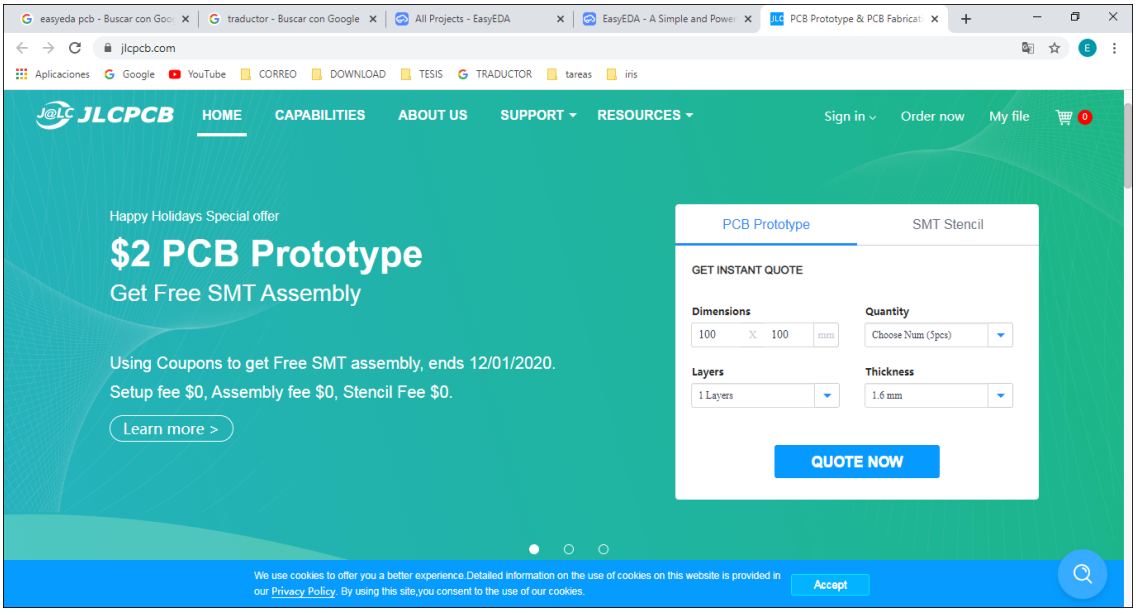


Figura 71. Pedido del PCB.

Con un clic puede solicitar la lista de materiales de LCSC. Además, cuando agrega una parte a su diseño, puede agregar partes que ya están disponibles en el stock de LCSC, que es una característica muy útil.

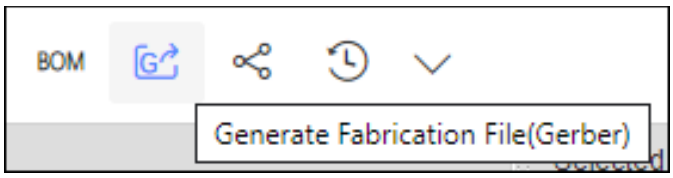


Figura 72. Archivo Gerber.

Además, cuando desee generar el archivo Gerber desde el editor de diseño. Se abrirá una nueva pestaña del navegador para descargar los archivos Gerber y ofrecerle la fabricación de la PCB utilizando el servicio JLCPCB con una herramienta de cotización muy útil.

Tienen una oferta interesante y económica llamada "Gran caída de precios". Por $ 2 puede obtener 10 piezas para PCB con 2 capas y con tamaños inferiores a 10 cm x 10 cm. Si está interesado en cómo realizar un pedido en JLCPCB, puede consultar esta guía.

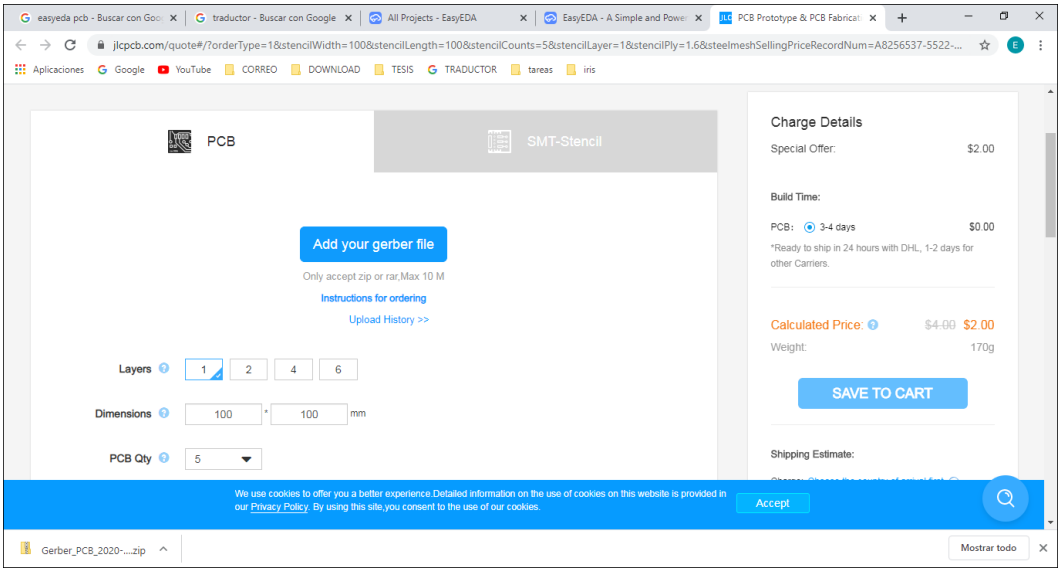


Figura 73. Selección de características de la PCB para su fabricación.

El intercambio, las colaboraciones y los sistemas de control de versiones son características fundamentales para herramientas en la nube como EasyEDA. Primero, puede agregar otro miembro para ver o editar el proyecto. Puede ver en la imagen a continuación cómo se agregó un usuario con permiso de lectura y escritura. En segundo

lugar, EasyEDA te permite crear equipos. Cada equipo tiene miembros y proyectos. Después de crear el equipo, invita a los miembros a él. Para mostrar y editar los proyectos del equipo, primero debe cambiar al perfil de su equipo. Tercero, un sistema de control de versiones simple. Donde una lista completa de versiones de su comprometido (guardado) cambia a su proyecto. Además, puede bifurcar los proyectos públicos de otros usuarios. Sería bueno si esta lista tiene información adicional como el nombre del modificador (en caso de uso del equipo). Además, puede elegir entre las propiedades del proyecto para hacer que su proyecto sea público o privado y seleccionar la licencia adecuada y el estado actual del proyecto (en progreso - completado).

2.8 DISEÑO Y CONSTRUCCIÓN DE ENVOLVENTE PARA MEDIDOR TRIFÁSICO MULTIPLEXADO.

Se utilizará un gabinete metálico ARGOS con las siguientes dimensiones 600x400x200mm, adentro se instalará las tarjetas electrónicas de instrumentación y control.

Figura 74. Gabinete metálico, medidor trifásico.

El gabinete consta de una longitud de largo de 600mm, utilizaremos 100mm para cada medidor en total 6 medidores, con una profundidad de 200mm aquí se distribuirá la borneras como se muestra en la figura.

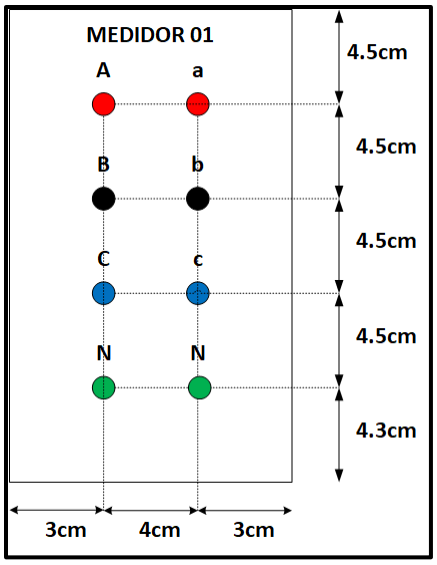


Figura 75. Dimensiones de la distribución de un medidor.

Se realizará una cuadricula internamente con las medidas de la figura anterior para realizar la distribución, utilizaremos regla y un marcador para realizar la cuadricula se utilizará un centro punto y un martillo para marcar las perforaciones, utilizaremos una broca de la misma dimensión que el aislante de la bornera.

Figura 76. Conector tipo hembra bornera banana 1.9x1.1x1.5cm.

A continuación, se muestra como quedara finalizado el ensamble del equipo en el área frontal.

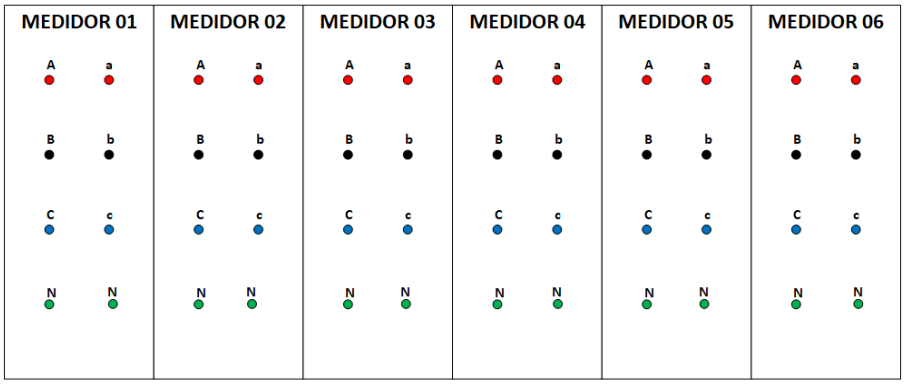


Figura 77. Diseño final de envolvente para medidor trifásico.

Para el circuito de voltaje se ha utilizado cable blindado se ha soldado a las borneras y al circuito dichos cables se han sujetado con chinchas plásticas sujetas al gabinete, se ha utilizado cable con blindaje por los campos magnéticos y agrupados por medidor.

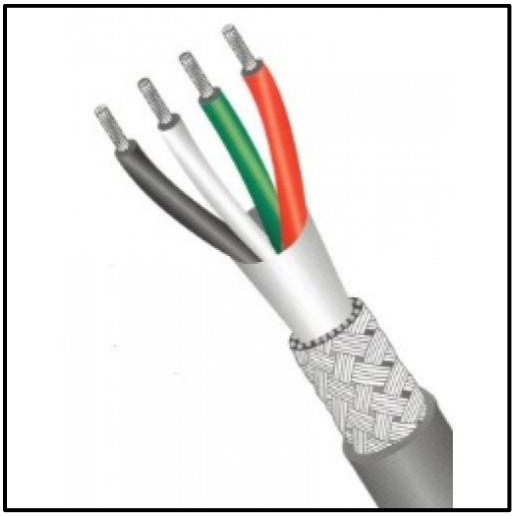


Figura 78. Cable blindado.

Los transformadores de corriente se conectan a un cable numero 14 THHN, el cable está conectado en la borneras de alimentación y la de carga por un conector de ojo número 14-16 de color amarillo el cable esta doblado para encajar en las borneras y sostener el transformador de corriente, el cable de los transformadores de corriente esta ordenados con bases adhesivas y chinchas plásticas, colocadas lateralmente y unidas por medidor.

Figura 79. Transformador de corriente.

Diagrama esquemático de la distribución interna, este diagrama muestra la ubicación de los circuitos dentro del gabinete y también el rectángulo de color gris corresponde al gabinete de la computadora para obtener aislamiento magnético de la cantidad de transformadores, las flechas indican la comunicación entra las tarjetas y la obtención de las mediciones.

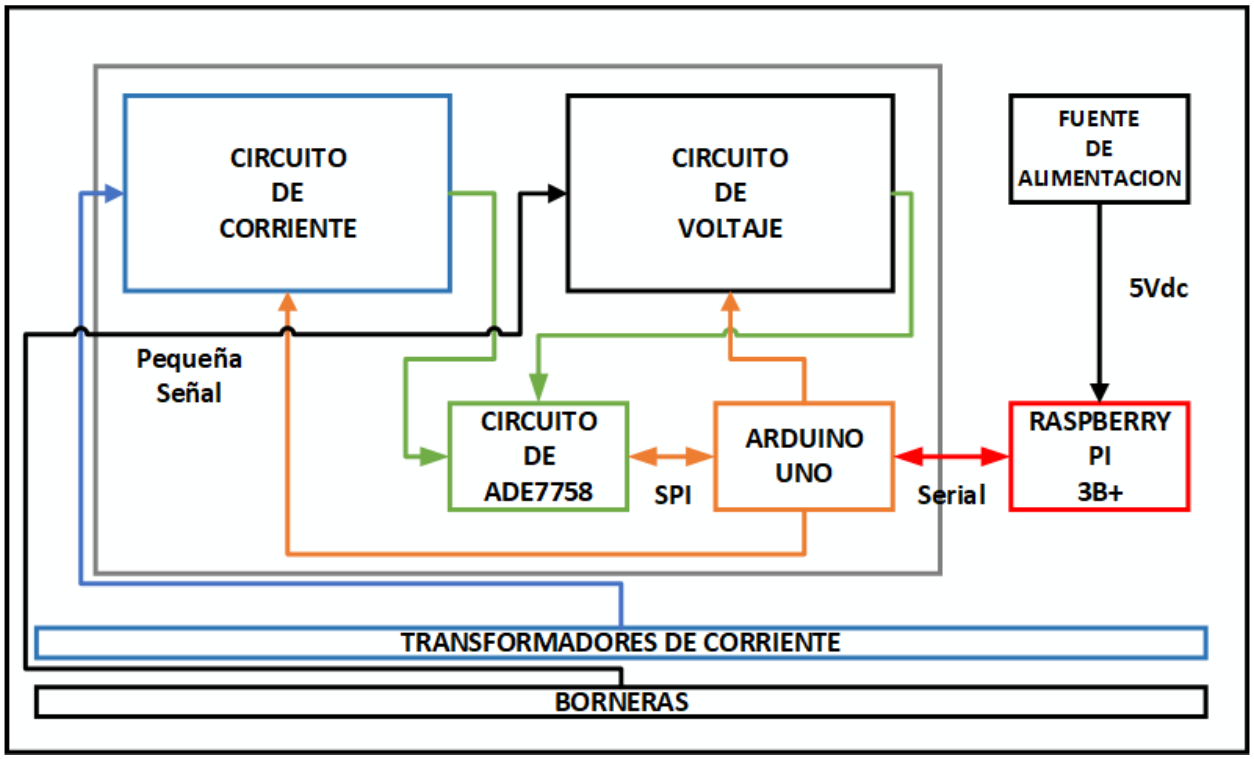


Figura 80. Diagrama esquemático de la distribución interna.

Se a utilizado un gabinete de computadora Dell Optiplex para un mayor aislamiento de los campos magnéticos para evitar ruido y tener lecturas erróneas, dentro de este se encuentra los circuitos de corriente, circuito de voltaje, circuito del ADE7758 y el Arduino.

Figura 81. Gabinete Dell Optiplex reciclado.

3 DISEÑO E IMPLEMENTACIÓN DE INTERFAZ GRAFICA DE USUARIO DE EL EQUIPO PARA MEDICIÓN.

3.1 INTRODUCCIÓN.

En este capítulo se analiza con detalle el diseño e implementación de la etapa de interfaz gráfica de usuario, la cual ha desarrollada para llevar a cabo el proceso para la adquisición y presentación de datos que provienen del circuito medidor de energía ADE7758.

Se presenta cada etapa del diseño en el diagrama de bloque de la Figura 23.

A continuación, se presentan las etapas con las que se ha efectuado el diseño e implementación del equipo.

- Adquisición de señales para cada medición, las cuales son recibidas a través de la comunicación SPI (Serial Peripheral Interface) utilizando la tarjeta Arduino Mega.
- Instalación de software de Arduino (Arduino IDE) en el sistema operativo Raspbian de la Raspberry Pi 3.
- Desarrollo de interfaz gráfica de usuario con el intérprete Python, utilizando la librería Tk Inter.

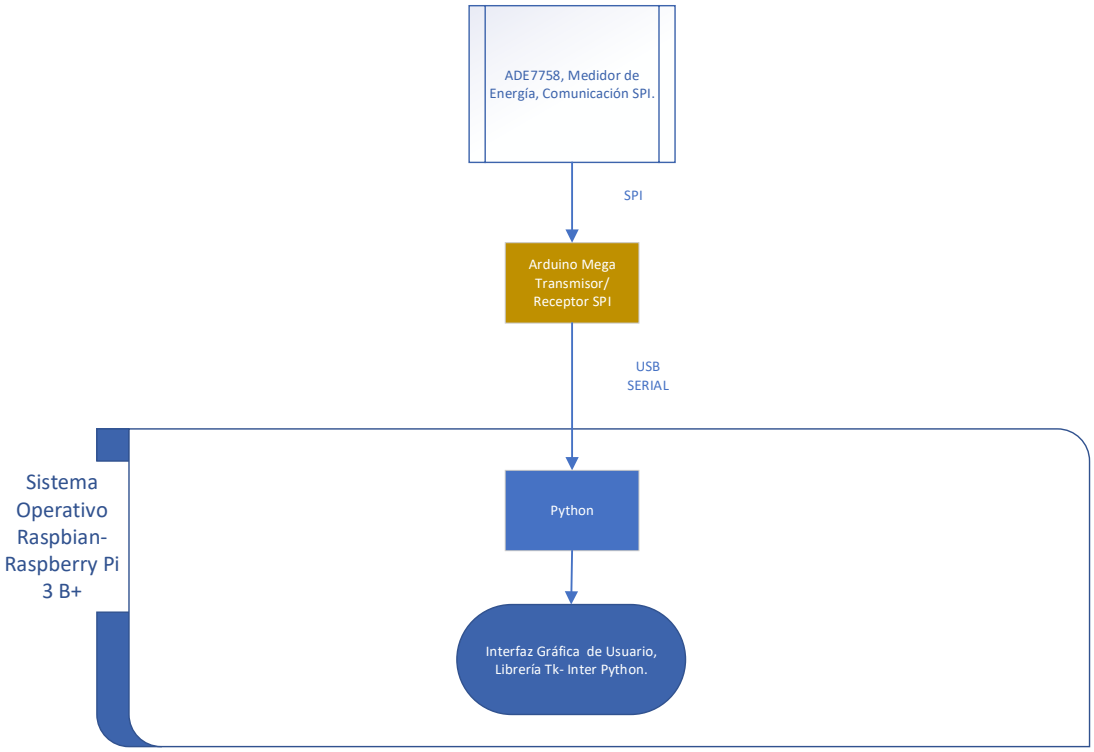


Figura 82. Diagrama en bloques de la interfaz gráfica.

3.2 ARDUINO MEGA, COMO TRANSMISOR/RECEPTOR SPI.

3.2.1 ARDUINO.

La plataforma Arduino pertenece al concepto de hardware y software libre y está abierto para su uso y contribución para una diversidad de aplicaciones de aporte a nuestra sociedad. Arduino como tal es una plataforma de prototipos electrónicos, creado en Italia, que consiste básicamente en una placa microcontrolador, con un lenguaje de programación en un entorno de desarrollo que soporta la entrada y salida de datos y señales.

Este proyecto fue desarrollado en el año 2005 con el objetivo de servir como base para proyectos de bajo coste y es lo suficientemente simple para ser utilizado por los desarrolladores. Este es una plataforma de computadora física (sistemas digitales conectados a sensores y actuadores, que permiten construir sistemas que perciben la realidad y responden con acciones físicas), dicho dispositivo está basado en un simple PCB (Printed Circuit Board) que contienen un microcontrolador de entradas y salidas y

desarrollado sobre una biblioteca que simplifica la escritura de la escritura de la programación en C/C++.

3.2.1.1 HARDWARE ARDUINO.

Tal como se hacía mención la placa Arduino contiene un microcontrolador (también denominado MCU), este es un sistema de cómputo que contiene procesador, memoria y periféricos de entrada/salida. A diferencia de los microprocesadores de propósito general (como los utilizados en las computadoras), el microcontrolador puede ser programado para funciones específicas, estos son embarcados en el interior de algún dispositivo, en nuestro caso Arduino, para que puedan controlar sus funciones o acciones.

Arduino está basado en un microcontrolador (Atmega) y de esta forma se puede programar lógicamente, esto permite la creación de programas utilizando un lenguaje propio basado en C/C++, que, cuando se implementan hacen que el hardware ejecute ciertas acciones, así de esta manera, estamos configurando la etapa de procesamiento.

Teniendo los dispositivos de desarrollo , este es capaz de interpretar variables físicas en el entorno y transformarlas en las señales eléctricas correspondientes a través de sensores conectados a sus terminales de entrada y actuar sobre el control o accionamiento de otro elemento electrónico conectado a la/las terminales de salida, lo cual le amerita a como una herramienta de control de entrada y salida de datos , que puede ser accionada por un sensor y que, después de pasar un etapa de procesamiento, el microcontrolador, podrá accionar un actuador.

Los microcontroladores que tiene las placas de Arduino más usados por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños son:

- **Atmega168**
- **Atmega328**
- **Atmega1280**
- **ATmega8**

Arduino también se usa también con microcontroladores **CortexM3 ARM** de **32 bits**

ARM y **AVR** no son plataformas compatibles a nivel binario, pero **se pueden programar con el mismo IDE** de Arduino.

- Los microcontroladores CortexM3 usan **3.3V**.
- La mayoría de las placas con AVR generalmente usan **5V**.

3.2.1.2 MODELOS DE ARDUINOS.

Hay muchos tipos de Arduino que se pueden utilizar dependiendo de los que se quiera hacer, con diferentes formas y configuraciones de hardware.

El Arduino Uno es el más utilizado pero el Arduino Mega (el cual hemos utilizado para nuestro desarrollo del medidor de energía), tiene más puertos de entrada, posibilitando la creación de sistemas más grandes y complejos.

El Arduino Nano, como el nombre dice, es una versión abreviada de un Arduino común, para la creación de objetos de electrónica más pequeña. A continuación, en la Figura 83, Figura 84 y Figura 85 , se observan los modelos de Arduino mencionados.

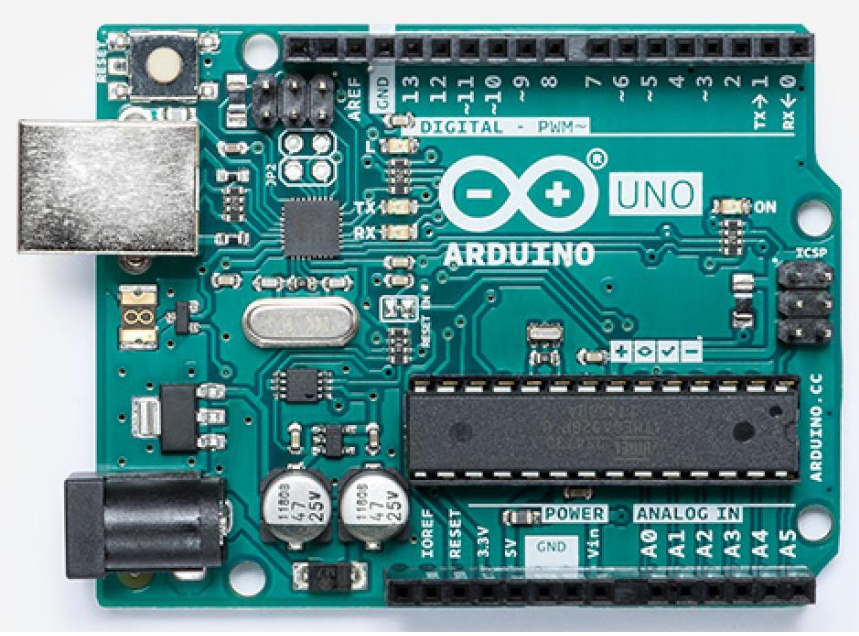


Figura 83. Arduino Uno.

Figura 84. Arduino Mega.

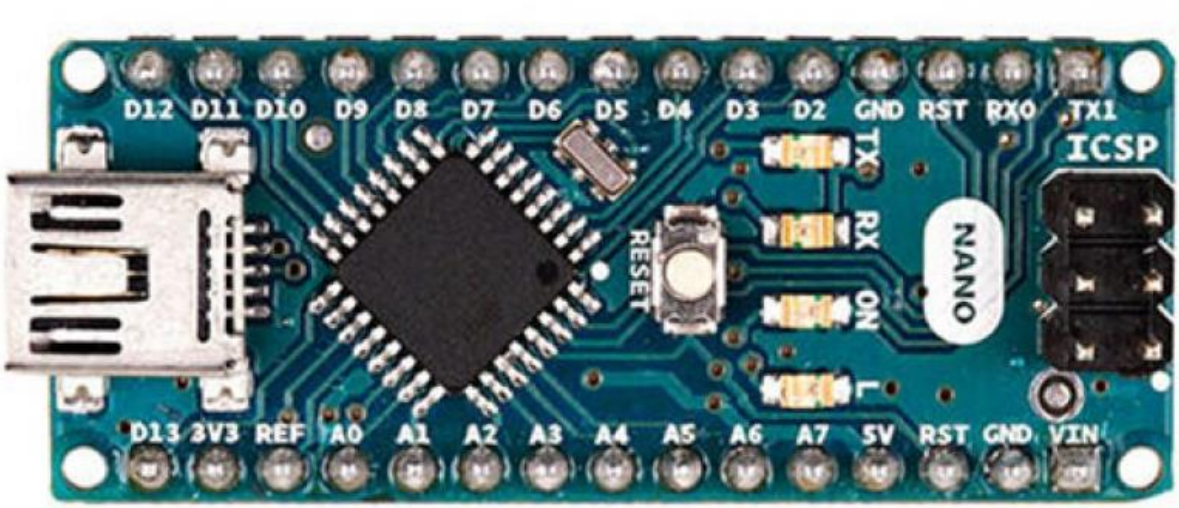


Figura 85. Arduino Nano.

3.3 INSTALAR RASPBIAN EN RASPBERRY PI 3 B+.

Iniciaremos en nuestro navegador web en la siguiente dirección https://www.raspberrypi.org/ daremos un clic en "Downloads".

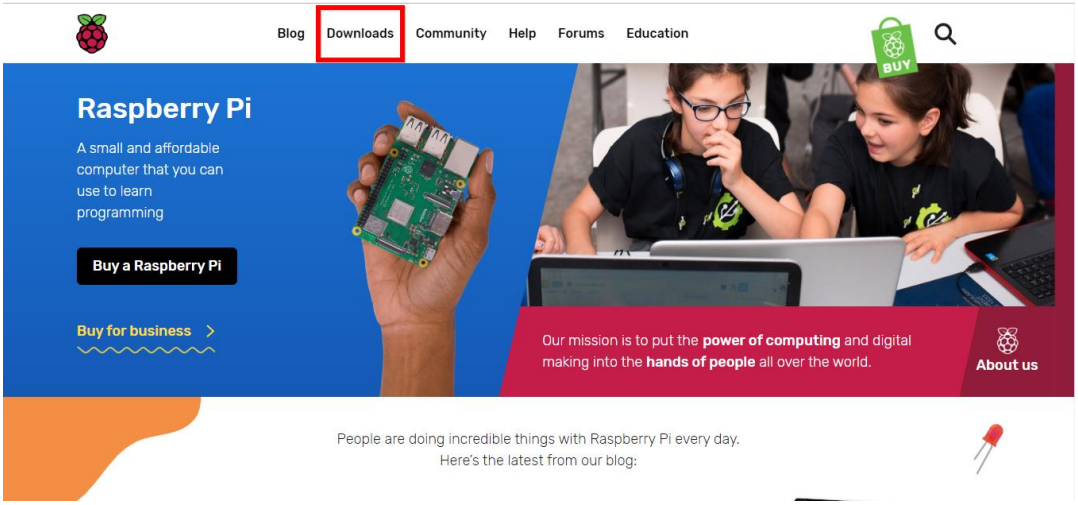


Figura 86. Página oficial de Raspberry Pi.

Nos mostrara la siguiente página web, nos muestran dos formas de instalar Raspbian, una es la versión a instalar por defecto la cual es "RASPBIAN" pero nos puede mostrar ciertos conflictos, como librerías corruptas las cuales hay que recompilar.

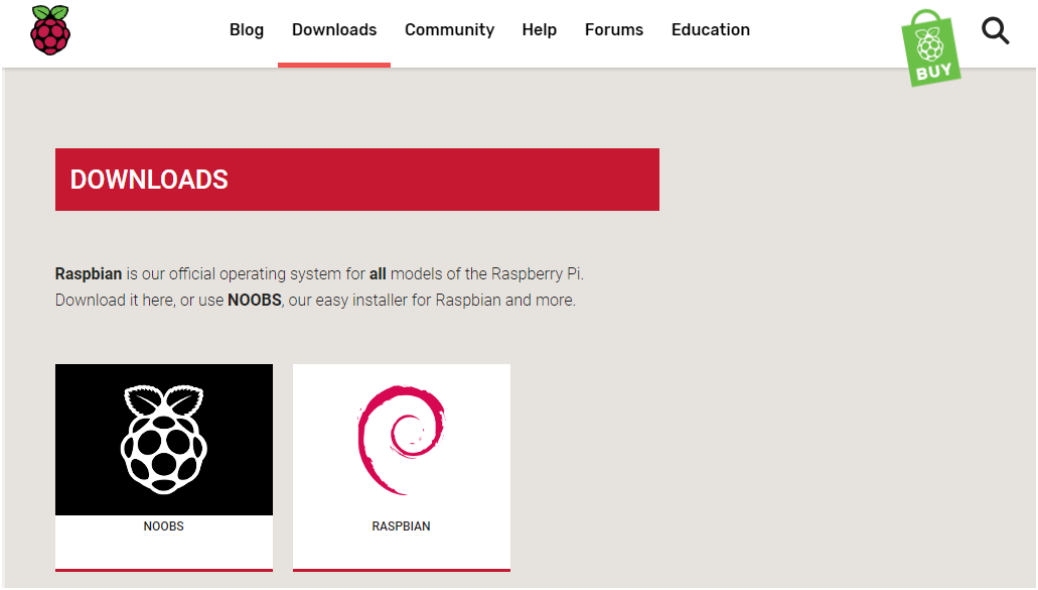


Figura 87. Página de descarga de Raspbian.

La otra forma de instalar raspbian es por medio de "NOOBS", la cual realizamos nosotros, daremos un click en NOOBS y se nos mostrara la siguiente página web. Descargaremos NOOBS LITE, daremos un clic en DOWNLOAD ZIP.

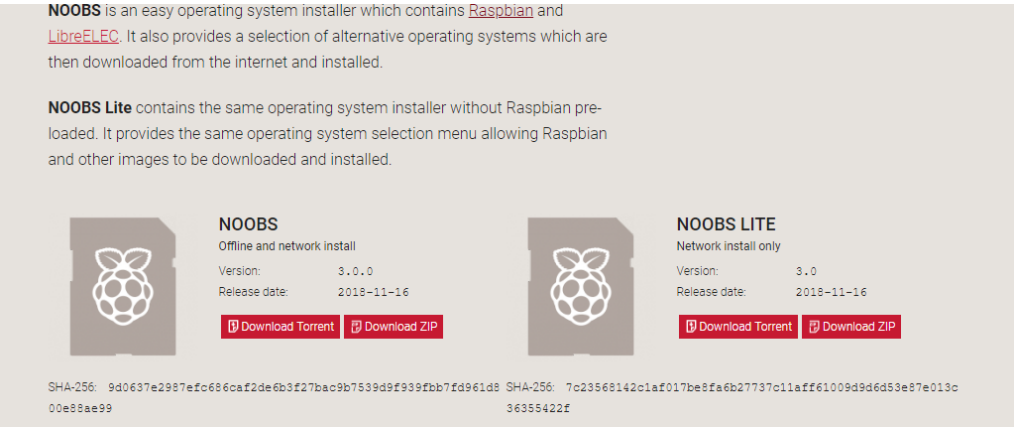


Figura 88. Archivos a descargar para la instalación de Raspbian.

Extraeremos los ficheros de la carpeta comprimida como se muestra a continuación.

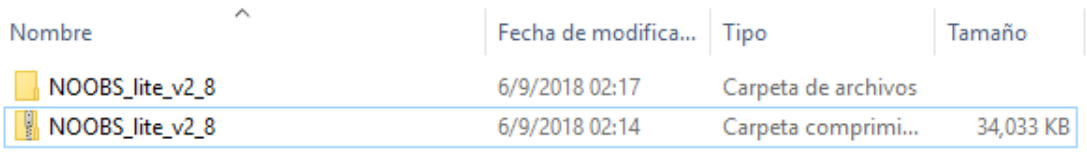


Figura 89. Extracción de los ficheros de Raspbian.

Utilizaremos una microSD, se recomienda utilizar una SanDisk Ultra Clase10 como se muestra en la Figura 82, colocaremos la microSD en la PC, formatearemos la microSD.

Figura 90. Micro SD SanDisk Ultra.

Copiaremos todos los archivos que extraemos del archivo zip y los pegaremos en la microSD.

Nombre	Fecha de modifica...	Tipo	Tamaño
defaults	14/3/2018 08:38	Carpeta de archivos	
os	14/3/2018 08:38	Carpeta de archivos	
overlays	18/4/2018 09:18	Carpeta de archivos	
bcm2708-rpi-0-w.dtb	18/4/2018 09:19	Archivo DTB	22 KB
bcm2708-rpi-b.dtb	18/4/2018 09:19	Archivo DTB	22 KB
bcm2708-rpi-b-plus.dtb	18/4/2018 09:19	Archivo DTB	22 KB
bcm2708-rpi-cm.dtb	18/4/2018 09:19	Archivo DTB	21 KB
bcm2709-rpi-2-b.dtb	18/4/2018 09:19	Archivo DTB	23 KB
bcm2710-rpi-3-b.dtb	18/4/2018 09:19	Archivo DTB	24 KB
bcm2710-rpi-3-b-plus.dtb	18/4/2018 09:19	Archivo DTB	24 KB
bcm2710-rpi-cm3.dtb	18/4/2018 09:19	Archivo DTB	23 KB
bootcode.bin	18/4/2018 09:19	Archivo BIN	51 KB
BUILD-DATA	18/4/2018 09:19	Archivo	1 KB
INSTRUCTIONS-README	18/4/2018 09:18	Archivo TXT	3 KB
recovery.cmdline	18/4/2018 09:19	Archivo CMDLINE	1 KB
recovery.elf	18/4/2018 09:19	Archivo ELF	658 KB
recovery	18/4/2018 09:19	Archivo de image...	2,918 KB
recovery.rfs	18/4/2018 09:19	Archivo RFS	27,896 KB
RECOVERY_FILES_DO_NOT_EDIT	18/4/2018 09:19	Archivo	0 KB
recovery7	18/4/2018 09:19	Archivo de image...	2,981 KB
riscos-boot.bin	18/4/2018 09:18	Archivo BIN	10 KB

Figura 91. Archivos Raspberry Pi.

Colocaremos la microSD en la Raspberry Pi 3 B+ también colocaremos los demás periféricos como es el teclado, ratón, el cable HDMI para el monitor, se recomienda conectar el cable de internet RJ45 para la instalación y conectaremos la alimentación de la Raspberry Pi.

Figura 92. Raspberry Pi.

Al iniciar aparece un menú con los distintos sistemas operativos disponibles. Elegimos el sistema que queremos y pulsamos la tecla 'I' para instalarlo.

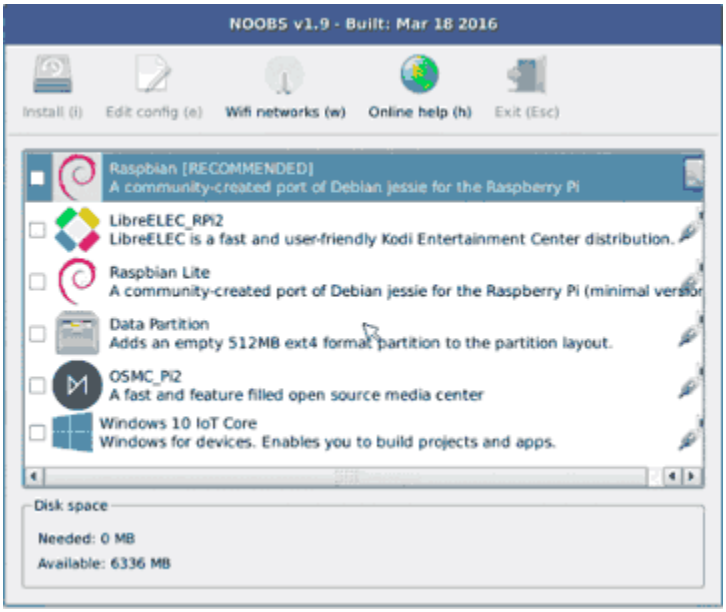


Figura 93. Instalación de Raspbian.

Esperamos pacientemente mientras Noobs instala el sistema operativo seleccionado en la tarjeta de memoria.

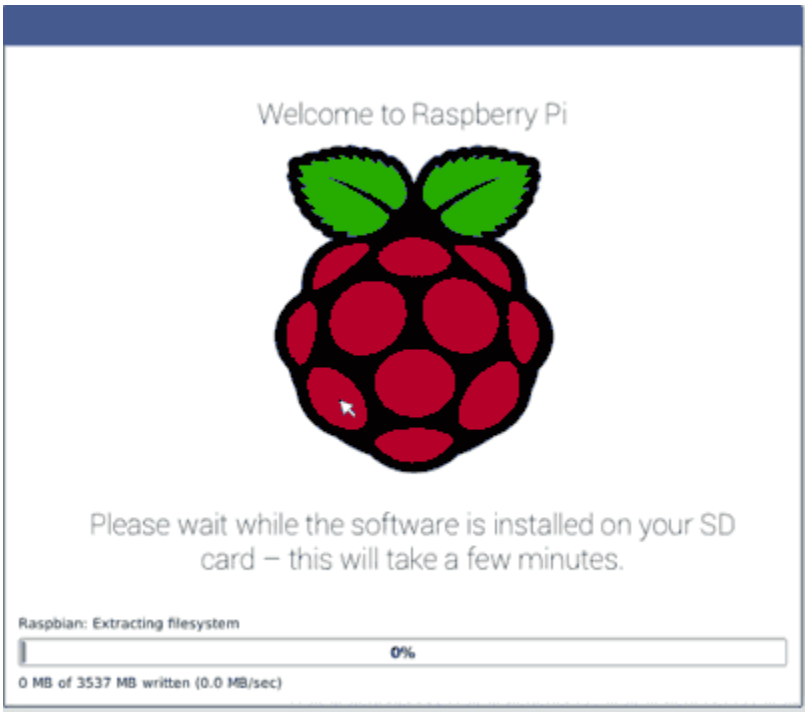


Figura 94. Proceso de instalación.

Una vez finalizado el proceso, hacemos click en el botón para reiniciar Raspberry Pi.

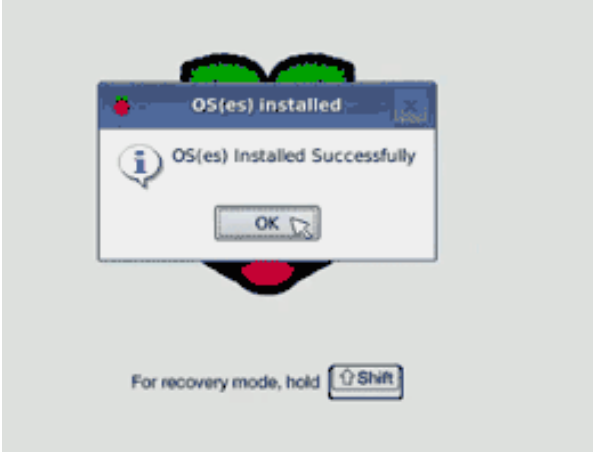


Figura 95. Finalización de la instalación.

Raspberry Pi se reinicia y ejecutará el sistema operativo instalado y ¡ya podemos empezar a jugar con nuestra Raspberry Pi.

Figura 96. Escritorio de Raspbian.

3.4 INSTALACIÓN DE ARDUINO IDE EN RASPBIAN.

Pasos para instalar Arduino IDE en Raspbian, utilizaremos nuestro navegador web, iremos a la siguiente dirección web https://www.arduino.cc/, daremos un clic en SOFTWARE, se desplegará un menú y daremos un clic en DOWNLOADS.

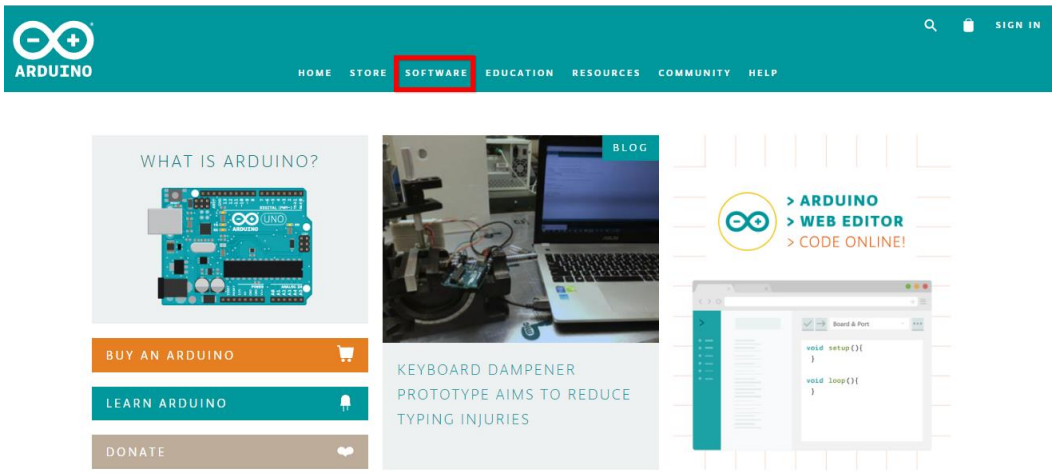


Figura 97. Página oficial de Arduino.

Damos un clic en Linux ARM y se iniciara una descarga del archivo arduino-1.8.7-r1-linuxarm.tar.xz se descomprime el archivo.

Figura 98. Archivo a descargar.

En la terminal buscamos la carpeta creada por el archivo ingresos a la carpeta y ejecutamos un archivo install.sh

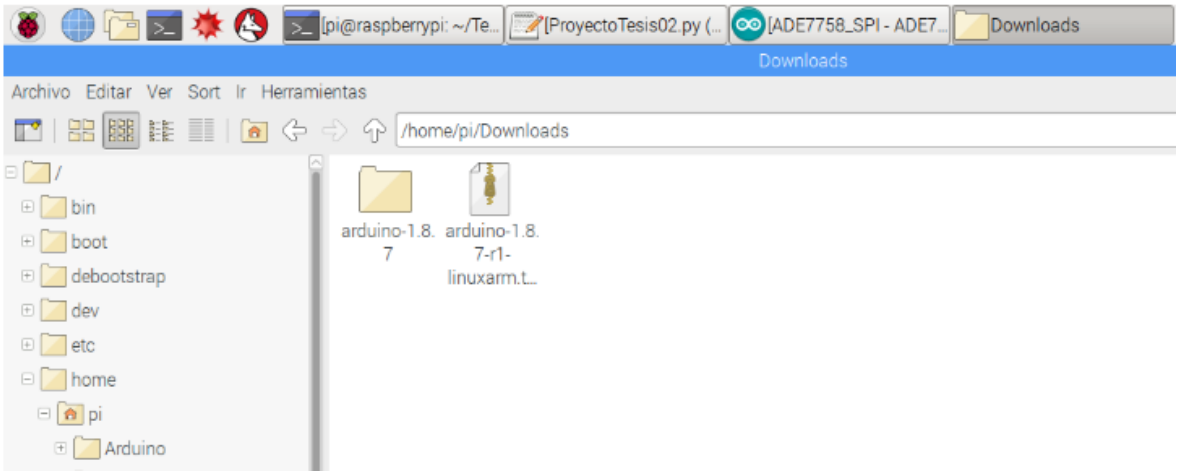


Figura 99. Ejecutamos en la terminal el comando.

Ejecutamos en la terminal sudo ./install.sh

3.5 PYTHON INTERFAZ GRÁFICA DE USUARIO.

Arduino Comunicación SPI ADE7758.

Para iniciar el programa principal de Arduino necesitamos utilizar librerías propias de Arduino y otra diseña especialmente para el ADE7758.

ADE7758_SPI | ADE7758.cpp | ADE7758.h

```
1  //LIBRERIA ADE7758
2  #ifndef ADE7758_h
3  #define ADE7758_h
4  #include "Arduino.h"
```

Figura 100. Librería del ADE7758.

El archivo .h o header

El header (cabecera) o archivo .h, es el archivo que representará a la librería cuando la integremos. En él se colocan declaraciones, pero no el código en si mismo. En nuestro caso el archivo se llamará ADE7758.h.

#define: sirve para definir macros. Las macros suministran un sistema para la sustitución de palabras, con y sin parámetros.

#ifndef: Esta directiva permiten comprobar si un identificador está o no actualmente definido, es decir, si un #define ha sido previamente procesado para el identificador y si sigue definido.

Esto evita problemas si alguien accidentalmente incluye dos veces la librería, lo que provocaría un error de compilación. A esto se llama guardián de inclusión múltiple.

En el fichero de cabecera de una librería es necesario la declaración #include que de acceso a los tipos y constantes estándar del lenguaje de Arduino Esta declaración debe ponerse antes de la definición de la clase (#include ”Arduio.h”).

```
6  #define AWATTHR 0x01
7  #define BWATTHR 0x02
8  #define CWATTHR 0x03
```

Figura 101. Definición de parámetros.

Definimos unas constantes con valores hexadecimales, las cuales las utilizamos para establecer comunicación con el ADE7758.

```
116 class ADE7758{
117   public:
118     ADE7758(int _CS);
119     void begin();
120     long getVRMS(char phase);
121     long getIRMS(char phase);
122     long getPwatt(char phase);
123     long getQvar(char phase);
124     long getSva(char phase);
125     long getFHz(char phase);
126 
127     unsigned char read8bits(char reg);
128     unsigned int read16bits(char reg);
129     unsigned long read24bits(char reg);
130     //ESCRITURA
131     void write8(char reg, unsigned char data);
132     void write16(char reg, unsigned int data);
133     void write24(char reg, unsigned long data);
134 
135   private:
136     int CS;
```

Figura 102. Clase de la librería del ADE7758.

Class: sirve para definir una clase y para declarar objetos de esa clase.

Public: Cualquier miembro público de una clase es accesible desde cualquier parte donde sea accesible el propio objeto, declaramos funciones públicas para la lectura de las mediciones.

Private: Los miembros privados de una clase sólo son accesibles por los propios miembros de la clase y en general por objetos de la misma clase, pero no desde funciones externas o desde funciones de clases derivadas.

Archivo principal CPP.
Usualmente a todo archivo de cabecera .h lo acompaña un archivo de código que puede tener la extención .c o .cpp. Es una buena práctica tener un archivo con el mismo nombre que la cabecera y por lo tanto el nuestro se llamará ADE7758.cpp, y su código se verá asi:

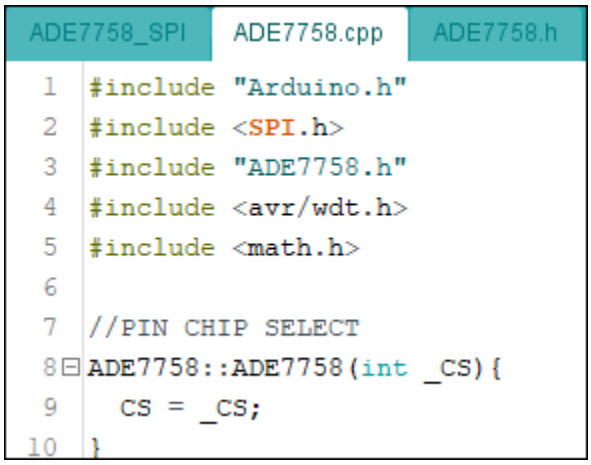


Figura 103. Constructor de la clase.

#include "Arduino.h": Una declaración que le da acceso a los tipos y constantes estándar del lenguaje Arduino.

#include <SPI.h>: Esta biblioteca le permite comunicarse con dispositivos SPI, con el Arduino como dispositivo maestro y esclavo.

#include "ADE7758.h": Librería que se utiliza para la comunicación SPI con el ADE7758, previamente explicada.

#include <avr/wdt.h>: Este archivo de encabezado declara la interfaz a algunas macros en línea que manejan el temporizador de vigilancia presente en muchos dispositivos AVR. Para evitar que la configuración del temporizador de vigilancia se altere accidentalmente por una aplicación que falla, se requiere una secuencia temporizada especial para cambiarla. Las macros dentro de este archivo de encabezado manejan la secuencia requerida automáticamente antes de cambiar cualquier valor. Las interrupciones se desactivarán durante la manipulación.

Lo siguiente es poner el constructor de la clase. Esto define que ocurre cuando se crea una instancia de la clase. En este caso especificamos cual es el pin que se va a usar y los guardamos en una variable privada para usarlo desde otras funciones.

Resolución de ámbito: El "::"es llamado el operador de resolución de ámbito. Esencialmente le dice al compilador que esta versión de "begin()" pertenece a la clase

ADE7758. Dicho de otra forma, :: declara que "begin()" se encuentra en el ámbito de ADE7758. Varias clases diferentes pueden usar los mismos nombres de función. El compilador sabe cuál función pertenece a cuál clase y esto es posible por el operador de resolución de ámbito y el nombre de la clase.

```
11  void ADE7758::begin(){
12    //DEFINO PIN CS COMO SALIDA
13    pinMode(CS,OUTPUT);
14    //DESACTIVO ADE7758
15    digitalWrite (CS,HIGH);
16    //MODO DE COMUNICACION
17    SPI.setDataMode(SPI_MODE2);
18    //FRECUENCIA DEL RELOJ
19    SPI.setClockDivider(SPI_CLOCK_DIV32);
20    //MSB PRIMERO
21    SPI.setBitOrder(MSBFIRST);
22    //INICIALIZO SPI
23    SPI.begin();
24    delay(10);
25  }
```

Figura 104. Configuración inicial del ADE7758 desde Arduino.

myADE.begin: establece la configuración básica para la comunicación SPI entre el Arduino y el ADE7758.

pinMode configura el pin especificado para que se comporte como una entrada o una salida. pinMode(CS,OUTPUT) define el pin chip select como salida.

digitalWrite(CS,HIGH), activa el pin de salida en alto (5Vdc) para desactivar el ADE7758,. digitalWrite(CS,LOW), desactiva el pin de salida en bajo (0Vdc) para activar el ADE7758.

SPI.setDataMode(SPI_MODE2): En términos generales, hay cuatro modos de transmisión. Estos modos controlan si los datos se desplazan hacia adentro y hacia afuera en el flanco ascendente o descendente de la señal del reloj de datos (llamada fase de reloj), y si el reloj está inactivo cuando está alto o bajo (llamada polaridad del reloj). Los cuatro modos combinan polaridad y fase, como se muestra en la tabla:

Mode	Clock Polarity (CPOL)	Clock Phase (CPHA)	Output Edge	Data Capture
SPI_MODE0	0	0	Falling	Rising
SPI_MODE1	0	1	Rising	Falling
SPI_MODE2	1	0	Rising	Falling
SPI_MODE3	1	1	Falling	Rising

Figura 105. Modos de comunicación SPI.

SERIAL INTERFACE

The ADE7758 has a built-in SPI interface. The serial interface of the ADE7758 is made of four signals: SCLK, DIN, DOUT, and $\overline{CS}$. The serial clock for a data transfer is applied at the SCLK logic input. This logic input has a Schmitt trigger input structure that allows slow rising (and falling) clock edges to be used. All data transfer operations are synchronized to the serial clock. Data is shifted into the ADE7758 at the DIN logic input on the falling edge of SCLK. Data is shifted out of the ADE7758 at the DOUT logic output on a rising edge of SCLK.

Figura 106. El modo de operación ADE7758 como lo describe el fabricante en su hoja de datos

SPI.setClockDivider(SPI_CLOCK_DIV32): Establece el divisor de reloj SPI en relación con el reloj del sistema. En las tarjetas basadas en AVR, los divisores disponibles son 2, 4, 8, 16, 32, 64 o 128. La configuración predeterminada es SPI_CLOCK_DIV4, que establece el reloj SPI en un cuarto de la frecuencia del reloj del sistema (4 Mhz para las tarjetas a 16 MHz).

SPI.setBitOrder(MSBFIRST): definimos que los primeros bytes a enviar son los más significativos.

SPI.begin(): función propia de Arduino para inicializar el modo SPI.

```
45⊟unsigned char ADE7758::read8bits(char reg){
46    enable();
47      unsigned char b0;
48    //FUNCION ARDUINO:PAUSA MICROSEGUNDOS
49    delayMicroseconds(TREAD);
50    //FUNCION ARDUINO: SPI
51    SPI.transfer(reg);
52    delayMicroseconds(TREAD);
53    b0=SPI.transfer(READ);
54    delayMicroseconds(TREAD);
55      disable();
56    return b0;
57  }
```

Figura 107. Función lectura de 8 bits.

Para poder realizar una lectura al ADE7758, necesitamos interpretar el protocolo de comunicación SPI y el diagrama de tiempo que nos ofrece la hoja de datos, iniciaremos con el diagrama de tiempo, que se muestra a continuación:

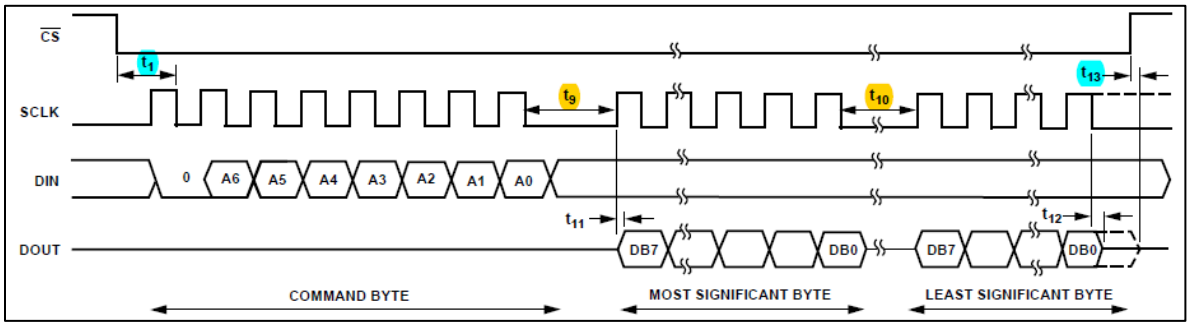


Figura 108. Diagrama de tiempo de lectura.

t1: es el tiempo en el borde descendente del CS al primer borde descendente al SCLK, con un tiempo mínimo de 50 nano segundos.
t9: Tiempo mínimo entre el comando de lectura (es decir, un registro de escritura en comunicación) y datos leídos, con un tiempo mínimo de 4 microsegundos.
t10: Tiempo mínimo entre transferencias de bytes de datos durante una lectura múltiples byte, el tiempo mínimo es de 10 nano segundos.
t13: se termina él envió de datos en el borde creciente de CS, con un tiempo mínimo de 10 nano segundos y máximo de 100 nano segundos.

A continuación, se describe cada una de las partes de la función que se utiliza para leer 8 bits:

Línea 45:se define una función la cual utilizaremos para leer datos del ADE7758 es del tipo unsigned char (1 byte), pero el valor que retornara es de 1 bytes, se toma un valor como registro tipo char (1 byte) el cual será un valor en hexadecimal.

Línea 46: se llama a una función para habilitar el ADE7758, lo habilitaremos con el pin CS (chip select) el cual se activa con un estado bajo.

Línea 47: se definen una variable del tipo unsigned char (1 byte), para poder realizar la lectura.

Línea 49: es el tiempo de pausa que se necesita entre el CS (chip select) en el borde descendente y el SCLK (la entrada de reloj serie para la interfaz serial síncrona) en el primer borde descendente ver diagrama de tiempo "t1".

Línea 51: se utiliza la función "SPI.transfer" esta función es de la librería de comunicación SPI de Arduino, se utiliza para enviar y recibir datos, en esta línea de código la utilizamos para enviar el registro al cual deseamos leer, el registro se envía MOSI (DIN diagrama de tiempo).
Línea 52: es el tiempo de pausa que se utiliza entre el envió de un registro y los datos que deseamos leer, ver diagrama tiempo "t9".

Línea 53: se utiliza la función "SPI.transfer" y como parámetro para poder leer los datos se envía 0x00 este dato se envía en la línea de MOSI (DIN diagrama de tiempo), se obtiene de la función el MSB de los bytes, este dato lo capturamos en la línea de MISO (DOUT diagrama de tiempo).

Table 16. Communications Register

Bit Location	Bit Mnemonic	Description
0 to 6	A0 to A6	The seven LSBs of the communications register specify the register for the data transfer operation. Table 17 lists the address of each ADE7758 on-chip register.
7	W/$\overline{R}$	When this bit is a Logic 1, the data transfer operation immediately following the write to the communications register is interpreted as a write to the ADE7758. When this bit is a Logic 0, the data transfer operation immediately following the write to the communications register is interpreted as a read operation.

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
W/$\overline{R}$	A6	A5	A4	A3	A2	A1	A0

Figura 109. Registro de comunicación.

Línea 54: es el tiempo de pausa para finalizar él envió de datos, ver diagrama de tiempo "t13".

Línea 55: se llama a una función para deshabilitar el ADE7758, lo deshabilitaremos con el pin CS (chip select) el cual se desactiva con un estado alto.

Línea 56: retornamos los valores recibidos del ADE7758.

```
91  void ADE7758::write8(char reg, unsigned char data){
92          enable();
93          unsigned char data0 = 0;
94          data0 = (unsigned char)data;
95          reg |= WRITE;
96
97          delayMicroseconds(50);
98          SPI.transfer((unsigned char)reg);
99          delayMicroseconds(50);
100         SPI.transfer((unsigned char)data0);//dato
101         delayMicroseconds(50);
102
103         disable();
104 }
```

Figura 110. Función de escritura de 8 bits.

Línea 91: definimos una función de escritura de 1BYTE del tipo void, con dos parámetros una es a la dirección de memoria a la que deseamos escribir y el otro es el valor a escribir.

Línea 92: se llama a una función para habilitar el ADE7758, lo habilitaremos con el pin CS (chip select) el cual se activa con un estado bajo.

Línea 93: inicializamos una variable de 8 bits del tipo unsigned char.

Línea 94: definimos el valor a escribir en el registro de memoria.

Línea 95: definimos la dirección de memoria a la cual deseamos realizar una escritura y agregamos un bit uno indicando que se realizara una escritura, como esta en el diagrama de tiempo del protocolo de comunicación SPI.

Línea 97: tiempo durante una transferencia de escritura en lo que se activa el ADE7758.

Línea 98: se utiliza la función "SPI.transfer" y como parámetro para poder escribir los datos se envía 0x80 este dato se envía en la línea de DIN (diagrama de tiempo).

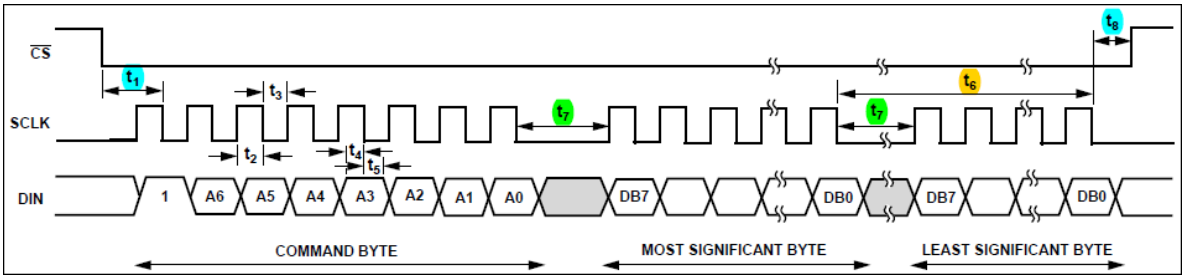


Figura 111. Diagrama de tiempo de escritura serial.

Línea 99: tiempo durante una transferencia de escritura.

Línea 100: se utiliza la función "SPI.transfer" para escribir en la memoria del ADE7758.

Línea 101: tiempo antes desactivar el ADE7758.

Línea 103: se llama a una función para deshabilitar el ADE7758, lo deshabilitaremos con el pin CS (chip select) el cual se desactiva con un estado alto.

Lectura de voltaje RMS.

getVRMS: es una función que realiza las lecturas de voltaje RMS del ADE7758, la precisión del valor RMS actual es típicamente 0.5% error de la entrada de escala completa hasta 1/500 de la entrada de escala completa. Adicionalmente, esta medida tiene un ancho de banda de 14 kHz.

Es recomendado para leer los registros RMS sincronizar con los cruces por ceros para garantizar la estabilidad.
Las interrupciones se pueden usar para indicar cuando ha ocurrido un cruce por cero (ver la sección Interrupciones).

INTERRUPT STATUS REGISTER (0x19)
RESET INTERRUPT STATUS REGISTER (0x1A)

El estado de los registros de interrupción se usa para determinar el origen de un evento de una interrupción.

Table 24. Interrupt Status Register

Bit Location	Interrupt Flag	Default Value	Event Description
0	AEHF	0	Indicates that an interrupt was caused by a change in Bit 14 among any one of the three WATTHR registers, that is, the WATTHR register is half full.
1	REHF	0	Indicates that an interrupt was caused by a change in Bit 14 among any one of the three VARHR registers, that is, the VARHR register is half full.
2	VAEHF	0	Indicates that an interrupt was caused by a 0 to 1 transition in Bit 15 among any one of the three VAHR registers, that is, the VAHR register is half full.
3	SAGA	0	Indicates that an interrupt was caused by a SAG on the line voltage of the Phase A.
4	SAGB	0	Indicates that an interrupt was caused by a SAG on the line voltage of the Phase B.
5	SAGC	0	Indicates that an interrupt was caused by a SAG on the line voltage of the Phase C.
6	ZXTOA	0	Indicates that an interrupt was caused by a missing zero crossing on the line voltage of the Phase A.
7	ZXTOB	0	Indicates that an interrupt was caused by a missing zero crossing on the line voltage of the Phase B.
8	ZXTOC	0	Indicates that an interrupt was caused by a missing zero crossing on the line voltage of the Phase C.
9	ZXA	0	Indicates a detection of a rising edge zero crossing in the voltage channel of Phase A.
10	ZXB	0	Indicates a detection of a rising edge zero crossing in the voltage channel of Phase B.
11	ZXC	0	Indicates a detection of a rising edge zero crossing in the voltage channel of Phase C.

Figura 112. Tabla de registro de las interrupciones.

El bit número nueve nos indica una detección de cruce por cero en el borde ascendente en el canal de voltaje de la fase A.

```
long ADE7758::getVRMS(char phase){
    int N = 20;
    unsigned long VRMS = 0;
    long lastupdate = 0;
    for (int i = 0; i<N; i++){
        getResetInterruptStatus();
        lastupdate = millis();
        while(!(getInterruptStatus() & (ZXA<<phase))){
                if((millis()-lastupdate)>100){return 0;}
        }
        VRMS += read24bits(AVRMS+phase);
    }
    return VRMS/N;
}
```

Figura 113. Función de lectura Vrms.

La función getVRMS recibe un para metro por defecto el cual indica la fase a realizar la medición, estos valores están definidos en la librería "ADE7758" como constantes, la

constante “N”, nos sirve para determinar una cantidad de mediciones a realizar, todas las mediciones se suman en una variable “VRMS”, el valor a retornar son las mediciones acumuladas entre el valor de “N”.

Lastupdate: esta variable guarda el tiempo inicial de una medición.

El ciclo for realiza N interacciones, después reiniciamos los registros y guardamos el tiempo inicial en milisegundos en la variable “lasupdate”, el ciclo while hace una comparación booleana, si el registro de interrupciones es igual con una constante definida en la librería “ADE7758.h” indica que hay un cruce por cero, sin son diferentes continuara preguntando hasta la que el tiempo actual menos el tiempo inicial se han menor a 100 ciclos de la señal senoidal, si se cumple esta condición retornaremos cero por quien fue llamado la función. Caso contrario realizaremos una lectura del voltaje RMS, llamaremos a la función de lectura de 3 Bytes (24 bits) y sumaremos las mediciones, el valor de retorno es el valor promedio.

<<: operador de desplazamiento a la izquierda, utilizando esta herramienta podemos comparar los cruces por cero de las tres fases.

```
#define ZXA 0x0200
#define ZXB 0x0400
#define ZXC 0x0800
```

Figura 114. Definición de constantes para los cruces por cero de las tres fases.

```
#define AVRMS 0x0D
#define BVRMS 0x0E
#define CVRMS 0x0F
```

Figura 115. Constantes que definimos para la lectura de voltaje RMS de las tres fases.

```
212 int ADE7758::getPwatt(char phase){
213     int Pwatt = 0;
214     long lastupdate = 0;
215     getResetInterruptStatus();
216     lastupdate = millis();
217     while(!(getInterruptStatus() & (ZXA<<phase))){
218       if((millis()-lastupdate)>100)return 0;
219     }
220     Pwatt=readl6bits(AWATTHR+phase);
221     return Pwatt;
222 }
```

Figura 116. Función de lectura de Potencia Activa.

```
6  #define AWATTHR 0x01
7  #define BWATTHR 0x02
8  #define CWATTHR 0x03
```

Figura 117. Dirección de memoria de la potencia activa.

```
108 #define PHASE_A 0
109 #define PHASE_B 1
110 #define PHASE_C 2
```

Figura 118. Constantes para el programa identifique cada una de las fases.

La función de potencia activa "getPwatt" realiza una lectura de 16 bits de los registros de memoria del ADE7758, el parámetro que recibe la función es la fase de la cual queremos realizar la lectura, primero revisamos si hay cruces por cero de la fase respectiva si no hay cruce por ceros retornamos el valor de cero de lo contrario realizamos una lectura con el operador de desplazamiento a la izquierda podemos verificar las tres fases "<<", para poder realizar la lectura de las diferentes fases solo se le suma el valor de la constante mas el valor de la dirección de la fase A.

```
int ADE7758::getQvar(char phase){
    int Qvar = 0;
    long lastupdate = 0;
    getResetInterruptStatus();
    lastupdate = millis();
    while(!(getInterruptStatus() & (ZXA<<phase))){
      if((millis()-lastupdate)>100)return 0;
    }
    Qvar=read16bits(AVARHR+phase);
    return Qvar;
}
```

Figura 119. Función de lectura de Potencia Reactiva.

La función de potencia reactiva "getQvar" realiza una lectura de 16 bits de los registros de memoria del ADE7758, la funcio es muy similar a la del apotencia activa.

```
int ADE7758::getSva(char phase){
    int Sva = 0;
    long lastupdate = 0;
    getResetInterruptStatus();
    lastupdate = millis();
    while(!(getInterruptStatus() & (ZXA<<phase))){
      if((millis()-lastupdate)>100)return 0;
    }
    Sva=read16bits(AVAHR+phase);
    return Sva;
}
```

Figura 120. Función de lectura de Potencia Aparente.

La función de potencia aparente "getSva" realiza una lectura de 16 bits de los registros de memoria del ADE7758, la función es muy similar a la de la potencia activa.

```
247 int ADE7758::getFHz(char phase){
248     int FHz = 0;
249     long lastupdate = 0;
250     getResetInterruptStatus();
251     lastupdate = millis();
252     while(!(getInterruptStatus() & (ZXA<<phase))){
253       if((millis()-lastupdate)>100)return 0;
254     }
255     write8(MMODE,(0xFC|phase));
256     FHz=read8bits(FREC);
257     return 60;
258 }
```

Figura 121. Función de lectura de la frecuencia.

La función de la frecuencia "getFHz" realiza una lectura de 8 bits de los registros de memoria del ADE7758, la función es muy similar a la de la potencia activa, para realizar la lectura por fase se cambia la configuración de mediciones "MMODE".

```
42   if(Serial.available()>0){
43     var=Serial.read();
44     switch (var) {
```

Figura 122. Comunicación serial.

Serial.available: Obtenemos el número de bytes (caracteres) disponibles para leer desde el puerto serie. Estos son datos que ya llegaron y se almacenaron en el búfer de recepción en serie (que contiene 64 bytes).

Serial.read: se reliza la lectura de los datos entrantes.

Switch: Al igual que las instrucciones if, switch case controla el flujo de programas al permitir que los programadores especifiquen código diferente que se debe ejecutar en varias condiciones. En particular, una instrucción switch compara el valor de una variable con los valores especificados en las instrucciones case. Cuando se encuentra una instrucción case cuyo valor coincide con el de la variable, se ejecuta el código de esa instrucción case.

La palabra clave break sale de la instrucción switch y se utiliza normalmente al final de cada caso. Sin una instrucción break, la instrucción switch continuará ejecutando las

siguientes expresiones ("falling-through") hasta que se alcance un break o se alcance el final de la instrucción switch.

```
45        case '1':
46          digitalWrite(5,HIGH);
47          digitalWrite(6,LOW);
48          digitalWrite(7,LOW);
49        break;
```

Figura 123. Selección del canal del multiplexor.

DigitalWrite: Escriba un valor ALTO o BAJO en un pin digital. Si el pin se ha configurado como una SALIDA con pinMode (), su voltaje se establecerá en el valor correspondiente: 5V para ALTO, 0V (tierra) para BAJO, así seleccionamos la dirección del multiplexor.

```
75        case 'A':
76          Serial.println(myADE.getVRMS(PHASE_A)/constV);
77          Serial.println(myADE.getIRMS(PHASE_A)/constI);
78          if(myADE.setPotLine(PHASE_A)){
79            Pa=myADE.getPwatt(PHASE_A);
80            Qa=myADE.getQvar(PHASE_A);
81            Sa=myADE.getSva(PHASE_A);
82          }
83          Serial.println(Pa);
84          Serial.println(Qa);
85          Serial.println(Sa);
86          Serial.println(myADE.getFHz(PHASE_A)*constF);
87          freeRam();
88        break;
```

Figura 124. Envió de datos de la fase por el puerto serial.

Serial.println: Imprime datos en el puerto serie como texto ASCII legible por humanos seguido de un carácter de retorno de carro (ASCII 13 o '\ r') y un carácter de nueva línea (ASCII 10 o '\ n').

3.6 PYTHON COMUNICACIÓN SERIAL.

Python es un lenguaje de programación de alto nivel, interpretado, interactivo, orientado a objetos y de propósito general. Fue creado por Guido van Rossum durante 1985-1990. Al igual que Perl, el código fuente de Python también está disponible bajo la Licencia Pública General de GNU (GPL).

Import serial: Este módulo encapsula el acceso para el puerto serie. Proporcionada para Python que se ejecuta en Windows, OSX, Linux, BSD (posiblemente cualquier sistema compatible con POSIX) y IronPython.

El módulo serial se instala de la siguiente forma:

```
C:\Users\tuyuyos>pip install pyserial
Collecting pyserial
  Cache entry deserialization failed, e
  Using cached https://files.pythonhost
Installing collected packages: pyserial
Successfully installed pyserial-3.4
```

Figura 125. Instalación pyserial.

Import time: Hay un módulo de tiempo, disponible en Python que proporciona funciones para trabajar con los tiempos y para convertir entre representaciones.

```
def funcion01(Mdd,MD02):
        t01=['A','B','C']
        arduino = serial.Serial('/dev/ttyACM0', 9600)
        #arduino = serial.Serial('COM5',9600)
        time.sleep(1)
        arduino.write(str(Mdd))
        time.sleep(1)
```

Figura 126. Función de lectura serial.

Fucion01: recibe dos parámetros Mdd es el medidor seleccionado, MD02 es una lista que nos sirve para obtener el numero de elementos a guardar y esta lista es el valor de retorno de la función por quien fue llamado.

Serial: establecemos comunicación con el puerto serial, definimos el puerto a utilizar y la velocidad de transmisión en baudio.

El baudio (en inglés baud) es una unidad de medida utilizada en telecomunicaciones, que representa el número de símbolos por segundo en un medio de transmisión digital. Cada símbolo puede comprender 1 o más bits, dependiendo del esquema de modulación.

Time.sleep: suspende la ejecución por el número de segundos dado. El argumento puede ser un número de punto flotante para indicar un tiempo de suspensión más preciso.

Arduino.write: escribo en el puerto serial y le indicamos el medidor seleccionado, esperamos un segundo.

```
for i in t01:
        arduino.write(i)
        time.sleep(1)
        while arduino.inWaiting() > 0:
                valorA=arduino.read(1)
                if valorA=='\n':
                        MD02[k]=float(valorB)
                        valorB,k='',k+1
                        arduino.flush()
                else:
                        valorB+=valorA
arduino.close()
return MD02
```

Figura 127. Lectura del puerto serial.

Arduino.inWaiting: Obtener el número de bytes en el búfer de entrada si es mayor que cero continua la lectura.

Arduino.read: lectura del puerto serie, la catidad de bytes a leer se establecen en el valor que está en el paréntesis, nosotros estamos leyendo 1Byte.

valorA, valorB: valorA esta variable guarda las lecturas del puerto serie y se acumulan en valorB, y si la lectura es nueva línea (‘\n’) se guarda el valor en el vector de retorno.

Arduino.close: cerramos la comunicación serial.

Return: retornamos el valor de la lista MD02, a onde fue llamada la función.

3.7 PYTHON TKINTER.

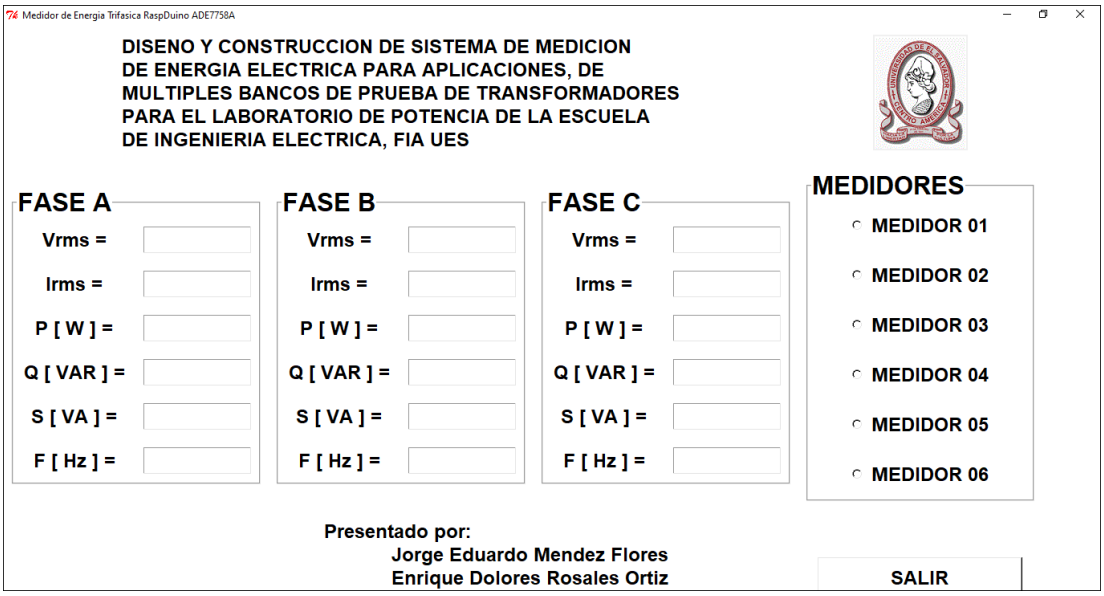


Figura 128. Interfaz Gráfica de Usuario.

Programación Tkinter, es la biblioteca de GUI (interfaz gráfica de usuario) estándar para Python. Python cuando se combina con Tkinter proporciona una forma rápida y fácil de crear aplicaciones GUI. Tkinter proporciona una potente interfaz orientada a objetos para el kit de herramientas GUI de Tk.

Todo lo que necesitas hacer es seguir los siguientes pasos:

- ✓ Importar el módulo Tkinter.
- ✓ Crear la ventana principal de la aplicación GUI.
- ✓ Agregue uno o más de los widgets a la aplicación GUI.
- ✓ Ingrese al ciclo principal del evento para tomar acción contra cada evento desencadenado por el usuario.

Modulo: Un módulo le permite organizar lógicamente su código Python. Agrupar el código relacionado en un módulo hace que el código sea más fácil de entender y usar. Un

módulo es un objeto de Python con atributos nombrados arbitrariamente que puede enlazar y hacer referencia.
Simplemente, un módulo es un archivo que consta de código Python. Un módulo puede definir funciones, clases y variables. Un módulo también puede incluir código ejecutable.

La declaración de import: Puede usar cualquier archivo fuente de Python como módulo ejecutando una declaración de importación en algún otro archivo fuente de Python.
Cuando el intérprete encuentra una declaración de "import", importa el módulo si el módulo está presente en la ruta de búsqueda. Una ruta de búsqueda es una lista de directorios que el intérprete busca antes de importar un módulo.

```
import time
import sys
```

Figura 129. Importación de módulo de Python.

Import time: Hay un módulo de tiempo, disponible en Python que proporciona funciones para trabajar con los tiempos y para convertir entre representaciones.

Import sys: Este módulo proporciona acceso a algunas variables utilizadas o mantenidas por el intérprete ya funciones que interactúan fuertemente con el intérprete. Siempre está disponible.

Por ejemplo, para importar el módulo SerialADE7758.py.

Figura 130. Python serial.

Se debe colocar el siguiente comando en la parte superior de la secuencia de comandos:

```
import SerialADE7758
```

Figura 131. Modulo serial.

La declaración from: Le permite importar atributos específicos de un módulo al espacio de nombres actual.

Esta declaración no importa todo el módulo "threading" en el espacio de nombres; simplemente introduce el elemento "Timer y Thread" del módulo "threading" en la tabla de símbolos globales del módulo de importación.

```
from threading import Timer,Thread
```

Figura 132. Modulo multihilo.

From name import*: También es posible importar todos los nombres de un módulo al espacio de nombres actual usando la siguiente declaración de importación.

```
if sys.version_info[0] < 3:
        from Tkinter import*
else:
        from tkinter import*
```

Figura 133. Selección de la versión de Python.

Utilizando "import sys", podemos saber la versión de Python que estamos utilizando, nuestro programa puede correr con versiones de Python 2.x.x o Python 3.x.x

```
>>> import sys
>>> sys.version
'2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MSC v.1500 32 bit (Intel)]'
>>> sys.version_info[0]
2
>>>
```

Figura 134. Versión de Python 2.7.14.

```
>>> import sys
>>> sys.version
'3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)]'
>>> sys.version_info[0]
3
>>>
```

Figura 135. Versión de Python 3.7.1.

Class: Un prototipo definido por el usuario para un objeto que define un conjunto de atributos que caracterizan cualquier objeto de la clase. Los atributos son miembros de datos (variables de clase y variables de instancia) y métodos, a los que se accede mediante notación de puntos.

Class variable: Una variable que comparten todas las instancias de una clase. Las variables de clase se definen dentro de una clase pero fuera de cualquiera de los métodos de la clase. Las variables de clase no se utilizan tan frecuentemente como las variables de instancia.

El primer método __init __ (): Es un método especial, que se llama constructor de clase o método de inicialización al que Python llama cuando crea una nueva instancia de esta clase.

Self: usted declara otros métodos de clase como las funciones normales, con la excepción de que el primer argumento de cada método es "self.".

Self.Master: método de inicialización dando el parámetro de tk().

Self.MASTER.protocol(): utilizamos para que antes de cerrarla ventana llame a una función y se puede cerrar los métodos utilizados.

MASTER.title: título de la interfaz grafica de usuario.

MASTER.configure: definimos color de fondo, en RGB hexadecimal.

```
class Ventana:
    def __init__(self, MASTER):
        self.MASTER = MASTER
        self.MASTER.protocol("WM_DELETE_WINDOW", self.cancel)
        self.MASTER.title("Medidor de Energia Trifasica RaspDuino ADE7758A")
        self.MASTER.configure(bg ='#333333')
```

Figura 136. Clase de GUI.

Self.thread: variable utilizada para el multiprocesamiento e inicializada, con la palabra reservada de Python None (ninguno).

Self.var: variable utilizada para los RadioButton, variable tipo entero.

Cl01: color de letra utilizado para los LabelFrame, utilizaremos una cadena que especifique la proporción de rojo, verde y azul en dígitos hexadecimales. Por ejemplo, "#fff" es blanco, "# 000000" es negro, "# 000fff000" es verde puro, y "# 00ffff" es cian puro (verde más azul).

Cl02: color de letra utilizado en los Label y Entry.

Cf01: color de fondo utilizado en los Label y Entry.

Font01 y font02: tipo de fuente utilizado.

```
self.thread = None
self.var = IntVar()
cl01="#c8f945" #color letra
cl02="#000000" #color letra
cf01="#4D4D4D" #color fondo
font01="Helvetica 24 bold"
font02="Helvetica 16 bold"
```

Figura 137. Tipo de fuente y color.

Las listas son buenas para hacer un seguimiento de las cosas por su orden, especialmente cuando el orden y los contenidos pueden cambiar A diferencia de las cadenas, las listas son mutables. Puede cambiar una lista en el lugar, agregar nuevos elementos y eliminar o sobrescribir elementos existentes. El mismo valor puede ocurrir Más de una vez en una lista.

Una lista está hecha de cero o más elementos, separados por comas y rodeados por corchetes:

Self.texto01: lista utilizada para definir la cantidad de LabelFrame.
Self.texto02: lista utilizada para definir la cantidad de Label.
Self.texto03: lista utilizada para definir la cantidad de RadioButton.
Self.VctLblFrm: esta lista guarda cada una de los LabelFrame creados para la interfaz gráfica y poder agregar los Label y Entry correspondientes.
Self.VctEntry: guarda cada uno de los Entry creados, para poder modificar su valor.

```
self.texto01=["FASE A","FASE B","FASE C","MEDIDORES"]
self.texto02= ['Vrms =','Irms =','P [ W ] =','Q [ VAR ] =','S [ VA ] =','F [ Hz ] =']
self.texto03=["MEDIDOR 01","MEDIDOR 02","MEDIDOR 03","MEDIDOR 04","MEDIDOR 05","MEDIDOR 06"]
self.VctLblFrm=self.texto01# vector label frame
self.VctEntry=(len(self.texto01)-1)*self.texto02# vector Entry
```

Figura 138. Listas principales de la interfaz gráfica.

Creamos los LabelFrame, con el ciclo for que recorre self.texto01 y lo guardamos en su respectiva lista, le agregamos atributos de:

Text: etiqueta del LabeFrame.
Font: tipo de fuente a utilizar.
Bg: color de fondo.
Fg: color del texto.
Grid: el organizador geométrico donde definimos fila y columna correspondiente.
Row: la fila para colocar el elemento por defecto la primera fila que está vacía.
Column: la columna para colocar el elemento por defecto la columna más vacía a la izquierda.

```
cl=0 # columna
for self.j in self.texto01:
      self.VctLblFrm[cl]= LabelFrame(MASTER, text=self.j,font=font01,bg=cf01,fg=cl01)
      self.VctLblFrm[cl].grid(column=cl,row=0,padx=10,pady=10)
      cl=cl+1
```

Figura 139. Guía del marco.

Label: Este widget implementa un cuadro de visualización donde puede colocar texto o imágenes. El texto mostrado por este widget se puede actualizar en cualquier momento que desee.

También es posible subrayar parte del texto (como identificar un método abreviado de teclado) y abarcar el texto en varias líneas.

Widget: es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.

Padx y pady: cuántos píxeles para rellenar el widget, horizontal y verticalmente, afuera de los bordes widget.

```
self.LB=Label(self.VctLblFrm[cl], text=self.j,font=font02,bg=cf01,fg=cl02)
self.LB.grid(column=0,row=fl,padx=10,pady=10)
```

Figura 140. Widget Label.

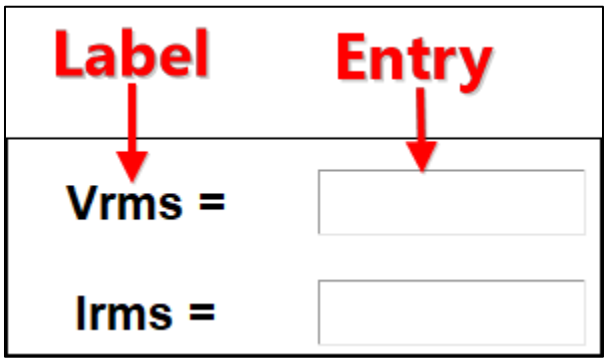


Figura 141. Widget Label y Widget Entry.

Entry: El widget de entrada se utiliza para aceptar cadenas de texto de una línea de un usuario. Nosotros lo utilizaremos para mostrar datos y guardaremos los Entry creados en self.VctEntry.

Self.VctEntry[N].insert(0,i): Inserta la cadena "i" caracteres antes del carácter en el índice dado.

```
self.VctEntry[N]=Entry(self.VctLblFrm[cl],width=10,font=font02,bg=cf01,fg=cl02)
self.VctEntry[N].grid(column=1,row=fl,padx=10,pady=10)
self.VctEntry[N].insert(0,0.0)
```

Figura 142. Widget Entry.

El primer ciclo for recorrerá la longitud de la lista self.texto01 menos uno para agrupar los widget al LabelFrame.

El segundo ciclo for recorrerá la lista self.texto02, y creara los widget Label y Entry, los agrupara a su respectivo LabelFrame.

Fl y cl: son variable que representan las filas y columnas, las utilizamos para ordenar los widget.

```
cl,fl,N=0,0,0
for i in range(len(self.texto01)-1):
      for self.j in self.texto02:
            self.LB=Label(self.VctLblFrm[cl], text=self.j,font=font02,bg=cf01,fg=cl02)
            self.LB.grid(column=0,row=fl,padx=10,pady=10)
            self.VctEntry[N]=Entry(self.VctLblFrm[cl],width=10,font=font02,bg=cf01,fg=cl02)
            self.VctEntry[N].grid(column=1,row=fl,padx=10,pady=10)
            self.VctEntry[N].insert(0,0.0)
            fl,N=fl+1,N+1
      cl=cl+1
```

Figura 143. Ciclos anidados.

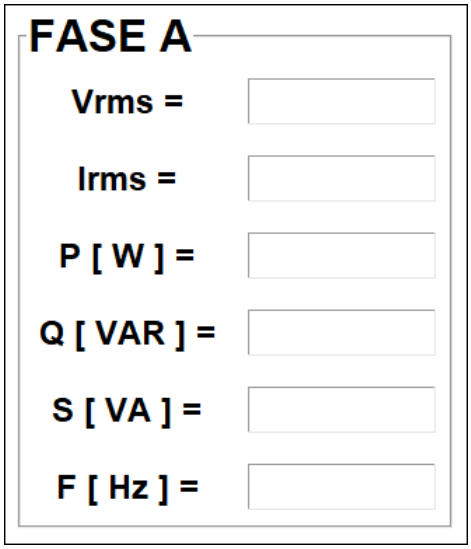


Figura 144. Widget utilizando el LabelFrame, Label y Entry.

RadioButton: Este widget implementa un botón de opción múltiple, que es una forma de ofrecer muchas selecciones posibles al usuario y le permite al usuario elegir solo una de ellas.

Para implementar esta funcionalidad, cada grupo de botones de radio debe estar asociado a la misma variable (variable=self.var) y cada uno de los botones debe simbolizar un solo valor (value=fl iniciando con uno y terminado con seis, esto con respecto al número de medidores).

Command: Un procedimiento a ser llamado cada vez que el usuario cambia el estado del RadioButton, lo utilizamos para llamar a la función self.start().

ipadx e ipady: Cuántos píxeles para rellenar el widget, horizontal y verticalmente, dentro de los bordes del widget.

```
fl=1
for self.j in self.texto03:
      self.Rdbttn = Radiobutton(self.VctLblFrm[3], text=self.j,
        font=font02,bg=cf01,fg=cl02,variable=self.var, value=fl,command=self.start)
      self.Rdbttn.grid(column=0,row=fl,padx=10,pady=10,ipadx=40)
      fl+=1
```

Figura 145. Widget Radio Button.

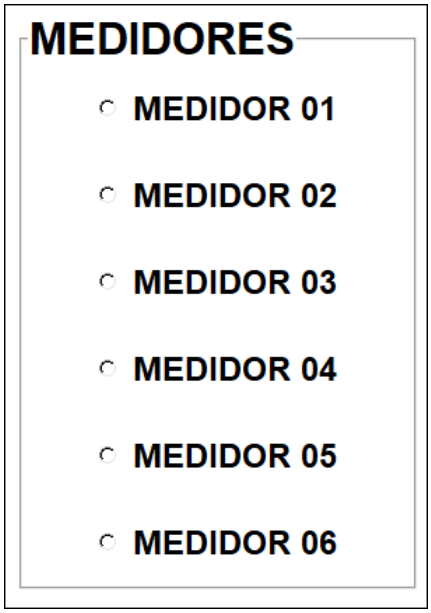


Figura 146. Label Frame y Radio Button.

Button: El widget de botón se usa para agregar botones en una aplicación de Python. Estos botones pueden mostrar texto o imágenes que transmiten el propósito de los botones. Puede adjuntar una función o un método a un botón que se llama automáticamente al hacer clic en el botón.

```
self.Bsalir = Button(MASTER, text="SALIR",font=font02,bg=cf01,fg=cl02,command=self.cancel)
self.Bsalir.grid(column=2,row=1,padx=10,pady=5,ipadx=80)
```

Figura 147. Widget Button salir.

Ejecutar varios subprocesos es similar a ejecutar varios programas diferentes al mismo tiempo, pero con los siguientes beneficios:

Varios subprocesos dentro de un proceso comparten el mismo espacio de datos con el subproceso principal y, por lo tanto, pueden compartir información o comunicarse entre sí más fácilmente que si fueran procesos separados.

Los subprocesos a veces se denominan procesos ligeros y no requieren mucha sobrecarga de memoria;

Self.thread.start: El método start inicia un hilo llamando al método run.

Run: El método run () es el punto de entrada para un hilo.

El modo más sencillo para usar un hilo es instanciar un objeto de la clase Thread con una función objetivo y hacer una llamada a su método start().

```
def start(self):
    if self.thread == None:
        self.thread = Thread(target=self.multiHilo)
        self.thread.start()
```

Figura 148. Inicio del multihilo.

La función multihilo llama al módulo SerialADE7758 y a la funcio01 para establecer una comunicación serial con arduino, mandamos dos parámetros el primero el medidor seleccionado y el otro parámetro es la creación de una lista con el número de valores de la medición, el valor de retorno son los valores de las mediciones.

Self.VctEntry[N].delete(0,END): Borra los caracteres del widget, comenzando con el del índice primero, hasta el último. Si se omite el segundo argumento, solo se borra el único carácter en la posición primero.

Self.VctEntry[N].insert(0,i): Inserta la cadena "i" caracteres antes del carácter en el índice dado, los valores insertados son las mediciones realizadas.

Timer: llama a la función multihilo cada diez segundos, actualizando los datos de las mediciones. Esta es iniciada con el método start.

```
def multiHilo(self):
        N=0
        a=SerialADE7758.funcion01(self.var.get(),(len(self.texto01)-1)*self.texto02)
        for i in a:
          self.VctEntry[N].delete(0,END)
          self.VctEntry[N].insert(0,i)
          N=N+1
        print("Medidor="),str(self.var.get())
        self.thread = Timer(10,self.multiHilo)
        self.thread.start()
```

Figura 149. Función multihilo.

La función cancel desaparece mainloop o detiene el ciclo infinito, el if es para cancelar el Timer si este ha sido iniciado.

```
def cancel(self):
        if self.thread != None:
                self.thread.cancel()
        self.MASTER.destroy()
```

Figura 150. Función destructora.

Creación de instancia a objetos.

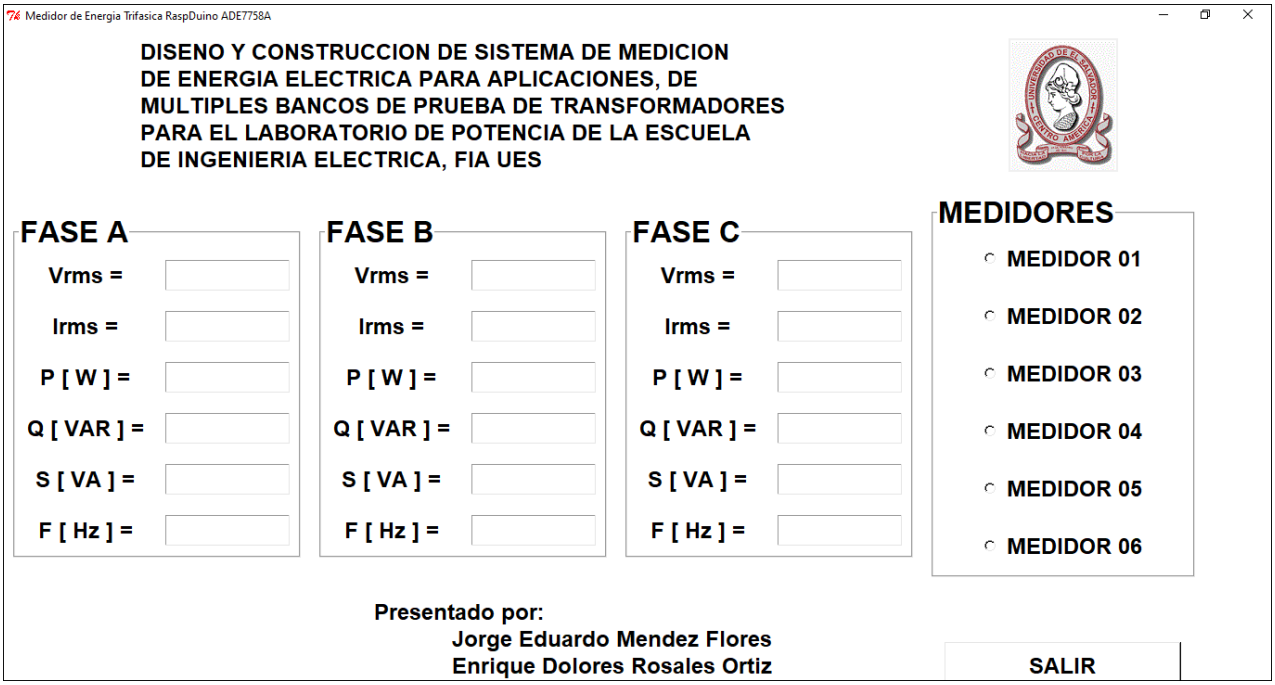
Crearemos una instancia de la clase Ventana, llame a la clase usando el nombre de la clase y pase los argumentos que acepte el método __init__. Para nuestro caso ROOT.

If__name__=='__main__': Esto está ligado al modo de funcionamiento del intérprete Python, cuando el intérprete lee un archivo de código, ejecuta todo el código que se encuentra en él. Todo módulo en python tiene un atributo especial llamado __name__ que define el espacio de nombres en el que se está ejecutando. Es usado para identificar de forma única un módulo en el sistema de importaciones.
Por su parte __main__ es el nombre del ámbito en el que se ejecuta el código de nivel superior (tu programa principal).

ROOT.mainloop: ciclo infinito de la interfaz gráfica.

```
if __name__ == '__main__':
    ROOT = Tk()
    miVentana = Ventana(ROOT)
    ROOT.mainloop()
```

Figura 151. Ciclo infinito de la interfaz gráfica de usuario.



4 MANUAL DE USUARIO, PUEBAS DE MEDICIÓN Y PRESUPUESTO.

4.1 MANUAL DE USUARIO DEL MEDIDOR TRIFÁSICO MULTIPLEXADO.

A continuación, en la Figura 152, se muestra el medidor trifásico multiplexado, los seis medidores y sus correspondientes borneras para cada fase.

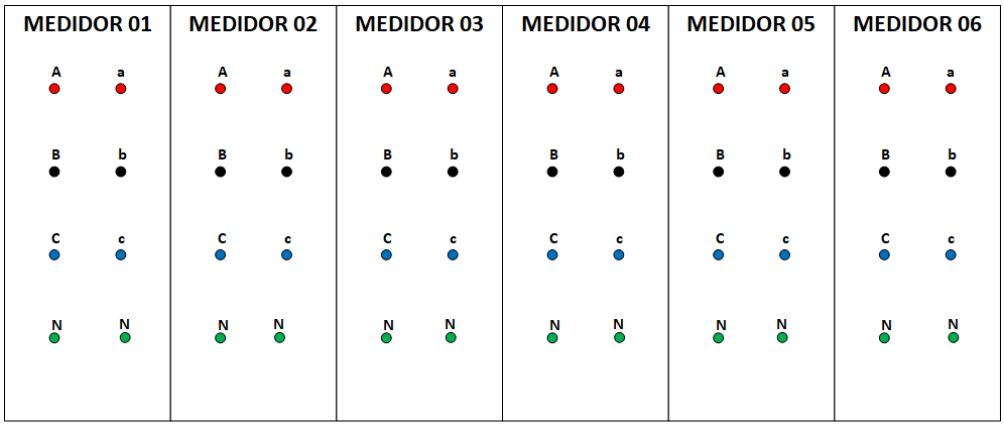


Figura 152. Medidor trifásico multiplexado.

Como se muestra en la Figura 153, la alimentación se conecta en las borneras de la izquierda (vista de frente) y la carga se conecta en las borneras de la derecha.

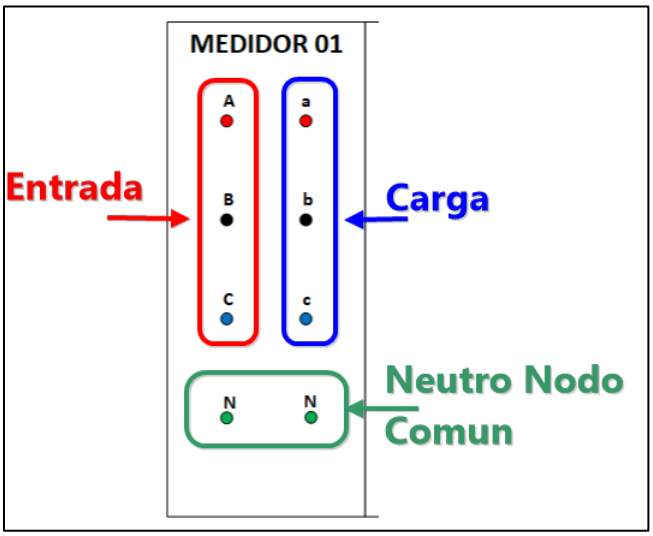


Figura 153. Descripción del Medidor 01.

Pasos para utilizar el medidor:

1- Conecte el conector de alimentación de 120Vac que se encuentra en el lado derecho.
2- Conecte el conector HDMI al equipo y al medidor.
3- Active el interruptor que se encuentra en el lado derecho del medidor.
4- Cuando se encuentre iniciado el sistema operativo Raspbian de click derecho en el icono del rayo ejecute el programa.

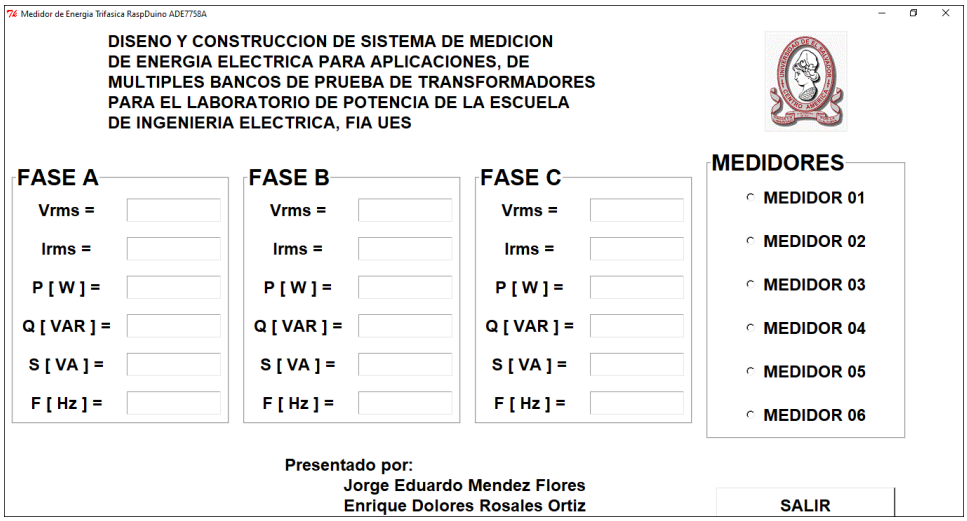


Figura 154. Interfaz Gráfica de Usuario.

5- Verifique que el Breaker de la alimentación trifásica se encuentre desactivado (se recomienda realizarlo para modificaciones en la conexión).
6- Realice las conexiones como se muestra en la Figura 155, para conectar el motor trifásico en estrella (Breaker en off), verificar conexión activar el Breaker y subir el voltaje lentamente y revisar las mediciones en la interfaz gráfica de usuario.

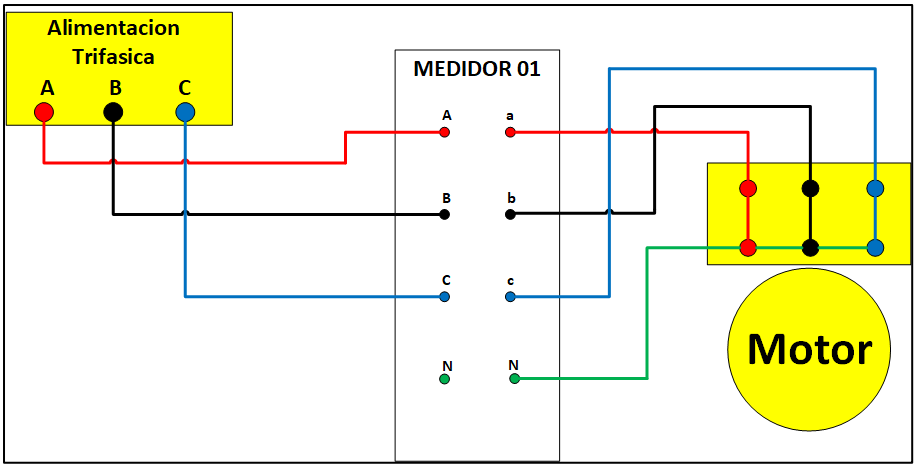


Figura 155. Motor conectado en estrella Yn.

7- Conexión de una carga monofásica, antes de iniciar las conexiones verifique que el Breaker este off, realizamos las conexiones como se muestran en la Figura 156, verificamos y activamos el Breaker.

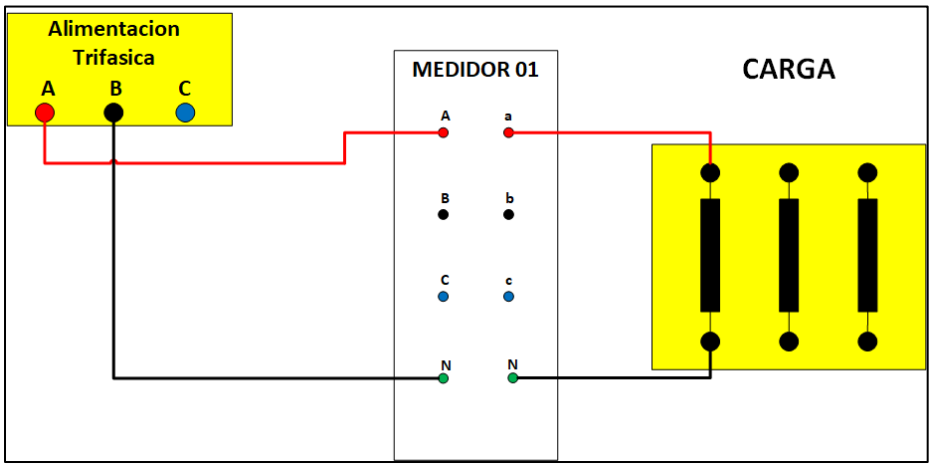


Figura 156. Conexión de una carga monofásica.

8- Conexión de dos cargas monofásicas utilizando un medidor, conectaremos la fase a para la alimentación de las dos cargas y la fase b la conectaremos en el neutro del medidor, utilizaremos el medidor 1 fase a y b como se muestra en la Figura 157, después daremos clic en el medidor que deseamos para ver la lectura de datos.

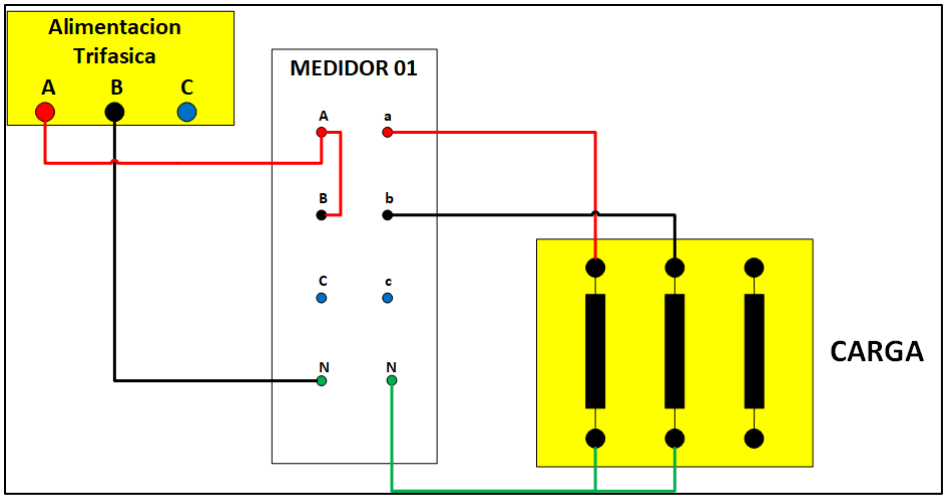


Figura 157. Conexiones de dos cargas monofásicas.

9- Conexión de dos cargas monofásicas utilizando dos medidores, conectaremos la fase a para la alimentación de las dos cargas y la fase b la conectaremos en el neutro del medidor, utilizaremos el medidor 1 fase a y el medidor2 fase b como se muestra en la Figura 158, después daremos clic en el medidor que deseamos para ver la lectura de datos.

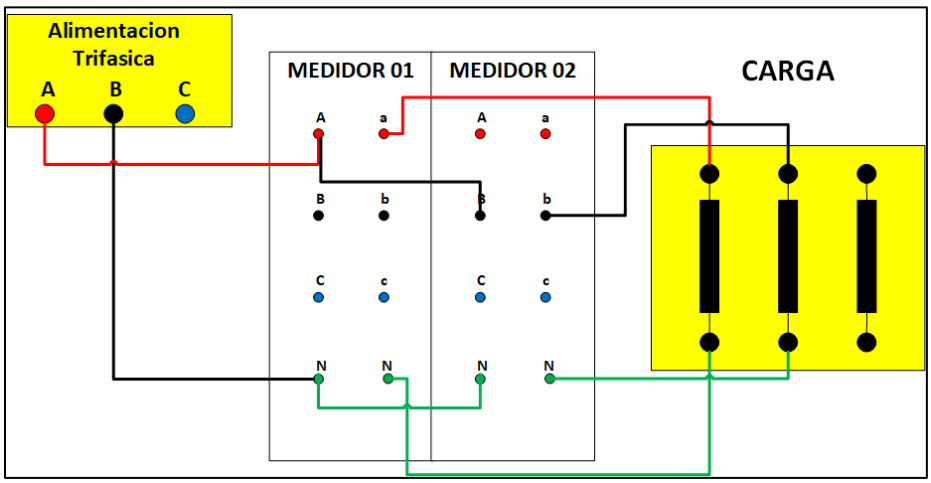


Figura 158. Conexión de dos cargas monofásicas.

10- Realizaremos una conexión trifásica como se muestra en la Figura 159 verificar las conexiones activar el Breaker dar click en el medidor uno en la interfaz gráfica.

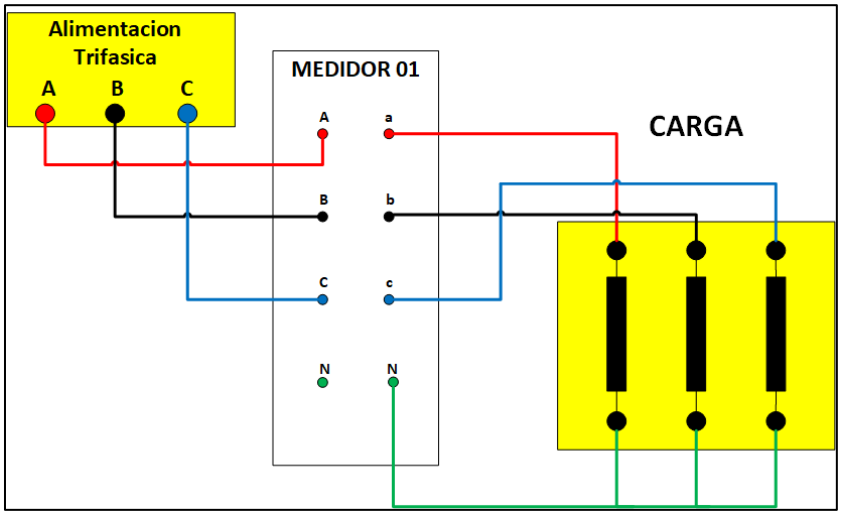


Figura 159. Conexión de una carga trifásica.

4.2 ANÁLISIS DE FALLAS.

Daño de la interfaz gráfica de usuario: la raspberry pi 3b+ tiene en sus archivos una copia de seguridad de todos los programas solo se copia la carpeta y se sustituye.

Lecturas erróneas: las lecturas de los datos son valores muy grandes en las tres fases, posible causa daño en el Arduino, apague todo el equipo y reinícielo. La falla persiste posible daño en el Arduino hay que cambiar el Arduino se realiza el cambio y en las raspberry pi 3b+ se encuentra el programa del Arduino y se sube el programa.

Nota: el cambio del Arduino hay que revisar q tipo de Arduino es el que estamos colocando, hay un Arduino de fabricación china el cual no establece comunicación serial con las raspberry pi 3b+.

Lecturas erróneas en diferentes lecturas: al obtener este error se encuentra en la comunicación SPI posibles causas un alambre con soldadura fría, un alambre suelto o no hay continuidad en un cable de comunicación se remplaza dicho cable.

Una fase no muestra voltaje: todas las fases A pasan las pequeñas señales por un multiplexor posible daño del multiplexor se remplaza multiplexor, sin un multiplexor de

voltaje se daña no mostraría datos de corriente para realizar una lectura de corriente se toma en cuenta los cruce por cero del voltaje.

Una fase no muestra corriente: sin un multiplexor de corriente se daña no muestra corriente en todos los medidores en la misma fase, tampoco mostraría los datos de potencia, pero si lecturas de voltaje.

4.3 PRUEBAS DE MEDICION.

Para las pruebas de medición se utiliza la comparación con medidor patrón el cual se efectúa la comparación con el medidor de calidad de energía para redes monofásicas AEMC 8230, el cual se muestra en la Figura 160 y dispone de las características descritas en la Figura 161, lo cual nos brinda los resultados en la Tabla 4, teniendo nuestro medidor trifásico una medición aceptable.

Figura 160. Medidor patrón AEMC 8230.

MODELO	8230
CARACTERÍSTICAS ELÉCTRICAS	
Tensión (TRMS)	Fase - Fase: 660 V Fase - Neutro: 600 V
Corriente (TRMS)	Sonda MN: 5 mA a 6 A/120 A ó 2 a 240 A Sonda MR: 10 a 1000 Aca, 10 a 1400 Acc Sonda SR: 3 a 1200 A Sensor MiniFlex®: 0,1 a 1000 A Sensor AmpFlex®: 10 a 6500 A(1)
Frecuencia (Hz)	40 a 70 Hz
Otras mediciones	kW, kVAR, factor de potencia (PF), factor de potencia de desplazamiento (DPF), kWh, kVARh, kVAh, factor K, flicker, ángulo de fase de armónico, secuencia de fase
Armónicos	THD-R, THD-F, V, A, VA 1º a 50º, dirección, secuencia
Frecuencia de muestreo	256 muestras/ciclo
Almacenamiento de datos	1,5 MB con partición para formas de ondas, alarmas y registro de tendencias
Fuente de alimentación	Baterías NiMH recargables (incluidas) Adaptador de corriente CA: 120/230 Vca (50/60 Hz)
Autonomía de la batería	≥8 horas con la pantalla encendida ≥40 horas con la pantalla apagada (modo de registro)
CARACTERÍSTICAS MECÁNICAS	
Puerto de comunicaciones	USB con aislamiento óptico
Pantalla	LCD a color ¼ VGA (320 x 240)
Dimensiones	211 x 108 x 60 mm (8,3 x 4,3 x 2,4 pulg.)
Peso	0,88 kg (1,9 lbs)
Clasificación de seguridad	EN 61010, 600 V CAT III, Grado de contaminación ambiental 2

Figura 161. Características técnicas del medidor AEMC 8230.

	Medidor AEMC Model 8230					Trabajo de graduación.					Tipo de carga
Medición	VA	IA	PA	QA[VAR}	SA	VA	IA	PA	QA	SA	
1	13.8	1.1	15.1	1.6	15.2	14.42	1.11	10	1	10	**Carga Resistiva**
2	15.5	1.2	19.1	2	19.2	16.04	1.25	12	1	13	Carga Resistiva
3	18	1.4	25.7	2.4	25.9	18.36	1.45	16	2	17	Carga Resistiva
4	21.2	1.7	35.8	2.5	35.9	21.41	1.7	23	2	24	Carga Resistiva
5	25.6	2	52.2	3	52.3	25.7	2.04	34	3	24	Carga Resistiva
6	27.8	2.2	61.3	3.7	61.2	27.74	2.2	39	3	40	Carga Resistiva
7	30.3	2.4	73.6	4.2	73.6	30.25	2.4	47	4	48	Carga Resistiva
8	33.3	2.7	90.4	4.2	90.6	33.2	2.63	57	4	58	Carga Resistiva
9	37.4	3.1	114.4	5.2	114.3	37.19	2.97	72	6	73	Carga Resistiva
10	12.5	1.1	12.7	3.9	13.3	13.2	1.09	8	-1	8	Carga LR
11	20.1	1.7	33.3	9.8	24.7	20.4	1.73	22	-4	23	Carga LR
12	25	2.1	51.2	15.3	53.5	25.5	2.14	34	-6	35	Carga LR
13	29.9	2.6	74.7	22.8	78	29.84	2.54	49	-9	50	Carga LR
14	35	3.1	103.5	33	108.6	34.87	3	67	-15	68	Carga LR
15	38.5	3.4	125.4	42.8	132.8	38.22	3.33	81	-20	84	Carga LR
16	14.3	1.2	16.4	4.9	7.1	14.85	1.22	11	3	12	Carga CR
17	20.3	1.7	23.1	9.5	34.5	20.53	1.71	20	8	22	Carga CR
18	25.4	2.1	51.7	14.7	53.8	25.47	2.12	32	13	35	Carga CR
19	30.2	2.6	74.4	21.3	77.4	30.16	2.51	47	18	49	Carga CR
20	34.9	3	101	28.8	105.1	34.86	2.91	62	23	67	Carga CR
21	40.2	3.4	132.3	38	138	39.91	3.34	82	31	87	Carga CR

Tabla 4. Comparación de mediciones, Medidor AEMC 8230 vs Medidor desarrollado en trabajo de graduación.

4.4 PRESUPUESTO DE LA INVERSIÓN.

En la tabla 5 se muestra la inversión para el diseño y construcción de sistema de medición de energía eléctrica para aplicaciones, de múltiples bancos de prueba de transformadores para el laboratorio de potencia de la escuela de ingeniería eléctrica, FIA-UES.

Item	Cantidad	Descripción	Precio Unit	Costo de Envio Total	Total	Forma de Envio
1	1	Raspberry PI 3B+ PLUS	$ 62.00	$ 50.00	$ 112.00	Transxpress / Amazon
2	1	Arduino Mega	$ 20.00	$ -	$ 20.00	Compra Local
3	2	IC ADE7758	$ 20.00	$ 10.00	$ 50.00	Transxpress / Amazon
4	10	Cristal 10MHz	$ 0.50	$ 5.00	$ 10.00	Transxpress / Amazon
5	10	Capacitor 22pF	$ 0.50	$ 5.00	$ 10.00	Transxpress / Amazon
6	10	Capacitor 10uF	$ 0.50	$ 5.00	$ 10.00	Transxpress / Amazon
7	10	Capacitor 100nF	$ 0.50	$ 5.00	$ 10.00	Transxpress / Amazon
8	10	Capacitor 33nF	$ 0.50	$ 5.00	$ 10.00	Transxpress / Amazon
9	19	Resistencias 1M ohm	$ 0.30	$ -	$ 5.70	Compra Local
10	26	Resistencias 1k ohm	$ 0.30	$ -	$ 7.80	Compra Local
11	10	Resistencia de 10 ohm	$ 0.30	$ -	$ 3.00	Compra Local
12	8	IC 47HC4051	$ 8.94	$ 30.00	$ 101.52	Transxpress / Amazon
13	10	Bases de I.C.	$ 0.55	$ -	$ 5.50	Compra Local
14	18	Borner hembra negro	$ 0.35	$ -	$ 6.30	Compra Local
15	18	Borner hembra rojo	$ 0.35	$ -	$ 6.30	Compra Local
16	3	Cable THHN #12 rojo	$ 0.39	$ -	$ 1.17	Compra Local
17	3	Cable THHN #12 negro	$ 0.39	$ -	$ 1.17	Compra Local
18	3	Cable THHN #12 verde	$ 0.39	$ -	$ 1.17	Compra Local
19	6	Cable blindado 5 Hilos	$ 7.00	$ -	$ 42.00	Compra Local
20	36	Conector de ojo M6#14-16 amarillo	$ 0.38	$ -	$ 13.68	Compra Local
21	4	Circuitos Impresos ADE7758	$ 5.00	$ 20.00	$ 40.00	DHL Express
22	4	Circuito Impresos Voltaje	$ 5.00	$ 20.00	$ 40.00	DHL Express
23	4	Circuito Impresos Corriente	$ 5.00	$ 20.00	$ 40.00	DHL Express
24	1	Gabinete	$ 60.00	$ -	$ 60.00	Compra Local
25	1	Juego de brocas	$ 4.95	$ -	$ 4.95	Compra Local
26	1	Cautin	$ 21.95	$ -	$ 21.95	Compra Local
27	1	Estaño	$ 5.00	$ -	$ 5.00	Compra Local
28	1	Pasta	$ 1.95	$ -	$ 1.95	Compra Local
29	1	Flux	$ 10.00	$ -	$ 10.00	Compra Local
30	20	Base adhesiva	$ 0.15	$ -	$ 3.00	Compra Local
31	20	Cinchos plasticos pequeños	$ 0.10	$ -	$ 2.00	Compra Local
32	18	SCT-013-000 , Transformador de nucleo partido de 100A / 50 mA	$ 5.50	$ 5.00	$ 104.00	Correo Nacional/ Ebay
					$ 760.16	

Tabla 5. Presupuesto, materiales y dispositivos.

Cabe destacar que en algunos casos se elevó el costo debido a las formas de importación a la que se recurrieron que en base a la experiencia pueden reducirse con una mejor planificación e investigación de formas de envío de productos a nuestro país.

5 CONCLUSIONES Y RECOMENDACIONES.

5.1 CONCLUSIONES.

A. La compañía Analog Devices, es líder en el desarrollo de Circuitos Integrados de procesamiento de señales análogas, digitales y mixtas, con un numero extenso de productos que recaen en múltiples industrias lo cual hace desarrollar el diseño de diversidad de aplicaciones como en nuestro caso el de la medición de energía.

B. El desarrollo de las comunicaciones Chip to Chip como la Serial Peripheral Interface, la cual hemos utilizado para la comunicación con nuestra plataforma Arduino, nos permite desarrollar de bloque en bloque la interrelación de diversos dispositivos, lo que nos permite crear proyectos modulares con dispositivos desarrollados por diversos fabricantes y unificados por un protocolo de comunicación para el manejo de los datos.

C. La plataforma de Hardware Libre Raspberry Pi, nos permitió contar con una computadora portátil con los recursos necesarios para el desarrollo de nuestra interfaz gráfica, ahorrando inversión en una computadora tradicional y disminuyendo en gran medida el espacio, además cabe mencionar que esta posee comunicación Chip to Chip SPI y I2C lo cual permite conectarse a otros dispositivos.

D. La interfaz gráfica de usuario se diseñó flexible de tal forma que se le pueda agregar o quitar elementos, que esta dependa de las mediciones a realizar, así como el número de medidores, se tomó en cuenta la memoria de la Raspberry Pi, para que no utilizara muchos recursos.

5.2 RECOMENDACIONES.

A. Para tener una mayor opción a las configuraciones que brinda el ADE7758 se debe de diseñar un sistema de multiplexado que permita la independencia de todos los pines de entrada del ADE7758 para tener la posibilidad de selección del tipo de red ya sea esta Delta o Estrella de 3 o 4 hilos.

B. El ser la Raspberry Pi un sistema más completo ya que cuenta con la ventaja de un sistema operativo desarrollado, con acceso al control de las entradas y salidas de propósito general GPIO y contar con comunicaciones Chip to Chip como SPI e I2C, se recomienda trabajarlo directamente comunicado al ADE7758 para mejorar el tiempo de respuesta para la lectura de datos.

C. Se recomienda para el desarrollo del envolvelmente (Case o Enclousure) donde van los dispositivos, sea normado con fabricación industrial que posea la eliminación de ruido, ya que puede generar distorsión en las pequeñas señales utilizadas.

D. Dado a los problemas generados por señales parasitas y a la cantidad de datos que se están manipulando en las comunicaciones Chip to Chip como SPI e I2C, se recomienda que todos los componentes sean implementados en una única tarjeta de circuito impreso, ya que utilizar cables generar perdida y aumento en la distorsión de la señal, en nuestro caso se solventó utilizando cableado con blindaje y colocando los dispositivos a distancias muy cortas.

REFERENCIAS

[1] Analog Devices, «Analog Device», mayo 2018. [En línea]. Available: https://www.analog.com/media/cn/technical-documentation/data-sheets/ADE7758.pdf. [Último acceso: 8 febrero 2020].

[2] R. B. Galán Parras," Estudio de demanda energética utilizando software y hardware libre en el Edificio de Ingeniería Industrial, UES.", Universidad de El Salvador, San Salvador, El Salvador, 2014.

[3] J. O. Murcia Flores, "Equipo concentrador de vatímetros para uso en el sector doméstico e industrial", Universidad de El Salvador, San Salvador, El Salvador, 2014.

[4] Tim Cox, April 2014, Raspberry Pi Cookbook for Python Programmers.

[5] Bill Lubanovic, 20-02-2015, Introducing Python.

[6] Burkhard A. Meier, November 2015, Python GUI Programming Cookbook.

[7] Mike Ohlson de Fine, November, 2010, Python 2.6 Graphics Cookbook.

[8] Simon Monk, December 2013, Raspberry Pi Cookbook.

[9] Arduino, «Arduino», mayo 2018. [En línea]. Available: https://www.arduino.cc/en/reference/SPI, [Último acceso: 8 febrero 2020].

ANEXOS

74HC4051; 74HCT4051

8-channel analog multiplexer/demultiplexer

Rev. 9 — 26 September 2017 Product data sheet

1 General description

The 74HC4051; 74HCT4051 is a single-pole octal-throw analog switch (SP8T) suitable for use in analog or digital 8:1 multiplexer/demultiplexer applications. The switch features three digital select inputs (S0, S1 and S2), eight independent inputs/outputs (Yn), a common input/output (Z) and a digital enable input ($\overline{E}$). When $\overline{E}$ is HIGH, the switches are turned off. Inputs include clamp diodes. This enables the use of current limiting resistors to interface inputs to voltages in excess of V_{CC}.

2 Features and benefits

- Wide analog input voltage range from -5 V to +5 V
- Complies with JEDEC standard no. 7A
- Low ON resistance:
  - 80 Ω (typical) at V_{CC} - V_{EE} = 4.5 V
  - 70 Ω (typical) at V_{CC} - V_{EE} = 6.0 V
  - 60 Ω (typical) at V_{CC} - V_{EE} = 9.0 V
- Logic level translation: to enable 5 V logic to communicate with ±5 V analog signals
- Typical ‘break before make’ built-in
- ESD protection:
  - HBM JESD22-A114F exceeds 2000 V
  - MM JESD22-A115-A exceeds 200 V
  - CDM JESD22-C101E exceeds 1000 V
- Multiple package options
- Specified from -40 °C to +85 °C and -40 °C to +125 °C

3 Applications

- Analog multiplexing and demultiplexing
- Digital multiplexing and demultiplexing
- Signal gating

nexperia

4 Ordering information

Table 1. Ordering information

Type number	Package			
	Temperature range	Name	Description	Version
74HC4051D	-40 °C to +125 °C	SO16	plastic small outline package; 16 leads; body width 3.9 mm	SOT109-1
74HCT4051D				
74HC4051DB	-40 °C to +125 °C	SSOP16	plastic shrink small outline package; 16 leads; body width 5.3 mm	SOT338-1
74HCT4051DB				
74HC4051PW	-40 °C to +125 °C	TSSOP16	plastic thin shrink small outline package; 16 leads; body width 4.4 mm	SOT403-1
74HCT4051PW				
74HC4051BQ	-40 °C to +125 °C	DHVQFN16	plastic dual in-line compatible thermal enhanced very thin quad flat package; no leads; 16 terminals; body 2.5 x 3.5 x 0.85 mm	SOT763-1
74HCT4051BQ				

5 Functional diagram

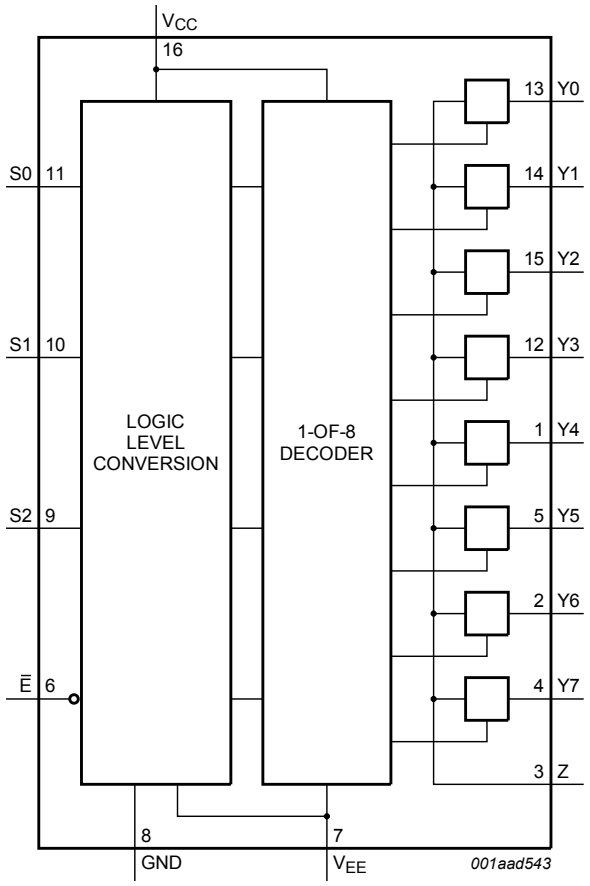


Figure 1. Functional diagram

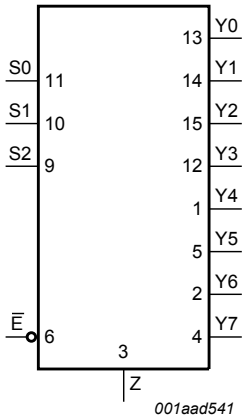


Figure 2. Logic symbol

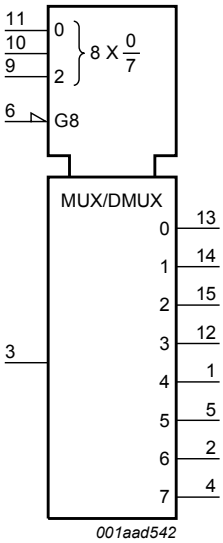


Figure 3. IEC logic symbol

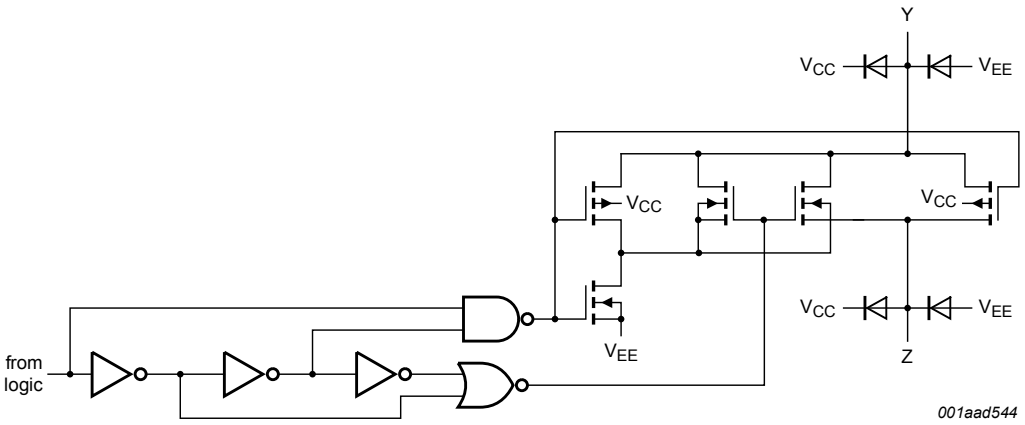


Figure 4. Schematic diagram (one switch)

6 Pinning information

6.1 Pinning

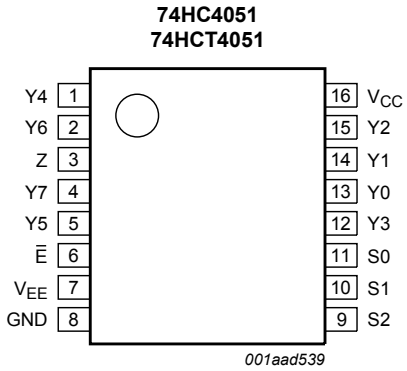
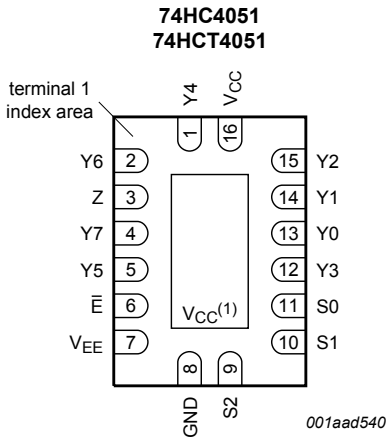


Figure 5. Pin configuration SO16, and (T)SSOP16



Transparent top view

(1) This is not a supply pin. The substrate is attached to this pad using conductive die attach material. There is no electrical or mechanical requirement to solder this pad.However, if it is soldered, the solder land should remain floating or be connected to V_{CC}.

Figure 6. Pin configuration DHVQFN16

6.2 Pin description

Table 2. Pin description

Symbol	Pin	Description
$\overline{E}$	6	enable input (active LOW)
V_{EE}	7	supply voltage
GND	8	ground supply voltage
S0, S1, S2	11, 10, 9	select input
Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7	13, 14, 15, 12, 1, 5, 2, 4	independent input or output
Z	3	common output or input
V_{CC}	16	supply voltage

7 Function description

Table 3. Function table [1]

Input				Channel ON
$\overline{E}$	S2	S1	S0	
L	L	L	L	Y0 to Z
L	L	L	H	Y1 to Z
L	L	H	L	Y2 to Z
L	L	H	H	Y3 to Z
L	H	L	L	Y4 to Z
L	H	L	H	Y5 to Z
L	H	H	L	Y6 to Z
L	H	H	H	Y7 to Z
H	X	X	X	switches off

[1] H = HIGH voltage level; L = LOW voltage level; X = don't care.

8 Limiting values

Table 4. Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134). Voltages are referenced to V_{SS} = 0 V (ground).

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CC}	supply voltage	[1]	-0.5	+11.0	V
I_{IK}	input clamping current	$V_I < -0.5$ V or $V_I > V_{CC} + 0.5$ V	-	±20	mA
I_{SK}	switch clamping current	$V_{SW} < -0.5$ V or $V_{SW} > V_{CC} + 0.5$ V	-	±20	mA
I_{SW}	switch current	-0.5 V $< V_{SW} < V_{CC} + 0.5$ V	-	±25	mA
I_{EE}	supply current		-	±20	mA
I_{CC}	supply current		-	50	mA
I_{GND}	ground current		-	-50	mA
T_{stg}	storage temperature		-65	+150	°C
P_{tot}	total power dissipation	[2] SO16, (T)SSOP16, and DHVQFN16 package	-	500	mW
P	power dissipation	per switch	-	100	mW

[1] To avoid drawing V_{CC} current out of terminal Z, when switch current flows into terminals Yn, the voltage drop across the bidirectional switch must not exceed 0.4 V. If the switch current flows into terminal Z, no V_{CC} current will flow out of terminals Yn, and in this case there is no limit for the voltage drop across the switch, but the voltages at Yn and Z may not exceed V_{CC} or V_{EE}.

[2] For SO16 packages: above 70 °C the value of P_{tot} derates linearly with 8 mW/K.
For SSOP16 and TSSOP16 packages: above 60 °C the value of P_{tot} derates linearly with 5.5 mW/K.
For DHVQFN16 packages: above 60 °C the value of P_{tot} derates linearly with 4.5 mW/K.

9 Recommended operating conditions

Table 5. Recommended operating conditions

Symbol	Parameter	Conditions	74HC4051			74HCT4051			Unit
			Min	Typ	Max	Min	Typ	Max	
V_{CC}	supply voltage	see Figure 7 and Figure 8							
		V_{CC} - GND	2.0	5.0	10.0	4.5	5.0	5.5	V
		V_{CC} - V_{EE}	2.0	5.0	10.0	2.0	5.0	10.0	V
V_I	input voltage		GND	-	V_{CC}	GND	-	V_{CC}	V
V_{SW}	switch voltage		V_{EE}	-	V_{CC}	V_{EE}	-	V_{CC}	V
T_{amb}	ambient temperature		-40	+25	+125	-40	+25	+125	°C
$\Delta t/\Delta V$	input transition rise and fall rate	V_{CC} = 2.0 V	-	-	625	-	-	-	ns/V
		V_{CC} = 4.5 V	-	1.67	139	-	1.67	139	ns/V
		V_{CC} = 6.0 V	-	-	83	-	-	-	ns/V
		V_{CC} = 10.0 V	-	-	31	-	-	-	ns/V

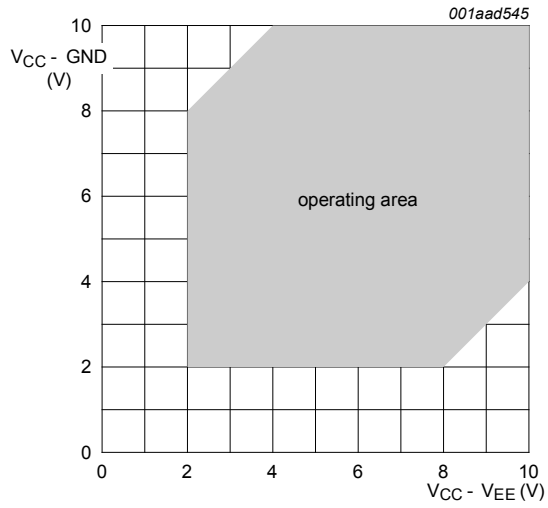


Figure 7. Guaranteed operating area as a function of the supply voltages for 74HC4051

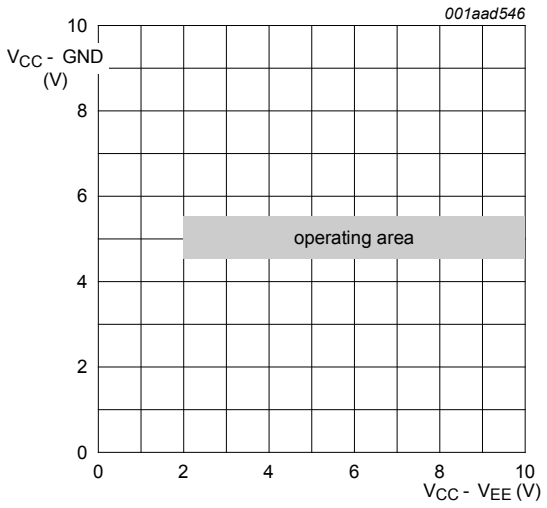


Figure 8. Guaranteed operating area as a function of the supply voltages for 74HCT4051

10 Static characteristics

Table 6. R_{ON} resistance per switch for 74HC4051 and 74HCT4051

V_I = V_{IH} or V_{IL}; for test circuit see Figure 9.

V_{is} is the input voltage at a Yn or Z terminal, whichever is assigned as an input.

V_{os} is the output voltage at a Yn or Z terminal, whichever is assigned as an output.

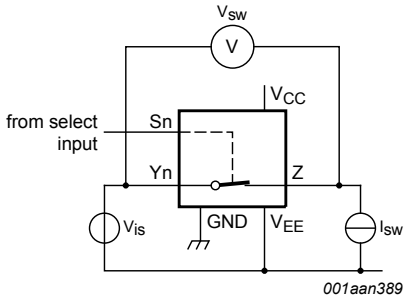
For 74HC4051: V_{CC} - GND or V_{CC} - V_{EE} = 2.0 V, 4.5 V, 6.0 V and 9.0 V.

For 74HCT4051: V_{CC} - GND = 4.5 V and 5.5 V, V_{CC} - V_{EE} = 2.0 V, 4.5 V, 6.0 V and 9.0 V.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
T_{amb} = 25 °C							
$R_{ON(peak)}$	ON resistance (peak)	V_{is} = V_{CC} to V_{EE}					
		V_{CC} = 2.0 V; V_{EE} = 0 V; I_{SW} = 100 µA	[1]	-	-	-	Ω
		V_{CC} = 4.5 V; V_{EE} = 0 V; I_{SW} = 1000 µA		-	100	180	Ω
		V_{CC} = 6.0 V; V_{EE} = 0 V; I_{SW} = 1000 µA		-	90	160	Ω
		V_{CC} = 4.5 V; V_{EE} = -4.5 V; I_{SW} = 1000 µA		-	70	130	Ω
$R_{ON(rail)}$	ON resistance (rail)	V_{is} = V_{EE}					
		V_{CC} = 2.0 V; V_{EE} = 0 V; I_{SW} = 100 µA	[1]	-	150	-	Ω
		V_{CC} = 4.5 V; V_{EE} = 0 V; I_{SW} = 1000 µA		-	80	140	Ω
		V_{CC} = 6.0 V; V_{EE} = 0 V; I_{SW} = 1000 µA		-	70	120	Ω
		V_{CC} = 4.5 V; V_{EE} = -4.5 V; I_{SW} = 1000 µA		-	60	105	Ω
		V_{is} = V_{CC}					
		V_{CC} = 2.0 V; V_{EE} = 0 V; I_{SW} = 100 µA	[1]	-	150	-	Ω
		V_{CC} = 4.5 V; V_{EE} = 0 V; I_{SW} = 1000 µA		-	90	160	Ω
		V_{CC} = 6.0 V; V_{EE} = 0 V; I_{SW} = 1000 µA		-	80	140	Ω
		V_{CC} = 4.5 V; V_{EE} = -4.5 V; I_{SW} = 1000 µA		-	65	120	Ω
ΔR_{ON}	ON resistance mismatch between channels	V_{is} = V_{CC} to V_{EE}					
		V_{CC} = 2.0 V; V_{EE} = 0 V	[1]	-	-	-	Ω
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	9	-	Ω
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	8	-	Ω
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	6	-	Ω

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T_{amb} = -40 °C to +85 °C						
$R_{ON(peak)}$	ON resistance (peak)	V_{is} = V_{CC} to V_{EE}				
		V_{CC} = 2.0 V; V_{EE} = 0 V; I_{SW} = 100 µA [1]	-	-	-	Ω
		V_{CC} = 4.5 V; V_{EE} = 0 V; I_{SW} = 1000 µA	-	-	225	Ω
		V_{CC} = 6.0 V; V_{EE} = 0 V; I_{SW} = 1000 µA	-	-	200	Ω
		V_{CC} = 4.5 V; V_{EE} = -4.5 V; I_{SW} = 1000 µA	-	-	165	Ω
$R_{ON(rail)}$	ON resistance (rail)	V_{is} = V_{EE}				
		V_{CC} = 2.0 V; V_{EE} = 0 V; I_{SW} = 100 µA [1]	-	-	-	Ω
		V_{CC} = 4.5 V; V_{EE} = 0 V; I_{SW} = 1000 µA	-	-	175	Ω
		V_{CC} = 6.0 V; V_{EE} = 0 V; I_{SW} = 1000 µA	-	-	150	Ω
		V_{CC} = 4.5 V; V_{EE} = -4.5 V; I_{SW} = 1000 µA	-	-	130	Ω
		V_{is} = V_{CC}				
		V_{CC} = 2.0 V; V_{EE} = 0 V; I_{SW} = 100 µA [1]	-	-	-	Ω
		V_{CC} = 4.5 V; V_{EE} = 0 V; I_{SW} = 1000 µA	-	-	200	Ω
		V_{CC} = 6.0 V; V_{EE} = 0 V; I_{SW} = 1000 µA	-	-	175	Ω
		V_{CC} = 4.5 V; V_{EE} = -4.5 V; I_{SW} = 1000 µA	-	-	150	Ω
T_{amb} = -40 °C to +125 °C						
$R_{ON(peak)}$	ON resistance (peak)	V_{is} = V_{CC} to V_{EE}				
		V_{CC} = 2.0 V; V_{EE} = 0 V; I_{SW} = 100 µA [1]	-	-	-	Ω
		V_{CC} = 4.5 V; V_{EE} = 0 V; I_{SW} = 1000 µA	-	-	270	Ω
		V_{CC} = 6.0 V; V_{EE} = 0 V; I_{SW} = 1000 µA	-	-	240	Ω
		V_{CC} = 4.5 V; V_{EE} = -4.5 V; I_{SW} = 1000 µA	-	-	195	Ω
$R_{ON(rail)}$	ON resistance (rail)	V_{is} = V_{EE}				
		V_{CC} = 2.0 V; V_{EE} = 0 V; I_{SW} = 100 µA [1]	-	-	-	Ω
		V_{CC} = 4.5 V; V_{EE} = 0 V; I_{SW} = 1000 µA	-	-	210	Ω
		V_{CC} = 6.0 V; V_{EE} = 0 V; I_{SW} = 1000 µA	-	-	180	Ω
		V_{CC} = 4.5 V; V_{EE} = -4.5 V; I_{SW} = 1000 µA	-	-	160	Ω
		V_{is} = V_{CC}				
		V_{CC} = 2.0 V; V_{EE} = 0 V; I_{SW} = 100 µA [1]	-	-	-	Ω
		V_{CC} = 4.5 V; V_{EE} = 0 V; I_{SW} = 1000 µA	-	-	240	Ω
		V_{CC} = 6.0 V; V_{EE} = 0 V; I_{SW} = 1000 µA	-	-	210	Ω
		V_{CC} = 4.5 V; V_{EE} = -4.5 V; I_{SW} = 1000 µA	-	-	180	Ω

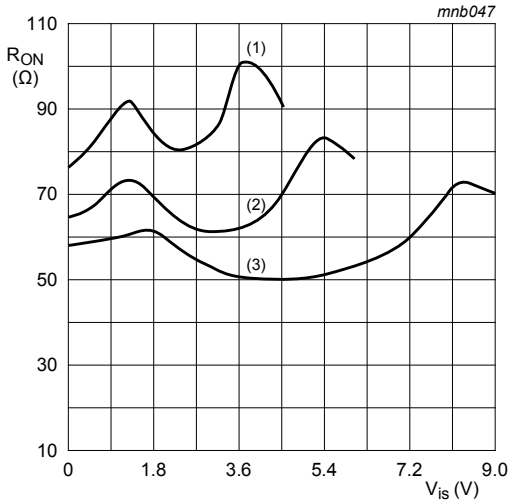
[1] When supply voltages (V_{CC} - V_{EE}) near 2.0 V the analog switch ON resistance becomes extremely non-linear. When using a supply of 2 V, it is recommended to use these devices only for transmitting digital signals.



V_{is} = 0 V to (V_{CC} - V_{EE}).

$$R_{ON} = \frac{V_{sw}}{I_{sw}}$$

Figure 9. Test circuit for measuring R_{ON}



V_{is} = 0 V to (V_{CC} - V_{EE}).

(1) V_{CC} = 4.5 V

(2) V_{CC} = 6 V

(3) V_{CC} = 9 V

Figure 10. Typical R_{ON} as a function of input voltage V_{is}

Table 7. Static characteristics for 74HC4051

Voltages are referenced to GND (ground = 0 V).

V_{is} is the input voltage at pins Yn or Z, whichever is assigned as an input.

V_{os} is the output voltage at pins Z or Yn, whichever is assigned as an output.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T_{amb} = 25 °C						
V_{IH}	HIGH-level input voltage	V_{CC} = 2.0 V	1.5	1.2	-	V
		V_{CC} = 4.5 V	3.15	2.4	-	V
		V_{CC} = 6.0 V	4.2	3.2	-	V
		V_{CC} = 9.0 V	6.3	4.7	-	V
V_{IL}	LOW-level input voltage	V_{CC} = 2.0 V	-	0.8	0.5	V
		V_{CC} = 4.5 V	-	2.1	1.35	V
		V_{CC} = 6.0 V	-	2.8	1.8	V
		V_{CC} = 9.0 V	-	4.3	2.7	V
I_I	input leakage current	V_{EE} = 0 V; V_I = V_{CC} or GND				
		V_{CC} = 6.0 V	-	-	±0.1	µA
		V_{CC} = 10.0 V	-	-	±0.2	µA
$I_{S(OFF)}$	OFF-state leakage current	V_{CC} = 10.0 V; V_{EE} = 0 V; V_I = V_{IH} or V_{IL}; $\lvert V_{SW}\rvert$ = V_{CC} - V_{EE}; see Figure 11				
		per channel	-	-	±0.1	µA
		all channels	-	-	±0.4	µA

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$I_{S(ON)}$	ON-state leakage current	$V_I = V_{IH}$ or V_{IL}; $\|V_{SW}\| = V_{CC} - V_{EE}$; V_{CC} = 10.0 V; V_{EE} = 0 V; see Figure 12	-	-	±0.4	µA
I_{CC}	supply current	V_{EE} = 0 V; $V_I = V_{CC}$ or GND; $V_{is} = V_{EE}$ or V_{CC}; $V_{os} = V_{CC}$ or V_{EE}				
		V_{CC} = 6.0 V	-	-	8.0	µA
		V_{CC} = 10.0 V	-	-	16.0	µA
C_I	input capacitance		-	3.5	-	pF
C_{sw}	switch capacitance	independent pins Yn	-	5	-	pF
		common pins Z	-	25	-	pF
T_{amb} = -40 °C to +85 °C						
V_{IH}	HIGH-level input voltage	V_{CC} = 2.0 V	1.5	-	-	V
		V_{CC} = 4.5 V	3.15	-	-	V
		V_{CC} = 6.0 V	4.2	-	-	V
		V_{CC} = 9.0 V	6.3	-	-	V
V_{IL}	LOW-level input voltage	V_{CC} = 2.0 V	-	-	0.5	V
		V_{CC} = 4.5 V	-	-	1.35	V
		V_{CC} = 6.0 V	-	-	1.8	V
		V_{CC} = 9.0 V	-	-	2.7	V
I_I	input leakage current	V_{EE} = 0 V; $V_I = V_{CC}$ or GND				
		V_{CC} = 6.0 V	-	-	±1.0	µA
		V_{CC} = 10.0 V	-	-	±2.0	µA
$I_{S(OFF)}$	OFF-state leakage current	V_{CC} = 10.0 V; V_{EE} = 0 V; $V_I = V_{IH}$ or V_{IL}; $\|V_{SW}\| = V_{CC} - V_{EE}$; see Figure 11				
		per channel	-	-	±1.0	µA
		all channels	-	-	±4.0	µA
$I_{S(ON)}$	ON-state leakage current	$V_I = V_{IH}$ or V_{IL}; $\|V_{SW}\| = V_{CC} - V_{EE}$; V_{CC} = 10.0 V; V_{EE} = 0 V; see Figure 12	-	-	±4.0	µA
I_{CC}	supply current	V_{EE} = 0 V; $V_I = V_{CC}$ or GND; $V_{is} = V_{EE}$ or V_{CC}; $V_{os} = V_{CC}$ or V_{EE}				
		V_{CC} = 6.0 V	-	-	80.0	µA
		V_{CC} = 10.0 V	-	-	160.0	µA

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T_{amb} = -40 °C to +125 °C						
V_{IH}	HIGH-level input voltage	V_{CC} = 2.0 V	1.5	-	-	V
		V_{CC} = 4.5 V	3.15	-	-	V
		V_{CC} = 6.0 V	4.2	-	-	V
		V_{CC} = 9.0 V	6.3	-	-	V
V_{IL}	LOW-level input voltage	V_{CC} = 2.0 V	-	-	0.5	V
		V_{CC} = 4.5 V	-	-	1.35	V
		V_{CC} = 6.0 V	-	-	1.8	V
		V_{CC} = 9.0 V	-	-	2.7	V
I_I	input leakage current	V_{EE} = 0 V; V_I = V_{CC} or GND				
		V_{CC} = 6.0 V	-	-	±1.0	µA
		V_{CC} = 10.0 V	-	-	±2.0	µA
$I_{S(OFF)}$	OFF-state leakage current	V_{CC} = 10.0 V; V_{EE} = 0 V; V_I = V_{IH} or V_{IL}; $\|V_{SW}\|$ = V_{CC} - V_{EE}; see Figure 11				
		per channel	-	-	±1.0	µA
		all channels	-	-	±4.0	µA
$I_{S(ON)}$	ON-state leakage current	V_I = V_{IH} or V_{IL}; $\|V_{SW}\|$ = V_{CC} - V_{EE}; V_{CC} = 10.0 V; V_{EE} = 0 V; see Figure 12	-	-	±4.0	µA
I_{CC}	supply current	V_{EE} = 0 V; V_I = V_{CC} or GND; V_{is} = V_{EE} or V_{CC}; V_{os} = V_{CC} or V_{EE}				
		V_{CC} = 6.0 V	-	-	160.0	µA
		V_{CC} = 10.0 V	-	-	320.0	µA

Table 8. Static characteristics for 74HCT4051

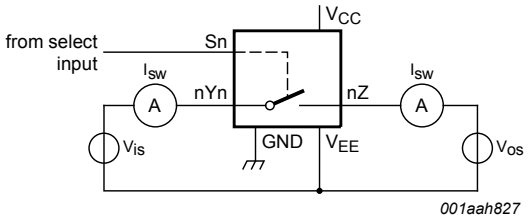
Voltages are referenced to GND (ground = 0 V).

V_{is} is the input voltage at pins Yn or Z, whichever is assigned as an input.

V_{os} is the output voltage at pins Z or Yn, whichever is assigned as an output.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T_{amb} = 25 °C						
V_{IH}	HIGH-level input voltage	V_{CC} = 4.5 V to 5.5 V	2.0	1.6	-	V
V_{IL}	LOW-level input voltage	V_{CC} = 4.5 V to 5.5 V	-	1.2	0.8	V
I_I	input leakage current	V_I = V_{CC} or GND; V_{CC} = 5.5 V; V_{EE} = 0 V	-	-	±0.1	µA
$I_{S(OFF)}$	OFF-state leakage current	V_{CC} = 10.0 V; V_{EE} = 0 V; V_I = V_{IH} or V_{IL}; $\|V_{SW}\|$ = V_{CC} - V_{EE}; see Figure 11				
		per channel	-	-	±0.1	µA
		all channels	-	-	±0.4	µA
$I_{S(ON)}$	ON-state leakage current	V_{CC} = 10.0 V; V_{EE} = 0 V; V_I = V_{IH} or V_{IL}; $\|V_{SW}\|$ = V_{CC} - V_{EE}; see Figure 12	-	-	±0.4	µA
I_{CC}	supply current	V_I = V_{CC} or GND; V_{is} = V_{EE} or V_{CC}; V_{os} = V_{CC} or V_{EE}				
		V_{CC} = 5.5 V; V_{EE} = 0 V	-	-	8.0	µA
		V_{CC} = 5.0 V; V_{EE} = -5.0 V	-	-	16.0	µA
ΔI_{CC}	additional supply current	per input; V_I = V_{CC} - 2.1 V; other inputs at V_{CC} or GND; V_{CC} = 4.5 V to 5.5 V; V_{EE} = 0 V	-	50	180	µA
C_I	input capacitance		-	3.5	-	pF
C_{sw}	switch capacitance	independent pins Yn	-	5	-	pF
		common pins Z	-	25	-	pF

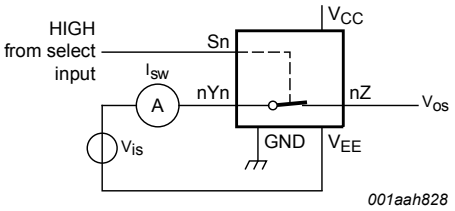
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T_{amb} = -40 °C to +85 °C						
V_{IH}	HIGH-level input voltage	V_{CC} = 4.5 V to 5.5 V	2.0	-	-	V
V_{IL}	LOW-level input voltage	V_{CC} = 4.5 V to 5.5 V	-	-	0.8	V
I_I	input leakage current	V_I = V_{CC} or GND; V_{CC} = 5.5 V; V_{EE} = 0 V	-	-	±1.0	µA
$I_{S(OFF)}$	OFF-state leakage current	V_{CC} = 10.0 V; V_{EE} = 0 V; V_I = V_{IH} or V_{IL}; $\|V_{SW}\|$ = V_{CC} - V_{EE}; see Figure 11				
		per channel	-	-	±1.0	µA
		all channels	-	-	±4.0	µA
$I_{S(ON)}$	ON-state leakage current	V_{CC} = 10.0 V; V_{EE} = 0 V; V_I = V_{IH} or V_{IL}; $\|V_{SW}\|$ = V_{CC} - V_{EE}; see Figure 12	-	-	±4.0	µA
I_{CC}	supply current	V_I = V_{CC} or GND; V_{is} = V_{EE} or V_{CC}; V_{os} = V_{CC} or V_{EE}				
		V_{CC} = 5.5 V; V_{EE} = 0 V	-	-	80.0	µA
		V_{CC} = 5.0 V; V_{EE} = -5.0 V	-	-	160.0	µA
ΔI_{CC}	additional supply current	per input; V_I = V_{CC} - 2.1 V; other inputs at V_{CC} or GND; V_{CC} = 4.5 V to 5.5 V; V_{EE} = 0 V	-	-	225	µA
T_{amb} = -40 °C to +125 °C						
V_{IH}	HIGH-level input voltage	V_{CC} = 4.5 V to 5.5 V	2.0	-	-	V
V_{IL}	LOW-level input voltage	V_{CC} = 4.5 V to 5.5 V	-	-	0.8	V
I_I	input leakage current	V_I = V_{CC} or GND; V_{CC} = 5.5 V; V_{EE} = 0 V	-	-	±1.0	µA
$I_{S(OFF)}$	OFF-state leakage current	V_{CC} = 10.0 V; V_{EE} = 0 V; V_I = V_{IH} or V_{IL}; $\|V_{SW}\|$ = V_{CC} - V_{EE}; see Figure 11				
		per channel	-	-	±1.0	µA
		all channels	-	-	±4.0	µA
$I_{S(ON)}$	ON-state leakage current	V_{CC} = 10.0 V; V_{EE} = 0 V; V_I = V_{IH} or V_{IL}; $\|V_{SW}\|$ = V_{CC} - V_{EE}; see Figure 12	-	-	±4.0	µA
I_{CC}	supply current	V_I = V_{CC} or GND; V_{is} = V_{EE} or V_{CC}; V_{os} = V_{CC} or V_{EE}				
		V_{CC} = 5.5 V; V_{EE} = 0 V	-	-	160.0	µA
		V_{CC} = 5.0 V; V_{EE} = -5.0 V	-	-	320.0	µA
ΔI_{CC}	additional supply current	per input; V_I = V_{CC} - 2.1 V; other inputs at V_{CC} or GND; V_{CC} = 4.5 V to 5.5 V; V_{EE} = 0 V	-	-	245	µA



$V_{is} = V_{CC}$ and $V_{os} = V_{EE}$.

$V_{is} = V_{EE}$ and $V_{os} = V_{CC}$.

Figure 11. Test circuit for measuring OFF-state current



$V_{is} = V_{CC}$ and V_{os} = open-circuit.

$V_{is} = V_{EE}$ and V_{os} = open-circuit.

Figure 12. Test circuit for measuring ON-state current

11 Dynamic characteristics

Table 9. Dynamic characteristics for 74HC4051

GND = 0 V; $t_r = t_f$ = 6 ns; C_L = 50 pF; for test circuit see Figure 15.

V_{is} is the input voltage at a Yn or Z terminal, whichever is assigned as an input.

V_{os} is the output voltage at a Yn or Z terminal, whichever is assigned as an output.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
T_{amb} = 25 °C							
t_{pd}	propagation delay	V_{is} to V_{os}; R_L = ∞ Ω; see Figure 13	[1]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	14	60	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	5	12	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	4	10	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	4	8	ns
t_{on}	turn-on time	$\overline{E}$ to V_{os}; R_L = ∞ Ω; see Figure 14	[2]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	72	345	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	29	69	ns
		V_{CC} = 5.0 V; V_{EE} = 0 V; C_L = 15 pF		-	22	-	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	21	59	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	18	51	ns
		Sn to V_{os}; R_L = ∞ Ω; see Figure 14	[2]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	66	345	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	28	69	ns
		V_{CC} = 5.0 V; V_{EE} = 0 V; C_L = 15 pF		-	20	-	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	19	59	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	16	51	ns
t_{off}	turn-off time	$\overline{E}$ to V_{os}; R_L = 1 kΩ; see Figure 14	[3]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	58	290	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	31	58	ns
		V_{CC} = 5.0 V; V_{EE} = 0 V; C_L = 15 pF		-	18	-	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	17	49	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	18	42	ns
		Sn to V_{os}; R_L = 1 kΩ; see Figure 14	[3]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	61	290	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	25	58	ns
		V_{CC} = 5.0 V; V_{EE} = 0 V; C_L = 15 pF		-	19	-	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	18	49	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	18	42	ns

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
C_{PD}	power dissipation capacitance	per switch; V_I = GND to V_{CC}	[4]	-	25	-	pF
T_{amb} = -40 °C to +85 °C							
t_{pd}	propagation delay	V_{is} to V_{os}; R_L = ∞ Ω; see Figure 13	[1]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	-	75	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	15	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	-	13	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	10	ns
t_{on}	turn-on time	$\overline{E}$ to V_{os}; R_L = ∞ Ω; see Figure 14	[2]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	-	430	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	86	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	-	73	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	64	ns
		Sn to V_{os}; R_L = ∞ Ω; see Figure 14	[2]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	-	430	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	86	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	-	73	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	64	ns
t_{off}	turn-off time	$\overline{E}$ to V_{os}; R_L = 1 kΩ; see Figure 14	[3]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	-	365	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	73	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	-	62	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	53	ns
		Sn to V_{os}; R_L = 1 kΩ; see Figure 14	[3]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	-	365	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	73	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	-	62	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	53	ns

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
T_{amb} = -40 °C to +125 °C							
t_{pd}	propagation delay	V_{is} to V_{os}; R_L = ∞ Ω; see Figure 13	[1]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	-	90	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	18	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	-	15	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	12	ns
t_{on}	turn-on time	$\overline{E}$ to V_{os}; R_L = ∞ Ω; see Figure 14	[2]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	-	520	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	104	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	-	88	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	77	ns
		Sn to V_{os}; R_L = ∞ Ω; see Figure 14	[2]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	-	520	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	104	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	-	88	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	77	ns
t_{off}	turn-off time	$\overline{E}$ to V_{os}; R_L = 1 kΩ; see Figure 14	[3]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	-	435	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	87	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	-	74	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	72	ns
		Sn to V_{os}; R_L = 1 kΩ; see Figure 14	[3]				
		V_{CC} = 2.0 V; V_{EE} = 0 V		-	-	435	ns
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	87	ns
		V_{CC} = 6.0 V; V_{EE} = 0 V		-	-	74	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	72	ns

[1] t_{pd} is the same as t_{PHL} and t_{PLH}.
[2] t_{on} is the same as t_{PZH} and t_{PZL}.
[3] t_{off} is the same as t_{PHZ} and t_{PLZ}.
[4] C_{PD} is used to determine the dynamic power dissipation (P_D in µW).
$P_D = C_{PD} \times V_{CC}^2 \times f_i \times N + \Sigma\{(C_L + C_{sw}) \times V_{CC}^2 \times f_o\}$ where:
f_i = input frequency in MHz;
f_o = output frequency in MHz;
N = number of inputs switching;
$\Sigma\{(C_L + C_{sw}) \times V_{CC}^2 \times f_o\}$ = sum of outputs;
C_L = output load capacitance in pF;
C_{sw} = switch capacitance in pF;
V_{CC} = supply voltage in V.

Table 10. Dynamic characteristics for 74HCT4051

GND = 0 V; $t_r = t_f$ = 6 ns; C_L = 50 pF; for test circuit see Figure 15.

V_{is} is the input voltage at a Yn or Z terminal, whichever is assigned as an input.

V_{os} is the output voltage at a Yn or Z terminal, whichever is assigned as an output.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
T_{amb} = 25 °C							
t_{pd}	propagation delay	V_{is} to V_{os}; R_L = ∞ Ω; see Figure 13	[1]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	5	12	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	4	8	ns
t_{on}	turn-on time	$\overline{E}$ to V_{os}; R_L = 1 kΩ; see Figure 14	[2]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	26	55	ns
		V_{CC} = 5.0 V; V_{EE} = 0 V; C_L = 15 pF		-	22	-	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	16	39	ns
		Sn to V_{os}; R_L = 1 kΩ; see Figure 14	[2]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	28	55	ns
		V_{CC} = 5.0 V; V_{EE} = 0 V; C_L = 15 pF		-	24	-	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	16	39	ns
t_{off}	turn-off time	$\overline{E}$ to V_{os}; R_L = 1 kΩ; see Figure 14	[3]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	19	45	ns
		V_{CC} = 5.0 V; V_{EE} = 0 V; C_L = 15 pF		-	16	-	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	16	32	ns
		Sn to V_{os}; R_L = 1 kΩ; see Figure 14	[3]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	23	45	ns
		V_{CC} = 5.0 V; V_{EE} = 0 V; C_L = 15 pF		-	20	-	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	16	32	ns
C_{PD}	power dissipation capacitance	per switch; V_I = GND to V_{CC} - 1.5 V	[4]	-	25	-	pF
T_{amb} = -40 °C to +85 °C							
t_{pd}	propagation delay	V_{is} to V_{os}; R_L = ∞ Ω; see Figure 13	[1]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	15	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	10	ns
t_{on}	turn-on time	$\overline{E}$ to V_{os}; R_L = 1 kΩ; see Figure 14	[2]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	69	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	49	ns
		Sn to V_{os}; R_L = 1 kΩ; see Figure 14	[2]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	69	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	49	ns

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
t_{off}	turn-off time	$\overline{E}$ to V_{os}; R_L = 1 kΩ; see Figure 14	[3]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	56	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	40	ns
		Sn to V_{os}; R_L = 1 kΩ; see Figure 14	[3]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	56	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	40	ns
T_{amb} = -40 °C to +125 °C							
t_{pd}	propagation delay	V_{is} to V_{os}; R_L = ∞ Ω; see Figure 13	[1]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	18	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	12	ns
t_{on}	turn-on time	$\overline{E}$ to V_{os}; R_L = 1 kΩ; see Figure 14	[2]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	83	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	59	ns
		Sn to V_{os}; R_L = 1 kΩ; see Figure 14	[2]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	83	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	59	ns
t_{off}	turn-off time	$\overline{E}$ to V_{os}; R_L = 1 kΩ; see Figure 14	[3]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	68	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	48	ns
		Sn to V_{os}; R_L = 1 kΩ; see Figure 14	[3]				
		V_{CC} = 4.5 V; V_{EE} = 0 V		-	-	68	ns
		V_{CC} = 4.5 V; V_{EE} = -4.5 V		-	-	48	ns

[1] t_{pd} is the same as t_{PHL} and t_{PLH}.
[2] t_{on} is the same as t_{PZH} and t_{PZL}.
[3] t_{off} is the same as t_{PHZ} and t_{PLZ}.
[4] C_{PD} is used to determine the dynamic power dissipation (P_D in µW).
$P_D = C_{PD} \times V_{CC}^2 \times f_i \times N + \Sigma\{(C_L + C_{sw}) \times V_{CC}^2 \times f_o\}$ where:
f_i = input frequency in MHz;
f_o = output frequency in MHz;
N = number of inputs switching;
$\Sigma\{(C_L + C_{sw}) \times V_{CC}^2 \times f_o\}$ = sum of outputs;
C_L = output load capacitance in pF;
C_{sw} = switch capacitance in pF;
V_{CC} = supply voltage in V.

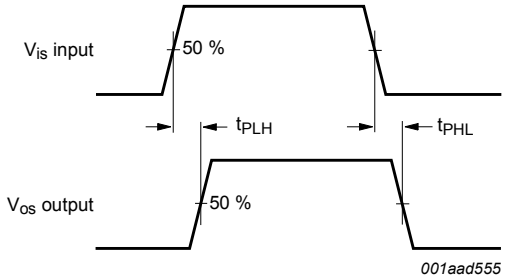
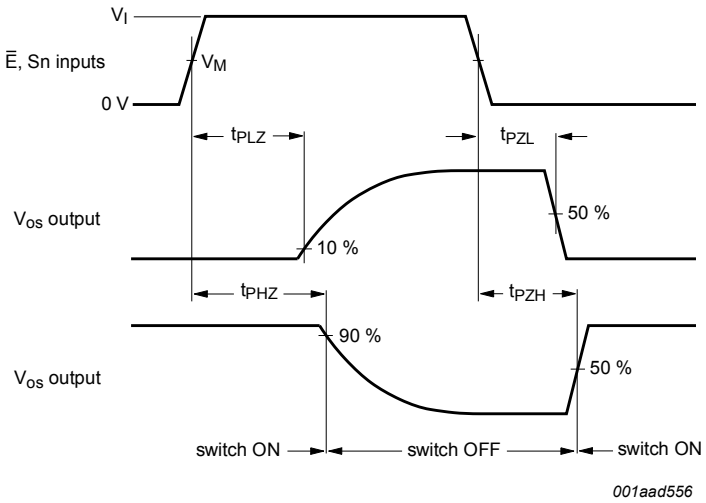


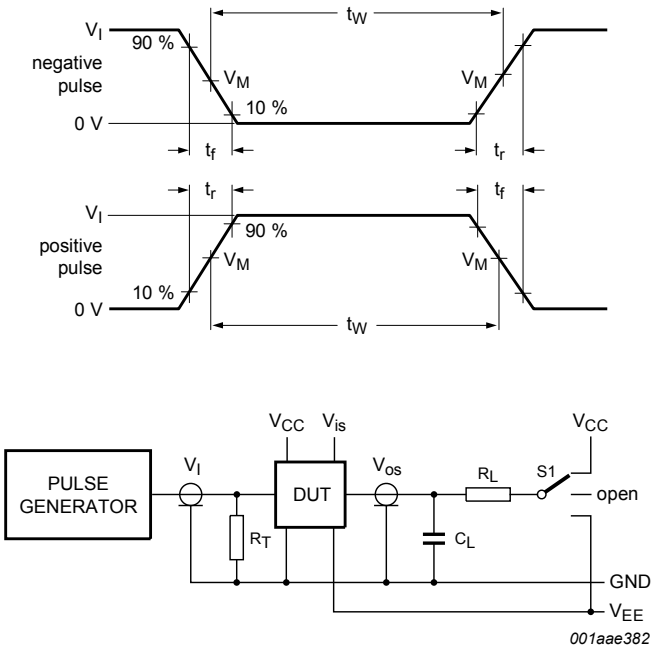
Figure 13. Input (V_{is}) to output (V_{os}) propagation delays



For 74HC4051: V_M = 0.5 x V_{CC}.

For 74HCT4051: V_M = 1.3 V.

Figure 14. Turn-on and turn-off times



Definitions for test circuit; see Table 11:

R_T = termination resistance should be equal to the output impedance Z_o of the pulse generator.

C_L = load capacitance including jig and probe capacitance.

R_L = load resistance.

S1 = Test selection switch.

Figure 15. Test circuit for measuring switching times

Table 11. Test data

Test	Input				Load		S1 position
	V_I	V_{is}	t_r, t_f		C_L	R_L	
			at f_{max}	other [1]			
t_{PHL}, t_{PLH}	[2]	pulse	< 2 ns	6 ns	50 pF	1 kΩ	open
t_{PZH}, t_{PHZ}	[2]	V_{CC}	< 2 ns	6 ns	50 pF	1 kΩ	V_{EE}
t_{PZL}, t_{PLZ}	[2]	V_{EE}	< 2 ns	6 ns	50 pF	1 kΩ	V_{CC}

[1] $t_r = t_f$ = 6 ns; when measuring f_{max}, there is no constraint to t_r and t_f with 50 % duty factor.

[2] V_I values:
For 74HC4051: $V_I = V_{CC}$
For 74HCT4051: V_I = 3 V

11.1 Additional dynamic characteristics

Table 12. Additional dynamic characteristics

Recommended conditions and typical values; GND = 0 V; T_{amb} = 25 °C; C_L = 50 pF.

V_{is} is the input voltage at pins nYn or nZ, whichever is assigned as an input.

V_{os} is the output voltage at pins nYn or nZ, whichever is assigned as an output.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
d_{sin}	sine-wave distortion	f_i = 1 kHz; R_L = 10 kΩ; see Figure 16				
		V_{is} = 4.0 V (p-p); V_{CC} = 2.25 V; V_{EE} = -2.25 V	-	0.04	-	%
		V_{is} = 8.0 V (p-p); V_{CC} = 4.5 V; V_{EE} = -4.5 V	-	0.02	-	%
		f_i = 10 kHz; R_L = 10 kΩ; see Figure 16				
		V_{is} = 4.0 V (p-p); V_{CC} = 2.25 V; V_{EE} = -2.25 V	-	0.12	-	%
		V_{is} = 8.0 V (p-p); V_{CC} = 4.5 V; V_{EE} = -4.5 V	-	0.06	-	%
α_{iso}	isolation (OFF-state)	R_L = 600 Ω; f_i = 1 MHz; see Figure 17				
		V_{CC} = 2.25 V; V_{EE} = -2.25 V [1]	-	-50	-	dB
		V_{CC} = 4.5 V; V_{EE} = -4.5 V [1]	-	-50	-	dB
V_{ct}	crosstalk voltage	peak-to-peak value; between control and any switch; R_L = 600 Ω; f_i = 1 MHz; $\overline{E}$ or Sn square wave between V_{CC} and GND; $t_r = t_f$ = 6 ns; see Figure 18				
		V_{CC} = 4.5 V; V_{EE} = 0 V	-	110	-	mV
		V_{CC} = 4.5 V; V_{EE} = -4.5 V	-	220	-	mV
$f_{(-3dB)}$	-3 dB frequency response	R_L = 50 Ω; see Figure 19				
		V_{CC} = 2.25 V; V_{EE} = -2.25 V [2]	-	170	-	MHz
		V_{CC} = 4.5 V; V_{EE} = -4.5 V [2]	-	180	-	MHz

[1] Adjust input voltage V_{is} to 0 dBm level (0 dBm = 1 mW into 600 Ω).
[2] Adjust input voltage V_{is} to 0 dBm level at V_{os} for 1 MHz (0 dBm = 1 mW into 50 Ω).

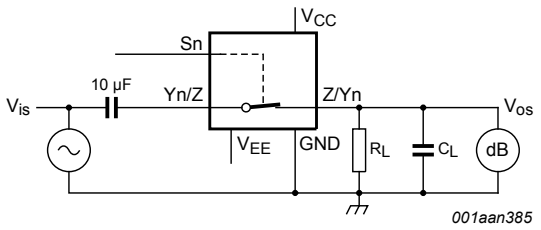
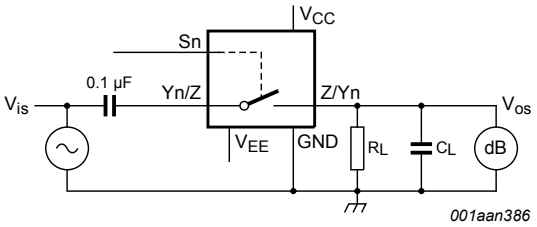
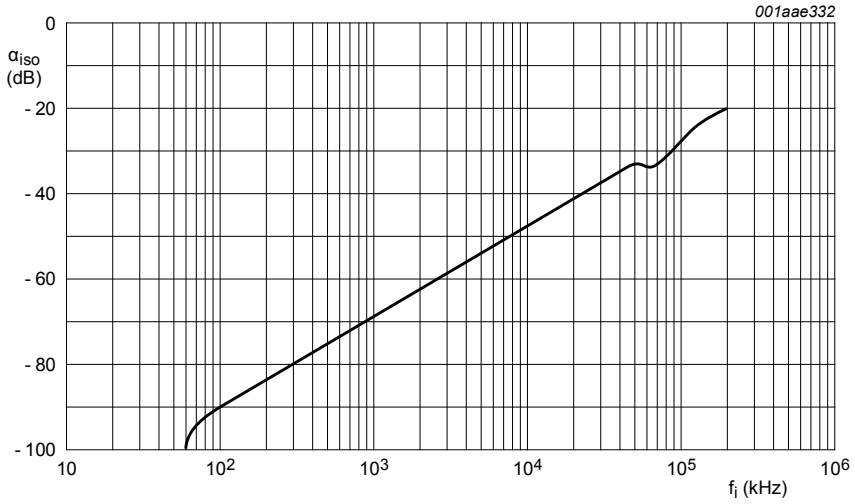


Figure 16. Test circuit for measuring sine-wave distortion



V_{CC} = 4.5 V; GND = 0 V; V_{EE} = -4.5 V; R_L = 600 Ω; R_S = 1 kΩ.

a. Test circuit



b. Isolation (OFF-state) as a function of frequency

Figure 17. Test circuit for measuring isolation (OFF-state)

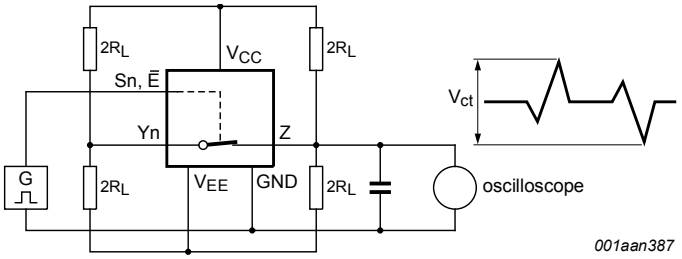
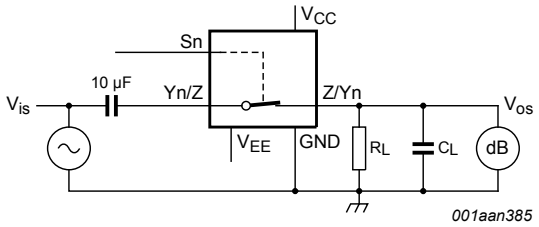
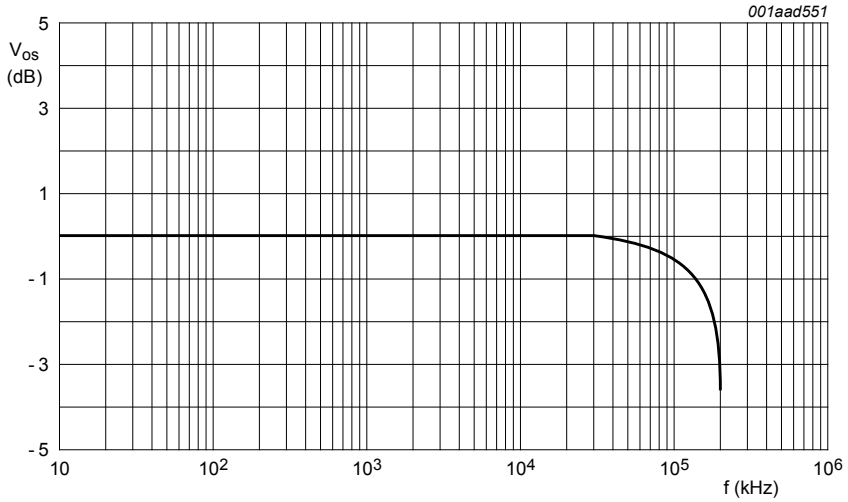


Figure 18. Test circuit for measuring crosstalk between control input and any switch



V_{CC} = 4.5 V; GND = 0 V; V_{EE} = -4.5 V; R_L = 50 Ω; R_S = 1 kΩ.

a. Test circuit

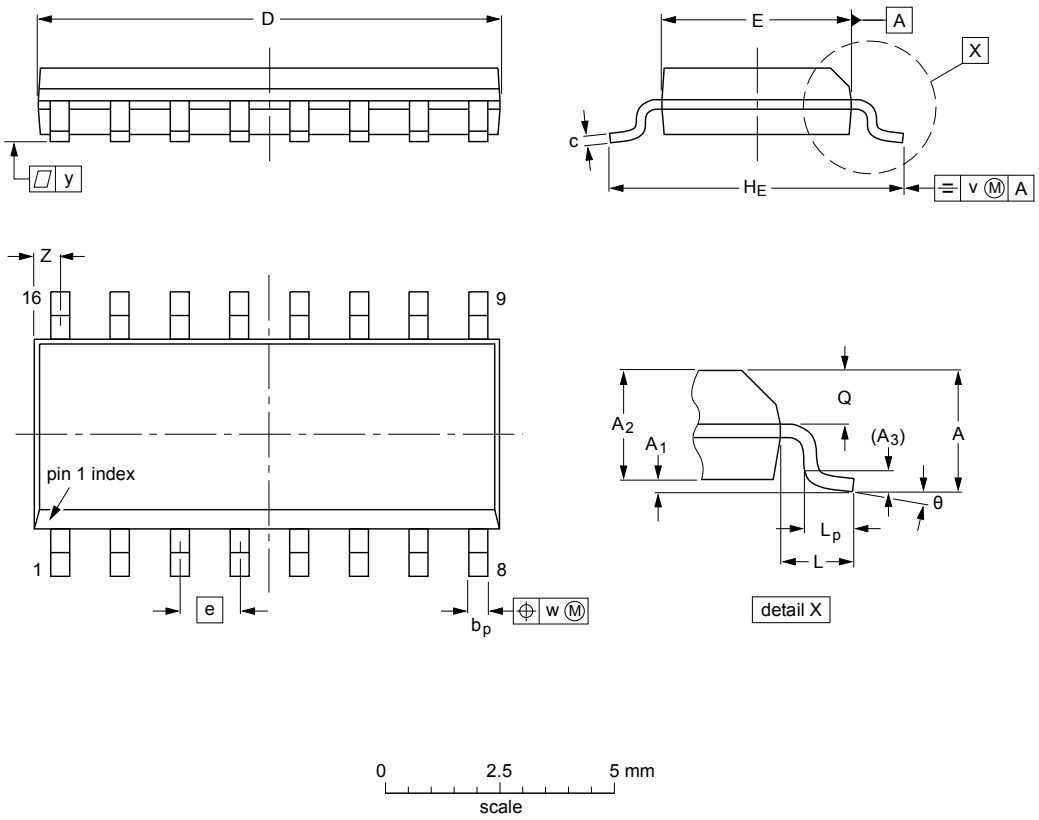


b. Typical frequency response

Figure 19. Test circuit for frequency response

12 Package outline

SO16: plastic small outline package; 16 leads; body width 3.9 mm **SOT109-1**



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	1.75	0.25 0.10	1.45 1.25	0.25	0.49 0.36	0.25 0.19	10.0 9.8	4.0 3.8	1.27	6.2 5.8	1.05	1.0 0.4	0.7 0.6	0.25	0.25	0.1	0.7 0.3	8° 0°
inches	0.069	0.010 0.004	0.057 0.049	0.01	0.019 0.014	0.0100 0.0075	0.39 0.38	0.16 0.15	0.05	0.244 0.228	0.041	0.039 0.016	0.028 0.020	0.01	0.01	0.004	0.028 0.012	

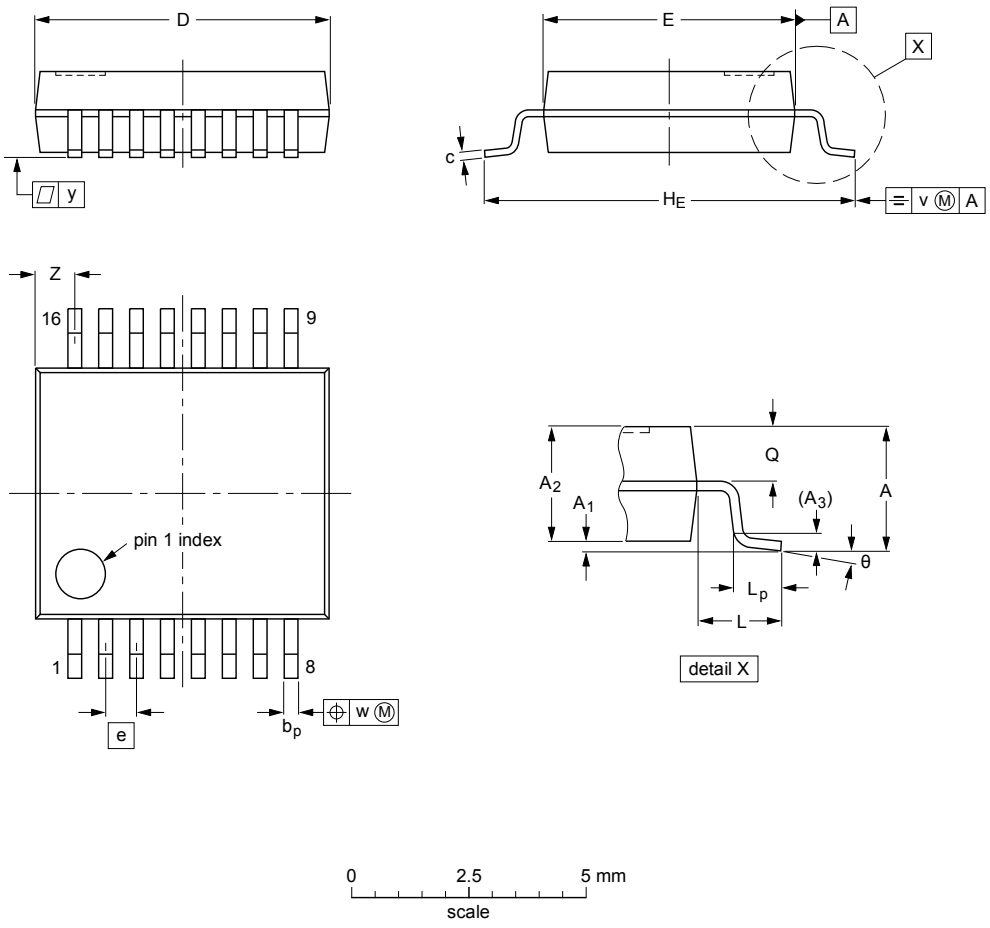
Note

1. Plastic or metal protrusions of 0.15 mm (0.006 inch) maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT109-1	076E07	MS-012				~~99-12-27~~ 03-02-19

Figure 20. Package outline SOT109-1 (SO16)

SSOP16: plastic shrink small outline package; 16 leads; body width 5.3 mm **SOT338-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	2	0.21 0.05	1.80 1.65	0.25	0.38 0.25	0.20 0.09	6.4 6.0	5.4 5.2	0.65	7.9 7.6	1.25	1.03 0.63	0.9 0.7	0.2	0.13	0.1	1.00 0.55	8^o 0^o

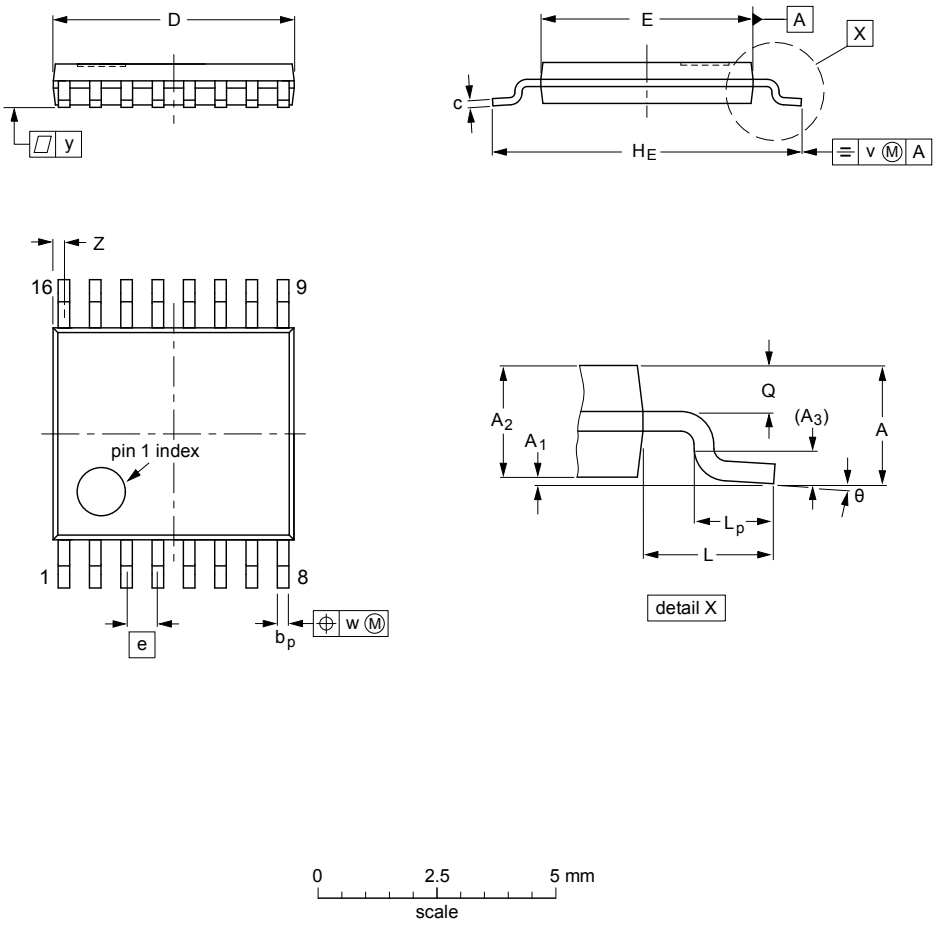
Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT338-1		MO-150				~~99-12-27~~ 03-02-19

Figure 21. Package outline SOT338-1 (SSOP16)

TSSOP16: plastic thin shrink small outline package; 16 leads; body width 4.4 mm **SOT403-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	D (1)	E (2)	e	H_E	L	L_p	Q	v	w	y	Z (1)	θ
mm	1.1	0.15 0.05	0.95 0.80	0.25	0.30 0.19	0.2 0.1	5.1 4.9	4.5 4.3	0.65	6.6 6.2	1	0.75 0.50	0.4 0.3	0.2	0.13	0.1	0.40 0.06	8^o 0^o

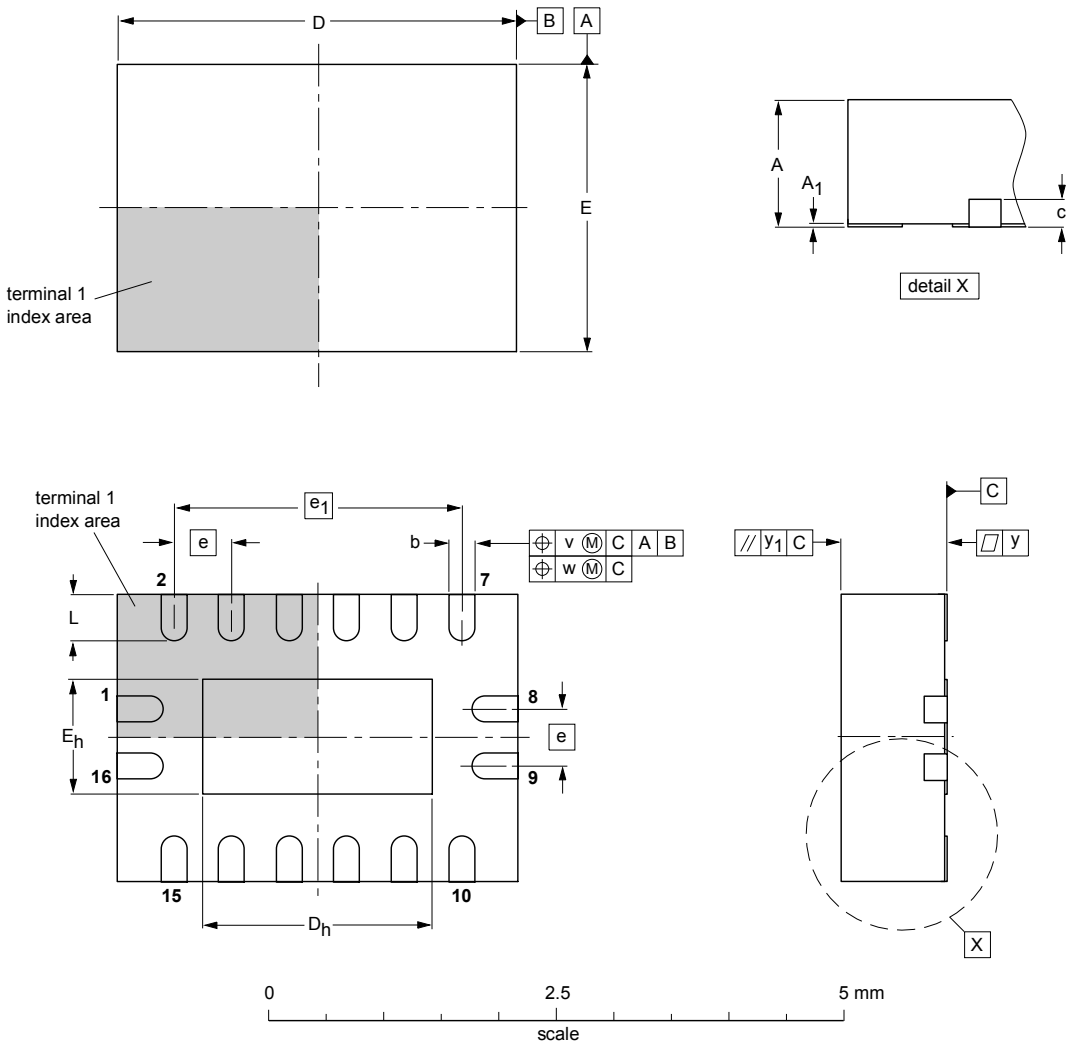
Notes

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.
2. Plastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT403-1		MO-153				~~99-12-27~~ 03-02-18

Figure 22. Package outline SOT403-1 (TSSOP16)

DHVQFN16: plastic dual in-line compatible thermal enhanced very thin quad flat package; no leads; 16 terminals; body 2.5 x 3.5 x 0.85 mm **SOT763-1**



DIMENSIONS (mm are the original dimensions)

UNIT	$A^{(1)}$ max.	A_1	b	c	$D^{(1)}$	D_h	$E^{(1)}$	E_h	e	e_1	L	v	w	y	y_1
mm	1	0.05 0.00	0.30 0.18	0.2	3.6 3.4	2.15 1.85	2.6 2.4	1.15 0.85	0.5	2.5	0.5 0.3	0.1	0.05	0.05	0.1

Note

1. Plastic or metal protrusions of 0.075 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT763-1	- - -	MO-241	- - -			~~02-10-17~~ 03-01-27

Figure 23. Package outline SOT763-1 (DHVQFN16)

13 Abbreviations

Table 13. Abbreviations

Acronym	Description
CDM	Charged Device Model
DUT	Device Under Test
ESD	ElectroStatic Discharge
HBM	Human Body Model
MM	Machine Model

14 Revision history

Table 14. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
74HC_HCT4051 v.9	20170926	Product data sheet	-	74HC_HCT4051 v.8
Modifications:	• The format of this data sheet has been redesigned to comply with the identity guidelines of Nexperia. • Legal texts have been adapted to the new company name where appropriate.			
74HC_HCT4051 v.8	20160205	Product data sheet	-	74HC_HCT4051 v.7
Modifications:	• Type numbers 74HC4051N and 74HCT4051N (SOT38-4) removed.			
74HC_HCT4051 v.7	20120719	Product data sheet	-	74HC_HCT4051 v.6
Modifications:	• CDM added to features.			
74HC_HCT4051 v.6	20111213	Product data sheet	-	74HC_HCT4051 v.5
Modifications:	• Legal pages updated.			
74HC_HCT4051 v.5	20110513	Product data sheet	-	74HC_HCT4051 v.4
74HC_HCT4051 v.4	20110117	Product data sheet	-	74HC_HCT4051 v.3
74HC_HCT4051 v.3	20051219	Product specification	-	74HC_HCT4051_CNV_2

15 Legal information

15.1 Data sheet status

Document status[1][2]	Product status[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1] Please consult the most recently issued document before initiating or completing a design.
[2] The term 'short data sheet' is explained in section "Definitions".
[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL http://www.nexperia.com.

15.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. Nexperia does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local Nexperia sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

Product specification — The information and data provided in a Product data sheet shall define the specification of the product as agreed between Nexperia and its customer, unless Nexperia and customer have explicitly agreed otherwise in writing. In no event however, shall an agreement be valid in which the Nexperia product is deemed to offer functions and qualities beyond those described in the Product data sheet.

15.3 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, Nexperia does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. Nexperia takes no responsibility for the content in this document if provided by an information source outside of Nexperia. In no event shall Nexperia be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory. Notwithstanding any damages that customer might incur for any reason whatsoever, Nexperia's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of Nexperia.

Right to make changes — Nexperia reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — Nexperia products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an Nexperia product can reasonably be expected to result in personal injury, death or severe property or environmental damage. Nexperia and its suppliers accept no liability for inclusion and/or use of Nexperia products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. Nexperia makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their applications and products using Nexperia products, and Nexperia accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the Nexperia product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. Nexperia does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using Nexperia products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). Nexperia does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — Nexperia products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nexperia.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. Nexperia hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of Nexperia products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Non-automotive qualified products — Unless this data sheet expressly states that this specific Nexperia product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. Nexperia accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications. In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without Nexperia's warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond Nexperia's specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies Nexperia for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond Nexperia's standard warranty and Nexperia's product specifications.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

15.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

Contents

For more information, please visit: http://www.nexperia.com
For sales office addresses, please send an email to: salesaddresses@nexperia.com

Date of release: 26 September 2017
Document identifier: 74HC_HCT4051

Poly Phase Multifunction Energy Metering IC with Per Phase Information

Data Sheet

ADE7758

FEATURES

- **Highly accurate; supports IEC 60687, IEC 61036, IEC 61268, IEC 62053-21, IEC 62053-22, and IEC 62053-23**
- **Compatible with 3-phase/3-wire, 3-phase/4-wire, and other 3-phase services**
- **Less than 0.1% active energy error over a dynamic range of 1000 to 1 at 25°C**
- **Supplies active/reactive/apparent energy, voltage rms, current rms, and sampled waveform data**
- **Two pulse outputs, one for active power and the other selectable between reactive and apparent power with programmable frequency**
- **Digital power, phase, and rms offset calibration**
- **On-chip, user-programmable thresholds for line voltage SAG and overvoltage detections**
- **An on-chip, digital integrator enables direct interface-to-current sensors with di/dt output**
- **A PGA in the current channel allows direct interface to current transformers**
- **An SPI®-compatible serial interface with $\overline{IRQ}$**
- **Proprietary ADCs and DSP provide high accuracy over large variations in environmental conditions and time**
- **Reference 2.4 V (drift 30 ppm/°C typical) with external overdrive capability**
- **Single 5 V supply, low power (70 mW typical)**

GENERAL DESCRIPTION

The ADE7758 is a high accuracy, 3-phase electrical energy measurement IC with a serial interface and two pulse outputs. The ADE7758 incorporates second-order Σ-Δ ADCs, a digital integrator, reference circuitry, a temperature sensor, and all the signal processing required to perform active, reactive, and apparent energy measurement and rms calculations.

The ADE7758 is suitable to measure active, reactive, and apparent energy in various 3-phase configurations, such as WYE or DELTA services, with both three and four wires. The ADE7758 provides system calibration features for each phase, that is, rms offset correction, phase calibration, and power calibration. The APCF logic output gives active power information, and the VARCF logic output provides instantaneous reactive or apparent power information.

FUNCTIONAL BLOCK DIAGRAM

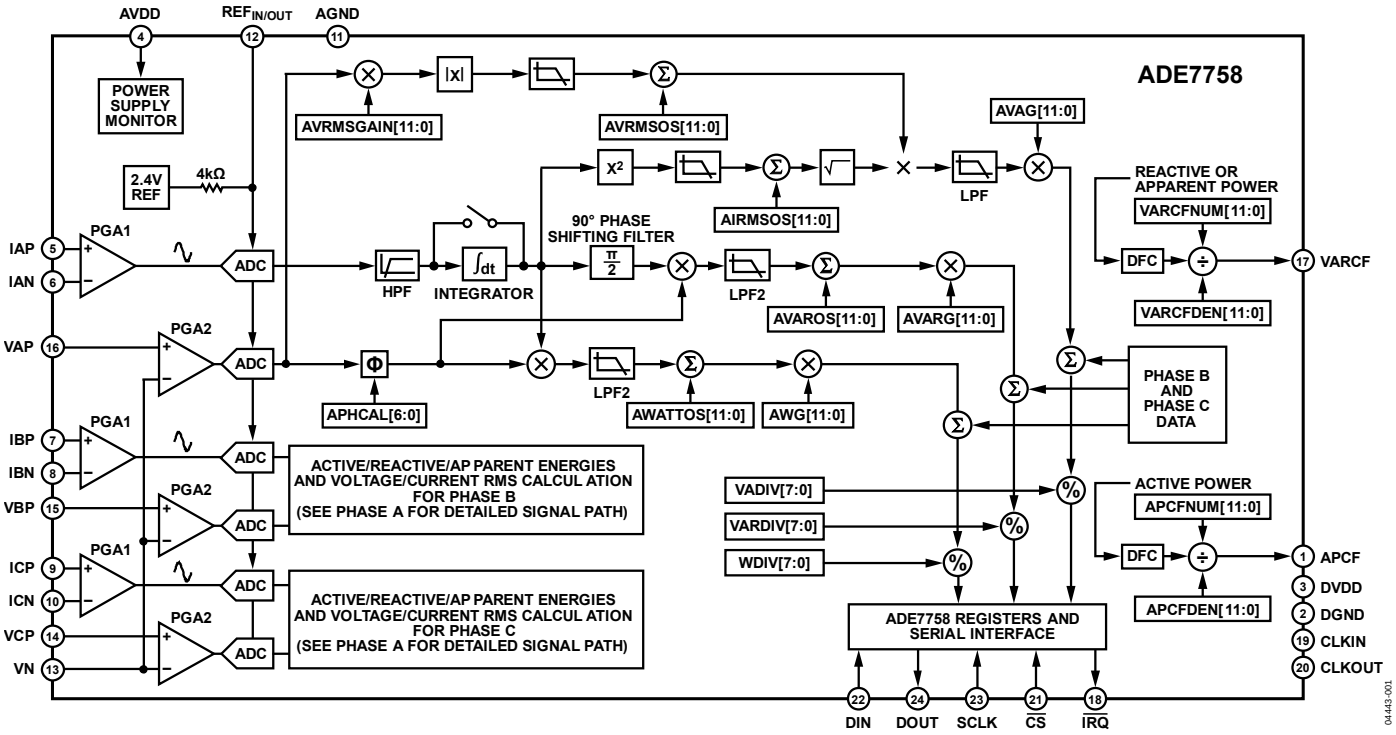


Figure 1.

Rev. E

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113

TABLE OF CONTENTS

Revision History

10/11—Rev. D to Rev. E

Changes to Figure 1 ... 1
Changes to Figure 41 ... 19
Changes to Figure 60 ... 27
Added Figure 61; Renumbered Sequentially ... 27
Changes to Phase Sequence Detection Section ... 27
Changes to Power-Supply Monitor Section ... 27
Changes to Figure 62 ... 28
Changes to Figure 67 ... 32
Changes to Figure 68 ... 32
Changes to Equation 25 ... 34
Changes to Figure 69 ... 34
Changes to Table 17 ... 62
Change to Table 18 ... 64
Changes to Table 24 ... 69
Changes to Ordering Guide ... 70

10/08—Rev. C to Rev. D

Changes to Figure 1 ... 1
Changes to Phase Sequence Detection Section and Figure 60 ... 27

7/06—Rev. B to Rev. C

11/05—Rev. A to Rev. B

9/04—Rev. 0 to Rev. A

1/04—Revision 0: Initial Version

GENERAL DESCRIPTION

The ADE7758 has a waveform sample register that allows access to the ADC outputs. The part also incorporates a detection circuit for short duration low or high voltage variations. The voltage threshold levels and the duration (number of half-line cycles) of the variation are user programmable. A zero-crossing detection is synchronized with the zero-crossing point of the line voltage of any of the three phases. This information can be used to measure the period of any one of the three voltage inputs. The zero-crossing detection is used inside the chip for the line cycle energy accumulation mode. This mode permits faster and more accurate calibration by synchronizing the energy accumulation with an integer number of line cycles.

Data is read from the ADE7758 via the SPI serial interface. The interrupt request output ($\overline{\text{IRQ}}$) is an open-drain, active low logic output. The $\overline{\text{IRQ}}$ output goes active low when one or more interrupt events have occurred in the ADE7758. A status register indicates the nature of the interrupt. The ADE7758 is available in a 24-lead SOIC package.

SPECIFICATIONS

AVDD = DVDD = 5 V ± 5%, AGND = DGND = 0 V, on-chip reference, CLKIN = 10 MHz XTAL, T_{MIN} to T_{MAX} = −40°C to +85°C.

Table 1.

Parameter[1, 2]	Specification	Unit	Test Conditions/Comments
ACCURACY			
Active Energy Measurement Error (per Phase)	0.1	% typ	Over a dynamic range of 1000 to 1
Phase Error Between Channels			Line frequency = 45 Hz to 65 Hz, HPF on
PF = 0.8 Capacitive	±0.05	°max	Phase lead 37°
PF = 0.5 Inductive	±0.05	°max	Phase lag 60°
AC Power Supply Rejection			AVDD = DVDD = 5 V + 175 mV rms/120 Hz
Output Frequency Variation	0.01	% typ	V1P = V2P = V3P = 100 mV rms
DC Power Supply Rejection			AVDD = DVDD = 5 V ± 250 mV dc
Output Frequency Variation	0.01	% typ	V1P = V2P = V3P = 100 mV rms
Active Energy Measurement Bandwidth	14	kHz	
IRMS Measurement Error	0.5	% typ	Over a dynamic range of 500:1
IRMS Measurement Bandwidth	14	kHz	
VRMS Measurement Error	0.5	% typ	Over a dynamic range of 20:1
VRMS Measurement Bandwidth	260	Hz	
ANALOG INPUTS			See the Analog Inputs section
Maximum Signal Levels	±500	mV max	Differential input
Input Impedance (DC)	380	kΩ min	
ADC Offset Error[3]	±30	mV max	Uncalibrated error, see the Terminology section
Gain Error[3]	±6	% typ	External 2.5 V reference
WAVEFORM SAMPLING			Sampling CLKIN/128, 10 MHz/128 = 78.1 kSPS
Current Channels			See the Current Channel ADC section
Signal-to-Noise Plus Distortion	62	dB typ	
Bandwidth (−3 dB)	14	kHz	
Voltage Channels			See the Voltage Channel ADC section
Signal-to-Noise Plus Distortion	62	dB typ	
Bandwidth (−3 dB)	260	Hz	
REFERENCE INPUT			
$REF_{IN/OUT}$ Input Voltage Range	2.6	V max	2.4 V + 8%
	2.2	V min	2.4 V − 8%
Input Capacitance	10	pF max	
ON-CHIP REFERENCE			Nominal 2.4 V at $REF_{IN/OUT}$ pin
Reference Error	±200	mV max	
Current Source	6	µA max	
Output Impedance	4	kΩ min	
Temperature Coefficient	30	ppm/°C typ	
CLKIN			All specifications CLKIN of 10 MHz
Input Clock Frequency	15	MHz max	
	5	MHz min	
LOGIC INPUTS			
DIN, SCLK, CLKIN, and $\overline{CS}$			
Input High Voltage, V_{INH}	2.4	V min	DVDD = 5 V ± 5%
Input Low Voltage, V_{INL}	0.8	V max	DVDD = 5 V ± 5%
Input Current, I_{IN}	±3	µA max	Typical 10 nA, V_{IN} = 0 V to DVDD
Input Capacitance, C_{IN}	10	pF max	

Parameter[1, 2]	Specification	Unit	Test Conditions/Comments
LOGIC OUTPUTS			DVDD = 5 V ± 5%
$\overline{IRQ}$, DOUT, and CLKOUT			$\overline{IRQ}$ is open-drain, 10 kΩ pull-up resistor
Output High Voltage, V_{OH}	4	V min	I_{SOURCE} = 5 mA
Output Low Voltage, V_{OL}	0.4	V max	I_{SINK} = 1 mA
APCF and VARCF			
Output High Voltage, V_{OH}	4	V min	I_{SOURCE} = 8 mA
Output Low Voltage, V_{OL}	1	V max	I_{SINK} = 5 mA
POWER SUPPLY			For specified performance
AVDD	4.75	V min	5 V − 5%
	5.25	V max	5 V + 5%
DVDD	4.75	V min	5 V − 5%
	5.25	V max	5 V + 5%
AI_{DD}	8	mA max	Typically 5 mA
DI_{DD}	13	mA max	Typically 9 mA

[1] See the Typical Performance Characteristics.
[2] See the Terminology section for a definition of the parameters.
[3] See the Analog Inputs section.

TIMING CHARACTERISTICS

AVDD = DVDD = 5 V ± 5%, AGND = DGND = 0 V, on-chip reference, CLKIN = 10 MHz XTAL, T_{MIN} to T_{MAX} = −40°C to +85°C.

Table 2.

Parameter[1, 2]	Specification	Unit	Test Conditions/Comments
WRITE TIMING			
t_1	50	ns (min)	$\overline{CS}$ falling edge to first SCLK falling edge
t_2	50	ns (min)	SCLK logic high pulse width
t_3	50	ns (min)	SCLK logic low pulse width
t_4	10	ns (min)	Valid data setup time before falling edge of SCLK
t_5	5	ns (min)	Data hold time after SCLK falling edge
t_6	1200	ns (min)	Minimum time between the end of data byte transfers
t_7	400	ns (min)	Minimum time between byte transfers during a serial write
t_8	100	ns (min)	$\overline{CS}$ hold time after SCLK falling edge
READ TIMING			
t_9[3]	4	µs (min)	Minimum time between read command (that is, a write to communication register) and data read
t_{10}	50	ns (min)	Minimum time between data byte transfers during a multibyte read
t_{11}[4]	30	ns (min)	Data access time after SCLK rising edge following a write to the communications register
t_{12}[5]	100	ns (max)	Bus relinquish time after falling edge of SCLK
	10	ns (min)	
t_{13}[5]	100	ns (max)	Bus relinquish time after rising edge of $\overline{CS}$
	10	ns (min)	

[1] Sample tested during initial release and after any redesign or process change that may affect this parameter. All input signals are specified with tr = tf = 5 ns (10% to 90%) and timed from a voltage level of 1.6 V.
[2] See the timing diagrams in Figure 3 and Figure 4 and the Serial Interface section.
[3] Minimum time between read command and data read for all registers except waveform register, which is t_9 = 500 ns min.
[4] Measured with the load circuit in Figure 2 and defined as the time required for the output to cross 0.8 V or 2.4 V.
[5] Derived from the measured time taken by the data outputs to change 0.5 V when loaded with the circuit in Figure 2. The measured number is then extrapolated back to remove the effects of charging or discharging the 50 pF capacitor. This means that the time quoted here is the true bus relinquish time of the part and is independent of the bus loading.

TIMING DIAGRAMS

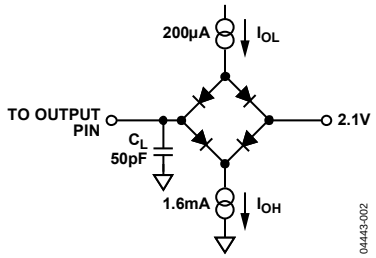


Figure 2. Load Circuit for Timing Specifications

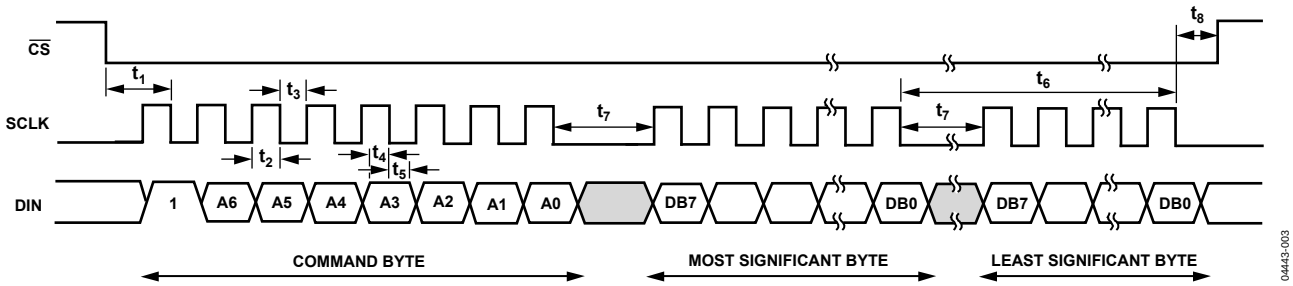


Figure 3. Serial Write Timing

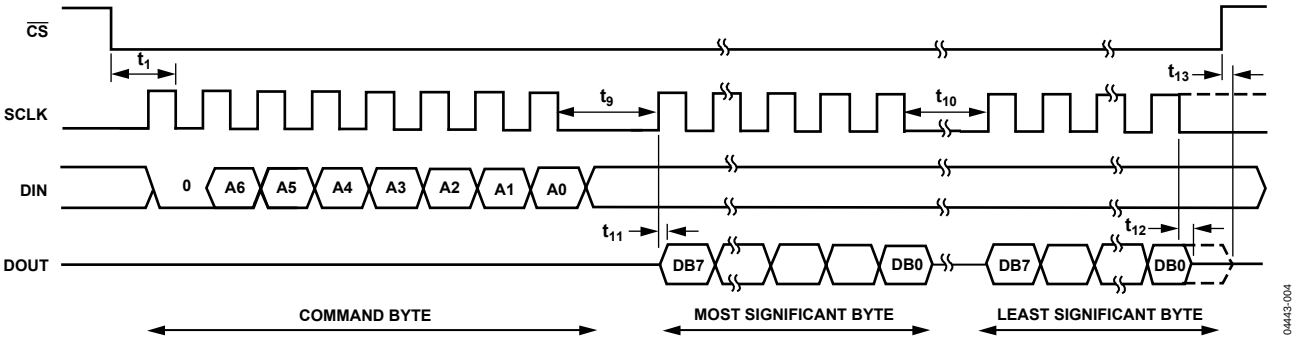


Figure 4. Serial Read Timing

ABSOLUTE MAXIMUM RATINGS

T_A = 25°C, unless otherwise noted.

Table 3.

Parameter	Rating
AVDD to AGND	–0.3 V to +7 V
DVDD to DGND	–0.3 V to +7 V
DVDD to AVDD	–0.3 V to +0.3 V
Analog Input Voltage to AGND, IAP, IAN, IBP, IBN, ICP, ICN, VAP, VBP, VCP, VN	–6 V to +6 V
Reference Input Voltage to AGND	–0.3 V to AVDD + 0.3 V
Digital Input Voltage to DGND	–0.3 V to DVDD + 0.3 V
Digital Output Voltage to DGND	–0.3 V to DVDD + 0.3 V
Operating Temperature	
Industrial Range	–40°C to +85°C
Storage Temperature Range	–65°C to +150°C
Junction Temperature	150°C
24-Lead SOIC, Power Dissipation	88 mW
θ_{JA} Thermal Impedance	53°C/W
Lead Temperature, Soldering	
Vapor Phase (60 sec)	215°C
Infrared (15 sec)	220°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

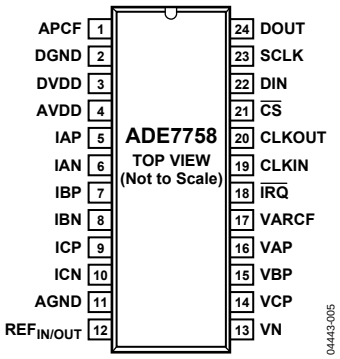


Figure 5. Pin Configuration

Table 4. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	APCF	Active Power Calibration Frequency (APCF) Logic Output. It provides active power information. This output is used for operational and calibration purposes. The full-scale output frequency can be scaled by writing to the APCFNUM and APCFDEN registers (see the Active Power Frequency Output section).
2	DGND	This provides the ground reference for the digital circuitry in the ADE7758, that is, the multiplier, filters, and digital-to-frequency converter. Because the digital return currents in the ADE7758 are small, it is acceptable to connect this pin to the analog ground plane of the whole system. However, high bus capacitance on the DOUT pin can result in noisy digital current that could affect performance.
3	DVDD	Digital Power Supply. This pin provides the supply voltage for the digital circuitry in the ADE7758. The supply voltage should be maintained at 5 V ± 5% for specified operation. This pin should be decoupled to DGND with a 10 µF capacitor in parallel with a ceramic 100 nF capacitor.
4	AVDD	Analog Power Supply. This pin provides the supply voltage for the analog circuitry in the ADE7758. The supply should be maintained at 5 V ± 5% for specified operation. Every effort should be made to minimize power supply ripple and noise at this pin by the use of proper decoupling. The Typical Performance Characteristics show the power supply rejection performance. This pin should be decoupled to AGND with a 10 µF capacitor in parallel with a ceramic 100 nF capacitor.
5, 6, 7, 8, 9, 10	IAP, IAN, IBP, IBN, ICP, ICN	Analog Inputs for Current Channel. This channel is used with the current transducer and is referenced in this document as the current channel. These inputs are fully differential voltage inputs with maximum differential input signal levels of ±0.5 V, ±0.25 V, and ±0.125 V, depending on the gain selections of the internal PGA (see the Analog Inputs section). All inputs have internal ESD protection circuitry. In addition, an overvoltage of ±6 V can be sustained on these inputs without risk of permanent damage.
11	AGND	This pin provides the ground reference for the analog circuitry in the ADE7758, that is, ADCs, temperature sensor, and reference. This pin should be tied to the analog ground plane or the quietest ground reference in the system. This quiet ground reference should be used for all analog circuitry, for example, antialiasing filters, current, and voltage transducers. To keep ground noise around the ADE7758 to a minimum, the quiet ground plane should be connected to the digital ground plane at only one point. It is acceptable to place the entire device on the analog ground plane.
12	$REF_{IN/OUT}$	This pin provides access to the on-chip voltage reference. The on-chip reference has a nominal value of 2.4 V ± 8% and a typical temperature coefficient of 30 ppm/°C. An external reference source can also be connected at this pin. In either case, this pin should be decoupled to AGND with a 1 µF ceramic capacitor.
13, 14, 15, 16	VN, VCP, VBP, VAP	Analog Inputs for the Voltage Channel. This channel is used with the voltage transducer and is referenced as the voltage channels in this document. These inputs are single-ended voltage inputs with the maximum signal level of ±0.5 V with respect to VN for specified operation. These inputs are voltage inputs with maximum input signal levels of ±0.5 V, ±0.25 V, and ±0.125 V, depending on the gain selections of the internal PGA (see the Analog Inputs section). All inputs have internal ESD protection circuitry, and in addition, an overvoltage of ±6 V can be sustained on these inputs without risk of permanent damage.

Pin No.	Mnemonic	Description
17	VARCF	Reactive Power Calibration Frequency Logic Output. It gives reactive power or apparent power information depending on the setting of the VACF bit of the WAVMODE register. This output is used for operational and calibration purposes. The full-scale output frequency can be scaled by writing to the VARCFNUM and VARCFDEN registers (see the Reactive Power Frequency Output section).
18	$\overline{IRQ}$	Interrupt Request Output. This is an active low open-drain logic output. Maskable interrupts include: an active energy register at half level, an apparent energy register at half level, and waveform sampling up to 26 kSPS (see the Interrupts section).
19	CLKIN	Master Clock for ADCs and Digital Signal Processing. An external clock can be provided at this logic input. Alternatively, a parallel resonant AT crystal can be connected across CLKIN and CLKOUT to provide a clock source for the ADE7758. The clock frequency for specified operation is 10 MHz. Ceramic load capacitors of a few tens of picofarad should be used with the gate oscillator circuit. Refer to the crystal manufacturer's data sheet for the load capacitance requirements
20	CLKOUT	A crystal can be connected across this pin and CLKIN as previously described to provide a clock source for the ADE7758. The CLKOUT pin can drive one CMOS load when either an external clock is supplied at CLKIN or a crystal is being used.
21	$\overline{CS}$	Chip Select. Part of the 4-wire serial interface. This active low logic input allows the ADE7758 to share the serial bus with several other devices (see the Serial Interface section).
22	DIN	Data Input for the Serial Interface. Data is shifted in at this pin on the falling edge of SCLK (see the Serial Interface section).
23	SCLK	Serial Clock Input for the Synchronous Serial Interface. All serial data transfers are synchronized to this clock (see the Serial Interface section). The SCLK has a Schmidt-trigger input for use with a clock source that has a slow edge transition time, for example, opto-isolator outputs.
24	DOUT	Data Output for the Serial Interface. Data is shifted out at this pin on the rising edge of SCLK. This logic output is normally in a high impedance state, unless it is driving data onto the serial data bus (see the Serial Interface section).

TERMINOLOGY

Measurement Error

The error associated with the energy measurement made by the ADE7758 is defined by

$$Measurement\ Error = \frac{Energy\ Registered\ by\ ADE7758 - True\ Energy}{True\ Energy} \times 100\% \quad (1)$$

Phase Error Between Channels

The high-pass filter (HPF) and digital integrator introduce a slight phase mismatch between the current and the voltage channel. The all-digital design ensures that the phase matching between the current channels and voltage channels in all three phases is within ±0.1° over a range of 45 Hz to 65 Hz and ±0.2° over a range of 40 Hz to 1 kHz. This internal phase mismatch can be combined with the external phase error (from current sensor or component tolerance) and calibrated with the phase calibration registers.

Power Supply Rejection (PSR)

This quantifies the ADE7758 measurement error as a percentage of reading when the power supplies are varied. For the ac PSR measurement, a reading at nominal supplies (5 V) is taken. A second reading is obtained with the same input signal levels when an ac signal (175 mV rms/100 Hz) is introduced onto the supplies. Any error introduced by this ac signal is expressed as a percentage of reading—see the Measurement Error definition.

For the dc PSR measurement, a reading at nominal supplies (5 V) is taken. A second reading is obtained with the same input signal levels when the power supplies are varied ±5%. Any error introduced is again expressed as a percentage of the reading.

ADC Offset Error

This refers to the dc offset associated with the analog inputs to the ADCs. It means that with the analog inputs connected to AGND that the ADCs still see a dc analog input signal. The magnitude of the offset depends on the gain and input range selection (see the Typical Performance Characteristics section). However, when HPFs are switched on, the offset is removed from the current channels and the power calculation is not affected by this offset.

Gain Error

The gain error in the ADCs of the ADE7758 is defined as the difference between the measured ADC output code (minus the offset) and the ideal output code (see the Current Channel ADC section and the Voltage Channel ADC section). The difference is expressed as a percentage of the ideal code.

Gain Error Match

The gain error match is defined as the gain error (minus the offset) obtained when switching between a gain of 1, 2, or 4. It is expressed as a percentage of the output ADC code obtained under a gain of 1.

TYPICAL PERFORMANCE CHARACTERISTICS

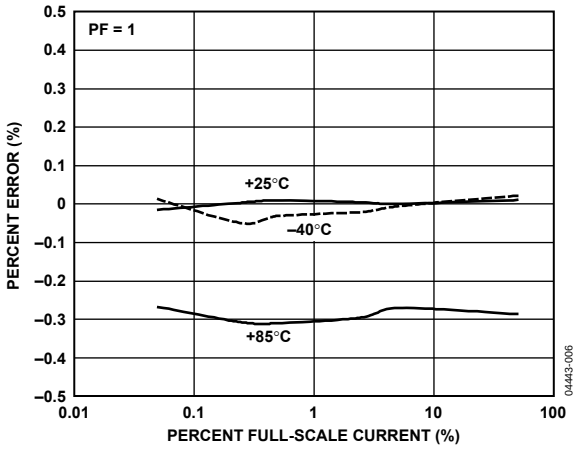


Figure 6. Active Energy Error as a Percentage of Reading (Gain = +1) over Temperature with Internal Reference and Integrator Off

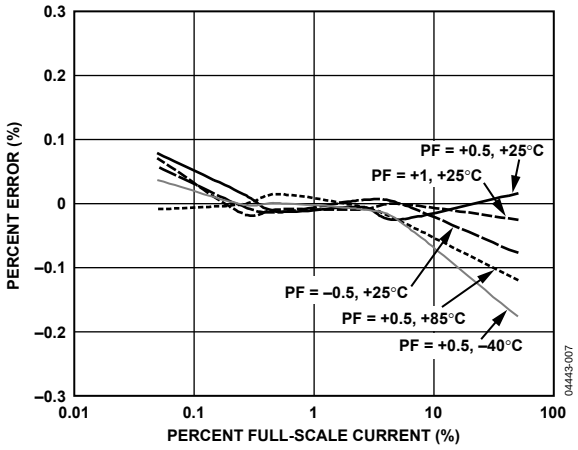


Figure 7. Active Energy Error as a Percentage of Reading (Gain = +1) over Power Factor with Internal Reference and Integrator Off

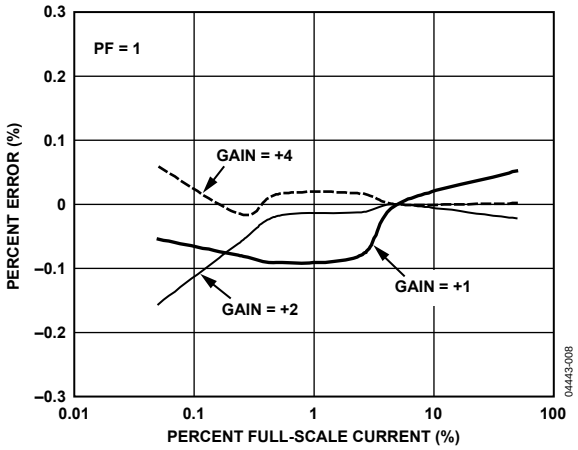


Figure 8. Active Energy Error as a Percentage of Reading over Gain with Internal Reference and Integrator Off

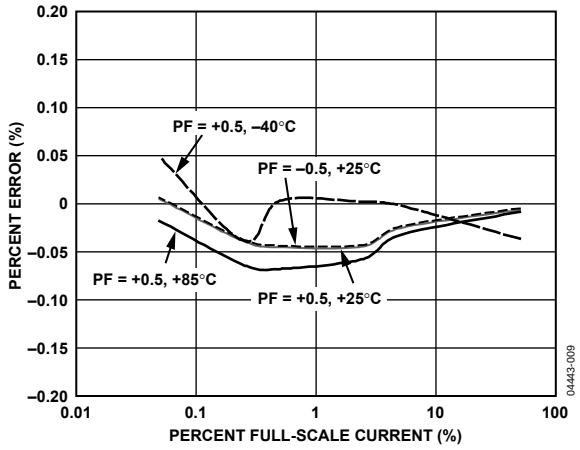


Figure 9. Active Energy Error as a Percentage of Reading (Gain = +1) over Temperature with External Reference and Integrator Off

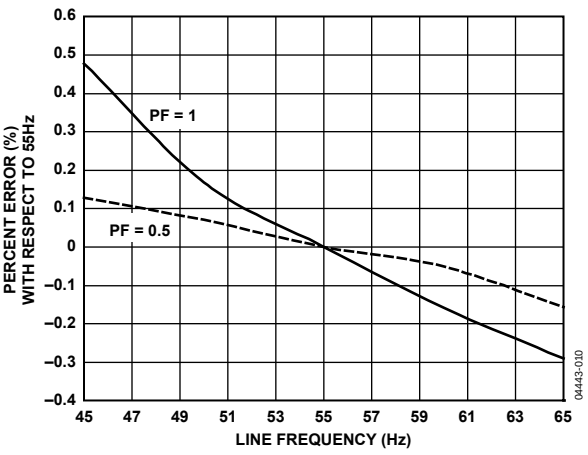


Figure 10. Active Energy Error as a Percentage of Reading (Gain = +1) over Frequency with Internal Reference and Integrator Off

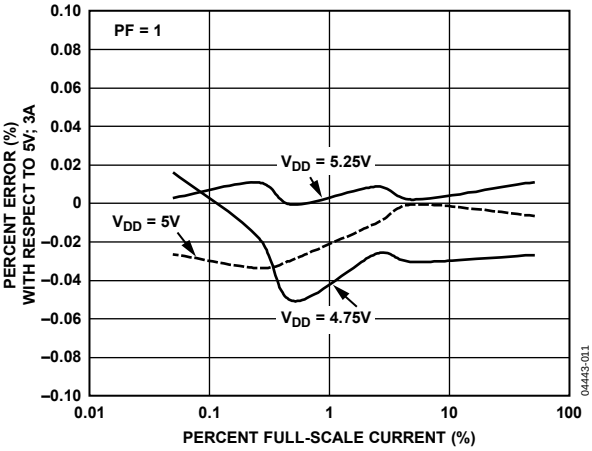


Figure 11. Active Energy Error as a Percentage of Reading (Gain = +1) over Power Supply with Internal Reference and Integrator Off

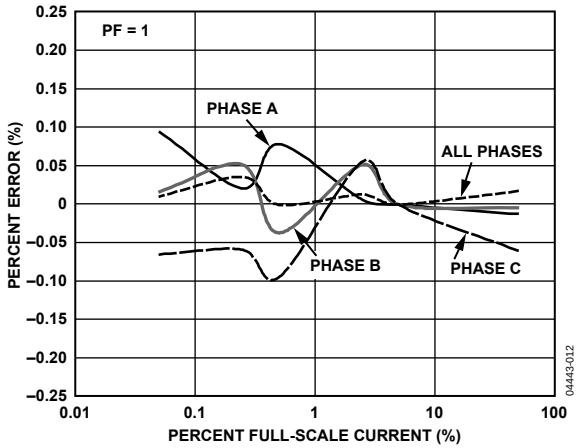


Figure 12. APCF Error as a Percentage of Reading (Gain = +1) with Internal Reference and Integrator Off

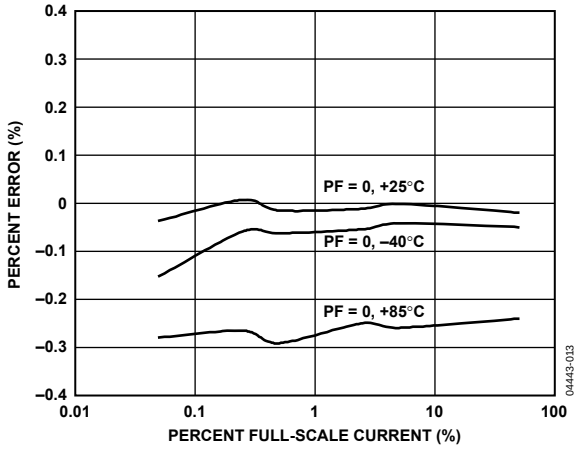


Figure 13. Reactive Energy Error as a Percentage of Reading (Gain = +1) over Temperature with Internal Reference and Integrator Off

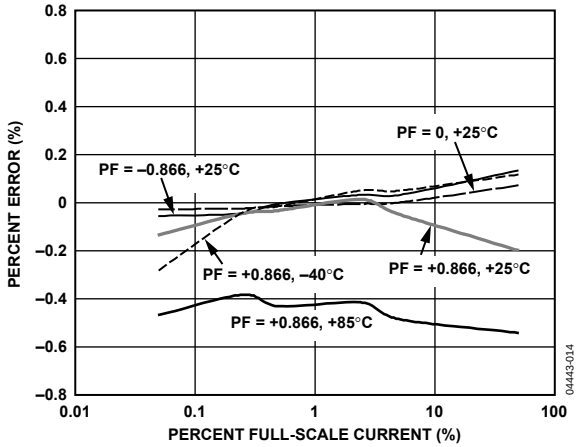


Figure 14. Reactive Energy Error as a Percentage of Reading (Gain = +1) over Power Factor with Internal Reference and Integrator Off

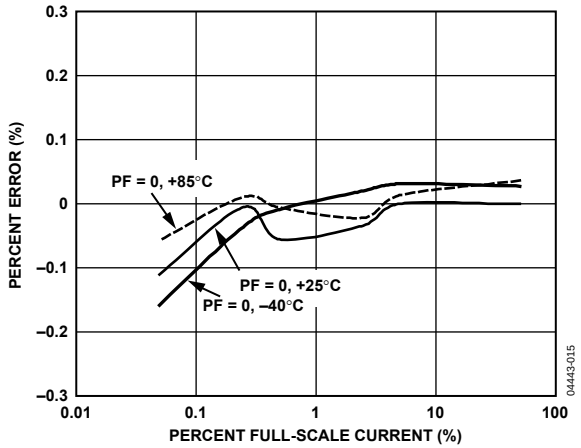


Figure 15. Reactive Energy Error as a Percentage of Reading (Gain = +1) over Temperature with External Reference and Integrator Off

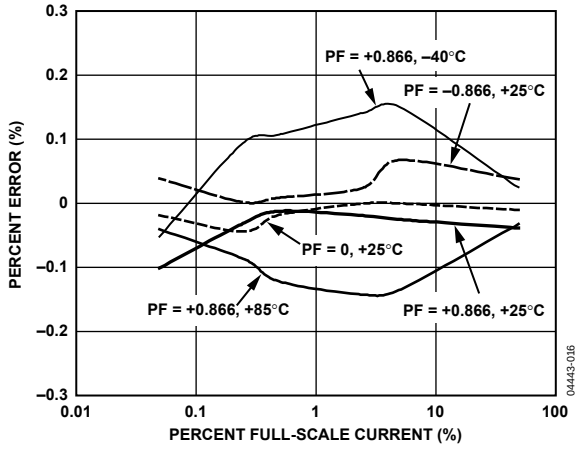


Figure 16. Reactive Energy Error as a Percentage of Reading (Gain = +1) over Power Factor with External Reference and Integrator Off

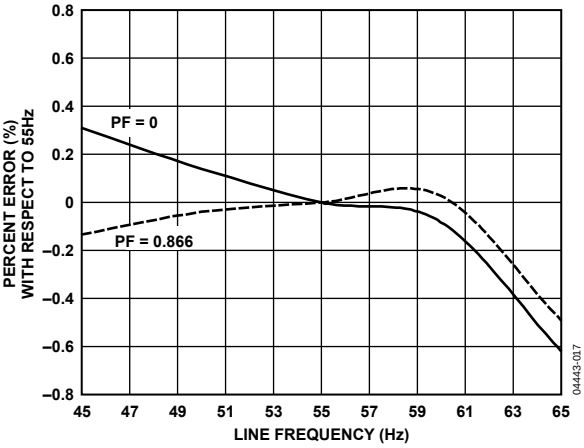


Figure 17. Reactive Energy Error as a Percentage of Reading (Gain = +1) over Frequency with Internal Reference and Integrator Off

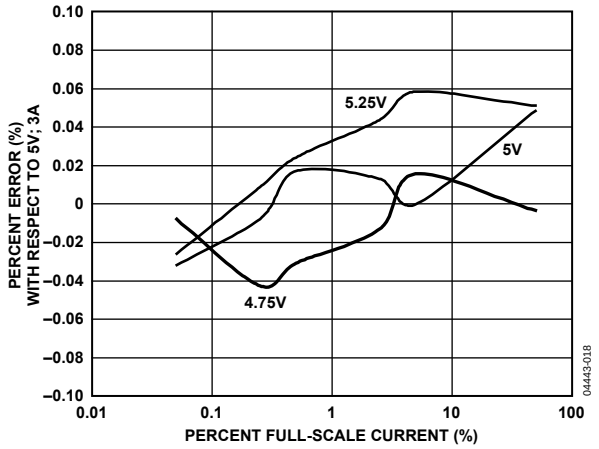


Figure 18. Reactive Energy Error as a Percentage of Reading (Gain = +1) over Supply with Internal Reference and Integrator Off

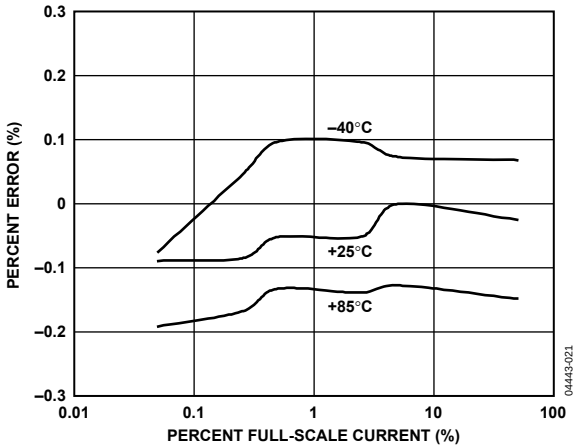


Figure 21. Active Energy Error as a Percentage of Reading (Gain = +4) over Temperature with Internal Reference and Integrator On

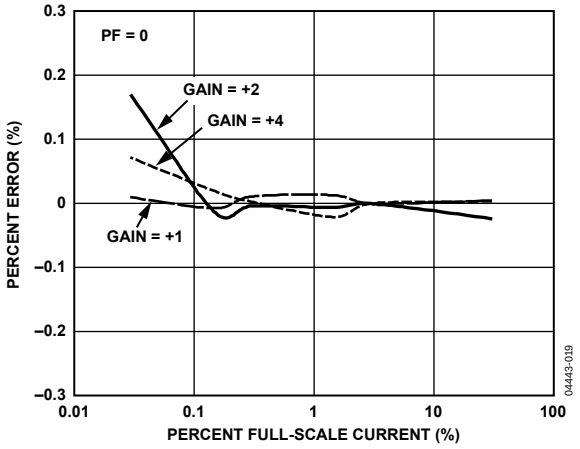


Figure 19. Reactive Energy Error as a Percentage of Reading over Gain with Internal Reference and Integrator Off

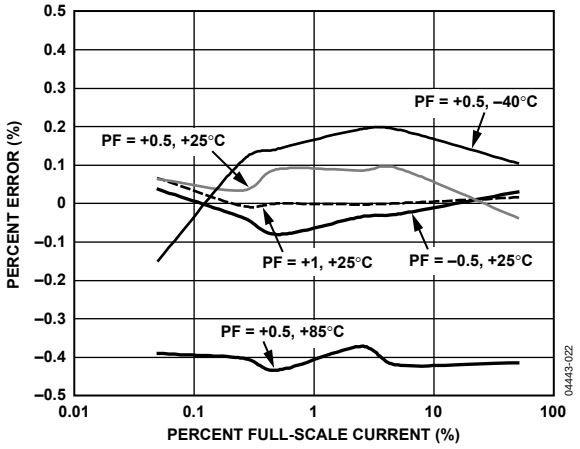


Figure 22. Active Energy Error as a Percentage of Reading (Gain = +4) over Power Factor with Internal Reference and Integrator On

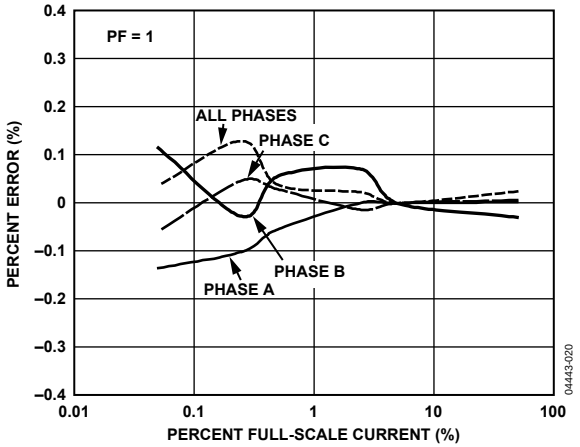


Figure 20. VARCF Error as a Percentage of Reading (Gain = +1) with Internal Reference and Integrator Off

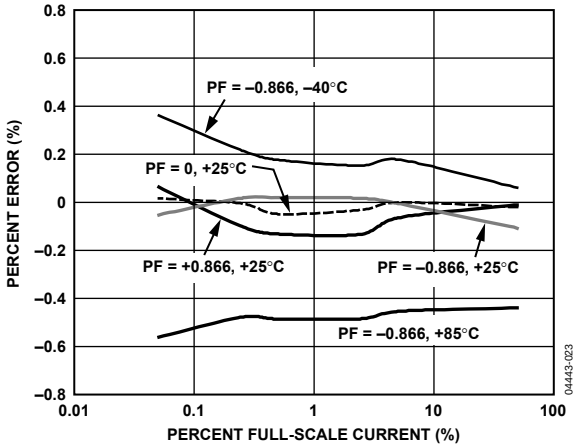


Figure 23. Reactive Energy Error as a Percentage of Reading (Gain = +4) over Power Factor with Internal Reference and Integrator On

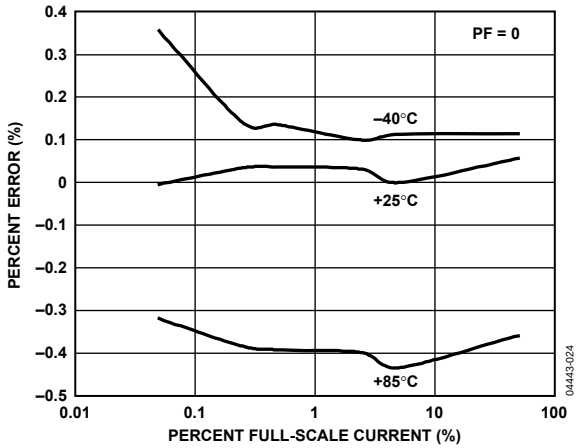


Figure 24. Reactive Energy Error as a Percentage of Reading (Gain = +4) over Temperature with Internal Reference and Integrator On

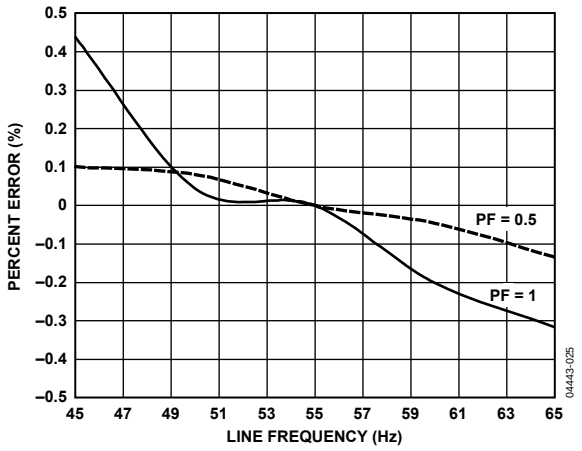


Figure 25. Active Energy Error as a Percentage of Reading (Gain = +4) over Frequency with Internal Reference and Integrator On

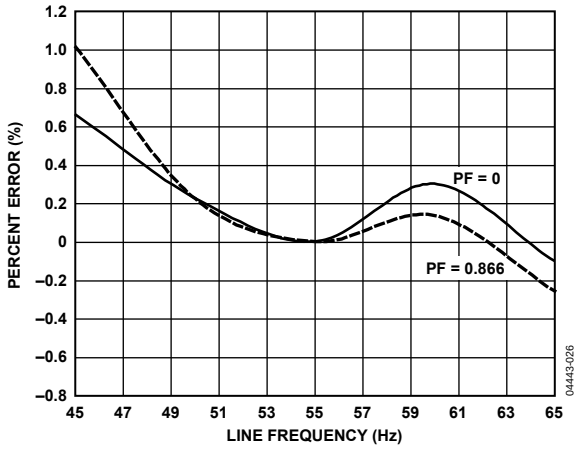


Figure 26. Reactive Energy Error as a Percentage of Reading (Gain = +4) over Frequency with Internal Reference and Integrator On

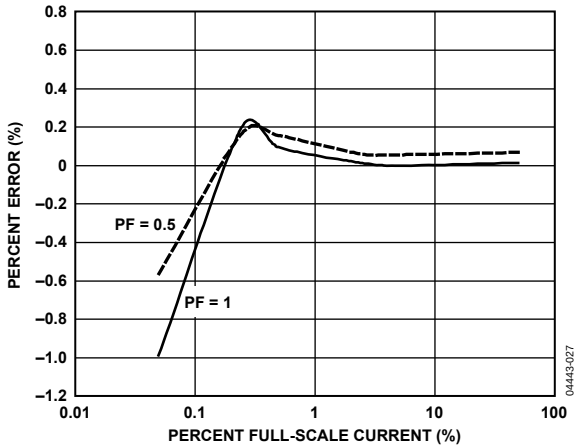


Figure 27. IRMS Error as a Percentage of Reading (Gain = +1) with Internal Reference and Integrator Off

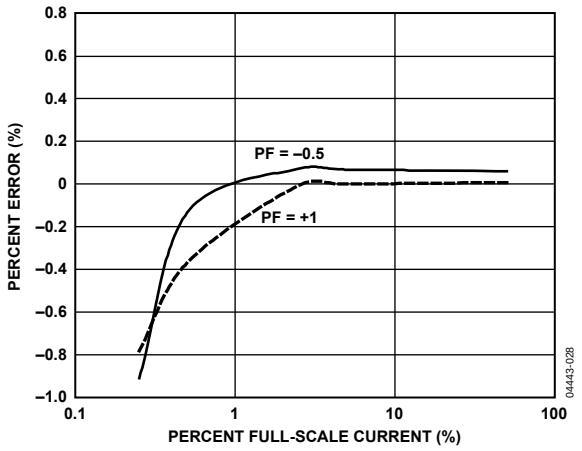


Figure 28. IRMS Error as a Percentage of Reading (Gain = +4) with Internal Reference and Integrator On

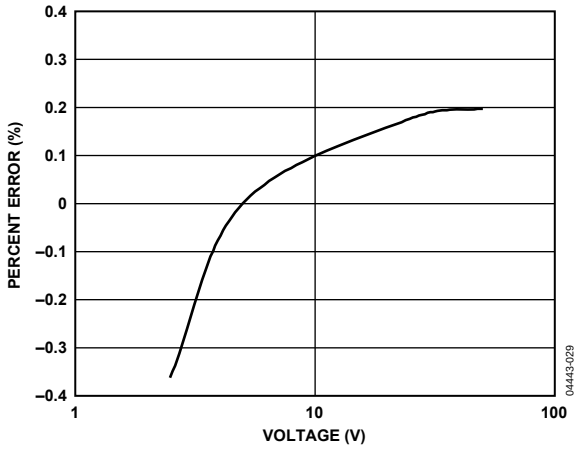


Figure 29. VRMS Error as a Percentage of Reading (Gain = +1) with Internal Reference

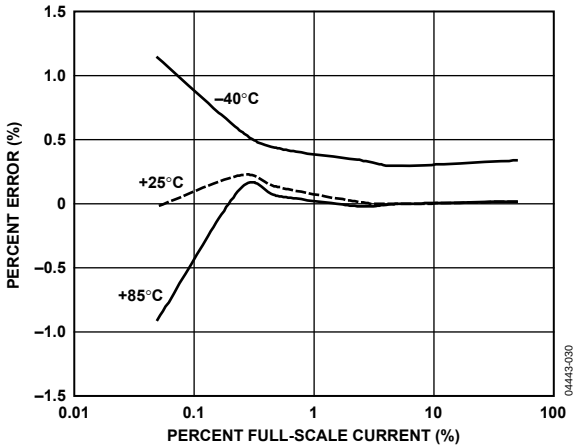


Figure 30. Apparent Energy Error as a Percentage of Reading (Gain = +1) over Temperature with Internal Reference and Integrator Off

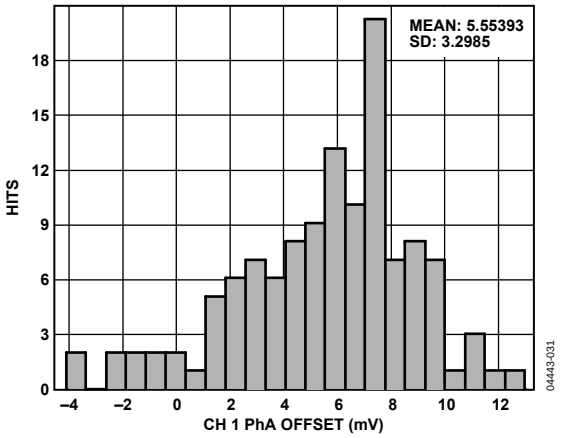


Figure 31. Phase A Channel 1 Offset Distribution

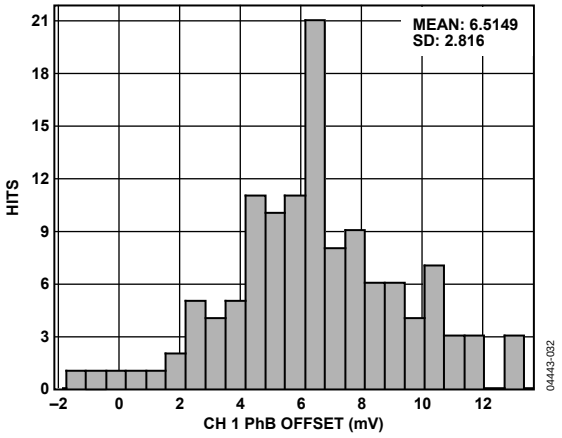


Figure 32. Phase B Channel 1 Offset Distribution

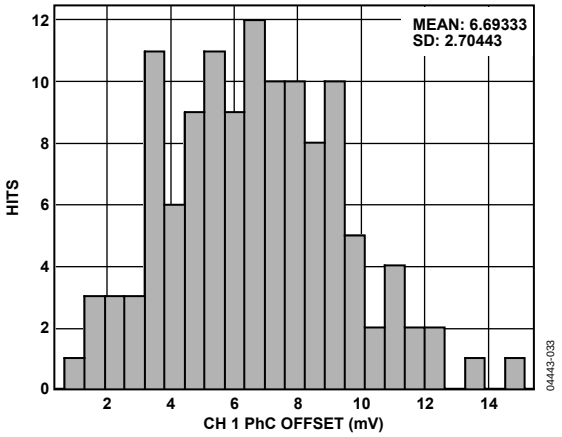


Figure 33. Phase C Channel 1 Offset Distribution

TEST CIRCUITS

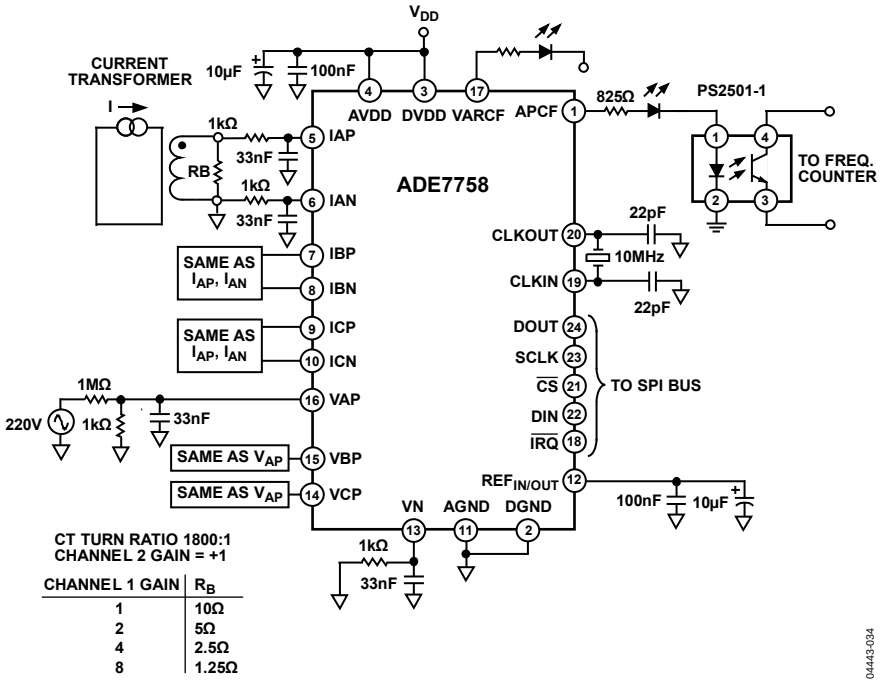


Figure 34. Test Circuit for Integrator Off

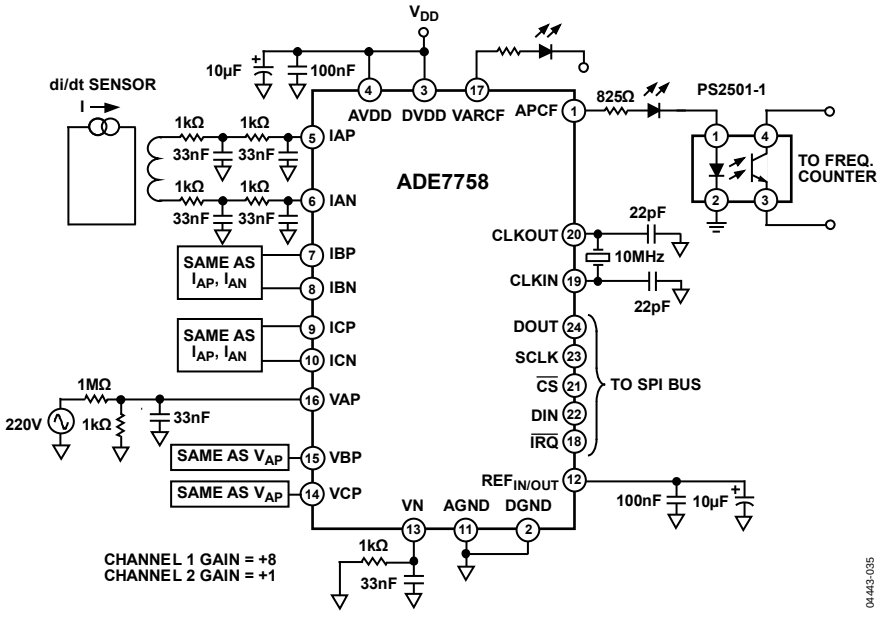


Figure 35. Test Circuit for Integrator On

THEORY OF OPERATION

ANTIALIASING FILTER

This filter prevents aliasing, which is an artifact of all sampled systems. Input signals with frequency components higher than half the ADC sampling rate distort the sampled signal at a frequency below half the sampling rate. This happens with all ADCs, regardless of the architecture. The combination of the high sampling rate Σ-Δ ADC used in the ADE7758 with the relatively low bandwidth of the energy meter allows a very simple low-pass filter (LPF) to be used as an antialiasing filter. A simple RC filter (single pole) with a corner frequency of 10 kHz produces an attenuation of approximately 40 dB at 833 kHz. This is usually sufficient to eliminate the effects of aliasing.

ANALOG INPUTS

The ADE7758 has six analog inputs divided into two channels: current and voltage. The current channel consists of three pairs of fully differential voltage inputs: IAP and IAN, IBP and IBN, and ICP and ICN. These fully differential voltage input pairs have a maximum differential signal of ±0.5 V. The current channel has a programmable gain amplifier (PGA) with possible gain selection of 1, 2, or 4. In addition to the PGA, the current channels also have a full-scale input range selection for the ADC. The ADC analog input range selection is also made using the gain register (see Figure 38). As mentioned previously, the maximum differential input voltage is ±0.5 V. However, by using Bit 3 and Bit 4 in the gain register, the maximum ADC input voltage can be set to ±0.5 V, ±0.25 V, or ±0.125 V on the current channels. This is achieved by adjusting the ADC reference (see the Reference Circuit section).

Figure 36 shows the maximum signal levels on the current channel inputs. The maximum common-mode signal is ±25 mV, as shown in Figure 37.

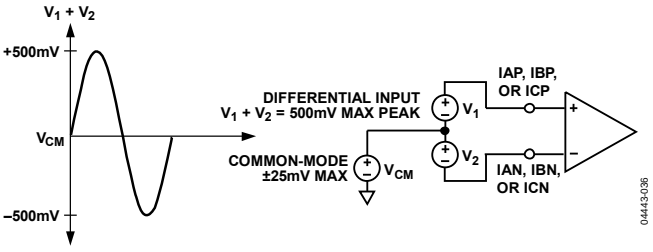


Figure 36. Maximum Signal Levels, Current Channels, Gain = 1

The voltage channel has three single-ended voltage inputs: VAP, VBP, and VCP. These single-ended voltage inputs have a maximum input voltage of ±0.5 V with respect to VN. Both the current and voltage channel have a PGA with possible gain selections of 1, 2, or 4. The same gain is applied to all the inputs of each channel.

Figure 37 shows the maximum signal levels on the voltage channel inputs. The maximum common-mode signal is ±25 mV, as shown in Figure 36.

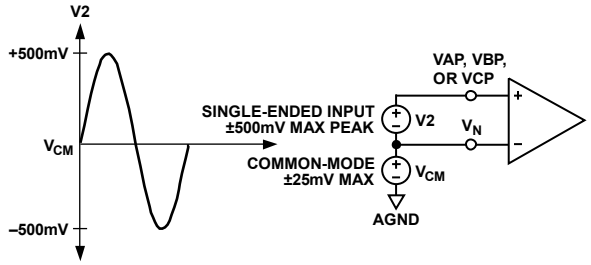


Figure 37. Maximum Signal Levels, Voltage Channels, Gain = 1

The gain selections are made by writing to the gain register. Bit 0 to Bit 1 select the gain for the PGA in the fully differential current channel. The gain selection for the PGA in the single-ended voltage channel is made via Bit 5 to Bit 6. Figure 38 shows how a gain selection for the current channel is made using the gain register.

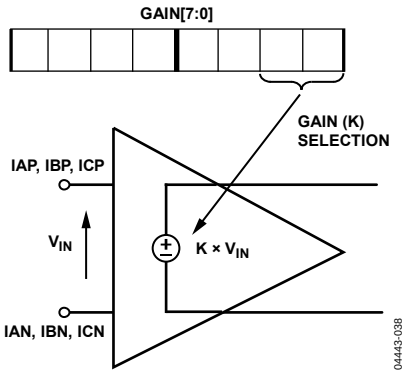


Figure 38. PGA in Current Channel

Figure 39 shows how the gain settings in PGA 1 (current channel) and PGA 2 (voltage channel) are selected by various bits in the gain register.

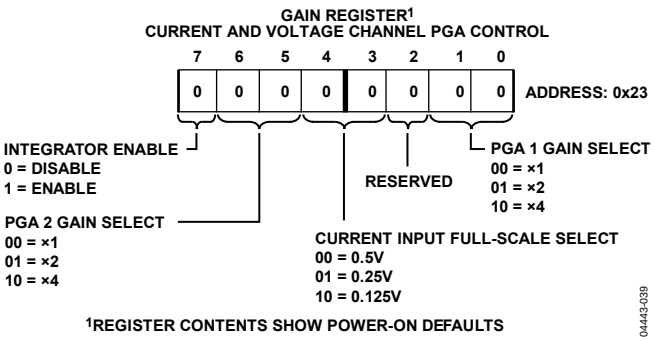


Figure 39. Analog Gain Register

Bit 7 of the gain register is used to enable the digital integrator in the current signal path. Setting this bit activates the digital integrator (see the DI/DT Current Sensor and Digital Integrator section).

CURRENT CHANNEL ADC

Figure 41 shows the ADC and signal processing path for the input IA of the current channels (same for IB and IC). In waveform sampling mode, the ADC outputs are signed twos complement 24-bit data-words at a maximum of 26.0 kSPS (thousand samples per second). With the specified full-scale analog input signal of ±0.5 V, the ADC produces its maximum output code value (see Figure 41). This diagram shows a full-scale voltage signal being applied to the differential inputs IAP and IAN. The ADC output swings between 0xD7AE14 (−2,642,412) and 0x2851EC (+2,642,412).

Current Channel Sampling

The waveform samples of the current channel can be routed to the WFORM register at fixed sampling rates by setting the WAVSEL[2:0] bit in the WAVMODE register to 000 (binary) (see Table 20). The phase in which the samples are routed is set by setting the PHSEL[1:0] bits in the WAVMODE register. Energy calculation remains uninterrupted during waveform sampling.

When in waveform sample mode, one of four output sample rates can be chosen by using Bit 5 and Bit 6 of the WAVMODE register (DTRT[1:0]). The output sample rate can be 26.04 kSPS, 13.02 kSPS, 6.51 kSPS, or 3.25 kSPS. By setting the WFSM bit in the interrupt mask register to Logic 1, the interrupt request output $\overline{\text{IRQ}}$ goes active low when a sample is available. The timing is shown in Figure 40. The 24-bit waveform samples are transferred from the ADE7758 one byte (8-bits) at a time, with the most significant byte shifted out first.

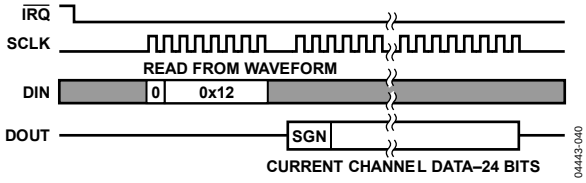


Figure 40. Current Channel Waveform Sampling

The interrupt request output $\overline{\text{IRQ}}$ stays low until the interrupt routine reads the reset status register (see the Interrupts section).

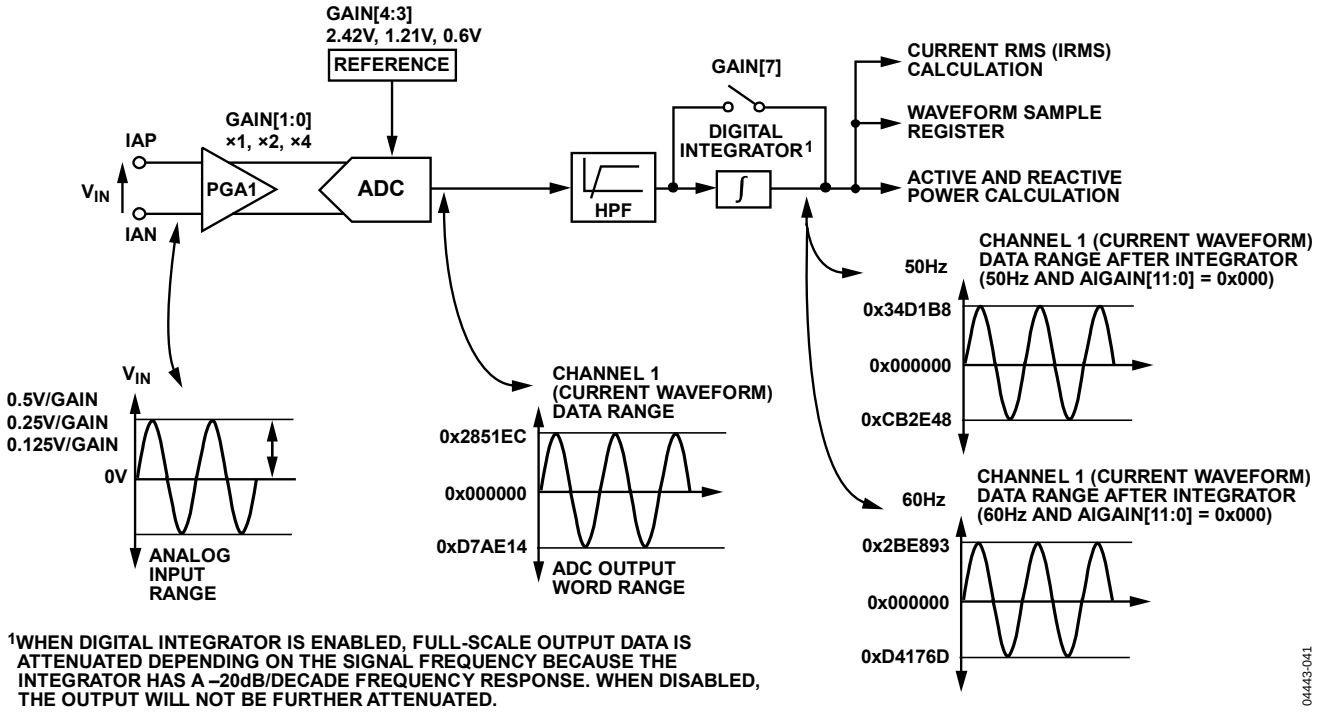


Figure 41. Current Channel Signal Path

DI/DT CURRENT SENSOR AND DIGITAL INTEGRATOR

The di/dt sensor detects changes in the magnetic field caused by the ac current. Figure 42 shows the principle of a di/dt current sensor.

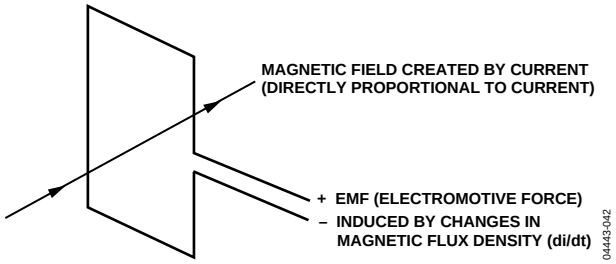


Figure 42. Principle of a di/dt Current Sensor

The flux density of a magnetic field induced by a current is directly proportional to the magnitude of the current. The changes in the magnetic flux density passing through a conductor loop generate an electromotive force (EMF) between the two ends of the loop. The EMF is a voltage signal that is proportional to the di/dt of the current. The voltage output from the di/dt current sensor is determined by the mutual inductance between the current carrying conductor and the di/dt sensor.

The current signal needs to be recovered from the di/dt signal before it can be used. An integrator is therefore necessary to restore the signal to its original form. The ADE7758 has a built-in digital integrator to recover the current signal from the di/dt sensor. The digital integrator on Channel 1 is disabled by default when the ADE7758 is powered up. Setting the MSB of the GAIN[7:0] register turns on the integrator. Figure 43 to Figure 46 show the magnitude and phase response of the digital integrator.

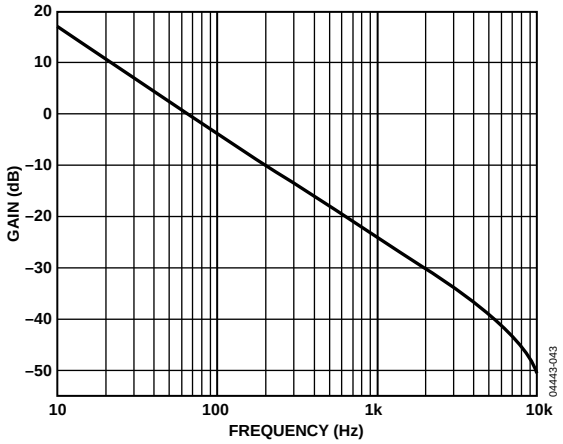


Figure 43. Combined Gain Response of the Digital Integrator and Phase Compensator

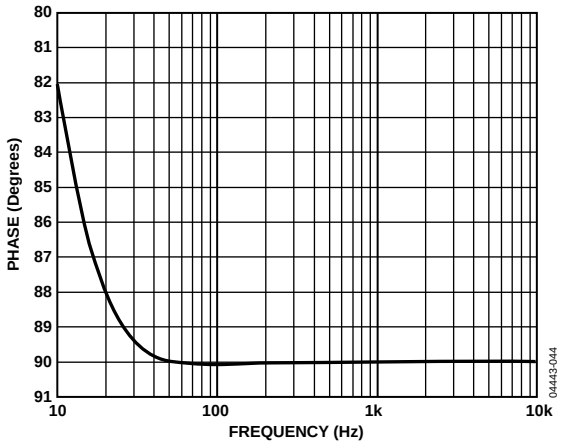


Figure 44. Combined Phase Response of the Digital Integrator and Phase Compensator

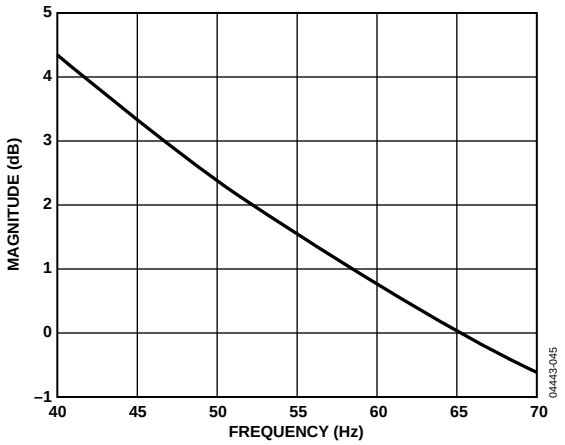


Figure 45. Combined Gain Response of the Digital Integrator and Phase Compensator (40 Hz to 70 Hz)

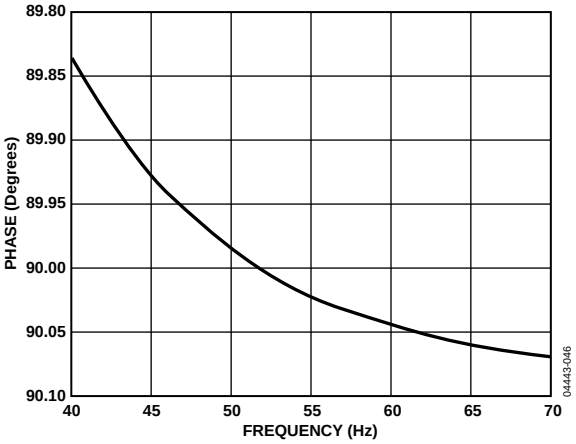


Figure 46. Combined Phase Response of the Digital Integrator and Phase Compensator (40 Hz to 70 Hz)

Note that the integrator has a −20 dB/dec attenuation and approximately −90° phase shift. When combined with a di/dt sensor, the resulting magnitude and phase response should be a flat gain over the frequency band of interest. However, the di/dt sensor has a 20 dB/dec gain associated with it and generates significant high frequency noise. A more effective antialiasing filter is needed to avoid noise due to aliasing (see the Theory of Operation section).

When the digital integrator is switched off, the ADE7758 can be used directly with a conventional current sensor, such as a current transformer (CT) or a low resistance current shunt.

PEAK CURRENT DETECTION

The ADE7758 can be programmed to record the peak of the current waveform and produce an interrupt if the current exceeds a preset limit.

Peak Current Detection Using the PEAK Register

The peak absolute value of the current waveform within a fixed number of half-line cycles is stored in the IPEAK register. Figure 47 illustrates the timing behavior of the peak current detection.

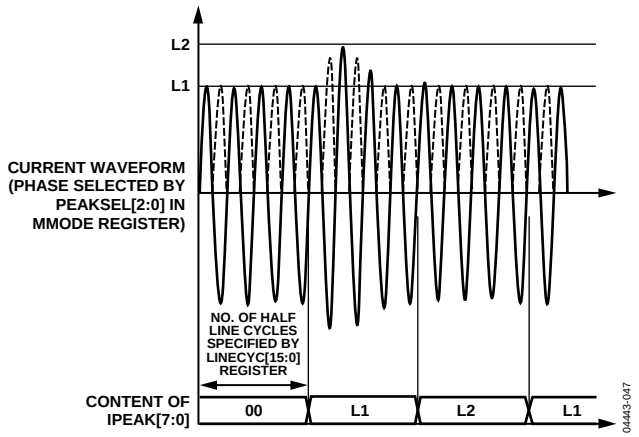


Figure 47. Peak Current Detection Using the IPEAK Register

Note that the content of the IPEAK register is equivalent to Bit 14 to Bit 21 of the current waveform sample. At full-scale analog input, the current waveform sample is 0x2851EC. The IPEAK at full-scale input is therefore expected to be 0xA1.

In addition, multiple phases can be activated for the peak detection simultaneously by setting more than one of the PEAKSEL[2:4] bits in the MMODE register to logic high. These bits select the phase for both voltage and current peak measurements. Note that if more than one bit is set, the VPEAK and IPEAK registers can hold values from two different phases, that is, the voltage and current peak are independently processed (see the Peak Current Detection section).

Note that the number of half-line cycles is based on counting the zero crossing of the voltage channel. The ZXSEL[2:0] bits in the LCYCMODE register determine which voltage channels are used for the zero-crossing detection. The same signal is also used for line cycle energy accumulation mode if activated (see the Line Cycle Accumulation Mode Register (0X17) section).

OVERCURRENT DETECTION INTERRUPT

Figure 48 illustrates the behavior of the overcurrent detection.

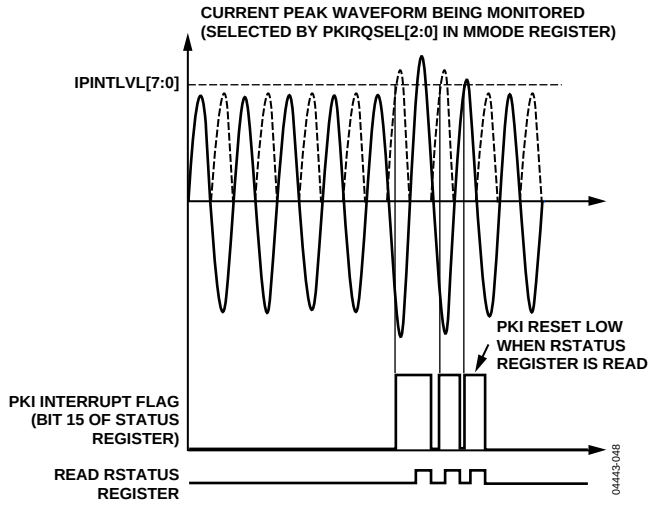


Figure 48. ADE7758 Overcurrent Detection

Note that the content of the IPINTLVL[7:0] register is equivalent to Bit 14 to Bit 21 of the current waveform sample. Therefore, setting this register to 0xA1 represents putting peak detection at full-scale analog input. Figure 48 shows a current exceeding a threshold. The overcurrent event is recorded by setting the PKI flag (Bit 15) in the interrupt status register. If the PKI enable bit is set to Logic 1 in the interrupt mask register, the $\overline{\text{IRQ}}$ logic output goes active low (see the Interrupts section).

Similar to peak level detection, multiple phases can be activated for peak detection. If any of the active phases produce waveform samples above the threshold, the PKI flag in the interrupt status register is set. The phase of which overcurrent is monitored is set by the PKIRQSEL[2:0] bits in the MMODE register (see Table 19).

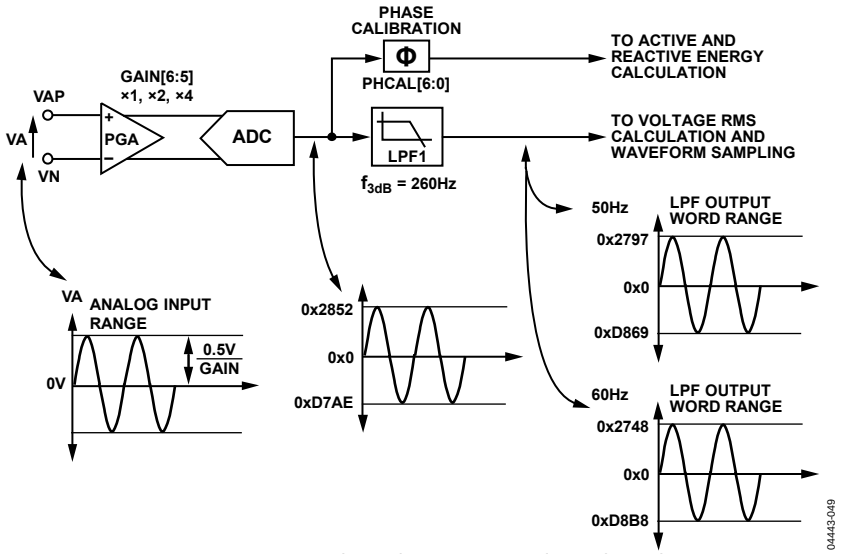


Figure 49. ADC and Signal Processing in Voltage Channel

VOLTAGE CHANNEL ADC

Figure 49 shows the ADC and signal processing chain for the input VA in the voltage channel. The VB and VC channels have similar processing chains.

For active and reactive energy measurements, the output of the ADC passes to the multipliers directly and is not filtered. This solution avoids the much larger multibit multiplier and does not affect the accuracy of the measurement. An HPF is not implemented on the voltage channel to remove the dc offset because the HPF on the current channel alone should be sufficient to eliminate error due to ADC offsets in the power calculation. However, ADC offset in the voltage channels produces large errors in the voltage rms calculation and affects the accuracy of the apparent energy calculation.

Voltage Channel Sampling

The waveform samples on the voltage channels can also be routed to the WFORM register. However, before passing to the WFORM register, the ADC outputs pass through a single-pole, low-pass filter (LPF1) with a cutoff frequency at 260 Hz. Figure 50 shows the magnitude and phase response of LPF1. This filter attenuates the signal slightly. For example, if the line frequency is 60 Hz, the signal at the output of LPF1 is attenuated by 3.575%. The waveform samples are 16-bit, twos complement data ranging between 0x2748 (+10,056d) and 0xD8B8 (−10,056d). The data is sign extended to 24-bit in the WFORM register.

$$H(f) = \frac{1}{\sqrt{1+\left(\frac{60\text{ Hz}}{260\text{ Hz}}\right)^2}} = 0.974 = -0.225\text{ dB} \quad (3)$$

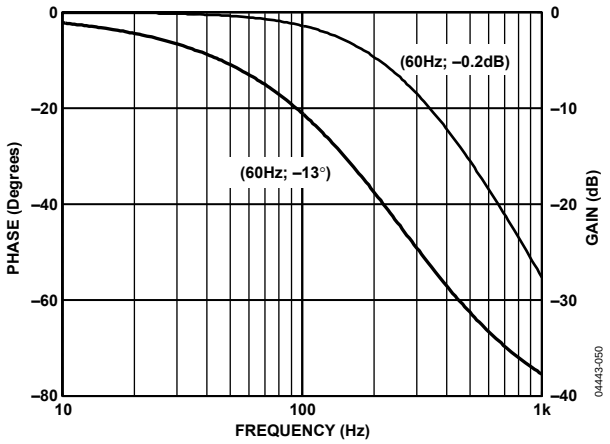


Figure 50. Magnitude and Phase Response of LPF1

Note that LPF1 does not affect the active and reactive energy calculation because it is only used in the waveform sampling signal path. However, waveform samples are used for the voltage rms calculation and the subsequent apparent energy accumulation.

The WAVSEL[2:0] bits in the WAVMODE register should be set to 001 (binary) to start the voltage waveform sampling. The PHSEL[1:0] bits control the phase from which the samples are routed. In waveform sampling mode, one of four output sample rates can be chosen by changing Bit 5 and Bit 6 of the WAVMODE register (see Table 20). The available output sample rates are 26.0 kSPS, 13.5 kSPS, 6.5 kSPS, or 3.3 kSPS. By setting the WFSM bit in the interrupt mask register to Logic 1, the interrupt request output $\overline{\text{IRQ}}$ goes active low when a sample is available. The 24-bit waveform samples are transferred from the ADE7758 one byte (8 bits) at a time, with the most significant byte shifted out first.

The sign of the register is extended in the upper 8 bits. The timing is the same as for the current channels, as seen in Figure 40.

ZERO-CROSSING DETECTION

The ADE7758 has zero-crossing detection circuits for each of the voltage channels (VAN, VBN, and VCN). Figure 51 shows how the zero-cross signal is generated from the output of the ADC of the voltage channel.

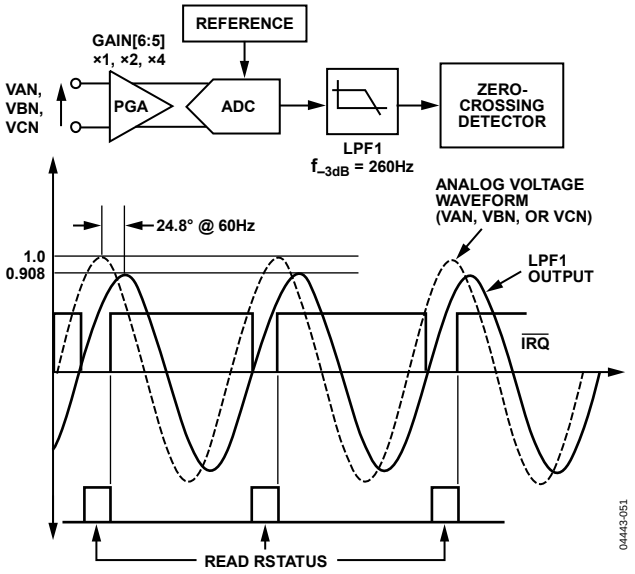


Figure 51. Zero-Crossing Detection on Voltage Channels

The zero-crossing interrupt is generated from the output of LPF1. LPF1 has a single pole at 260 Hz (CLKIN = 10 MHz). As a result, there is a phase lag between the analog input signal of the voltage channel and the output of LPF1. The phase response of this filter is shown in the Voltage Channel Sampling section. The phase lag response of LPF1 results in a time delay of approximately 1.1 ms (at 60 Hz) between the zero crossing on the voltage inputs and the resulting zero-crossing signal. Note that the zero-crossing signal is used for the line cycle accumulation mode, zero-crossing interrupt, and line period/frequency measurement.

When one phase crosses from negative to positive, the corresponding flag in the interrupt status register (Bit 9 to Bit 11) is set to Logic 1. An active low in the IRQ output also appears if the corresponding ZX bit in the interrupt mask register is set to Logic 1. Note that only zero crossing from negative to positive generates an interrupt.

The flag in the interrupt status register is reset to 0 when the interrupt status register with reset (RSTATUS) is read. Each phase has its own interrupt flag and mask bit in the interrupt register.

Zero-Crossing Timeout

Each zero-crossing detection has an associated internal timeout register (not accessible to the user). This unsigned, 16-bit register is decreased by 1 every 384/CLKIN seconds. The registers are reset to a common user-programmed value, that is, the zero-crossing timeout register (ZXTOUT[15:0], Address 0x1B), every time a zero crossing is detected on its associated input. The default value of ZXTOUT is 0xFFFF. If the internal register decrements to 0 before a zero crossing at the corresponding input is detected, it indicates an absence of a zero crossing in the time determined by the ZXTOUT[15:0]. The ZXTOx detection bit of the corresponding phase in the interrupt status register is then switched on (Bit 6 to Bit 8). An active low on the IRQ output also appears if the ZXTOx mask bit for the corresponding phase in the interrupt mask register is set to Logic 1. Figure 52 shows the mechanism of the zero-crossing timeout detection when the Line Voltage A stays at a fixed dc level for more than 384/CLKIN × ZXTOUT[15:0] seconds.

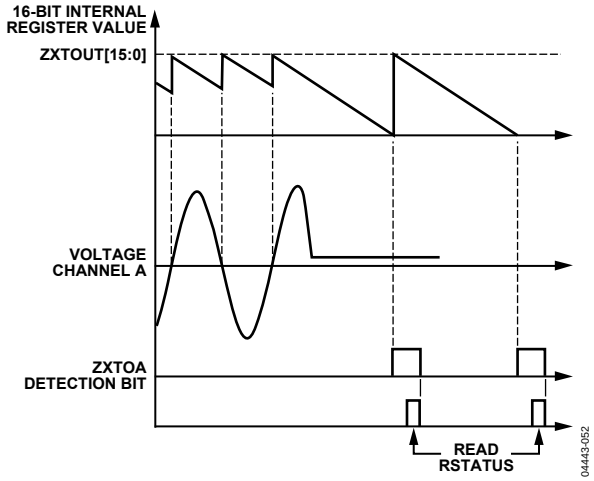


Figure 52. Zero-Crossing Timeout Detection

PHASE COMPENSATION

When the HPF in the current channel is disabled, the phase error between the current channel (IA, IB, or IC) and the corresponding voltage channel (VA, VB, or VC) is negligible. When the HPF is enabled, the current channels have phase response (see Figure 53 through Figure 55). The phase response is almost 0 from 45 Hz to 1 kHz. The frequency band is sufficient for the requirements of typical energy measurement applications.

However, despite being internally phase compensated, the ADE7758 must work with transducers that may have inherent phase errors. For example, a current transformer (CT) with a phase error of 0.1° to 0.3° is not uncommon. These phase errors can vary from part to part, and they must be corrected to perform accurate power calculations.

The errors associated with phase mismatch are particularly noticeable at low power factors. The ADE7758 provides a means of digitally calibrating these small phase errors. The ADE7758 allows a small time delay or time advance to be introduced into the signal processing chain to compensate for the small phase errors.

The phase calibration registers (APHCAL, BPHCAL, and CPHCAL) are twos complement, 7-bit sign-extended registers that can vary the time advance in the voltage channel signal path from +153.6 µs to −75.6 µs (CLKIN = 10 MHz),

respectively. Negative values written to the PHCAL registers represent a time advance, and positive values represent a time delay. One LSB is equivalent to 1.2 µs of time delay or 2.4 µs of time advance with a CLKIN of 10 MHz. With a line frequency of 60 Hz, this gives a phase resolution of 0.026° (360° × 1.2 µs × 60 Hz) at the fundamental in the positive direction (delay) and 0.052° in the negative direction (advance). This corresponds to a total correction range of −3.32° to +1.63° at 60 Hz.

Figure 56 illustrates how the phase compensation is used to remove a 0.1° phase lead in IA of the current channel from the external current transducer. To cancel the lead (0.1°) in the current channel of Phase A, a phase lead must be introduced into the corresponding voltage channel. The resolution of the phase adjustment allows the introduction of a phase lead of 0.104°. The phase lead is achieved by introducing a time advance into VA. A time advance of 4.8 µs is made by writing −2 (0x7E) to the time delay block (APHCAL[6:0]), thus reducing the amount of time delay by 4.8 µs or equivalently, 360° × 4.8 µs × 60 Hz = 0.104° at 60 Hz.

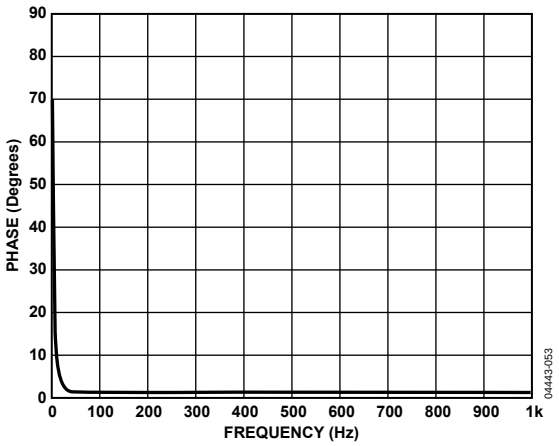


Figure 53. Phase Response of the HPF and Phase Compensation (10 Hz to 1 kHz)

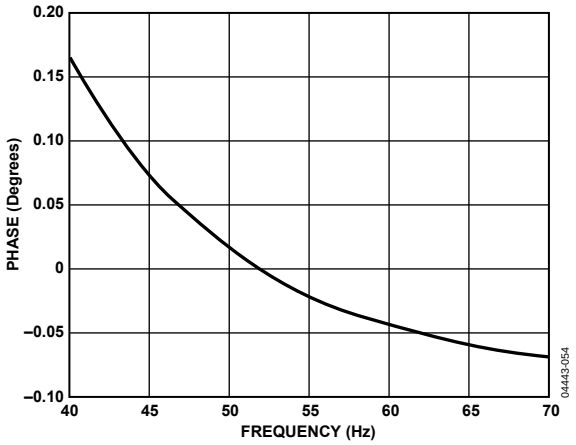


Figure 54. Phase Response of the HPF and Phase Compensation (40 Hz to 70 Hz)

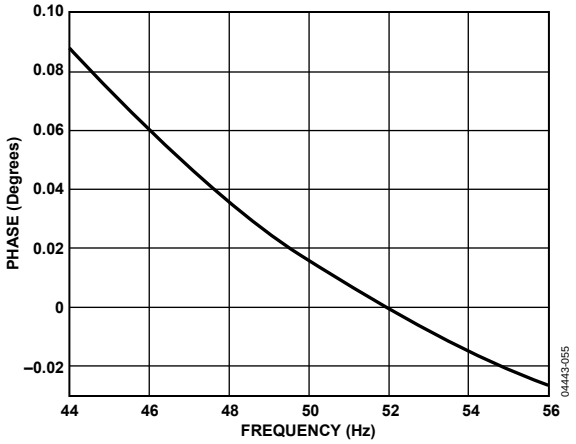


Figure 55. Phase Response of HPF and Phase Compensation (44 Hz to 56 Hz)

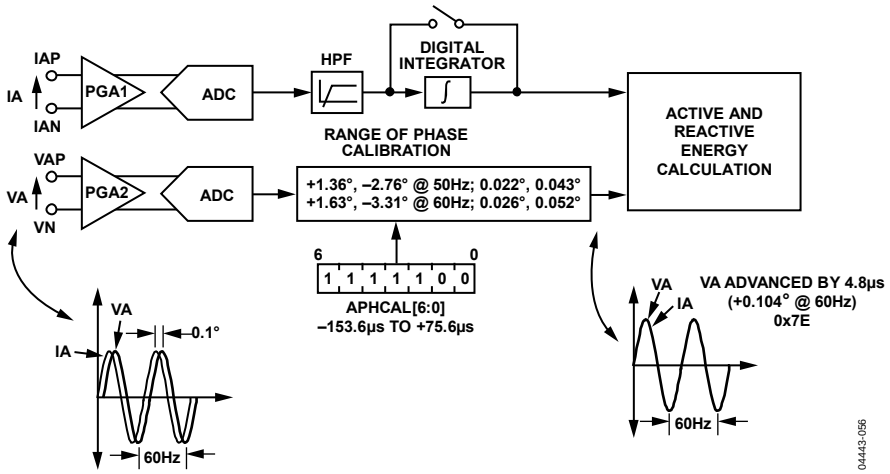


Figure 56. Phase Calibration on Voltage Channels

PERIOD MEASUREMENT

The ADE7758 provides the period or frequency measurement of the line voltage. The period is measured on the phase specified by Bit 0 to Bit 1 of the MMODE register. The period register is an unsigned 12-bit FREQ register and is updated every four periods of the selected phase.

Bit 7 of the LCYCMODE selects whether the period register displays the frequency or the period. Setting this bit causes the register to display the period. The default setting is logic low, which causes the register to display the frequency.

When set to measure the period, the resolution of this register is 96/CLKIN per LSB (9.6 µs/LSB when CLKIN is 10 MHz), which represents 0.06% when the line frequency is 60 Hz. At 60 Hz, the value of the period register is 1737d. At 50 Hz, the value of the period register is 2084d. When set to measure frequency, the value of the period register is approximately 960d at 60 Hz and 800d at 50 Hz. This is equivalent to 0.0625 Hz/LSB.

LINE VOLTAGE SAG DETECTION

The ADE7758 can be programmed to detect when the absolute value of the line voltage of any phase drops below a certain peak value for a number of half cycles. Each phase of the voltage channel is controlled simultaneously. This condition is illustrated in Figure 57.

Figure 57 shows a line voltage fall below a threshold, which is set in the SAG level register (SAGLVL[7:0]), for nine half cycles. Because the SAG cycle register indicates a six half-cycle threshold (SAGCYC[7:0] = 0x06), the SAG event is recorded at the end of the sixth half cycle by setting the SAG flag of the corresponding phase in the interrupt status register (Bit 1 to Bit 3 in the interrupt status register).

If the SAG enable bit is set to Logic 1 for this phase (Bit 1 to Bit 3 in the interrupt mask register), the $\overline{\text{IRQ}}$ logic output goes active low (see the Interrupts section). The phases are compared to the same parameters defined in the SAGLVL and SAGCYC registers.

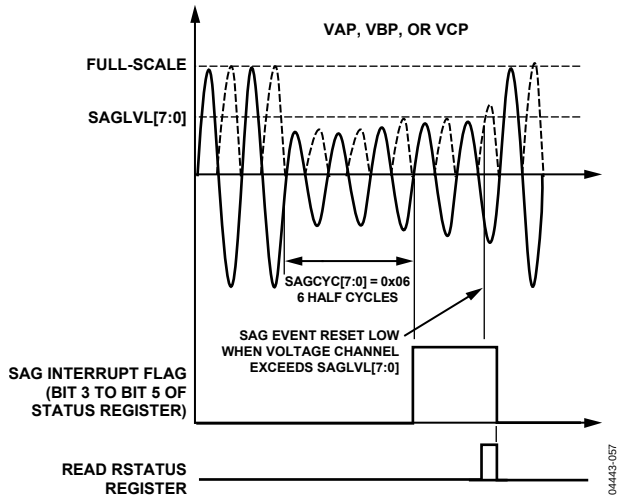


Figure 57. ADE7758 SAG Detection

Figure 57 shows a line voltage fall below a threshold, which is set in the SAG level register (SAGLVL[7:0]), for nine half cycles. Because the SAG cycle register indicates a six half-cycle threshold (SAGCYC[7:0] = 0x06), the SAG event is recorded at the end of the sixth half cycle by setting the SAG flag of the corresponding phase in the interrupt status register (Bit 1 to Bit 3 in the interrupt status register). If the SAG enable bit is set to Logic 1 for this phase (Bit 1 to Bit 3 in the interrupt mask register), the $\overline{\text{IRQ}}$ logic output goes active low (see the Interrupts section). The phases are compared to the same parameters defined in the SAGLVL and SAGCYC registers.

SAG LEVEL SET

The contents of the single-byte SAG level register, SAGLVL[0:7], are compared to the absolute value of Bit 6 to Bit 13 from the voltage waveform samples. For example, the nominal maximum code of the voltage channel waveform samples with a full-scale signal input at 60 Hz is 0x2748 (see the Voltage Channel Sampling section). Bit 13 to Bit 6 are 0x9D. Therefore, writing 0x9D to the SAG level register puts the SAG detection level at full scale and sets the SAG detection to its most sensitive value.

The detection is made when the content of the SAGLVL[7:0] register is greater than the incoming sample. Writing 0x00 puts the SAG detection level at 0. The detection of a decrease of an input voltage is disabled in this case.

PEAK VOLTAGE DETECTION

The ADE7758 can record the peak of the voltage waveform and produce an interrupt if the current exceeds a preset limit.

Peak Voltage Detection Using the VPEAK Register

The peak absolute value of the voltage waveform within a fixed number of half-line cycles is stored in the VPEAK register. Figure 58 illustrates the timing behavior of the peak voltage detection.

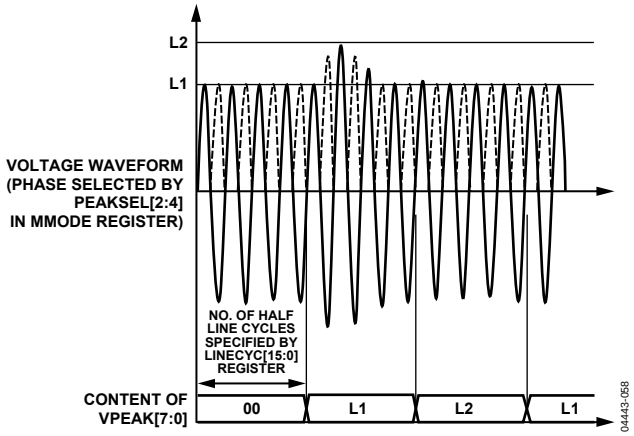


Figure 58. Peak Voltage Detection Using the VPEAK Register

Note that the content of the VPEAK register is equivalent to Bit 6 to Bit 13 of the 16-bit voltage waveform sample. At full-scale analog input, the voltage waveform sample at 60 Hz is 0x2748. The VPEAK at full-scale input is, therefore, expected to be 0x9D.

In addition, multiple phases can be activated for the peak detection simultaneously by setting multiple bits among the PEAKSEL[2:4] bits in the MMODE register. These bits select the phase for both voltage and current peak measurements.

Note that if more than one bit is set, the VPEAK and IPEAK registers can hold values from two different phases, that is, the voltage and current peak are independently processed (see the Peak Current Detection section).

Note that the number of half-line cycles is based on counting the zero crossing of the voltage channel. The ZXSEL[2:0] bits in the LCYCMODE register determine which voltage channels are used for the zero-crossing detection (see Table 22). The same signal is also used for line cycle energy accumulation mode if activated.

Overvoltage Detection Interrupt

Figure 59 illustrates the behavior of the overvoltage detection.

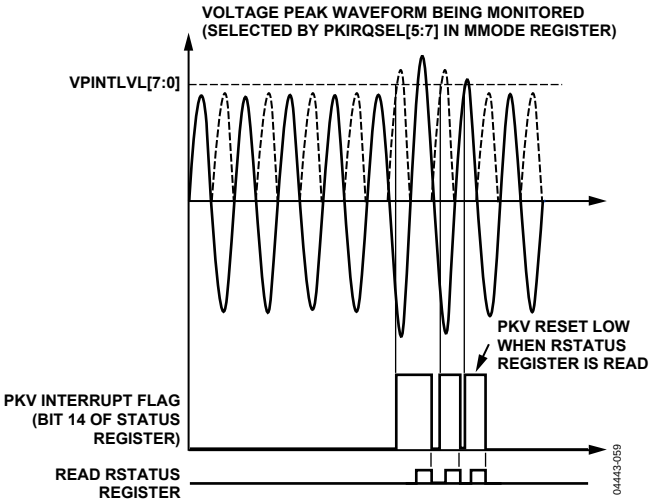


Figure 59. ADE7758 Overvoltage Detection

Note that the content of the VPINTLVL[7:0] register is equivalent to Bit 6 to Bit 13 of the 16-bit voltage waveform samples; therefore, setting this register to 0x9D represents putting the peak detection at full-scale analog input. Figure 59 shows a voltage exceeding a threshold. By setting the PKV flag (Bit 14) in the interrupt status register, the overvoltage event is recorded. If the PKV enable bit is set to Logic 1 in the interrupt mask register, the $\overline{\text{IRQ}}$ logic output goes active low (see the Interrupts section).

Multiple phases can be activated for peak detection. If any of the active phases produce waveform samples above the threshold, the PKV flag in the interrupt status register is set. The phase in which overvoltage is monitored is set by the PKIRQSEL[5:7] bits in the MMODE register (see Table 19).

PHASE SEQUENCE DETECTION

The ADE7758 has an on-chip phase sequence error detection interrupt. This detection works on phase voltages and considers all associated zero crossings. The regular succession of these zero crossings events is a negative to positive transition on Phase A, followed by a positive to negative transition on Phase C, followed by a negative to positive transition on Phase B, and so on.

On the ADE7758, if the regular succession of the zero crossings presented above happens, the SEQERR bit (Bit 19) in the STATUS register is set (Figure 60). If SEQERR is set in the mask register, the $\overline{\text{IRQ}}$ logic output goes active low (see the Interrupts section).

If the regular zero crossing succession does not occur, that is when a negative to positive transition on Phase A followed by a positive to negative transition on Phase B, followed by a negative to positive transition on Phase C, and so on, the SEQERR bit (Bit 19) in the STATUS register is cleared to 0.

To have the ADE7758 trigger SEQERR status bit when the zero crossing regular succession does not occur, the analog inputs for Phase C and Phase B should be swapped. In this case, the Phase B voltage input should be wired to the VCP pin, and the Phase C voltage input should be wired to the VBP pin.

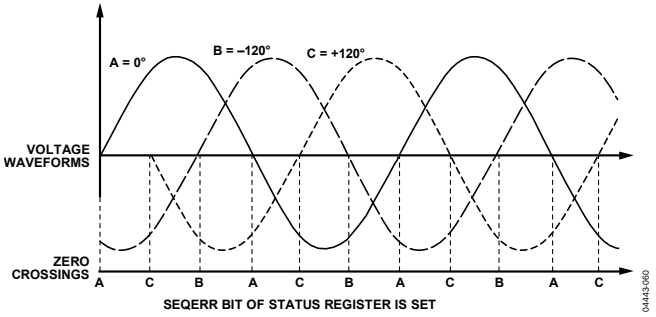


Figure 60. Regular Phase Sequence Sets SEQERR Bit to 1

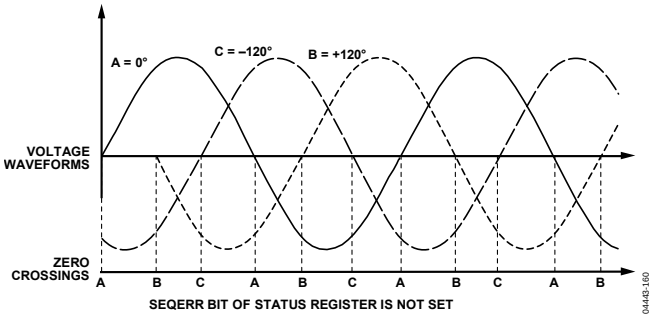


Figure 61. Erroneous Phase Sequence Clears SEQERR Bit to 0

POWER-SUPPLY MONITOR

The ADE7758 also contains an on-chip power-supply monitor. The analog supply (AVDD) is monitored continuously by the ADE7758. If the supply is less than 4 V ± 5%, the ADE7758 goes into an inactive state, that is, no energy is accumulated when the supply voltage is below 4 V. This is useful to ensure correct device operation at power-up and during power-down. The power-supply monitor has built-in hysteresis and filtering. This gives a high degree of immunity to false triggering due to noisy supplies. When AVDD returns above 4 V ± 5%, the ADE7758 waits 18 µs for the voltage to achieve the recommended voltage range, 5 V ± 5% and then becomes ready to function. Figure 62 shows the behavior of the ADE7758 when the voltage of AVDD falls below the power-supply monitor threshold. The power supply and decoupling for the part should be designed such that the ripple at AVDD does not exceed 5 V ± 5% as specified for normal operation.

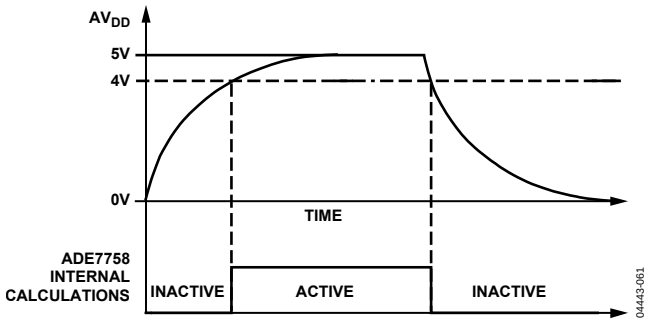


Figure 62. On-Chip, Power-Supply Monitoring

REFERENCE CIRCUIT

The nominal reference voltage at the $REF_{IN/OUT}$ pin is 2.42 V. This is the reference voltage used for the ADCs in the ADE7758. However, the current channels have three input range selections (full scale is selectable among 0.5 V, 0.25 V, and 0.125 V). This is achieved by dividing the reference internally by 1, ½, and ¼. The reference value is used for the ADC in the current channels. Note that the full-scale selection is only available for the current inputs.

The $REF_{IN/OUT}$ pin can be overdriven by an external source, for example, an external 2.5 V reference. Note that the nominal reference value supplied to the ADC is now 2.5 V and not 2.42 V. This has the effect of increasing the nominal analog input signal range by 2.5/2.42 × 100% = 3% or from 0.5 V to 0.5165 V.

The voltage of the ADE7758 reference drifts slightly with temperature; see the Specifications section for the temperature coefficient specification (in ppm/°C). The value of the temperature drift varies from part to part. Because the reference is used for all ADCs, any ×% drift in the reference results in a 2×% deviation of the meter accuracy. The reference drift resulting from temperature changes is usually very small and typically much smaller than the drift of other components on a meter. Alternatively, the meter can be calibrated at multiple temperatures.

TEMPERATURE MEASUREMENT

The ADE7758 also includes an on-chip temperature sensor. A temperature measurement is made every 4/CLKIN seconds. The output from the temperature sensing circuit is connected to an ADC for digitizing. The resultant code is processed and placed in the temperature register (TEMP[7:0]). This register can be read by the user and has an address of 0x11 (see the Serial Interface section). The contents of the temperature register are signed (twos complement) with a resolution of 3°C/LSB. The offset of this register may vary significantly from part to part. To calibrate this register, the nominal value should be measured, and the equation should be adjusted accordingly.

$$Temp\ (°C) = [(TEMP[7:0] - Offset) \times 3°C/LSB] + Ambient(°C) \quad (4)$$

For example, if the temperature register produces a code of 0x46 at ambient temperature (25°C), and the temperature register currently reads 0x50, then the temperature is 55°C :

$$Temp\ (°C) = [(0x50 - 0x46) \times 3°C/LSB] + 25°C = 55°C$$

Depending on the nominal value of the register, some finite temperature can cause the register to roll over. This should be compensated for in the system master (MCU).

The ADE7758 temperature register varies with power supply. It is recommended to use the temperature register only in applications with a fixed, stable power supply. Typical error with respect to power supply variation is show in Table 5.

Table 5. Temperature Register Error with Power Supply Variation

	4.5 V	4.75 V	5 V	5.25 V	5.5 V
Register Value	219	216	214	211	208
% Error	+2.34	+0.93	0	−1.40	−2.80

ROOT MEAN SQUARE MEASUREMENT

Root mean square (rms) is a fundamental measurement of the magnitude of an ac signal. Its definition can be both practical and mathematical. Defined practically, the rms value assigned to an ac signal is the amount of dc required to produce an equivalent amount of power in the load. Mathematically, the rms value of a continuous signal f(t) is defined as

$$FRMS = \sqrt{\frac{1}{T}\int_0^T f^2(t)\mathrm{d}t} \quad (5)$$

For time sampling signals, rms calculation involves squaring the signal, taking the average, and obtaining the square root.

$$FRMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} f^2[n]} \quad (6)$$

The method used to calculate the rms value in the ADE7758 is to low-pass filter the square of the input signal (LPF3) and take the square root of the result (see Figure 63).

$$i(t) = \sqrt{2} \times IRMS \times \sin(\omega t) \quad (7)$$

then

$$i^2(t) = IRMS^2 - IRMS^2 \times \cos(\omega t) \quad (8)$$

The rms calculation is simultaneously processed on the six analog input channels. Each result is available in separate registers.

While the ADE7758 measures nonsinusoidal signals, it should be noted that the voltage rms measurement, and therefore the apparent energy, are bandlimited to 260 Hz. The current rms as well as the active power have a bandwidth of 14 kHz.

Current RMS Calculation

Figure 63 shows the detail of the signal processing chain for the rms calculation on one of the phases of the current channel. The current channel rms value is processed from the samples used in the current channel waveform sampling mode. The current rms values are stored in 24-bit registers (AIRMS, BIRMS, and CIRMS). One LSB of the current rms register is equivalent to one LSB of the current waveform sample. The update rate of the current rms measurement is CLKIN/12.

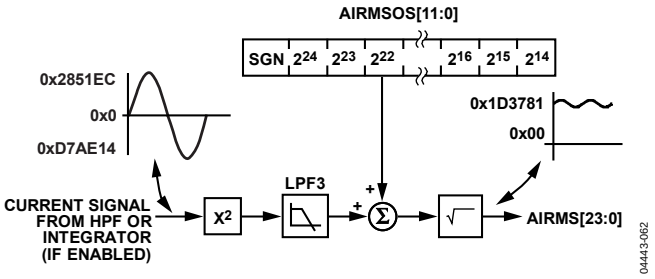


Figure 63. Current RMS Signal Processing

With the specified full-scale analog input signal of 0.5 V, the ADC produces an output code that is approximately ±2,642,412d (see the Current Channel ADC section). The equivalent rms value of a full-scale sinusoidal signal at 60 Hz is 1,914,753 (0x1D3781).

The accuracy of the current rms is typically 0.5% error from the full-scale input down to 1/500 of the full-scale input. Additionally, this measurement has a bandwidth of 14 kHz. It is recommended to read the rms registers synchronous to the voltage zero crossings to ensure stability. The IRQ can be used to indicate when a zero crossing has occurred (see the Interrupts section).

Table 6 shows the settling time for the IRMS measurement, which is the time it takes for the rms register to reflect the value at the input to the current channel.

Table 6. Settling Time for IRMS Measurement

	63%	100%
Integrator Off	80 ms	960 ms
Integrator On	40 ms	1.68 sec

Current RMS Offset Compensation

The ADE7758 incorporates a current rms offset compensation register for each phase (AIRMSOS, BIRMSOS, and CIRMSOS). These are 12-bit signed registers that can be used to remove offsets in the current rms calculations. An offset can exist in the rms calculation due to input noises that are integrated in the dc component of $I^2(t)$. Assuming that the maximum value from the current rms calculation is 1,914,753d with full-scale ac inputs (60 Hz), one LSB of the current rms offset represents 0.94% of the measurement error at 60 dB down from full scale. The IRMS measurement is undefined at zero input. Calibration of the offset should be done at low current and values at zero input should be ignored. For details on how to calibrate the current rms measurement, see the Calibration section.

$$IRMS = \sqrt{IRMS_0^2 + 16384 \times IRMSOS} \quad (9)$$

where $IRMS_0$ is the rms measurement without offset correction.

Table 7. Approximate IRMS Register Values

Frequency (Hz)	Integrator Off (d)	Integrator On (d)
50	1,921,472	2,489,581
60	1,914,752	2,067,210

Voltage Channel RMS Calculation

Figure 64 shows the details of the signal path for the rms estimation on Phase A of the voltage channel. This voltage rms estimation is done in the ADE7758 using the mean absolute value calculation, as shown in Figure 64.The voltage channel rms value is processed from the waveform samples after the low-pass filter LPF1. The output of the voltage channel ADC can be scaled by ±50% by changing VRMSGAIN[11:0] registers to perform an overall rms voltage calibration. The VRMSGAIN registers scale the rms calculations as well as the apparent energy calculation because apparent power is the product of the voltage and current rms values. The voltage rms values are stored in 24-bit registers (AVRMS, BVRMS, and CVRMS). One LSB of a voltage waveform sample is approximately equivalent to 256 LSBs of the voltage rms register. The update rate of the voltage rms measurement is CLKIN/12.

With the specified full-scale ac analog input signal of 0.5 V, the LPF1 produces an output code that is approximately 63% of its full-scale value, that is, ±9,372d, at 60 Hz (see the Voltage Channel ADC section). The equivalent rms value of a full-scale ac signal is approximately 1,639,101 (0x1902BD) in the VRMS register.

The accuracy of the VRMS measurement is typically 0.5% error from the full-scale input down to 1/20 of the full-scale input. Additionally, this measurement has a bandwidth of 260 Hz. It is recommended to read the rms registers synchronous to the voltage zero crossings to ensure stability. The IRQ can be used to indicate when a zero crossing has occurred (see the Interrupts section).

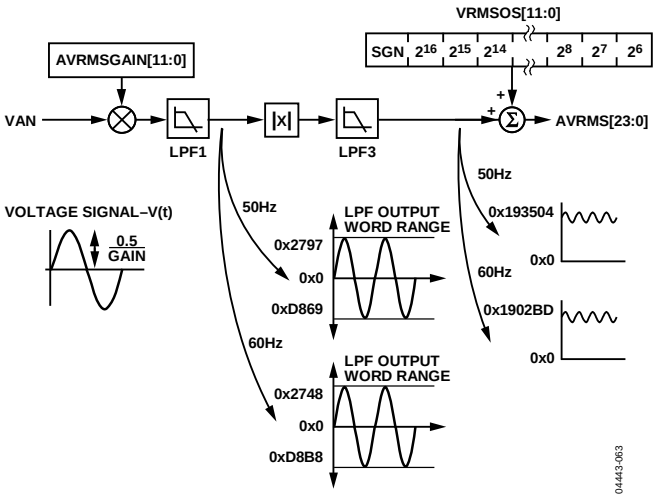


Figure 64. Voltage RMS Signal Processing

Table 8 shows the settling time for the VRMS measurement, which is the time it takes for the rms register to reflect the value at the input to the voltage channel.

Table 8. Settling Time for VRMS Measurement

63%	100%
100 ms	960 ms

Voltage RMS Offset Compensation

The ADE7758 incorporates a voltage rms offset compensation for each phase (AVRMSOS, BVRMSOS, and CVRMSOS). These are 12-bit signed registers that can be used to remove offsets in the voltage rms calculations. An offset can exist in the rms calculation due to input noises and offsets in the input samples. It should be noted that the offset calibration does not allow the contents of the VRMS registers to be maintained at 0 when no voltage is applied. This is caused by noise in the voltage rms calculation, which limits the usable range between full scale and 1/50th of full scale. One LSB of the voltage rms offset is equivalent to 64 LSBs of the voltage rms register.

Assuming that the maximum value from the voltage rms calculation is 1,639,101d with full-scale ac inputs, then 1 LSB of the voltage rms offset represents 0.042% of the measurement error at 1/10 of full scale.

$$VRMS = VRMS_0 + VRMSOS \times 64 \quad (10)$$

where $VRMS_0$ is the rms measurement without the offset correction.

Table 9. Approximate VRMS Register Values

Frequency (Hz)	Value (d)
50	1,678,210
60	1,665,118

Voltage RMS Gain Adjust

The ADC gain in each phase of the voltage channel can be adjusted for the rms calculation by using the voltage rms gain registers (AVRMSGAIN, BVRMSGAIN, and CVRMSGAIN). The gain of the voltage waveforms before LPF1 is adjusted by writing twos complement, 12-bit words to the voltage rms gain registers. Equation 11 shows how the gain adjustment is related to the contents of the voltage gain register.

$$Content\ of\ VRMS\ Register = Nominal\ RMS\ Values\ Without\ Gain \times \left(1 + \frac{VRMSGAIN}{2^{12}}\right) \quad (11)$$

For example, when 0x7FF is written to the voltage gain register, the RMS value is scaled up by 50%.

0x7FF = 2047d
$2047/2^{12} = 0.5$

Similarly, when 0x800, which equals –2047d (signed twos complement), is written the ADC output is scaled by –50%.

ACTIVE POWER CALCULATION

Electrical power is defined as the rate of energy flow from source to load. It is given by the product of the voltage and current waveforms. The resulting waveform is called the instantaneous power signal and it is equal to the rate of energy flow at every instant of time. The unit of power is the watt or joules/sec. Equation 14 gives an expression for the instantaneous power signal in an ac system.

$$v(t) = \sqrt{2} \times VRMS \times \sin(\omega t) \quad (12)$$

$$i(t) = \sqrt{2} \times IRMS \times \sin(\omega t) \quad (13)$$

where *VRMS* = rms voltage and *IRMS* = rms current.

$$p(t) = v(t) \times i(t)$$

$$p(t) = IRMS \times VRMS - IRMS \times VRMS \times cos(2\omega t) \quad (14)$$

The average power over an integral number of line cycles (n) is given by the expression in Equation 15.

$$p = \frac{1}{nT}\int_0^{nT} p(t)dt = VRMS \times IRMS \quad (15)$$

where:

t is the line cycle period.
P is referred to as the active or real power. Note that the active power is equal to the dc component of the instantaneous power signal *p*(*t*) in Equation 14, that is, *VRMS* × *IRMS*. This is the relationship used to calculate the active power in the ADE7758 for each phase.

The instantaneous power signal *p*(*t*) is generated by multiplying the current and voltage signals in each phase. The dc component of the instantaneous power signal in each phase (A, B, and C) is then extracted by LPF2 (the low-pass filter) to obtain the average active power information on each phase. Figure 65 shows this process. The active power of each phase accumulates in the corresponding 16-bit watt-hour register (AWATTHR, BWATTHR, or CWATTHR). The input to each active energy register can be changed depending on the accumulation mode setting (see Table 22).

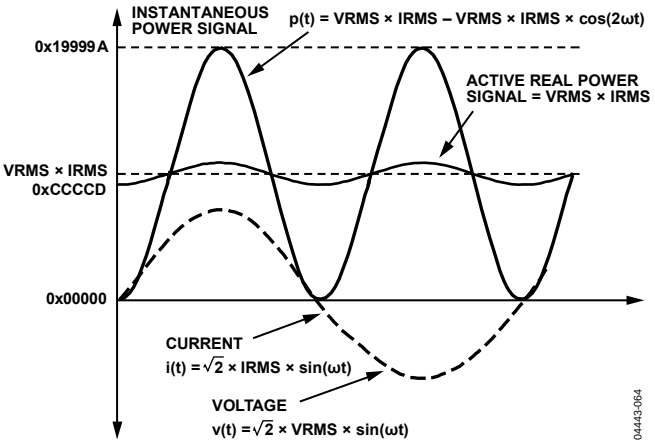


Figure 65. Active Power Calculation

Because LPF2 does not have an ideal brick wall frequency response (see Figure 66), the active power signal has some ripple due to the instantaneous power signal. This ripple is sinusoidal and has a frequency equal to twice the line frequency. Because the ripple is sinusoidal in nature, it is removed when the active power signal is integrated over time to calculate the energy.

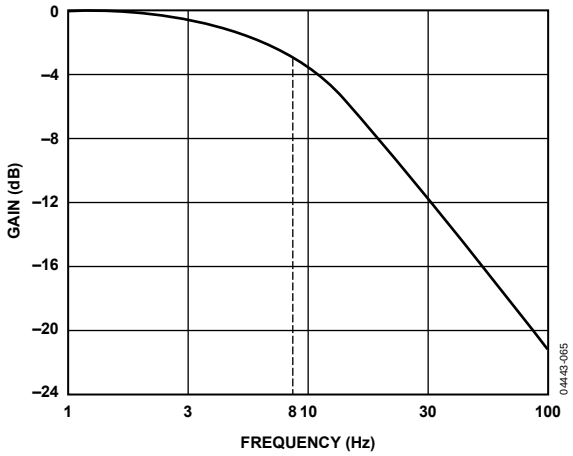


Figure 66. Frequency Response of the LPF Used to Filter Instantaneous Power in Each Phase

Active Power Gain Calibration

Note that the average active power result from the LPF output in each phase can be scaled by ±50% by writing to the phase's watt gain register (AWG, BWG, or CWG). The watt gain registers are twos complement, signed registers and have a resolution of 0.024%/LSB. Equation 16 describes mathematically the function of the watt gain registers.

$$Average\ Power\ Data = LPF2\ Output \times \left(1 + \frac{Watt\ Gain\ Register}{2^{12}}\right) \quad (16)$$

The output is scaled by −50% by writing 0x800 to the watt gain registers and increased by +50% by writing 0x7FF to them. These registers can be used to calibrate the active power (or energy) calculation in the ADE7758 for each phase.

Active Power Offset Calibration

The ADE7758 also incorporates a watt offset register on each phase (AWATTOS, BWATTOS, and CWATTOS). These are signed twos complement, 12-bit registers that are used to remove offsets in the active power calculations. An offset can exist in the power calculation due to crosstalk between channels on the PCB or in the chip itself. The offset calibration allows the contents of the active power register to be maintained at 0 when no power is being consumed. One LSB in the active power offset register is equivalent to 1/16 LSB in the active power multiplier output. At full-scale input, if the output from the multiplier is 0xCCCCD (838,861d), then 1 LSB in the LPF2 output is equivalent to 0.0075% of measurement error at 60 dB down from full scale on the current channel. At −60 dB down on full scale (the input signal level is 1/1000 of full-scale signal inputs), the average word value from LPF2 is 838.861 (838,861/1000). One LSB is equivalent to 1/838.861/16 × 100% = 0.0075% of the measured value. The active power offset register has a correction resolution equal to 0.0075% at −60 dB.

Sign of Active Power Calculation

Note that the average active power is a signed calculation. If the phase difference between the current and voltage waveform is more than 90°, the average power becomes negative. Negative power indicates that energy is being placed back on the grid. The ADE7758 has a sign detection circuitry for active power calculation.

The REVPAP bit (Bit 17) in the interrupt status register is set if the average power from any one of the phases changes sign. The phases monitored are selected by TERMSEL bits in the COMPMODE register (see Table 21). The TERMSEL bits are also used to select which phases are included in the APCF and VARCF pulse outputs. If the REVPAP bit is set in the mask register, the $\overline{IRQ}$ logic output goes active low (see the Interrupts section). Note that this bit is set whenever there are sign changes, that is, the REVPAP bit is set for both a positive-to-negative change and a negative-to-positive change of the sign bit. The response time of this bit is approximately 176 ms for a full-scale signal, which has an average value of 0xCCCCD at the low pass filter output. For smaller inputs, the time is longer.

$$Response\ \ Time \cong 160\ ms + \left[\frac{2^{25}}{Average\ Value}\right] \times \frac{4}{CLKIN} \quad (17)$$

The APCFNUM [15:13] indicate reverse power on each of the individual phases. Bit 15 is set if the sign of the power on Phase A is negative, Bit 14 for Phase B, and Bit 13 for Phase C.

No-Load Threshold

The ADE7758 has an internal no-load threshold on each phase. The no-load threshold can be activated by setting the NOLOAD bit (Bit 7) of the COMPMODE register. If the active power falls below 0.005% of full-scale input, the energy is not accumulated in that phase. As stated, the average multiplier output with full-scale input is 0xCCCCD. Therefore, if the average multiplier output falls below 0x2A, the power is not accumulated to avoid creep in the meter. The no-load threshold is implemented only on the active energy accumulation. The reactive and apparent energies do not have the no-load threshold option.

Active Energy Calculation

As previously stated, power is defined as the rate of energy flow. This relationship can be expressed mathematically as

$$Power = \frac{dEnergy}{dt} \quad (18)$$

Conversely, *Energy* is given as the integral of power.

$$Energy = \int p(t)dt \quad (19)$$

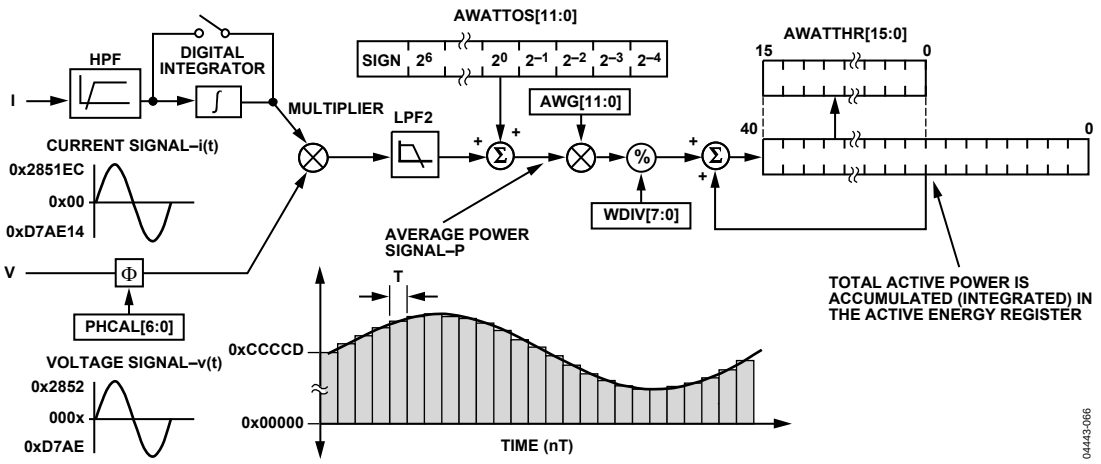


Figure 67. ADE7758 Active Energy Accumulation

The ADE7758 achieves the integration of the active power signal by continuously accumulating the active power signal in the internal 41-bit energy registers. The watt-hr registers (AWATTHR, BWATTHR, and CWATTHR) represent the upper 16 bits of these internal registers. This discrete time accumulation or summation is equivalent to integration in continuous time. Equation 20 expresses the relationship.

$$Energy = \int p(t)dt = \lim_{T \to 0} \left\{ \sum_{n=0}^{\infty} p(nT) \times T \right\} \quad (20)$$

where:

n is the discrete time sample number.
T is the sample period.

Figure 67 shows a signal path of this energy accumulation. The average active power signal is continuously added to the internal active energy register. This addition is a signed operation. Negative energy is subtracted from the active energy register. Note the values shown in Figure 67 are the nominal full-scale values, that is, the voltage and current inputs at the corresponding phase are at their full-scale input level. The average active power is divided by the content of the watt divider register before it is added to the corresponding watt-hr accumulation registers. When the value in the WDIV[7:0] register is 0 or 1, active power is accumulated without division. WDIV is an 8-bit unsigned register that is useful to lengthen the time it takes before the watt-hr accumulation registers overflow.

Figure 68 shows the energy accumulation for full-scale signals (sinusoidal) on the analog inputs. The three displayed curves show the minimum time it takes for the watt-hr accumulation register to overflow when the watt gain register of the corresponding phase equals to 0x7FF, 0x000, and 0x800. The watt gain registers are used to carry out a power calibration in the ADE7758. As shown, the fastest integration time occurs when the watt gain registers are set to maximum full scale, that is, 0x7FF. This is the time it takes before overflow can be scaled by writing to the WDIV register and therefore can be increased by a maximum factor of 255.

Note that the active energy register content can roll over to full-scale negative (0x8000) and continue increasing in value when the active power is positive (see Figure 67). Conversely, if the active power is negative, the energy register would under flow to full-scale positive (0x7FFF) and continue decreasing in value.

By setting the AEHF bit (Bit 0) of the interrupt mask register, the ADE7758 can be configured to issue an interrupt ($\overline{IRQ}$) when Bit 14 of any one of the three watt-hr accumulation registers has changed, indicating that the accumulation register is half full (positive or negative).

Setting the RSTREAD bit (Bit 6) of the LCYMODE register enables a read-with-reset for the watt-hr accumulation registers, that is, the registers are reset to 0 after a read operation.

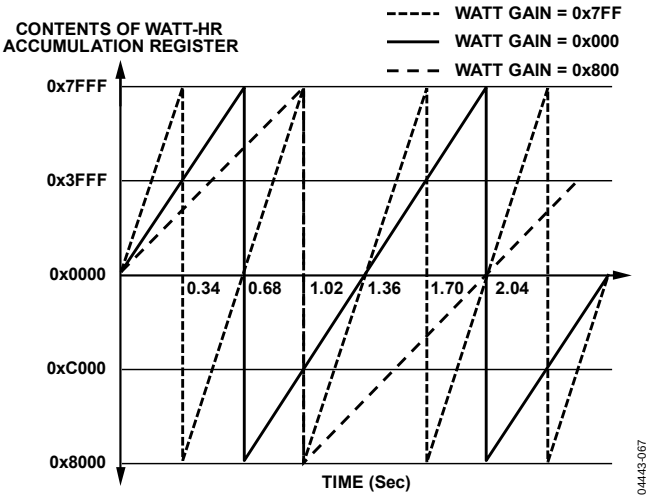


Figure 68. Energy Register Roll-Over Time for Full-Scale Power (Minimum and Maximum Power Gain)

Integration Time Under Steady Load

The discrete time sample period (T) for the accumulation register is 0.4 µs (4/CLKIN). With full-scale sinusoidal signals on the analog inputs and the watt gain registers set to 0x000, the average word value from each LPF2 is 0xCCCCD (see Figure 65 and Figure 67). The maximum value that can be stored in the watt-hr accumulation register before it overflows is $2^{15} - 1$ or 0x7FFF. Because the average word value is added to the internal register, which can store $2^{40} - 1$ or 0xFF, FFFF, FFFF before it overflows, the integration time under these conditions with WDIV = 0 is calculated as

$$Time = \frac{0xFF, FFFF, FFFF}{0xCCCCD} \times 0.4\ \mu s = 0.524\ sec \quad (21)$$

When *WDIV* is set to a value different from 0, the time before overflow is scaled accordingly as shown in Equation 22.

$$Time = Time\ (WDIV = 0) \times WDIV[7:0] \quad (22)$$

Energy Accumulation Mode

The active power accumulated in each watt-hr accumulation register (AWATTHR, BWATTHR, or CWATTHR) depends on the configuration of the CONSEL bits in the COMPMODE register (Bit 0 and Bit 1). The different configurations are described in Table 10.

Table 10. Inputs to Watt-Hr Accumulation Registers

CONSEL[1, 0]	AWATTHR	BWATTHR	CWATTHR
00	VA × IA	VB × IB	VC × IC
01	VA × (IA – IB)	0	VC × (IC – IB)
10	VA × (IA – IB)	0	VC × IC
11	Reserved	Reserved	Reserved

Depending on the poly phase meter service, the appropriate formula should be chosen to calculate the active energy. The American ANSI C12.10 Standard defines the different configurations of the meter.

Table 11 describes which mode should be chosen in these different configurations.

Table 11. Meter Form Configuration

ANSI Meter Form		CONSEL (d)	TERMSEL (d)
5S/13S	3-Wire Delta	0	3, 5, or 6
6S/14S	4-Wire Wye	1	7
8S/15S	4-Wire Delta	2	7
9S/16S	4-Wire Wye	0	7

Active Power Frequency Output

Pin 1 (APCF) of the ADE7758 provides frequency output for the total active power. After initial calibration during manufacturing, the manufacturer or end customer often verifies the energy meter calibration. One convenient way to verify the meter calibration is for the manufacturer to provide an output frequency that is proportional to the energy or active power under steady load conditions. This output frequency can provide a simple, single-wire, optically isolated interface to external calibration equipment. Figure 69 illustrates the energy-to-frequency conversion in the ADE7758.

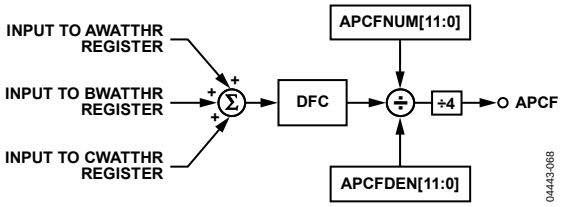


Figure 69. Active Power Frequency Output

A digital-to-frequency converter (DFC) is used to generate the APCF pulse output from the total active power. The TERMSEL bits (Bit 2 to Bit 4) of the COMPMODE register can be used to select which phases to include in the total power calculation. Setting Bit 2, Bit 3, and Bit 4 includes the input to the AWATTHR, BWATTHR, and CWATTHR registers in the total active power calculation. The total active power is signed addition. However, setting the ABS bit (Bit 5) in the COMPMODE register enables the absolute-only mode; that is, only the absolute value of the active power is considered.

The output from the DFC is divided down by a pair of frequency division registers before being sent to the APCF pulse output. Namely, APCFDEN/APCFNUM pulses are needed at the DFC output before the APCF pin outputs a pulse. Under steady load conditions, the output frequency is directly proportional to the total active power. The pulse width of APCF is 64/CLKIN if APCFNUM and APCFDEN are both equal. If APCFDEN is greater than APCFNUM, the pulse width depends on APCFDEN. The pulse width in this case is $T \times (APCFDEN/2)$, where T is the period of the APCF pulse and $APCFDEN/2$ is rounded to the nearest whole number. An exception to this is when the period is greater than 180 ms. In this case, the pulse width is fixed at 90 ms.

The maximum output frequency (APCFNUM = 0x00 and APCFDEN = 0x00) with full-scale ac signals on one phase is approximately 16 kHz.

The ADE7758 incorporates two registers to set the frequency of APCF (APCFNUM[11:0] and APCFDEN[11:0]). These are unsigned 12-bit registers that can be used to adjust the frequency of APCF by $1/2^{12}$ to 1 with a step of $1/2^{12}$. For example, if the output frequency is 1.562 kHz while the contents of APCFDEN are 0 (0x000), then the output frequency can be set to 6.103 Hz by writing 0xFF to the APCFDEN register.

If 0 were written to any of the frequency division registers, the divider would use 1 in the frequency division. In addition, the ratio APCFNUM/APCFDEN should be set not greater than 1 to ensure proper operation. In other words, the APCF output frequency cannot be higher than the frequency on the DFC output.

The output frequency has a slight ripple at a frequency equal to 2× the line frequency. This is due to imperfect filtering of the instantaneous power signal to generate the active power signal

(see the Active Power Calculation section). Equation 14 gives an expression for the instantaneous power signal. This is filtered by LPF2, which has a magnitude response given by Equation 23.

$$H(f) = \frac{1}{\sqrt{1 + \frac{f^2}{8^2}}} \quad (23)$$

The active power signal (output of the LPF2) can be rewritten as

$$p(t) = VRMS \times IRMS - \left[\frac{VRMS \times IRMS}{\sqrt{1 + \frac{(2f_1)^2}{8^2}}} \right] \times \cos(4\pi f_1 t) \quad (24)$$

where f_1 is the line frequency, for example, 60 Hz.

From Equation 24, E(t) equals

$$VRMS \times IRMS \times t - \left[\frac{VRMS \times IRMS}{4\pi f_1 \sqrt{1 + \frac{(2f_1)^2}{8^2}}} \right] \times \cos(4\pi f_1 t) \quad (25)$$

From Equation 25, it can be seen that there is a small ripple in the energy calculation due to the sin(2ωt) component (see Figure 70). The ripple gets larger with larger loads. Choosing a lower output frequency for APCF during calibration by using a large APCFDEN value and keeping APCFNUM relatively small can significantly reduce the ripple. Averaging the output frequency over a longer period achieves the same results.

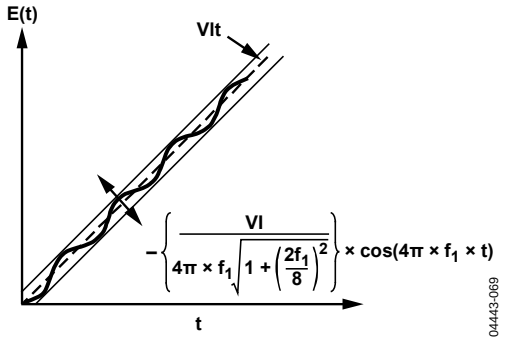


Figure 70. Output Frequency Ripple

Line Cycle Active Energy Accumulation Mode

The ADE7758 is designed with a special energy accumulation mode that simplifies the calibration process. By using the on-chip, zero-crossing detection, the ADE7758 updates the watt-hr accumulation registers after an integer number of zero crossings (see Figure 71). The line-active energy accumulation mode for watt-hr accumulation is activated by setting the LWATT bit (Bit 0) of the LCYCMODE register. The total energy accumulated over an integer number of half-line cycles is written to the watt-hr accumulation registers after the LINECYC number of zero crossings is detected. When using the line cycle accumulation mode, the RSTREAD bit (Bit 6) of the LCYCMODE register should be set to Logic 0.

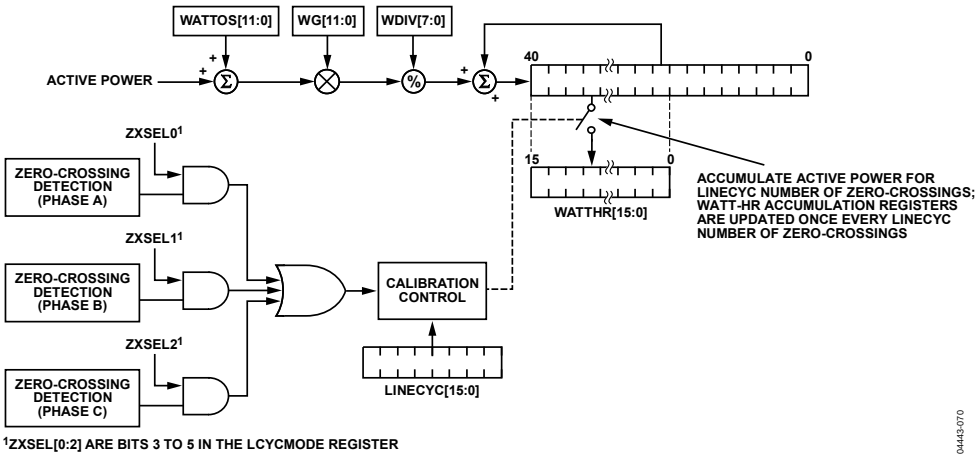


Figure 71. ADE7758 Line Cycle Active Energy Accumulation Mode

Phase A, Phase B, and Phase C zero crossings are, respectively, included when counting the number of half-line cycles by setting ZXSEL[0:2] bits (Bit 3 to Bit 5) in the LCYCMODE register. Any combination of the zero crossings from all three phases can be used for counting the zero crossing. Only one phase should be selected at a time for inclusion in the zero crossings count during calibration (see the Calibration section).

The number of zero crossings is specified by the LINECYC register. LINECYC is an unsigned 16-bit register. The ADE7758 can accumulate active power for up to 65535 combined zero crossings. Note that the internal zero-crossing counter is always active. By setting the LWATT bit, the first energy accumulation result is, therefore, incorrect. Writing to the LINECYC register when the LWATT bit is set resets the zero-crossing counter, thus ensuring that the first energy accumulation result is accurate.

At the end of an energy calibration cycle, the LENERGY bit (Bit 12) in the STATUS register is set. If the corresponding mask bit in the interrupt mask register is enabled, the $\overline{\text{IRQ}}$ output also goes active low; thus, the $\overline{\text{IRQ}}$ can also be used to signal the end of a calibration.

Because active power is integrated on an integer number of half-line cycles in this mode, the sinusoidal component is reduced to 0, eliminating any ripple in the energy calculation. Therefore, total energy accumulated using the line-cycle accumulation mode is

$$E(t) = VRMS \times IRMS \times t \quad (26)$$

where t is the accumulation time.

Note that line cycle active energy accumulation uses the same signal path as the active energy accumulation. The LSB size of these two methods is equivalent. Using the line cycle accumulation to calculate the kWh/LSB constant results in a value that can be applied to the WATTHR registers when the line accumulation mode is not selected (see the Calibration section).

REACTIVE POWER CALCULATION

A load that contains a reactive element (inductor or capacitor) produces a phase difference between the applied ac voltage and the resulting current. The power associated with reactive elements is called reactive power, and its unit is VAR. Reactive power is defined as the product of the voltage and current waveforms when one of these signals is phase shifted by 90°.

Equation 30 gives an expression for the instantaneous reactive power signal in an ac system when the phase of the current channel is shifted by +90°.

$$v(t) = \sqrt{2}\,V \sin(\omega t - \theta) \quad (27)$$

$$i(t) = \sqrt{2}\,I \sin(\omega t)$$

$$i'(t) = \sqrt{2}\,I \sin\left(\omega t + \frac{\pi}{2}\right) \quad (28)$$

where:

v = rms voltage.
i = rms current.
θ = total phase shift caused by the reactive elements in the load.

Then the instantaneous reactive power q(t) can be expressed as

$$q(t) = v(t) \times i'(t)$$

$$q(t) = VI \cos\left(-\theta - \frac{\pi}{2}\right) - VI \cos\left(2\omega t - \theta - \frac{\pi}{2}\right) \quad (29)$$

where $i'(t)$ is the current waveform phase shifted by 90°. Note that $q(t)$ can be rewritten as

$$q(t) = VI \sin(\theta) + VI \sin(2\omega t - \theta) \quad (30)$$

The average reactive power over an integral number of line cycles (n) is given by the expression in Equation 31.

$$Q = \frac{1}{nT} \int_0^{nT} q(t)dt = V \times I \times \sin(\theta) \quad (31)$$

where:

T is the period of the line cycle.
Q is referred to as the average reactive power. The instantaneous reactive power signal $q(t)$ is generated by multiplying the voltage signals and the 90° phase-shifted current in each phase.

The dc component of the instantaneous reactive power signal in each phase (A, B, and C) is then extracted by a low-pass filter to obtain the average reactive power information on each phase. This process is illustrated in Figure 72. The reactive power of each phase is accumulated in the corresponding 16-bit VAR-hour register (AVARHR, BVARHR, or CVARHR). The input to each reactive energy register can be changed depending on the accumulation mode setting (see Table 21).

The frequency response of the LPF in the reactive power signal path is identical to that of the LPF2 used in the average active power calculation (see Figure 66).

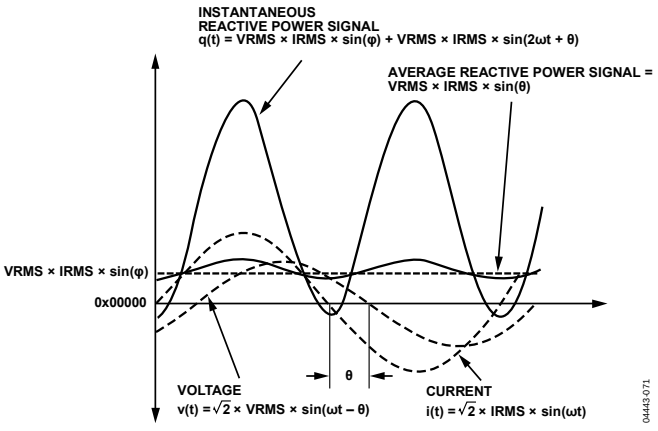


Figure 72. Reactive Power Calculation

The low-pass filter is nonideal, so the reactive power signal has some ripple. This ripple is sinusoidal and has a frequency equal to 2× the line frequency. Because the ripple is sinusoidal in nature, it is removed when the reactive power signal is integrated over time to calculate the reactive energy.

The phase-shift filter has –90° phase shift when the integrator is enabled and +90° phase shift when the integrator is disabled. In addition, the filter has a nonunity magnitude response. Because the phase-shift filter has a large attenuation at high frequency, the reactive power is primarily for the calculation at line frequency. The effect of harmonics is largely ignored in the reactive power calculation. Note that because of the magnitude characteristic of the phase shifting filter, the LSB weight of the reactive power calculation is slightly different from that of the active power calculation (see the Energy Registers Scaling section). The ADE7758 uses the line frequency of the phase selected in the FREQSEL[1:0] bits of the MMODE[1:0] to compensate for attenuation of the reactive energy phase shift filter over frequency (see the Period Measurement section).

Reactive Power Gain Calibration

The average reactive power from the LPF output in each phase can be scaled by ±50% by writing to the phase's VAR gain register (AVARG, BVARG, or CVARG). The VAR gain registers are twos complement, signed registers and have a resolution of 0.024%/LSB. The function of the VAR gain registers is expressed by

$$Average\ Reactive\ Power = LPF2\,Output \times \left(1 + \frac{VAR\ Gain\ Register}{2^{12}}\right) \quad (32)$$

The output is scaled by –50% by writing 0x800 to the VAR gain registers and increased by +50% by writing 0x7FF to them. These registers can be used to calibrate the reactive power (or energy) calculation in the ADE7758 for each phase.

Reactive Power Offset Calibration

The ADE7758 incorporates a VAR offset register on each phase (AVAROS, BVAROS, and CVAROS). These are signed twos complement, 12-bit registers that are used to remove offsets in the reactive power calculations. An offset can exist in the power calculation due to crosstalk between channels on the PCB or in the chip itself. The offset calibration allows the contents of the reactive power register to be maintained at 0 when no reactive power is being consumed. The offset registers' resolution is the same as the active power offset registers (see the Apparent Power Offset Calibration section).

Sign of Reactive Power Calculation

Note that the average reactive power is a signed calculation. As stated previously, the phase shift filter has –90° phase shift when the integrator is enabled and +90° phase shift when the integrator is disabled.

Table 12 summarizes the relationship between the phase difference between the voltage and the current and the sign of the resulting VAR calculation.

The ADE7758 has a sign detection circuit for the reactive power calculation. The REVPRP bit (Bit 18) in the interrupt status register is set if the average reactive power from any one of the phases changes. The phases monitored are selected by TERMSEL bits in the COMPMODE register (see Table 21). If the REVPRP bit is set in the mask register, the $\overline{IRQ}$ logic output goes active low (see the Interrupts section). Note that this bit is set whenever there is a sign change; that is, the bit is set for either a positive-to-negative change or a negative-to-positive change of the sign bit. The response time of this bit is approximately 176 ms for a full-scale signal, which has an average value of 0xCCCCD at the low-pass filter output. For smaller inputs, the time is longer.

$$ResponseTime \cong 160\ ms + \left[\frac{2^{25}}{AverageValue}\right] \times \frac{4}{CLKIN} \quad (33)$$

Table 12. Sign of Reactive Power Calculation

Φ[1]	Integrator	Sign of Reactive Power
Between 0 to +90	Off	Positive
Between –90 to 0	Off	Negative
Between 0 to +90	On	Positive
Between –90 to 0	On	Negative

[1] Φ is defined as the phase angle of the voltage signal minus the current signal; that is, Φ is positive if the load is inductive and negative if the load is capacitive.

Reactive Energy Calculation

Reactive energy is defined as the integral of reactive power.

$$Reactive\ Energy = \int q(t)dt \quad (34)$$

Similar to active power, the ADE7758 achieves the integration of the reactive power signal by continuously accumulating the reactive power signal in the internal 41-bit accumulation registers. The VAR-hr registers (AVARHR, BVARHR, and CVARHR) represent the upper 16 bits of these internal registers. This discrete time accumulation or summation is equivalent to integration in continuous time. Equation 35 expresses the relationship

$$Reactive\ Energy = \int q(t)\mathrm{d}t = \lim_{T \to 0}\left\{\sum_{n=0}^{\infty} q(nT) \times T\right\} \quad (35)$$

where:

n is the discrete time sample number.
T is the sample period.

Figure 73 shows the signal path of the reactive energy accumulation. The average reactive power signal is continuously added to the internal reactive energy register. This addition is a signed operation. Negative energy is subtracted from the reactive energy register. The average reactive power is divided by the content of the VAR divider register before it is added to the corresponding VAR-hr accumulation registers. When the value in the VARDIV[7:0] register is 0 or 1, the reactive power is accumulated without any division.

VARDIV is an 8-bit unsigned register that is useful to lengthen the time it takes before the VAR-hr accumulation registers overflow.

Similar to reactive power, the fastest integration time occurs when the VAR gain registers are set to maximum full scale, that is, 0x7FF. The time it takes before overflow can be scaled by writing to the VARDIV register; and, therefore, it can be increased by a maximum factor of 255.

When overflow occurs, the VAR-hr accumulation registers content can rollover to full-scale negative (0x8000) and continue increasing in value when the reactive power is positive. Conversely, if the reactive power is negative, the VAR-hr accumulation registers content can roll over to full-scale positive (0x7FFF) and continue decreasing in value.

By setting the REHF bit (Bit 1) of the interrupt mask register, the ADE7758 can be configured to issue an interrupt ($\overline{\text{IRQ}}$) when Bit 14 of any one of the three VAR-hr accumulation registers has changed, indicating that the accumulation register is half full (positive or negative).

Setting the RSTREAD bit (Bit 6) of the LCYMODE register enables a read-with-reset for the VAR-hr accumulation registers; that is, the registers are reset to 0 after a read operation.

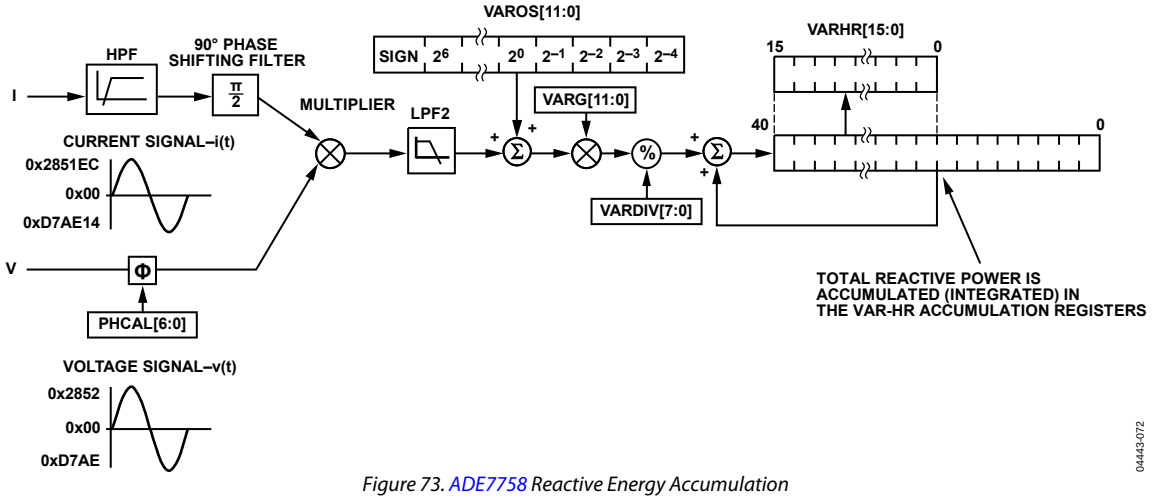


Figure 73. ADE7758 Reactive Energy Accumulation

Integration Time Under Steady Load

The discrete time sample period (T) for the accumulation register is 0.4 µs (4/CLKIN). With full-scale sinusoidal signals on the analog inputs, a 90° phase difference between the voltage and the current signal (the largest possible reactive power), and the VAR gain registers set to 0x000, the average word value from each LPF2 is 0xCCCCD.

The maximum value that can be stored in the reactive energy register before it overflows is $2^{15} - 1$ or 0x7FFF. Because the average word value is added to the internal register, which can store $2^{40} - 1$ or 0xFF, FFFF, FFFF before it overflows, the integration time under these conditions with VARDIV = 0 is calculated as

$$Time = \frac{0xFF,FFFF,FFFF}{0xCCCCD} \times 0.4\,\mu s = 0.5243\,sec \quad (36)$$

When VARDIV is set to a value different from 0, the time before overflow are scaled accordingly as shown in Equation 37.

$$Time = Time(VARDIV = 0) \times VARDIV \quad (37)$$

Energy Accumulation Mode

The reactive power accumulated in each VAR-hr accumulation register (AVARHR, BVARHR, or CVARHR) depends on the configuration of the CONSEL bits in the COMPMODE register (Bit 0 and Bit 1). The different configurations are described in Table 13. Note that IA′/IB′/IC′ are the current phase-shifted current waveform.

Table 13. Inputs to VAR-Hr Accumulation Registers

CONSEL[1, 0]	AVARHR	BVARHR	CVARHR
00	VA × IA′	VB × IB	VC × IC′
01	VA (IA′ – IB′)	0	VC (IC′ – IB′)
10	VA (IA′ – IB′)	0	VC × IC′
11	Reserved	Reserved	Reserved

Reactive Power Frequency Output

Pin 17 (VARCF) of the ADE7758 provides frequency output for the total reactive power. Similar to APCF, this pin provides an output frequency that is directly proportional to the total reactive power. The pulse width of VARPCF is 64/*CLKIN* if VARCFNUM and VARCFDEN are both equal. If VARCFDEN is greater than VARCFNUM, the pulse width depends on VARCFDEN. The pulse width in this case is $T \times (VARCFDEN/2)$, where T is the period of the VARCF pulse and $VARCFDEN/2$ is rounded to the nearest whole number. An exception to this is when the period is greater than 180 ms. In this case, the pulse width is fixed at 90 ms.

A digital-to-frequency converter (DFC) is used to generate the VARCF pulse output from the total reactive power. The TERMSEL bits (Bit 2 to Bit 4) of the COMPMODE register can be used to select which phases to include in the total reactive power calculation. Setting Bit 2, Bit 3, and Bit 4 includes the input to the AVARHR, BVARHR, and CVARHR registers in the total reactive power calculation. The total reactive power is signed addition. However, setting the SAVAR bit (Bit 6) in the COMPMODE register enables absolute value calculation. If the active power of that phase is positive, no change is made to the sign of the reactive power. However, if the sign of the active power is negative in that phase, the sign of its reactive power is inverted before summing and creating VARCF pulses. This mode should be used in conjunction with the absolute value mode for active power (Bit 5 in the COMPMODE register) for APCF pulses.

The effects of setting the ABS and SAVAR bits of the COMPMODE register are as follows when ABS = 1 and SAVAR = 1:

If watt > 0, APCF = Watts, VARCF = +VAR.

If watt < 0, APCF = |Watts|, VARCF = −VAR.

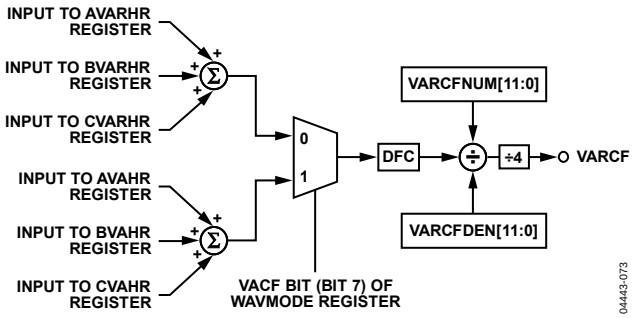


Figure 74. Reactive Power Frequency Output

The output from the DFC is divided down by a pair of frequency division registers before sending to the VARCF pulse output. Namely, VARCFDEN/VARCFNUM pulses are needed at the DFC output before the VARCF pin outputs a pulse. Under steady load conditions, the output frequency is directly proportional to the total reactive power.

Figure 74 illustrates the energy-to-frequency conversion in the ADE7758. Note that the input to the DFC can be selected between the total reactive power and total apparent power. Therefore, the VARCF pin can output frequency that is proportional to the total reactive power or total apparent power. The selection is made by setting the VACF bit (Bit 7) in the WAVMODE register. Setting this bit switches the input to the total apparent power. The default value of this bit is logic low. Therefore, the default output from the VARCF pin is the total reactive power.

All other operations of this frequency output are similar to that of the active power frequency output (see the Active Power Frequency Output section).

Line Cycle Reactive Energy Accumulation Mode

The line cycle reactive energy accumulation mode is activated by setting the LVAR bit (Bit 1) in the LCYCMODE register. The total reactive energy accumulated over an integer number of zero crossings is written to the VAR-hr accumulation registers after the LINECYC number of zero crossings is detected. The operation of this mode is similar to watt-hr accumulation (see the Line Cycle Active Energy Accumulation Mode section).

When using the line cycle accumulation mode, the RSTREAD bit (Bit 6) of the LCYCMODE register should be set to Logic 0.

APPARENT POWER CALCULATION

Apparent power is defined as the amplitude of the vector sum of the active and reactive powers. Figure 75 shows what is typically referred to as the power triangle.

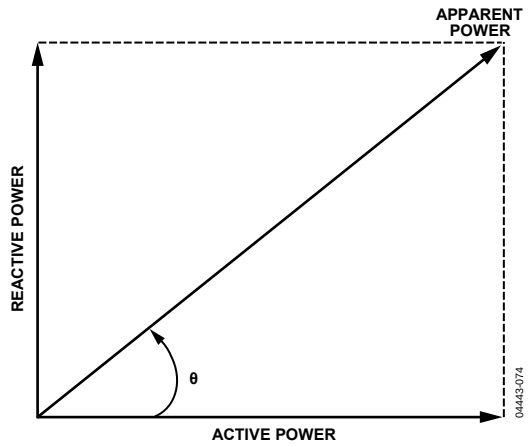


Figure 75. Power Triangle

There are two ways to calculate apparent power: the arithmetical approach or the vectorial method. The arithmetical approach uses the product of the voltage rms value and current rms value to calculate apparent power. Equation 38 describes the arithmetical approach mathematically.

$$S = VRMS \times IRMS \quad (38)$$

where S is the apparent power, and $VRMS$ and $IRMS$ are the rms voltage and current, respectively.

The vectorial method uses the square root of the sum of the active and reactive power, after the two are individually squared. Equation 39 shows the calculation used in the vectorial approach.

$$S = \sqrt{P^2 + Q^2} \quad (39)$$

where:

S is the apparent power.
P is the active power.
Q is the reactive power.

For a pure sinusoidal system, the two approaches should yield the same result. The apparent energy calculation in the ADE7758 uses the arithmetical approach. However, the line cycle energy accumulation mode in the ADE7758 enables energy accumulation between active and reactive energies over a synchronous period, thus the vectorial method can be easily implemented in the external MCU (see the Line Cycle Active Energy Accumulation Mode section).

Note that apparent power is always positive regardless of the direction of the active or reactive energy flows. The rms value of the current and voltage in each phase is multiplied to produce the apparent power of the corresponding phase.

The output from the multiplier is then low-pass filtered to obtain the average apparent power. The frequency response of the LPF in the apparent power signal path is identical to that of the LPF2 used in the average active power calculation (see Figure 66).

Apparent Power Gain Calibration

Note that the average active power result from the LPF output in each phase can be scaled by ±50% by writing to the phase's VAGAIN register (AVAG, BVAG, or CVAG). The VAGAIN registers are twos complement, signed registers and have a resolution of 0.024%/LSB. The function of the VAGAIN registers is expressed mathematically as

$$Average\ Apparent\ Power = LPF2\ Output \times \left(1 + \frac{VAGAIN\ Register}{2^{12}}\right) \quad (40)$$

The output is scaled by −50% by writing 0x800 to the VAR gain registers and increased by +50% by writing 0x7FF to them. These registers can be used to calibrate the apparent power (or energy) calculation in the ADE7758 for each phase.

Apparent Power Offset Calibration

Each rms measurement includes an offset compensation register to calibrate and eliminate the dc component in the rms value (see the Current RMS Calculation section and the Voltage Channel RMS Calculation section). The voltage and current rms values are then multiplied together in the apparent power signal processing. As no additional offsets are created in the multiplication of the rms values, there is no specific offset compensation in the apparent power signal processing. The offset compensation of the apparent power measurement in each phase should be done by calibrating each individual rms measurement (see the Calibration section).

Apparent Energy Calculation

Apparent energy is defined as the integral of apparent power.

$$Apparent\ Energy = \int S(t)\mathrm{dt} \quad (41)$$

Similar to active and reactive energy, the ADE7758 achieves the integration of the apparent power signal by continuously accumulating the apparent power signal in the internal 41-bit, unsigned accumulation registers. The VA-hr registers (AVAHR, BVAHR, and CVAHR) represent the upper 16 bits of these internal registers. This discrete time accumulation or summation is equivalent to integration in continuous time. Equation 42 expresses the relationship

$$Apparent\ Energy = \int S(t)\,\mathrm{dt} = \lim_{T\to 0}\left\{\sum_{n=0}^{\infty} S(nT) \times T\right\} \quad (42)$$

where:

n is the discrete time sample number.
T is the sample period.

Figure 76 shows the signal path of the apparent energy accumulation. The apparent power signal is continuously added to the internal apparent energy register. The average apparent power is divided by the content of the VA divider register before it is added to the corresponding VA-hr accumulation register. When the value in the VADIV[7:0] register is 0 or 1, apparent power is accumulated without any division. VADIV is an 8-bit unsigned register that is useful to lengthen the time it takes before the VA-hr accumulation registers overflow.

Similar to active or reactive power accumulation, the fastest integration time occurs when the VAGAIN registers are set to maximum full scale, that is, 0x7FF. When overflow occurs, the content of the VA-hr accumulation registers can roll over to 0 and continue increasing in value.

By setting the VAEHF bit (Bit 2) of the mask register, the ADE7758 can be configured to issue an interrupt ($\overline{\text{IRQ}}$) when the MSB of any one of the three VA-hr accumulation registers has changed, indicating that the accumulation register is half full.

Setting the RSTREAD bit (Bit 6) of the LCYMODE register enables a read-with-reset for the VA-hr accumulation registers; that is, the registers are reset to 0 after a read operation.

Integration Time Under Steady Load

The discrete time sample period (T) for the accumulation register is 0.4 µs (4/CLKIN). With full-scale, 60 Hz sinusoidal signals on the analog inputs and the VAGAIN registers set to 0x000, the average word value from each LPF2 is 0xB9954. The maximum value that can be stored in the apparent energy register before it overflows is $2^{16} - 1$ or 0xFFFF. As the average word value is first added to the internal register, which can store $2^{41} - 1$ or 0x1FF, FFFF, FFFF before it overflows, the integration time under these conditions with VADIV = 0 is calculated as

$$Time = \frac{0\text{x}1\text{FF, FFFF, FFFF}}{0\text{xB9954}} \times 0.4\ \mu\text{s} = 1.157\ \text{sec} \quad (43)$$

When VADIV is set to a value different from 0, the time before overflow is scaled accordingly, as shown in Equation 44.

$$Time = Time(VADIV = 0) \times VADIV \quad (44)$$

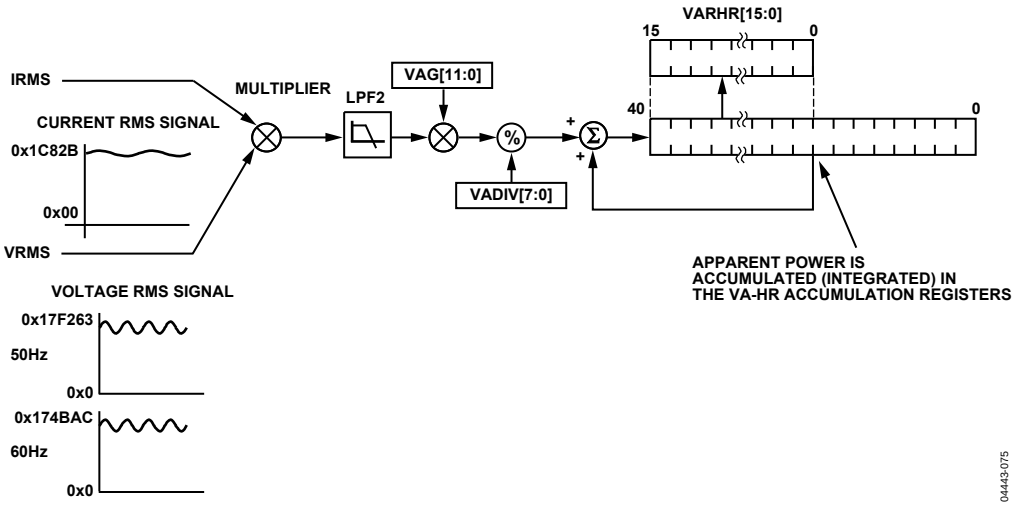


Figure 76. ADE7758 Apparent Energy Accumulation

Table 14. Inputs to VA-Hr Accumulation Registers

CONSEL[1, 0]	AVAHR[1]	BVAHR	CVAHR
00	AVRMS × AIRMS	BVRMS × BIRMS	CVRMS × CIRMS
01	AVRMS × AIRMS	AVRMS + CVRMS/2 × BIRMS	CVRMS × CIRMS
10	AVRMS × AIRMS	BVRMS × BIRMS	CVRMS × CIRMS
11	Reserved	Reserved	Reserved

[1] AVRMS/BVRMS/CVRMS are the rms voltage waveform, and AIRMS/BIRMS/CIRMS are the rms values of the current waveform.

Energy Accumulation Mode

The apparent power accumulated in each VA-hr accumulation register (AVAHR, BVAHR, or CVAHR) depends on the configuration of the CONSEL bits in the COMPMODE register (Bit 0 and Bit 1). The different configurations are described in Table 14.

The contents of the VA-hr accumulation registers are affected by both the registers for rms voltage gain (VRMSGAIN), as well as the VAGAIN register of the corresponding phase.

Apparent Power Frequency Output

Pin 17 (VARCF) of the ADE7758 provides frequency output for the total apparent power. By setting the VACF bit (Bit 7) of the WAVMODE register, this pin provides an output frequency that is directly proportional to the total apparent power.

A digital-to-frequency converter (DFC) is used to generate the pulse output from the total apparent power. The TERMSEL bits (Bit 2 to Bit 4) of the COMPMODE register can be used to select which phases to include in the total power calculation. Setting Bit 2, Bit 3, and Bit 4 includes the input to the AVAHR, BVAHR, and CVAHR registers in the total apparent power calculation. A pair of frequency divider registers, namely VARCFDEN and VARCFNUM, can be used to scale the output frequency of this pin. Note that either VAR or apparent power can be selected at one time for this frequency output (see the Reactive Power Frequency Output section).

Line Cycle Apparent Energy Accumulation Mode

The line cycle apparent energy accumulation mode is activated by setting the LVA bit (Bit 2) in the LCYCMODE register. The total apparent energy accumulated over an integer number of zero crossings is written to the VA-hr accumulation registers after the LINECYC number of zero crossings is detected. The operation of this mode is similar to watt-hr accumulation (see the Line Cycle Active Energy Accumulation Mode section). When using the line cycle accumulation mode, the RSTREAD bit (Bit 6) of the LCYCMODE register should be set to Logic 0.

Note that this mode is especially useful when the user chooses to perform the apparent energy calculation using the vectorial method.

By setting LWATT and LVAR bits (Bit 0 and Bit 1) of the LCYCMODE register, the active and reactive energies are accumulated over the same period. Therefore, the MCU can perform the squaring of the two terms and then take the square root of their sum to determine the apparent energy over the same period.

ENERGY REGISTERS SCALING

The ADE7758 provides measurements of active, reactive, and apparent energies that use separate signal paths and filtering for calculation. The differences in the datapaths can result in small differences in LSB weight between the active, reactive, and apparent energy registers. These measurements are internally compensated so that the scaling is nearly one to one. The relationship between the registers is shown in Table 15.

Table 15. Energy Registers Scaling

	Frequency	
	60 Hz	50 Hz
Integrator Off		
VAR	1.004 × WATT	1.0054 × WATT
VA	1.00058 × WATT	1.0085 × WATT
Integrator On		
VAR	1.0059 × WATT	1.0064 × WATT
VA	1.00058 × WATT	1.00845 × WATT

WAVEFORM SAMPLING MODE

The waveform samples of the current and voltage waveform, as well as the active, reactive, and apparent power multiplier outputs, can all be routed to the WAVEFORM register by setting the WAVSEL[2:0] bits (Bit 2 to Bit 4) in the WAVMODE register. The phase in which the samples are routed is set by setting the PHSEL[1:0] bits (Bit 0 and Bit 1) in the WAVMODE register. All energy calculation remains uninterrupted during waveform sampling. Four output sample rates can be chosen by using Bit 5 and Bit 6 of the WAVMODE register (DTRT[1:0]). The output sample rate can be 26.04 kSPS, 13.02 kSPS, 6.51 kSPS, or 3.25 kSPS (see Table 20).

By setting the WFSM bit in the interrupt mask register to Logic 1, the interrupt request output $\overline{\text{IRQ}}$ goes active low when a sample is available. The 24-bit waveform samples are transferred from the ADE7758 one byte (8 bits) at a time, with the most significant byte shifted out first.

The interrupt request output $\overline{\text{IRQ}}$ stays low until the interrupt routine reads the reset status register (see the Interrupts section).

CALIBRATION

A reference meter or an accurate source is required to calibrate the ADE7758 energy meter. When using a reference meter, the ADE7758 calibration output frequencies APCF and VARCF are adjusted to match the frequency output of the reference meter under the same load conditions. Each phase must be calibrated separately in this case. When using an accurate source for calibration, one can take advantage of the line cycle accumulation mode and calibrate the three phases simultaneously.

There are two objectives in calibrating the meter: to establish the correct impulses/kW-hr constant on the pulse output and to obtain a constant that relates the LSBs in the energy and rms registers to Watt/VA/VAR hours, amps, or volts. Additionally, calibration compensates for part-to-part variation in the meter design as well as phase shifts and offsets due to the current sensor and/or input networks.

Calibration Using Pulse Output

The ADE7758 provides a pulsed output proportional to the active power accumulated by all three phases, called APCF. Additionally, the VARCF output is proportional to either the reactive energy or apparent energy accumulated by all three phases. The following section describes how to calibrate the gain, offset, and phase angle using the pulsed output information. The equations are based on the pulse output from the ADE7758 (APCF or VARCF) and the pulse output of the reference meter or $CF_{EXPECTED}$.

Figure 77 shows a flowchart of how to calibrate the ADE7758 using the pulse output. Because the pulse outputs are proportional to the total energy in all three phases, each phase must be calibrated individually. Writing to the registers is fast to reconfigure the part for calibrating a different phase; therefore, Figure 77 shows a method that calibrates all phases at a given test condition before changing the test condition.

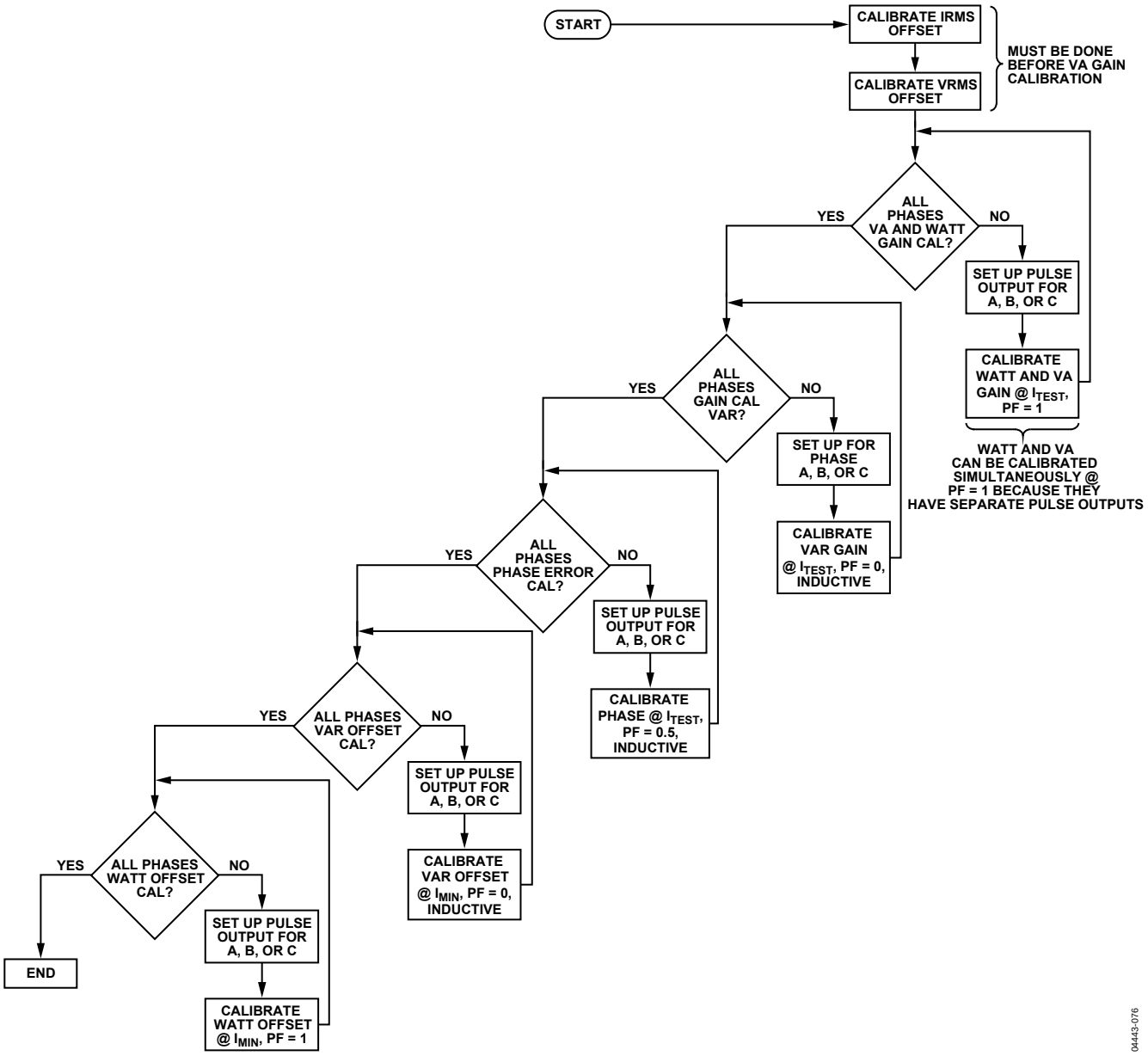


Figure 77. Calibration Using Pulse Output

Gain Calibration Using Pulse Output

Gain calibration is used for meter-to-meter gain adjustment, APCF or VARCF output rate calibration, and determining the Wh/LSB, VARh/LSB, and VAh/LSB constant. The registers used for watt gain calibration are APCFNUM (0x45), APCFDEN (0x46), and xWG (0x2A to 0x2C). Equation 50 through Equation 52 show how these registers affect the Wh/LSB constant and the APCF pulses.

For calibrating VAR gain, the registers in Equation 50 through Equation 52 should be replaced by VARCFNUM (0x47), VARCFDEN (0x48), and xVARG (0x2D to 0x2F). For VAGAIN, they should be replaced by VARCFNUM (0x47), VARCFDEN (0x48), and xVAG (0x30 to 0x32).

Figure 78 shows the steps for gain calibration of watts, VA, or VAR using the pulse outputs.

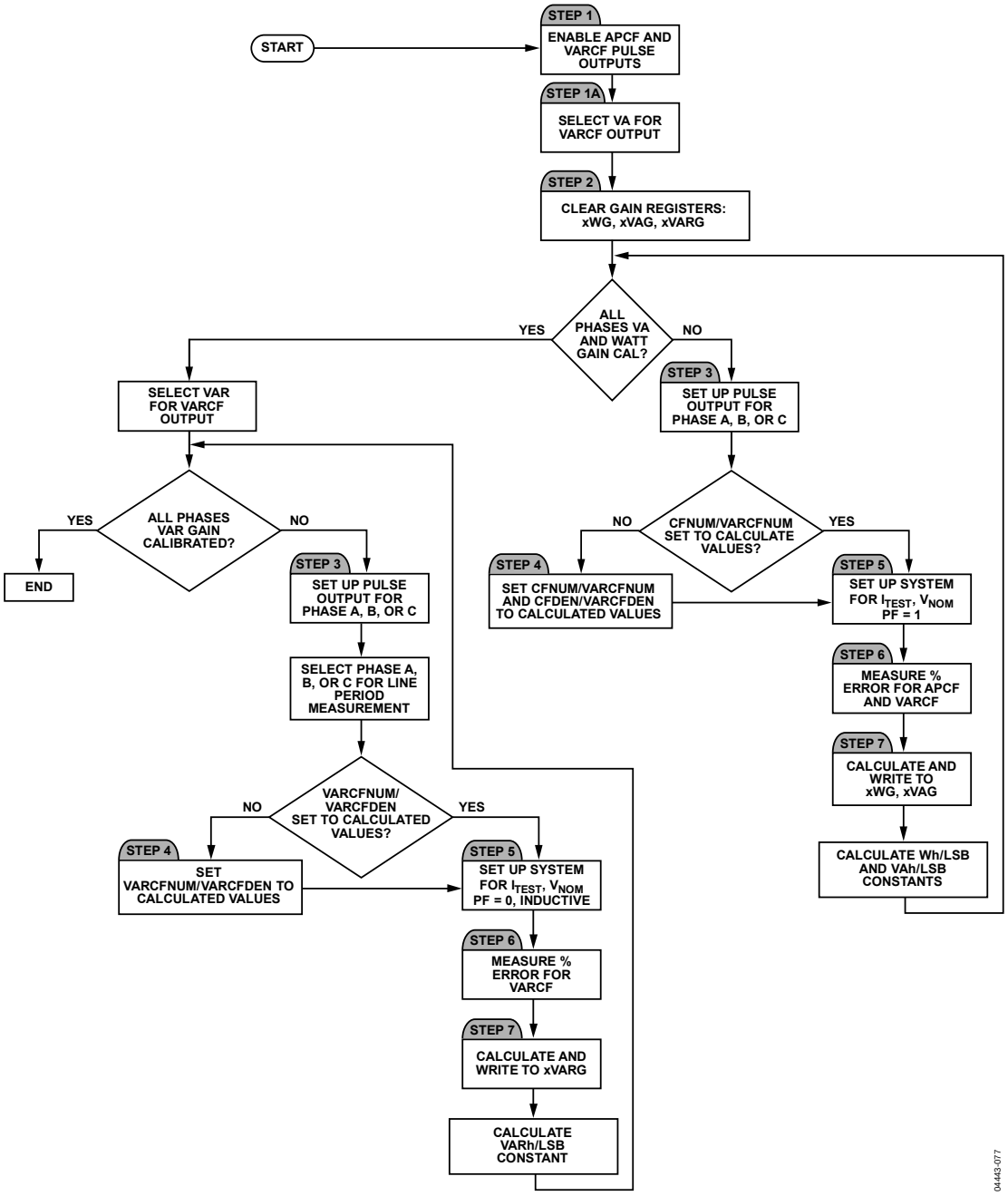


Figure 78. Gain Calibration Using Pulse Output

Step 1: Enable the pulse output by setting Bit 2 of the OPMODE register (0x13) to Logic 0. This bit enables both the APCF and VARCF pulses.

Step 1a: VAR and VA share the VARCF pulse output. WAVMODE[7], Address (0x15), should be set to choose between VAR or VA pulses on the output. Setting the bit to Logic 1 selects VA. The default is Logic 0 or VARCF pulse output.

Step 2: Ensure the xWG/xVARG/xVAG are zero.

Step 3: Disable the Phase B and Phase C contribution to the APCF and VARCF pulses. This is done by the TERMSEL[2:4] bits of the COMPMODE register (0x16). Setting Bit 2 to Logic 1 and Bit 3 and Bit 4 to Logic 0 allows only Phase A to be included in the pulse outputs. Select Phase A, Phase B, or Phase C for a line period measurement with the FREQSEL[1:0] bits in the MMODE register (0x14). For example, clearing Bit 1 and Bit 0 selects Phase A for line period measurement.

Step 4: Set APCFNUM (0x45) and APCFDEN (0x46) to the calculated value to perform a coarse adjustment on the imp/kWh ratio. For VAR/VA calibration, set VARCFNUM (0x47) and VARCFDEN (0x48) to the calculated value.

The pulse output frequency with one phase at full-scale inputs is approximately 16 kHz. A sample set of meters could be tested to find a more exact value of the pulse output at full scale in the user application.

To calculate the values for APCFNUM/APCFDEN and VARCFNUM/VARCFDEN, use the following formulas:

$$APCF_{NOMINAL} = 16\text{ kHz} \times \frac{V_{NOM}}{V_{FULLSCALE}} \times \frac{I_{TEST}}{I_{FULLSCALE}} \quad (45)$$

$$APCF_{EXPECTED} = \frac{MC \times I_{TEST} \times V_{NOM}}{1000 \times 3600} \times \cos(\theta) \quad (46)$$

$$APCFDEN = INT\left(\frac{APCF_{NOMINAL}}{APCF_{EXPECTED}}\right) \quad (47)$$

where:

MC is the meter constant.
I_{TEST} is the test current.
V_{NOM} is the nominal voltage at which the meter is tested.
$V_{FULLSCALE}$ and $I_{FULLSCALE}$ are the values of current and voltage, which correspond to the full-scale ADC inputs of the ADE7758.
θ is the angle between the current and the voltage channel.
$APCF_{EXPECTED}$ is equivalent to the reference meter output under the test conditions.
APCFNUM is written to 0 or 1.

The equations for calculating the VARCFNUM and VARCFDEN during VAR calibration are similar:

$$VARCF_{EXPECTED} = \frac{MC \times I_{TEST} \times V_{NOM}}{1000 \times 3600} \times \sin(\theta) \quad (48)$$

Because the APCFDEN and VARCFDEN values can be calculated from the meter design, these values can be written to the part automatically during production calibration.

Step 5: Set the test system for I_{TEST}, V_{NOM}, and the unity power factor. For VAR calibration, the power factor should be set to 0 inductive in this step. For watt and VA, the unity power factor should be used. VAGAIN can be calibrated at the same time as WGAIN because VAGAIN can be calibrated at the unity power factor, and both pulse outputs can be measured simultaneously. However, when calibrating VAGAIN at the same time as WGAIN, the rms offsets should be calibrated first (see the Calibration of IRMS and VRMS Offset section).

Step 6: Measure the percent error in the pulse output, APCF and/or VARCF, from the reference meter:

$$\%Error = \frac{APCF - CF_{REF}}{CF_{REF}} \times 100\% \quad (49)$$

where CF_{REF} = $APCF_{EXPECTED}$ = the pulse output of the reference meter.

Step 7: Calculate xWG adjustment. One LSB change in xWG (12 bits) changes the WATTHR register by 0.0244% and therefore APCF by 0.0244%. The same relationship holds true for VARCF.

$$APCF_{EXPECTED} = APCF_{NOMINAL} \times \frac{APCFNUM[11:0]}{APCFDEN[11:0]} \times \left(1 + \frac{xWG[11:0]}{2^{12}}\right) \quad (50)$$

$$xWG = -\frac{\%Error}{0.0244\%} \quad (51)$$

When APCF is calibrated, the xWATTHR registers have the same Wh/LSB from meter to meter if the meter constant and the APCFNUM/APCFDEN ratio remain the same. The Wh/LSB constant is

$$\frac{Wh}{LSB} = \frac{1}{4 \times \frac{MC}{1000} \times \frac{APCFDEN}{APCFNUM} \times \frac{1}{WDIV}} \quad (52)$$

Return to Step 2 to calibrate Phase B and Phase C gain.

Example: Watt Gain Calibration of Phase A Using Pulse Output

For this example, I_{TEST} = 10 A, V_{NOM} = 220 V, $V_{FULLSCALE}$ = 500 V, $I_{FULLSCALE}$ = 130 A, MC = 3200 impulses/kWh, Power Factor = 1, and Frequency = 50 Hz.

Clear APCFNUM (0x45) and write the calculated value to APCFDEN (0x46) to perform a coarse adjustment on the imp/kWh ratio, using Equation 45 through Equation 47.

$$APCF_{NOMINAL} = 16\text{ kHz} \times \frac{220}{500} \times \frac{10}{130} = 0.542\text{ kHz}$$

$$APCF_{EXPECTED} = \frac{3200 \times 10 \times 220}{1000 \times 3600} \times \cos(0) = 1.9556\text{ Hz}$$

$$APCFDEN = INT\left(\frac{542\text{ Hz}}{1.9556\text{ Hz}}\right) = 277$$

With Phase A contributing to CF, at I_{TEST}, V_{NOM}, and the unity power factor, the example ADE7758 meter shows 2.058 Hz on the pulse output. This is equivalent to a 5.26% error from the reference meter value using Equation 49.

$$\%Error = \frac{2.058\text{ Hz} - 1.9556\text{ Hz}}{1.9556\text{ Hz}} \times 100\% = 5.26\%$$

The AWG value is calculated to be −216 d using Equation 51, which means the value 0xF28 should be written to AWG.

$$AWG = -\frac{5.26\%}{0.0244\%} = -215.5 = -216 = 0xF28$$

PHASE CALIBRATION USING PULSE OUTPUT

The ADE7758 includes a phase calibration register on each phase to compensate for small phase errors. Large phase errors should be compensated by adjusting the antialiasing filters. The ADE7758 phase calibration is a time delay with different weights in the positive and negative direction (see the Phase Compensation section). Because a current transformer is a source of phase error, a fixed nominal value can be decided on to load into the xPHCAL registers at power-up. During calibration, this value can be adjusted for CT-to-CT error. Figure 79 shows the steps involved in calibrating the phase using the pulse output.

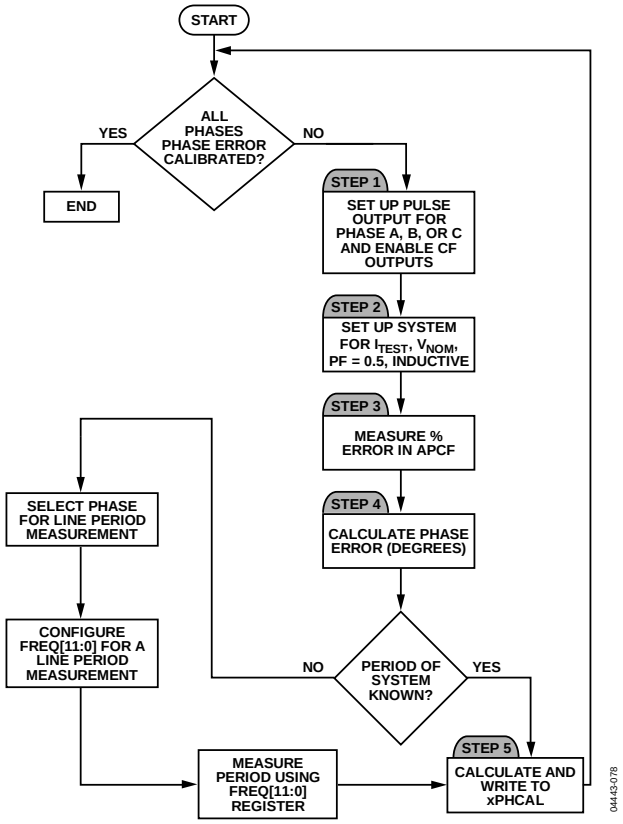


Figure 79. Phase Calibration Using Pulse Output

Step 1: Step 1 and Step 3 from the gain calibration should be repeated to configure the ADE7758 pulse output. Ensure the xPHCAL registers are zero.

Step 2: Set the test system for I_{TEST}, V_{NOM}, and 0.5 power factor inductive.

Step 3: Measure the percent error in the pulse output, APCF, from the reference meter using Equation 49.

Step 4: Calculate the Phase Error in degrees by

$$Phase\ Error(^\circ) = -Arcsin\left(\frac{\%Error}{100\% \times \sqrt{3}}\right) \quad (53)$$

Step 5: Calculate xPHCAL.

$$xPHCAL = Phase\ Error \times \frac{1}{PHCAL\_LSB\_Weight} \times \frac{1}{Line\ Period(s)} \times \frac{1}{360^\circ} \quad (54)$$

where *PHCAL_LSB_Weight* is 1.2 µs if the %Error is negative or 2.4 µs if the %Error is positive (see the Phase Compensation section).

If it is not known, the line period is available in the ADE7758 frequency register, FREQ (0x10). To configure line period measurement, select the phase for period measurement in the MMODE[1:0] and set LCYCMODE[7]. Equation 55 shows how to determine the value that needs to be written to *xPHCAL* using the period register measurement.

$$xPHCAL = Phase\ Error \times \frac{9.6\ \mu s}{PHCAL\_LSB\_Weight} \times \frac{FREQ[11:0]}{360^\circ} \quad (55)$$

Example: Phase Calibration of Phase A Using Pulse Output

For this example, I_{TEST} = 10 A, V_{NOM} = 220 V, $V_{FULLSCALE}$ = 500 V, $I_{FULLSCALE}$ = 130 A, MC = 3200 impulses/kWh, power factor = 0.5 inductive, and frequency = 50 Hz.

With Phase A contributing to CF, at I_{TEST}, V_{NOM}, and 0.5 inductive power factor, the example ADE7758 meter shows 0.9668 Hz on the pulse output. This is equivalent to −1.122% error from the reference meter value using Equation 49.

The *Phase Error* in degrees using Equation 53 is 0.3713°.

$$Phase\ Error(^\circ) = -Arcsin\left(\frac{-1.122\%}{100\% \times \sqrt{3}}\right) = 0.3713^\circ$$

If at 50 Hz the FREQ register = 2083d, the value that should be written to APHCAL is 17d, or 0x11 using Equation 55. Note that a PHCAL_LSB_Weight of 1.2 µs is used because the %Error is negative.

$$APHCAL = 0.3713^\circ \times \frac{9.6\ \mu s}{1.2\ \mu s} \times \frac{2083}{360^\circ} = 17.19 = 17 = 0x11$$

Power Offset Calibration Using Pulse Output

Power offset calibration should be used for outstanding performance over a wide dynamic range (1000:1). Calibration of the power offset is done at or close to the minimum current where the desired accuracy is required.

The ADE7758 has power offset registers for watts and VAR (xWATTOS and xVAROS). Offsets in the VA measurement are compensated by adjusting the rms offset registers (see the Calibration of IRMS and VRMS Offset section). Figure 80 shows the steps to calibrate the power offsets using the pulse outputs.

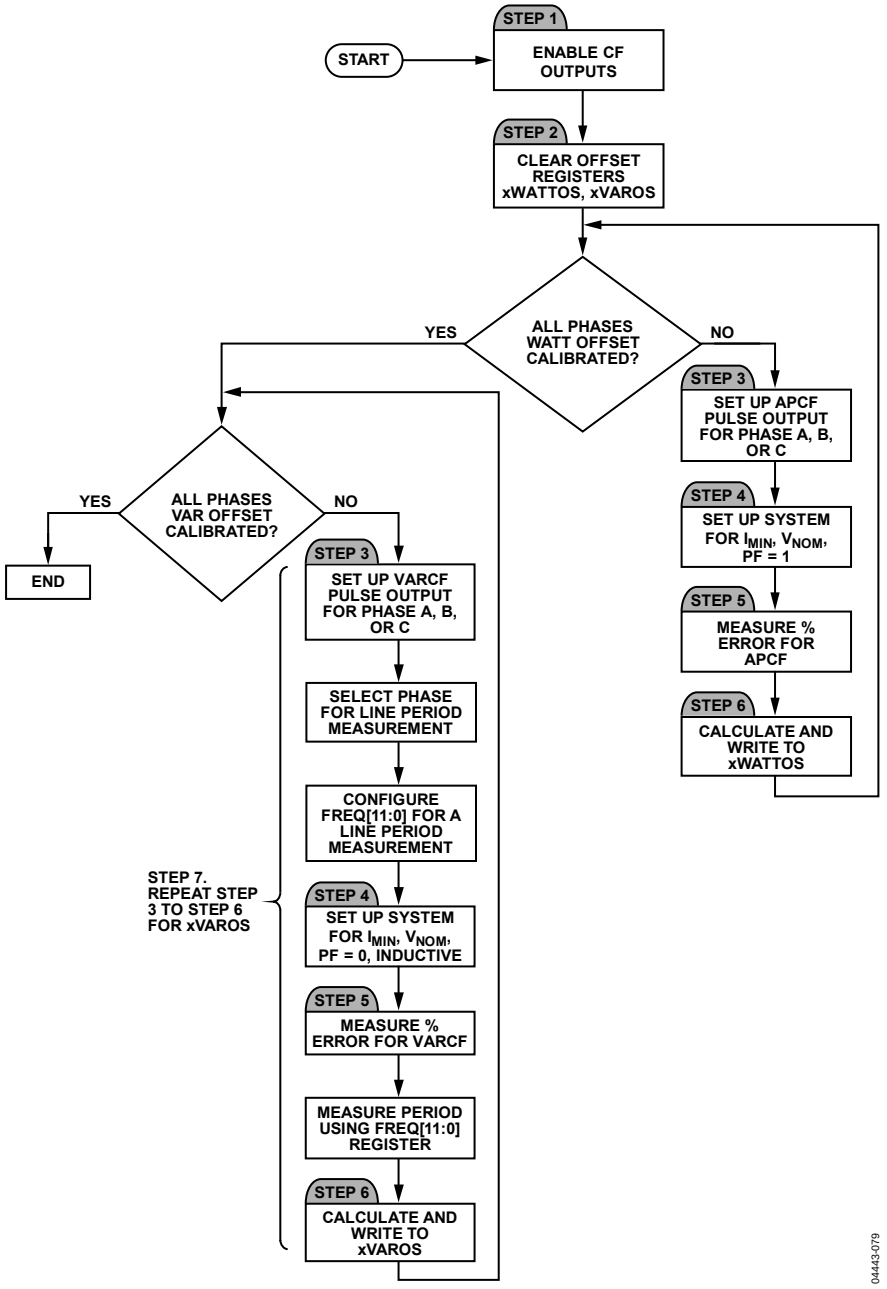


Figure 80. Offset Calibration Using Pulse Output

Step 1: Repeat Step 1 and Step 3 from the gain calibration to configure the ADE7758 pulse output.

Step 2: Clear the xWATTOS and xVAROS registers.

Step3: Disable the Phase B and Phase C contribution to the APCF and VARCF pulses. This is done by the TERMSEL[2:4] bits of the COMPMODE register (0x16). Setting Bit 2 to Logic 1 and Bit 3 and Bit 4 to Logic 0 allows only Phase A to be included in the pulse outputs. Select Phase A, Phase B, or Phase C for a line period measurement with the FREQSEL[1:0] bits in the MMODE register (0x14). For example, clearing Bit 1 and Bit 0 selects Phase A for line period measurement.

Step 4: Set the test system for I_{MIN}, V_{NOM}, and unity power factor. For Step 6, set the test system for I_{MIN}, V_{NOM}, and zero-power factor inductive.

Step 5: Measure the percent error in the pulse output, APCF or VARCF, from the reference meter using Equation 49.

Step 6: Calculate xWATTOS using Equation 56 (for xVAROS use Equation 57).

$$xWATTOS = -\left(\frac{\%APCF_{ERROR}}{100\%} \times APCF_{EXPECTED}\right) \times \frac{2^4}{Q} \times \frac{APCFDEN}{APCFNUM} \quad (56)$$

$$xVAROS = -\left(\frac{\%VARCF_{ERROR}}{100\%} \times VARCF_{EXPECTED}\right) \times \frac{2^4}{Q} \times \frac{VARCFDEN}{VARCFNUM} \quad (57)$$

where Q is defined in Equation 58 and Equation 59.

For xWATTOS,

$$Q = \frac{CLKIN}{4} \times \frac{1}{2^{25}} \times \frac{1}{4} \quad (58)$$

For xVAROS,

$$Q = \frac{CLKIN}{4} \times \frac{1}{2^{24}} \times \frac{202}{\left(\frac{FREQ[11:0]}{4}\right)} \times \frac{1}{4} \quad (59)$$

where the FREQ (0x10) register is configured for line period measurements.

Step 7: Repeat Step 3 to Step 6 for xVAROS calibration.

Example: Offset Calibration of Phase A Using Pulse Output

For this example, I_{MIN} = 50 mA, V_{NOM} = 220 V, $V_{FULLSCALE}$ = 500 V, $I_{FULLSCALE}$ = 130 A, MC = 3200 impulses/kWh, Power Factor = 1, Frequency = 50 Hz, and CLKIN = 10 MHz.

With I_{MIN}, V_{NOM}, and unity power factor, the example ADE7758 meter shows 0.009789 Hz on the APCF pulse output. When the power factor is changed to 0.5 inductive, the VARCF output is 0.009769 Hz.

This is equivalent to 0.1198% for the watt measurement and −0.0860% for the VAR measurement. Using Equation 56 through Equation 59, the values 0xFFD and 0x3 should be written to AWATTOS (0x39) and AVAROS (0x3C), respectively.

$$AWATTOS = -\left(\frac{0.1198\%}{100\%} \times 0.009778\right) \times \frac{2^4}{0.01863} \times \frac{277}{1} = -2.8 = -3 = 0xFFD$$

$$AVAROS = -\left(\frac{-0.0860\%}{100\%} \times 0.009778\right) \times \frac{2^4}{0.01444} \times \frac{277}{1} = 2.6 = 3$$

For AWATTOS,

$$Q = \frac{10E6}{4} \times \frac{1}{2^{25}} \times \frac{1}{4} = 0.01863$$

For AVAROS,

$$Q = \frac{10E6}{4} \times \frac{1}{2^{24}} \times \frac{202}{\frac{2083}{4}} \times \frac{1}{4} = 0.01444$$

Calibration Using Line Accumulation

Line cycle accumulation mode configures the nine energy registers such that the amount of energy accumulated over an integer number of half line cycles appears in the registers after the LENERGY interrupt. The benefit of using this mode is that the sinusoidal component of the active energy is eliminated.

Figure 81 shows a flowchart of how to calibrate the ADE7758 using the line accumulation mode. Calibration of all phases and energies can be done simultaneously using this mode to save time during calibration.

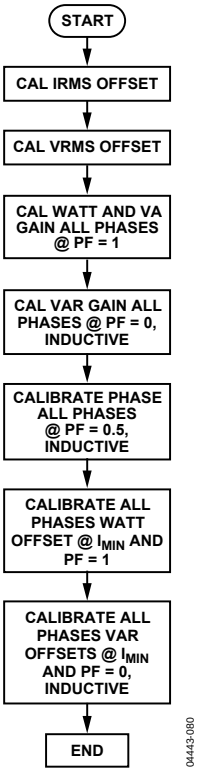


Figure 81. Calibration Using Line Accumulation

Gain Calibration Using Line Accumulation

Gain calibration is used for meter-to-meter gain adjustment, APCF or VARCF output rate calibration, and determining the Wh/LSB, VARh/LSB, and VAh/LSB constant.

Step 0: Before performing the gain calibration, the APCFNUM/ APCFDEN (0x45/0x46) and VARCFNUM/ VARCFDEN (0x47/0x48) values can be set to achieve the correct impulses/kWh, impulses/kVAh, or impulses/kVARh using the same method outlined in Step 4 in the Gain Calibration Using Pulse Output section. The calibration of xWG/xVARG/xVAG (0x2A through 0x32) is done with the line accumulation mode. Figure 82 shows the steps involved in calibrating the gain registers using the line accumulation mode.

Step 1: Clear xWG, xVARG, and xVAG.

Step 2: Select Phase A, Phase B, or Phase C for a line period measurement with the FREQSEL[1:0] bits in the MMODE register (0x14). For example, clearing Bit 1 and Bit 0 selects Phase A for line period measurement.

Step 3: Set up ADE7758 for line accumulation by writing 0xBF to LCYCMODE. This enables the line accumulation mode on the xWATTHR, xVARHR, and xVAHR (0x01 to 0x09) registers by setting the LWATT, LVAR, and LVA bits, LCYCMODE[0:2] (0x17), to Logic 1. It also sets the ZXSEL bits, LCYCMODE[3:5], to Logic 1 to enable the zero-crossing detection on all phases for line accumulation. Additionally, the FREQSEL bit, LCYCMODE[7], is set so that FREQ (0x10) stores the line period. When using the line accumulation mode, the RSTREAD bit of LCYCMODE should be set to 0 to disable the read with reset mode. Select the phase for line period measurement in MMODE[1:0].

Step 4: Set the number of half-line cycles for line accumulation by writing to LINECYC (0x1C).

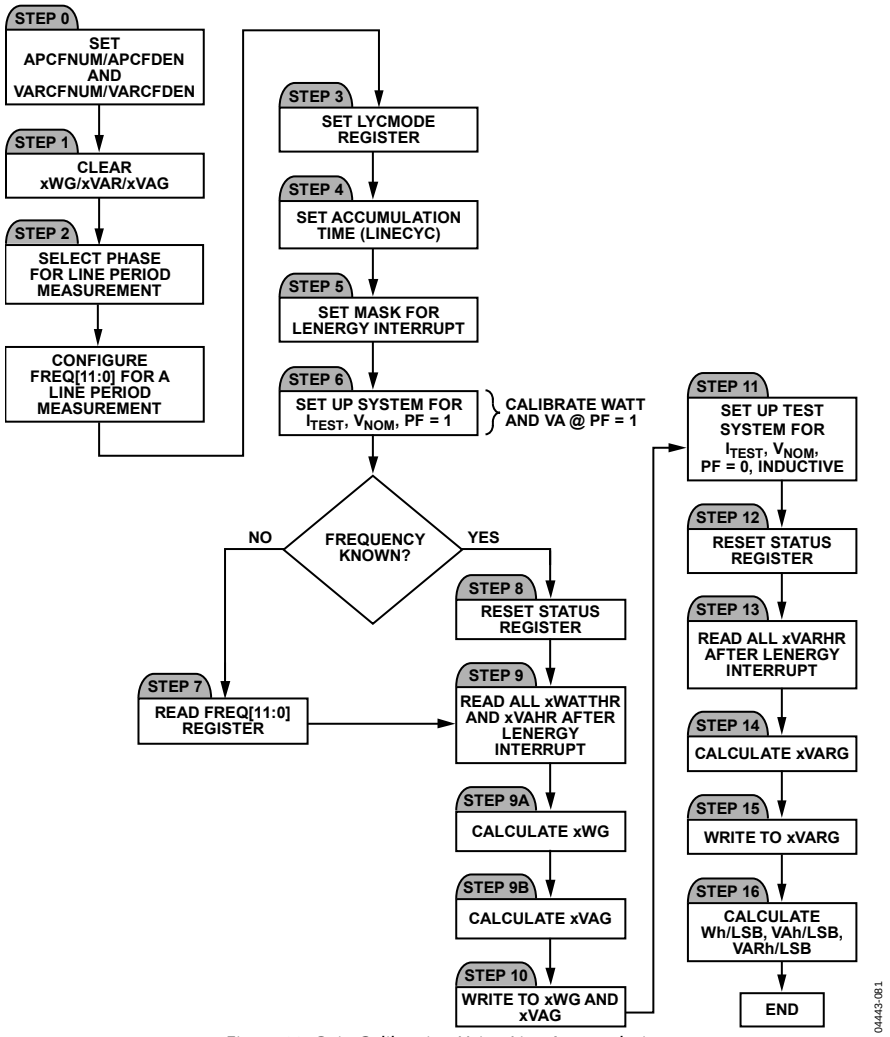


Figure 82. Gain Calibration Using Line Accumulation

Step 5: Set the LENERGY bit, MASK[12] (0x18), to Logic 1 to enable the interrupt signaling the end of the line cycle accumulation.

Step 6: Set the test system for I_{TEST}, V_{NOM}, and unity power factor (calibrate watt and VA simultaneously and first).

Step 7: Read the FREQ (0x10) register if the line frequency is unknown.

Step 8: Reset the interrupt status register by reading RSTATUS (0x1A).

Step 9: Read all six xWATTHR (0x01 to 0x03) and xVAHR (0x07 to 0x09) energy registers after the LENERGY interrupt and store the values.

Step 9a: Calculate the values to be written to xWG registers according to the following equations:

$$WATTHR_{EXPECTED} = \frac{4 \times MC \times I_{TEST} \times V_{NOM} \times cos(\theta) \times AccumTime}{1000 \times 3600} \times \frac{APCFDEN}{APCFNUM} \times \frac{1}{WDIV} \quad (60)$$

where *AccumTime* is

$$\frac{LINECYC[15:0]}{2 \times Line\ Frequency \times No.\ of\ Phases\ Selected} \quad (61)$$

where:

MC is the meter constant.

θ is the angle between the current and voltage.

Line Frequency is known or calculated from the FREQ[11:0] register. With the FREQ[11:0] register configured for line period measurements, the line frequency is calculated with Equation 62.

$$Line\ Frequency = \frac{1}{FREQ[11:0] \times 9.6 \times 10^{-6}} \quad (62)$$

No. of Phases Selected is the number of ZXSEL bits set to Logic 1 in LCYCMODE (0x17).

Then, *xWG* is calculated as

$$xWG = \left(\frac{WATTHR_{EXPECTED}}{WATTHR_{MEASURED}} - 1 \right) \times 2^{12} \quad (63)$$

Step 9b: Calculate the values to be written to the xVAG registers according to the following equation:

$$VAHR_{EXPECTED} = \frac{4 \times MC \times I_{TEST} \times V_{NOM} \times AccumTime}{1000 \times 3600} \times \frac{VARCFDEN}{VARCFNUM} \times \frac{1}{VADIV} \quad (64)$$

$$xVAG = \left(\frac{VAHR_{EXPECTED}}{VAHR_{MEASURED}} - 1 \right) \times 2^{12}$$

Step 10: Write to xWG and xVAG.

Step 11: Set the test system for I_{TEST}, V_{NOM}, and zero power factor inductive to calibrate VAR gain.

Step 12: Repeat Step 7.

Step 13: Read the xVARHR (0x04 to 0x06) after the LENERGY interrupt and store the values.

Step 14: Calculate the values to be written to the xVARG registers (to adjust VARCF to the expected value).

$$VARHR_{EXPECTED} = \frac{4 \times MC \times I_{TEST} \times V_{NOM} \times sin(\theta) \times AccumTime}{1000 \times 3600} \times \frac{VARCFDEN}{VARCFNUM} \times \frac{1}{VARDIV} \quad (65)$$

$$xVARG = \left(\frac{VARHR_{EXPECTED}}{VARHR_{MEASURED}} - 1 \right) \times 2^{12}$$

Step 15: Write to xVARG.

Step 16: Calculate the Wh/LSB, VARh/LSB, and VAh/LSB constants.

$$\frac{Wh}{LSB} = \frac{I_{TEST} \times V_{NOM} \times cos(\theta) \times AccumTime}{3600 \times xWATTHR} \quad (66)$$

$$\frac{VAh}{LSB} = \frac{I_{TEST} \times V_{NOM} \times AccumTime}{3600 \times xVAHR} \quad (67)$$

$$\frac{VARh}{LSB} = \frac{I_{TEST} \times V_{NOM} \times sin(\theta) \times AccumTime}{3600 \times xVARHR} \quad (68)$$

Example: Watt Gain Calibration Using Line Accumulation

This example shows only Phase A watt calibration. The steps outlined in the Gain Calibration Using Line Accumulation section show how to calibrate watt, VA, and VAR. All three phases can be calibrated simultaneously because there are nine energy registers.

For this example, I_{TEST} = 10 A, V_{NOM} = 220 V, Power Factor = 1, Frequency = 50 Hz, LINECYC (0x1C) is set to 0x800, and MC = 3200 imp/kWhr.

To set APCFNUM (0x45) and APCFDEN (0x46) to the calculated value to perform a coarse adjustment on the imp/kW-hr ratio, use Equation 45 to Equation 47.

$$APCF_{NOMINAL} = 16\text{ kHz} \times \frac{220}{500} \times \frac{10}{130} = 0.5415\text{ kHz}$$

$$APCF_{EXPECTED} = \frac{3200 \times 10 \times 220}{1000 \times 3600} \times \cos(\theta) = 1.956\text{ Hz}$$

$$APCFDEN = \text{INT}\left(\frac{541.5\text{ Hz}}{1.956\text{ Hz}}\right) = 277$$

Under the test conditions above, the AWATTHR register value is 15559d after the LENERGY interrupt. Using Equation 60 and Equation 61, the value to be written to AWG is −199d, 0xF39.

$$AccumTime = \frac{LINECYC[15{:}0]}{2 \times \frac{1}{FREQ[11:0] \times 9.6 \times 10^{-6}} \times No.\ of\ \ Phases\ Selected}$$

$$AccumTime = \frac{0x800}{2 \times \frac{1}{2085 \times 9.6 \times 10^{-6}} \times 3} = 6.832128\text{s}$$

$$WATTHR_{EXPECTED} = \frac{4 \times 3200 \times 10 \times 220 \times 1 \times 6.832}{1000 \times 3600} \times \frac{277}{1} \times 1 = 14804$$

$$xWG = \left(\frac{14804}{15559} - 1\right) \times 2^{12} = -198.87640 = -199 = 0\text{xF39}$$

Using Equation 66, the Wh/LSB constant is

$$\frac{Wh}{LSB} = \frac{10 \times 220 \times 6.832}{3600 \times 14804} = 0.0002820$$

Phase Calibration Using Line Accumulation

The ADE7758 includes a phase calibration register on each phase to compensate for small phase errors. Large phase errors should be compensated by adjusting the antialiasing filters. The ADE7758 phase calibration is a time delay with different weights in the positive and negative direction (see the Phase Compensation section). Because a current transformer is a source of phase error, a fixed nominal value can be decided on to load into the xPHCAL (0x3F to 0x41) registers at power-up. During calibration, this value can be adjusted for CT-to-CT error. Figure 83 shows the steps involved in calibrating the phase using the line accumulation mode.

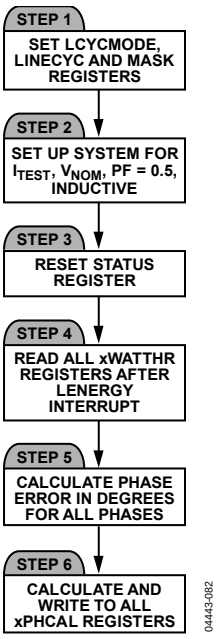


Figure 83. Phase Calibration Using Line Accumulation

Step 1: If the values were changed after gain calibration, Step 1, Step 3, and Step 4 from the gain calibration should be repeated to configure the LCYCMODE and LINECYC registers.

Step 2: Set the test system for I_{TEST}, V_{NOM}, and 0.5 power factor inductive.

Step 3: Reset the interrupt status register by reading RSTATUS (0x1A).

Step 4: The xWATTHR registers should be read after the LENERGY interrupt. Measure the percent error in the energy register readings (AWATTHR, BWATTHR, and CWATTHR) compared to the energy register readings at unity power factor (after gain calibration) using Equation 69. The readings at unity power factor should have been repeated after the gain calibration and stored for use in the phase calibration routine.

$$Error = \frac{xWATTHR_{PF=5} - \frac{xWATTHR_{PF=1}}{2}}{\frac{xWATTHR_{PF=1}}{2}} \quad (69)$$

Step 5: Calculate the *Phase Error* in degrees using the equation

$$Phase\ Error(^\circ) = -Arcsin\left(\frac{Error}{\sqrt{3}}\right) \quad (70)$$

Step 6: Calculate xPHCAL and write to the xPHCAL registers (0x3F to 0x41).

$$xPHCAL = Phase\ Error \times \frac{1}{PHCAL\_LSB\_Weight} \times \frac{1}{Line\ Period(s)} \times \frac{1}{360^\circ} \quad (71)$$

where PHCAL_LSB_Weight is 1.2 µs if the %Error is negative or 2.4 µs if the %Error is positive (see the Phase Compensation section).

If it is not known, the line period is available in the ADE7758 frequency register, FREQ (0x10). To configure line period measurement, select the phase for period measurement in the MMODE[1:0] and set LCYCMODE[7]. Equation 72 shows how to determine the value that needs to be written to *xPHCAL* using the period register measurement.

$$xPHCAL = Phase\ Error \times \frac{9.6\,\mu s}{PHCAL\_LSB\_Weight} \times \frac{FREQ[11:0]}{360^\circ} \quad (72)$$

Example: Phase Calibration Using Line Accumulation

This example shows only Phase A phase calibration. All three PHCAL registers can be calibrated simultaneously using the same method.

For this example, I_{TEST} = 10 A, V_{NOM} = 220 V, power factor = 0.5 inductive, and frequency = 50 Hz. Also, LINECYC = 0x800.

With I_{TEST}, V_{NOM}, and 0.5 inductive power factor, the example ADE7758 meter shows 7318d in the AWATTHR (0x01) register. For unity power factor (after gain calibration), the meter shows 14804d in the AWATTHR register. This is equivalent to −1.132% error.

$$Error = \frac{7318 - \frac{14804}{2}}{\frac{14804}{2}} = -0.01132 = -1.132\%$$

The *Phase Error* in degrees using Equation 66 is 0.374°.

$$Phase\ Error(^\circ) = -Arc\sin\left(\frac{-0.01132}{\sqrt{3}}\right) = 0.374^\circ$$

Using Equation 72, the value written to APHCAL (0x3F), if at 50 Hz, the FREQ (0x10) register = 2085d, is 17d. Note that a PHCAL_LSB_Weight of 1.2 µs is used because the %Error is negative.

$$APHCAL = 0.374^\circ \times \frac{9.6}{1.2} \times \frac{2085}{360} = 17 = 0x11$$

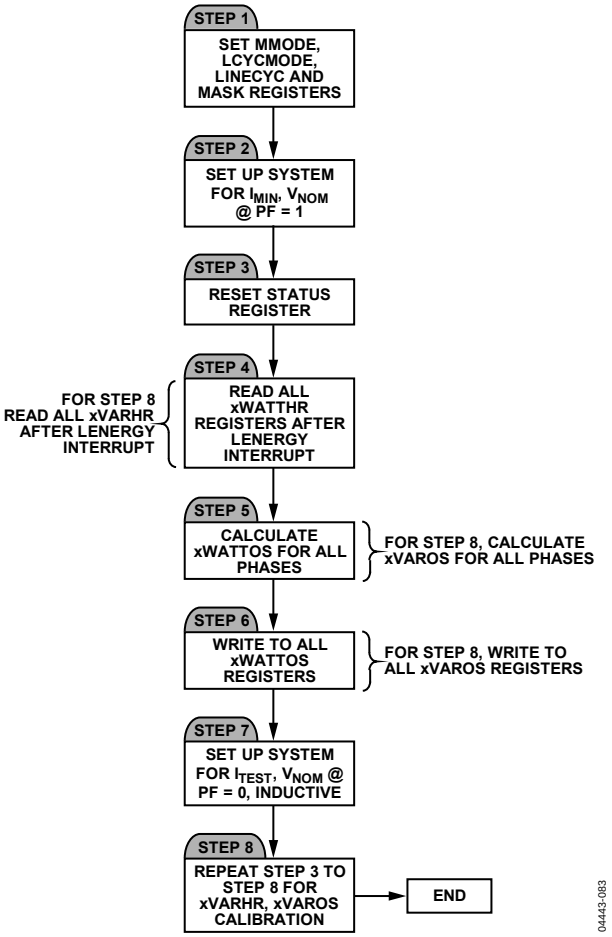


Figure 84. Power Offset Calibration Using Line Accumulation

Power Offset Calibration Using Line Accumulation

Power offset calibration should be used for outstanding performance over a wide dynamic range (1000:1). Calibration of the power offset is done at or close to the minimum current. The ADE7758 has power offset registers for watts and VAR, xWATTOS (0x39 to 0x3B) and xVAROS (0x3C to 0x3E). Offsets in the VA measurement are compensated by adjusting the rms offset registers (see the Calibration of IRMS and VRMS Offset section).

More line cycles could be required at the minimum current to minimize the effect of quantization error on the offset calibration. For example, if a current of 40 mA results in an active energy accumulation of 113 after 2000 half line cycles, one LSB variation in this reading represents an 0.8% error. This measurement does not provide enough resolution to calibrate out a <1% offset error. However, if the active energy is accumulated over 37,500 half line cycles, one LSB variation results in 0.05% error, reducing the quantization error.

Figure 84 shows the steps to calibrate the power offsets using the line accumulation mode.

Step 1: If the values change after gain calibration, Step 1, Step 3, and Step 4 from the gain calibration should be repeated to configure the LCYCMODE, LINECYC, and MASK registers. Select Phase A, Phase B, or Phase C for a line period measurement with the FREQSEL[1:0] bits in the MMODE register (0x14). For example, clearing Bit 1 and Bit 0 selects Phase A for line period measurement.

Step 2: Set the test system for I_{MIN}, V_{NOM}, and unity power factor.

Step 3: Reset the interrupt status register by reading RSTATUS (0x1A).

Step 4: Read all xWATTHR energy registers (0x01 to 0x03) after the LENERGY interrupt and store the values.

Step 4a: If it is not known, the line period is available in the ADE7758 frequency register, FREQ (0x10). To configure line period measurement, select the phase for period measurement in the MMODE[1:0] and set LCYCMODE[7].

Step 5: Calculate the value to be written to the xWATTOS registers according to the following equations:

$$Offset = \frac{xWATTHR_{I_{MIN}} \times I_{TEST} - \left(xWATTHR_{I_{TEST}} \times \frac{LINECYC_{IMIN}}{LINECYC_{ITEST}} \right) \times I_{MIN}}{I_{MIN} - I_{TEST}} \quad (73)$$

$$xWATTOS[11:0] = \frac{Offset \times 4}{AccumTime \times CLKIN} \times 2^{29} \quad (74)$$

where:

AccumTime is defined in Equation 61.

$xWATTHR_{I_{TEST}}$ is the value in the energy register at I_{TEST}.

$xWATTHR_{I_{MIN}}$ is the value in the energy register at I_{MIN}.

$LINECYC_{IMIN}$ is the number of line cycles accumulated at I_{MIN}.

$LINECYC_{IMAX}$ is the number of line cycles accumulated at I_{MAX}.

Step 6: Write to all xWATTOS registers (0x39 to 0x3B).

Step 7: Set the test system for I_{MIN}, V_{NOM}, and zero power factor inductive to calibrate VAR gain.

Step 8: Repeat Steps 3, 4, and 5.

Step 9: Calculate the value written to the xVAROS registers according to the following equations:

$$Offset = \frac{xVARHR_{I_{MIN}} \times I_{TEST} - \left(xVARHR_{I_{TEST}} \times \frac{LINECYC_{IMIN}}{LINECYC_{ITEST}} \right) \times I_{MIN}}{I_{MIN} - I_{TEST}} \quad (75)$$

$$xVAROS[11:0] = \frac{Offset \times 4}{AccumTime \times CLKIN} \times \frac{FREQ[11:0]}{202} \times 2^{26} \quad (76)$$

where the FREQ[11:0] register is configured for line period readings.

Example: Power Offset Calibration Using Line Accumulation

This example only shows Phase A of the phase active power offset calibration. Both active and reactive power offset for all phases can be calibrated simultaneously using the method explained in the Power Offset Calibration Using Line Accumulation section.

For this example, I_{MIN} = 50 mA, I_{TEST} = 10 A, V_{NOM} = 220 V, $V_{FULLSCALE}$ = 500 V, $I_{FULLSCALE}$ = 130 A, MC = 3200 impulses/kWh, Power Factor = 1, Frequency = 50 Hz, and CLKIN = 10 MHz. Also, $LINECYC_{ITEST}$ = 0x800 and $LINECYC_{IMIN}$ = 0x4000.

After accumulating over 0x800 line cycles for gain calibration at I_{TEST}, the example ADE7758 meter shows 14804d in the AWATTHR (0x01) register. At I_{MIN}, the meter shows 592d in the AWATTHR register. By using Equation 73, this is equivalent to 0.161 LSBs of offset; therefore, using Equation 61 and Equation 74, the value written to AWATTOS is 0d.

$$Offset = \frac{592 \times 10 - \left(14804 \times \frac{0x4000}{0x800} \right) \times 0.05}{0.05 - 10} = 0.16$$

$$AccumTime = \frac{0 \times 4000}{2 \times \frac{1}{2085 \times 9.6 \times 10^{-6}} \times 3} = 54.64s$$

$$AWATTOS = \frac{0.161 \times 4}{54.64 \times 10\ \text{MHz}} \times 2^{29} = -0.088 = 0$$

Calibration of IRMS and VRMS Offset

IRMSOS and VRMSOS are used to cancel noise and offset contributions from the inputs. The calibration method is the same whether calibrating using the pulse outputs or line accumulation. Reading the registers is required for this calibration because there is no rms pulse output. The rms offset calibration should be performed before VAGAIN calibration. The rms offset calibration also removes offset from the VA calculation. For this reason, no VA offset register exists in the ADE7758.

The low-pass filter used to obtain the rms measurements is not ideal; therefore, it is recommended to synchronize the readings with the zero crossings of the voltage waveform and to average a few measurements when reading the rms registers.

The ADE7758 IRMS measurement is linear over a 500:1 range, and the VRMS measurement is linear over a 20:1 range. To measure the voltage VRMS offset (xVRMSOS), measure rms values at two different nonzero current levels, for example, V_{NOM} and $V_{FULLSCALE}/20$.

To measure the current rms offset (IRMSOS), measure rms values at two different nonzero current levels, for example, I_{TEST} and $I_{FULLSCALE}/500$. This translates to two test conditions: I_{TEST} and V_{NOM}, and $I_{FULLSCALE}/500$ and $V_{FULLSCALE}/20$. Figure 85 shows a flowchart for calibrating the rms measurements.

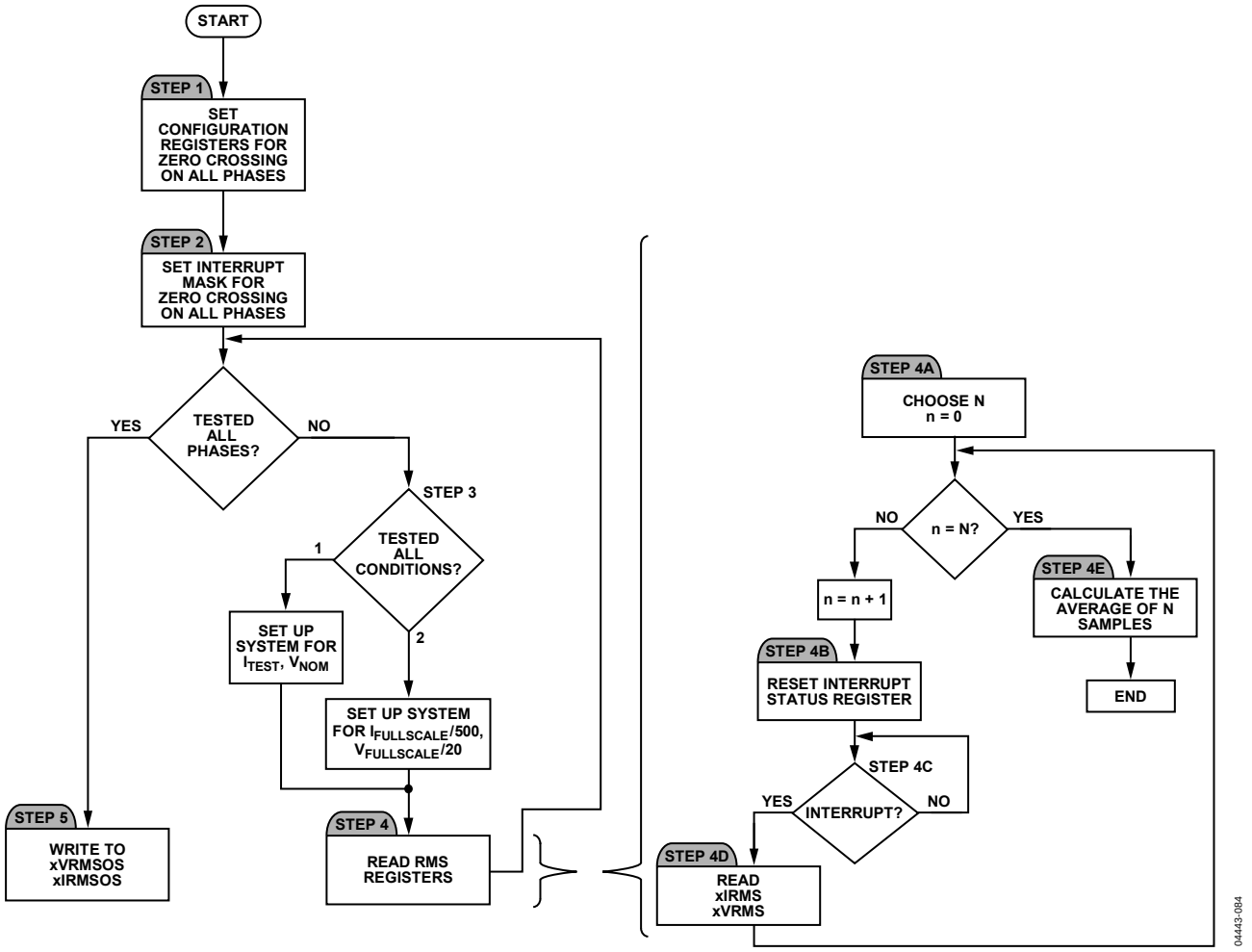


Figure 85. RMS Calibration Routine

Step 1: Set configuration registers for zero crossings on all phases by writing the value 0x38 to the LCYCMODE register (0x17). This sets all of the ZXSEL bits to Logic 1.

Step 2: Set the interrupt mask register for zero-crossing detection on all phases by writing 0xE00 to the MASK[0:24] register (0x18). This sets all of the ZX bits to Logic 1.

Step 3: Set up the calibration system for one of the two test conditions: I_{TEST} and V_{NOM}, and $I_{FULLSCALE}/500$ and $V_{FULLSCALE}/20$.

Step 4: Read the rms registers after the zero-crossing interrupt and take an average of N samples. This is recommended to get the most stable rms readings. This procedure is detailed in Figure 85: Steps 4a through 4e.

Step 4a. Choose the number of samples, N, to be averaged.

Step 4b. Reset the interrupt status register by reading RSTATUS (0x1A).

Step 4c. Wait for the zero-crossing interrupt. When the zero-crossing interrupt occurs, move to Step 4d.

Step 4d. Read the xIRMS and xVRMS registers. These values will be averaged in Step 4e.

Step 4e: Average the N samples of xIRMS and xVRMS. The averaged values will be used in Step 5.

Step 5: Write to the xVRMSOS (0x33 to 0x35) and xIRMSOS (0x36 to 0x38) registers according to the following equations:

$$xIRMSOS = \frac{1}{16384} \times \frac{\left(I_{TEST}^2 \times IRMS_{IMIN}^2\right) - \left(I_{MIN}^2 \times IRMS_{ITEST}^2\right)}{I_{MIN}^2 - I_{TEST}^2} \quad (77)$$

where:

I_{MIN} is the full scale current/500.
I_{TEST} is the test current.

$IRMS_{IMIN}$ and $IRMS_{ITEST}$ are the current rms register values without offset correction for the inputs I_{MIN} and I_{TEST}, respectively.

$$xVRMSOS = \frac{1}{64} \times \frac{V_{NOM} \times VRMS_{VMIN} - V_{MIN} \times VRMS_{VNOM}}{V_{MIN} - V_{NOM}} \quad (78)$$

where:

V_{MIN} is the full scale voltage/20
V_{NOM} is the nominal line voltage.

$VRMS_{VMIN}$ and $VRMS_{VNOM}$ are the voltage rms register values without offset correction for the input V_{MIN} and V_{NOM}, respectively.

Example: Calibration of RMS Offsets

For this example, I_{TEST} = 10 A, I_{MAX} = 100 A, V_{NOM} = 220 V, $V_{FULLSCALE}$ = 500 V, Power Factor = 1, and Frequency = 50 Hz.

Twenty readings are taken synchronous to the zero crossings of all three phases at each current and voltage to determine the average xIRMS and xVRMS readings. At I_{TEST} and V_{NOM}, the example ADE7758 meter gets an average AIRMS (0x0A) reading of 148242.2 and 744570.8 in the AVRMS (0x0D) register. Then the current is set to $I_{MIN} = I_{FULLSCALE}/500$ or 260 mA. At I_{MIN}, the average AIRMS reading is 3885.68. At $V_{MIN} = V_{FULLSCALE}/20$ or 25 V, the example meter gets an average AVRMS of 86362.36. Using this data, −15d is written to AIRMSOS (0x36) and −31d is written to AVRMSOS (0x33) registers according to the Equation 77 and Equation 78.

$$AIRMSOS = \frac{1}{16384} \times \frac{\left(10^2 \times 3885.68^2\right) - \left(0.260^2 \times 148242.2^2\right)}{\left(0.260 - 10^2\right)} =$$

$$-14.8 = -15 = 0xFF2$$

$$AVRMSOS = \frac{1}{64} \times \frac{\left(220 \times 86362.36\right) - \left(25 \times 744570.8\right)}{\left(25 - 220\right)} = -30.9 = -31 = 0xFE1$$

This example shows the calculations and measurements for Phase A only. However, all three xIRMS and xVRMS registers can be read simultaneously to compute the values for each xIRMSOS and xVRMSOS register.

CHECKSUM REGISTER

The ADE7758 has a checksum register CHKSUM[7:0] (0x7E) to ensure the data bits received in the last serial read operation are not corrupted. The 8-bit checksum register is reset before the first bit (MSB of the register to be read) is put on the DOUT pin. During a serial read operation, when each data bit becomes available on the rising edge of SCLK, the bit is added to the checksum register. In the end of the serial read operation, the contents of the checksum register are equal to the sum of all the 1s in the register previously read. Using the checksum register, the user can determine if an error has occurred during the last read operation. Note that a read to the checksum register also generates a checksum of the checksum register itself.

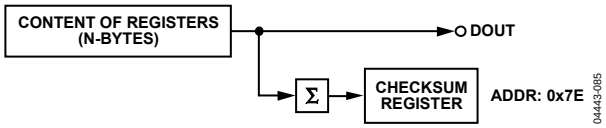


Figure 86. Checksum Register for Serial Interface Read

INTERRUPTS

The ADE7758 interrupts are managed through the interrupt status register (STATUS[23:0], Address 0x19) and the interrupt mask register (MASK[23:0], Address 0x18). When an interrupt event occurs in the ADE7758, the corresponding flag in the interrupt status register is set to a Logic 1 (see Table 24). If the mask bit for this interrupt in the interrupt mask register is Logic 1, then the $\overline{IRQ}$ logic output goes active low. The flag bits

in the interrupt status register are set irrespective of the state of the mask bits. To determine the source of the interrupt, the MCU should perform a read from the reset interrupt status register with reset. This is achieved by carrying out a read from RSTATUS, Address 0x1A. The $\overline{IRQ}$ output goes logic high on completion of the interrupt status register read command (see the Interrupt Timing section). When carrying out a read with reset, the ADE7758 is designed to ensure that no interrupt events are missed. If an interrupt event occurs just as the interrupt status register is being read, the event is not lost, and the $\overline{IRQ}$ logic output is guaranteed to go logic high for the duration of the interrupt status register data transfer before going logic low again to indicate the pending interrupt. Note that the reset interrupt bit in the status register is high for only one clock cycle, and it then goes back to 0.

USING THE INTERRUPTS WITH AN MCU

Figure 87 shows a timing diagram that illustrates a suggested implementation of ADE7758 interrupt management using an MCU. At time t_1, the $\overline{IRQ}$ line goes active low indicating that one or more interrupt events have occurred in the ADE7758. The $\overline{IRQ}$ logic output should be tied to a negative-edge-triggered external interrupt on the MCU. On detection of the negative edge, the MCU should be configured to start executing its interrupt service routine (ISR). On entering the ISR, all interrupts should be disabled using the global interrupt mask bit. At this point, the MCU external interrupt flag can be cleared to capture interrupt events that occur during the current ISR. When the MCU interrupt flag is cleared, a read from the reset interrupt status register with reset is carried out. (This causes the $\overline{IRQ}$ line to be reset logic high (t_2); see the Interrupt Timing section.) The reset interrupt status register contents are used to determine the source of the interrupt(s) and hence the appropriate action to be taken. If a subsequent interrupt event occurs during the ISR (t_3) that event is recorded by the MCU external interrupt flag being set again.

On returning from the ISR, the global interrupt mask bit is cleared (same instruction cycle) and the external interrupt flag uses the MCU to jump to its ISR once again. This ensures that the MCU does not miss any external interrupts. The reset bit in the status register is an exception to this and is only high for one clock cycle after a reset event.

INTERRUPT TIMING

The Serial Interface section should be reviewed before reviewing this section. As previously described, when the $\overline{IRQ}$ output goes low, the MCU ISR must read the interrupt status register to determine the source of the interrupt. When reading the interrupt status register contents, the $\overline{IRQ}$ output is set high on the last falling edge of SCLK of the first byte transfer (read interrupt status register command). The $\overline{IRQ}$ output is held high until the last bit of the next 8-bit transfer is shifted out (interrupt status register contents), as shown in Figure 88. If an interrupt is pending at this time, the $\overline{IRQ}$ output goes low again. If no interrupt is pending, the $\overline{IRQ}$ output remains high.

SERIAL INTERFACE

The ADE7758 has a built-in SPI interface. The serial interface of the ADE7758 is made of four signals: SCLK, DIN, DOUT, and $\overline{CS}$. The serial clock for a data transfer is applied at the SCLK logic input. This logic input has a Schmitt trigger input structure that allows slow rising (and falling) clock edges to be used. All data transfer operations are synchronized to the serial clock. Data is shifted into the ADE7758 at the DIN logic input on the falling edge of SCLK. Data is shifted out of the ADE7758 at the DOUT logic output on a rising edge of SCLK.

The $\overline{CS}$ logic input is the chip select input. This input is used when multiple devices share the serial bus. A falling edge on $\overline{CS}$ also resets the serial interface and places the ADE7758 in communications mode.

The $\overline{CS}$ input should be driven low for the entire data transfer operation. Bringing $\overline{CS}$ high during a data transfer operation aborts the transfer and places the serial bus in a high impedance state. The $\overline{CS}$ logic input can be tied low if the ADE7758 is the only device on the serial bus.

However, with $\overline{CS}$ tied low, all initiated data transfer operations must be fully completed. The LSB of each register must be transferred because there is no other way of bringing the ADE7758 back into communications mode without resetting the entire device, that is, performing a software reset using Bit 6 of the OPMODE[7:0] register, Address 0x13.

The functionality of the ADE7758 is accessible via several on-chip registers (see Figure 89). The contents of these registers can be updated or read using the on-chip serial interface. After a falling edge on $\overline{CS}$, the ADE7758 is placed in communications mode. In communications mode, the ADE7758 expects the first communication to be a write to the internal communications register. The data written to the communications register contains the address and specifies the next data transfer to be a read or a write command. Therefore, all data transfer operations with the ADE7758, whether a read or a write, must begin with a write to the communications register.

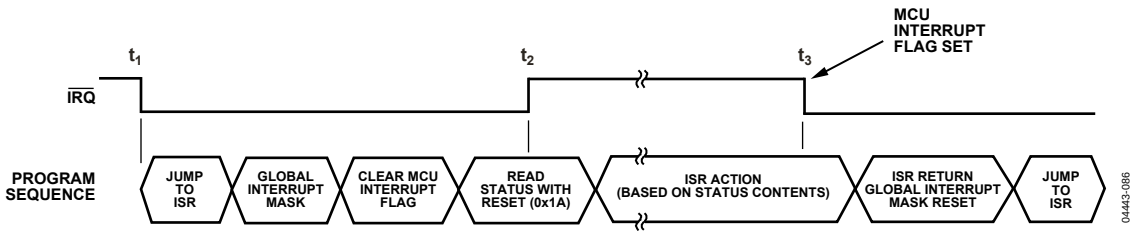


Figure 87. ADE7758 Interrupt Management

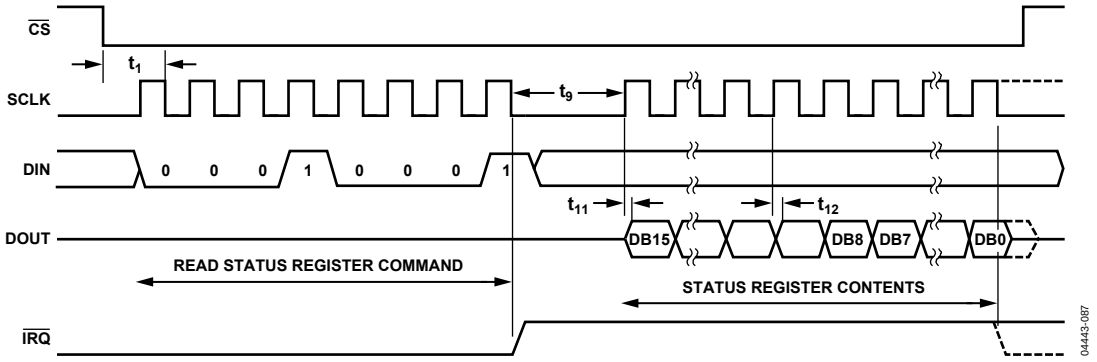


Figure 88. ADE7758 Interrupt Timing

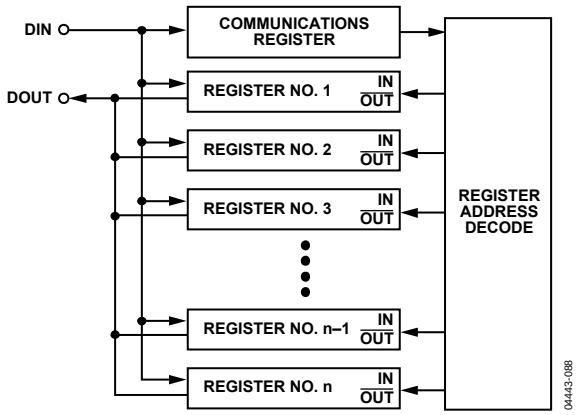


Figure 89. Addressing ADE7758 Registers via the Communications Register

The communications register is an 8-bit, write-only register. The MSB determines whether the next data transfer operation is a read or a write. The seven LSBs contain the address of the register to be accessed (see Table 16).

Figure 90 and Figure 91 show the data transfer sequences for a read and write operation, respectively.

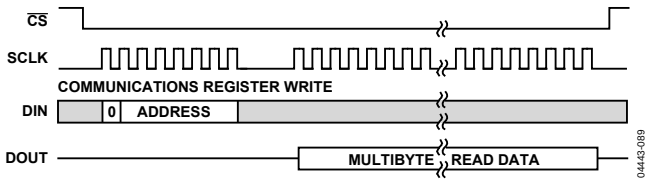


Figure 90. Reading Data from the ADE7758 via the Serial Interface

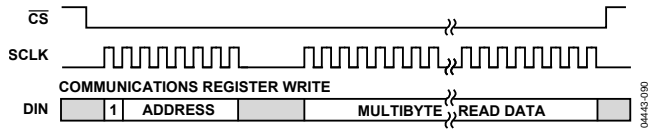


Figure 91. Writing Data to the ADE7758 via the Serial Interface

On completion of a data transfer (read or write), the ADE7758 once again enters into communications mode, that is, the next instruction followed must be a write to the communications register.

A data transfer is completed when the LSB of the ADE7758 register being addressed (for a write or a read) is transferred to or from the ADE7758.

SERIAL WRITE OPERATION

The serial write sequence takes place as follows. With the ADE7758 in communications mode and the $\overline{CS}$ input logic low, a write to the communications register takes place first. The MSB of this byte transfer must be set to 1, indicating that the next data transfer operation is a write to the register. The seven LSBs of this byte contain the address of the register to be written to. The ADE7758 starts shifting in the register data on the next falling edge of SCLK. All remaining bits of register data are shifted in on the falling edge of the subsequent SCLK pulses (see Figure 92).

As explained earlier, the data write is initiated by a write to the communications register followed by the data. During a data write operation to the ADE7758, data is transferred to all on-chip registers one byte at a time. After a byte is transferred into the serial port, there is a finite time duration before the content in the serial port buffer is transferred to one of the ADE7758 on-chip registers. Although another byte transfer to the serial port can start while the previous byte is being transferred to the destination register, this second-byte transfer should not finish until at least 900 ns after the end of the previous byte transfer. This functionality is expressed in the timing specification t_6 (see Figure 92). If a write operation is aborted during a byte transfer ($\overline{CS}$ brought high), then that byte is not written to the destination register.

Destination registers can be up to 3 bytes wide (see the Accessing the On-Chip Registers section). Therefore, the first byte shifted into the serial port at DIN is transferred to the most significant byte (MSB) of the destination register. If the destination register is 12 bits wide, for example, a two-byte data transfer must take place. The data is always assumed to be right justified; therefore, in this case, the four MSBs of the first byte would be ignored, and the four LSBs of the first byte written to the ADE7758 would be the four MSBs of the 12-bit word. Figure 93 illustrates this example.

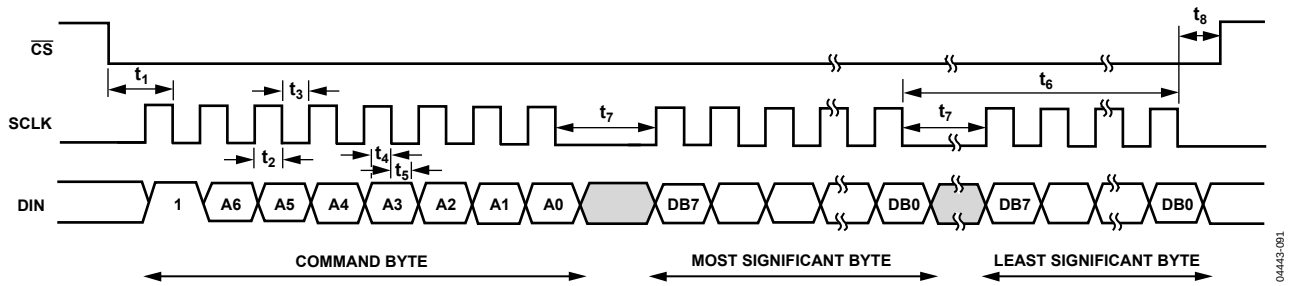


Figure 92. Serial Interface Write Timing Diagram

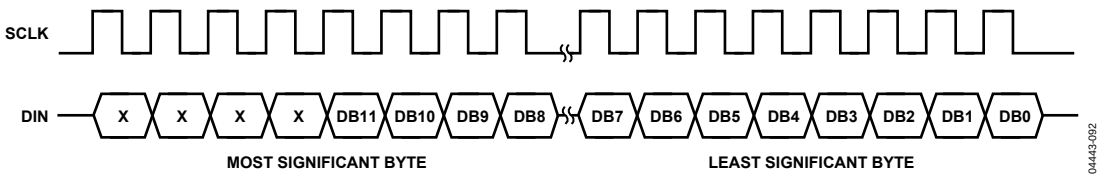


Figure 93. 12-Bit Serial Write Operation

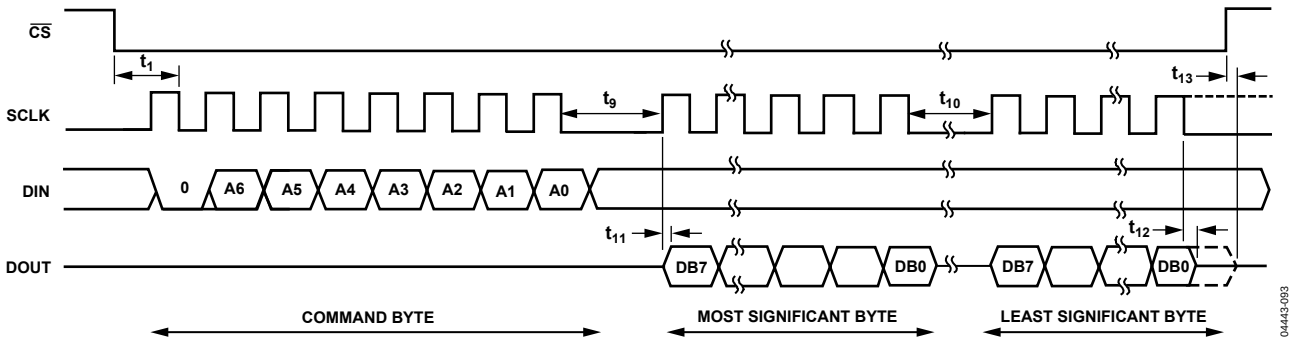


Figure 94. Serial Interface Read Timing Diagram

SERIAL READ OPERATION

During a data read operation from the ADE7758, data is shifted out at the DOUT logic output on the rising edge of SCLK. As was the case with the data write operation, a data read must be preceded with a write to the communications register.

With the ADE7758 in communications mode and $\overline{CS}$ logic low, an 8-bit write to the communications register takes place first. The MSB of this byte transfer must be a 0, indicating that the next data transfer operation is a read. The seven LSBs of this byte contain the address of the register that is to be read. The ADE7758 starts shifting out of the register data on the next rising edge of SCLK (see Figure 94). At this point, the DOUT logic output switches from a high impedance state and starts driving the data bus. All remaining bits of register data are shifted out on subsequent SCLK rising edges. The serial interface enters communications mode again as soon as the read is completed. The DOUT logic output enters a high impedance state on the falling edge of the last SCLK pulse.

The read operation can be aborted by bringing the $\overline{CS}$ logic input high before the data transfer is completed. The DOUT output enters a high impedance state on the rising edge of $\overline{CS}$.

When an ADE7758 register is addressed for a read operation, the entire contents of that register are transferred to the serial port. This allows the ADE7758 to modify its on-chip registers without the risk of corrupting data during a multibyte transfer.

Note that when a read operation follows a write operation, the read command (that is, write to communications register) should not happen for at least 1.1 μs after the end of the write operation. If the read command is sent within 1.1 μs of the write operation, the last byte of the write operation can be lost.

ACCESSING THE ON-CHIP REGISTERS

All ADE7758 functionality is accessed via the on-chip registers. Each register is accessed by first writing to the communications register and then transferring the register data. For a full description of the serial interface protocol, see the Serial Interface section.

REGISTERS

COMMUNICATIONS REGISTER

The communications register is an 8-bit, write-only register that controls the serial data transfer between the ADE7758 and the host processor. All data transfer operations must begin with a write to the communications register.

The data written to the communications register determines whether the next operation is a read or a write and which register is being accessed.

Table 16 outlines the bit designations for the communications register.

Table 16. Communications Register

Bit Location	Bit Mnemonic	Description
0 to 6	A0 to A6	The seven LSBs of the communications register specify the register for the data transfer operation. Table 17 lists the address of each ADE7758 on-chip register.
7	W/$\overline{R}$	When this bit is a Logic 1, the data transfer operation immediately following the write to the communications register is interpreted as a write to the ADE7758. When this bit is a Logic 0, the data transfer operation immediately following the write to the communications register is interpreted as a read operation.

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
W/$\overline{R}$	A6	A5	A4	A3	A2	A1	A0

Table 17. ADE7758 Register List

Address [A6:A0]	Name	R/W[1]	Length	Type[2]	Default Value	Description
0x00	Reserved	–				Reserved.
0x01	AWATTHR	R	16	S	0	Watt-Hour Accumulation Register for Phase A. Active power is accumulated over time in this read-only register. The AWATTHR register can hold a maximum of 0.52 seconds of active energy information with full-scale analog inputs before it overflows (see the Active Energy Calculation section). Bit 0 and Bit 1 of the COMPMODE register determine how the active energy is processed from the six analog inputs.
0x02	BWATTHR	R	16	S	0	Watt-Hour Accumulation Register for Phase B.
0x03	CWATTHR	R	16	S	0	Watt-Hour Accumulation Register for Phase C.
0x04	AVARHR	R	16	S	0	VAR-Hour Accumulation Register for Phase A. Reactive power is accumulated over time in this read-only register. The AVARHR register can hold a maximum of 0.52 seconds of reactive energy information with full-scale analog inputs before it overflows (see the Reactive Energy Calculation section). Bit 0 and Bit 1 of the COMPMODE register determine how the reactive energy is processed from the six analog inputs.
0x05	BVARHR	R	16	S	0	VAR-Hour Accumulation Register for Phase B.
0x06	CVARHR	R	16	S	0	VAR-Hour Accumulation Register for Phase C.
0x07	AVAHR	R	16	S	0	VA-Hour Accumulation Register for Phase A. Apparent power is accumulated over time in this read-only register. The AVAHR register can hold a maximum of 1.15 seconds of apparent energy information with full-scale analog inputs before it overflows (see the Apparent Energy Calculation section). Bit 0 and Bit 1 of the COMPMODE register determine how the apparent energy is processed from the six analog inputs.
0x08	BVAHR	R	16	S	0	VA-Hour Accumulation Register for Phase B.
0x09	CVAHR	R	16	S	0	VA-Hour Accumulation Register for Phase C.
0x0A	AIRMS	R	24	S	0	Phase A Current Channel RMS Register. The register contains the rms component of the Phase A input of the current channel. The source is selected by data bits in the mode register.
0x0B	BIRMS	R	24	S	0	Phase B Current Channel RMS Register.
0x0C	CIRMS	R	24	S	0	Phase C Current Channel RMS Register.
0x0D	AVRMS	R	24	S	0	Phase A Voltage Channel RMS Register.

Address [A6:A0]	Name	R/W[1]	Length	Type[2]	Default Value	Description
0x0E	BVRMS	R	24	S	0	Phase B Voltage Channel RMS Register.
0x0F	CVRMS	R	24	S	0	Phase C Voltage Channel RMS Register.
0x10	FREQ	R	12	U	0	Frequency of the Line Input Estimated by the Zero-Crossing Processing. It can also display the period of the line input. Bit 7 of the LCYCMODE register determines if the reading is frequency or period. Default is frequency. Data Bit 0 and Bit 1 of the MMODE register determine the voltage channel used for the frequency or period calculation.
0x11	TEMP	R	8	S	0	Temperature Register. This register contains the result of the latest temperature conversion. Refer to the Temperature Measurement section for details on how to interpret the content of this register.
0x12	WFORM	R	24	S	0	Waveform Register. This register contains the digitized waveform of one of the six analog inputs or the digitized power waveform. The source is selected by Data Bit 0 to Bit 4 in the WAVMODE register.
0x13	OPMODE	R/W	8	U	4	Operational Mode Register. This register defines the general configuration of the ADE7758 (see Table 18).
0x14	MMODE	R/W	8	U	0xFC	Measurement Mode Register. This register defines the channel used for period and peak detection measurements (see Table 19).
0x15	WAVMODE	R/W	8	U	0	Waveform Mode Register. This register defines the channel and sampling frequency used in the waveform sampling mode (see Table 20).
0x16	COMPMODE	R/W	8	U	0x1C	Computation Mode Register. This register configures the formula applied for the energy and line active energy measurements (see Table 22).
0x17	LCYCMODE	R/W	8	U	0x78	Line Cycle Mode Register. This register configures the line cycle accumulation mode for WATT-HR, VAR-HR, and VA-Hr (see Table 23).
0x18	Mask	R/W	24	U	0	$\overline{\text{IRQ}}$ Mask Register. It determines if an interrupt event generates an active-low output at the $\overline{\text{IRQ}}$ pin (see the Interrupts section).
0x19	Status	R	24	U	0	$\overline{\text{IRQ}}$ Status Register. This register contains information regarding the source of the ADE7758 interrupts (see the Interrupts section).
0x1A	RSTATUS	R	24	U	0	$\overline{\text{IRQ}}$ Reset Status Register. Same as the STATUS register, except that its contents are reset to 0 (all flags cleared) after a read operation.
0x1B	ZXTOUT	R/W	16	U	0xFFFF	Zero-Cross Timeout Register. If no zero crossing is detected within the time period specified by this register, the interrupt request line ($\overline{\text{IRQ}}$) goes active low for the corresponding line voltage. The maximum timeout period is 2.3 seconds (see the Zero-Crossing Detection section).
0x1C	LINECYC	R/W	16	U	0xFFFF	Line Cycle Register. The content of this register sets the number of half-line cycles that the active, reactive, and apparent energies are accumulated for in the line accumulation mode.
0x1D	SAGCYC	R/W	8	U	0xFF	SAG Line Cycle Register. This register specifies the number of consecutive half-line cycles where voltage channel input may fall below a threshold level. This register is common to the three line voltage SAG detection. The detection threshold is specified by the SAGLVL register (see the Line Voltage SAG Detection section).
0x1E	SAGLVL	R/W	8	U	0	SAG Voltage Level. This register specifies the detection threshold for the SAG event. This register is common to all three phases' line voltage SAG detections. See the description of the SAGCYC register for details.
0x1F	VPINTLVL	R/W	8	U	0xFF	Voltage Peak Level Interrupt Threshold Register. This register sets the level of the voltage peak detection. Bit 5 to Bit 7 of the MMODE register determine which phases are to be monitored. If the selected voltage phase exceeds this level, the PKV flag in the $\overline{\text{IRQ}}$ status register is set.
0x20	IPINTLVL	R/W	8	U	0xFF	Current Peak Level Interrupt Threshold Register. This register sets the level of the current peak detection. Bit 5 to Bit 7 of the MMODE register determine which phases are to be monitored. If the selected current phase exceeds this level, the PKI flag in the $\overline{\text{IRQ}}$ status register is set.
0x21	VPEAK	R	8	U	0	Voltage Peak Register. This register contains the value of the peak voltage waveform that has occurred within a fixed number of half-line cycles. The number of half-line cycles is set by the LINECYC register.

Address [A6:A0]	Name	R/W[1]	Length	Type[2]	Default Value	Description
0x22	IPEAK	R	8	U	0	Current Peak Register. This register holds the value of the peak current waveform that has occurred within a fixed number of half-line cycles. The number of half-line cycles is set by the LINECYC register.
0x23	Gain	R/W	8	U	0	PGA Gain Register. This register is used to adjust the gain selection for the PGA in the current and voltage channels (see the Analog Inputs section).
0x24	AVRMSGAIN	R/W	12	S	0	Phase A VRMS Gain Register. The range of the voltage rms calculation can be adjusted by writing to this register. It has an adjustment range of ±50% with a resolution of 0.0244%/LSB.
0x25	BVRMSGAIN	R/W	12	S	0	Phase B VRMS Gain Register.
0x26	CVRMSGAIN	R/W	12	S	0	Phase C VRMS Gain Register.
0x27	AIGAIN	R/W	12	S	0	Phase A Current Gain Register. This register is not recommended to be used and it should be kept at 0, its default value.
0x28	BIGAIN	R/W	12	S	0	Phase B Current Gain Register. This register is not recommended to be used and it should be kept at 0, its default value.
0x29	CIGAIN	R/W	12	S	0	Phase C Current Gain Register. This register is not recommended to be used and it should be kept at 0, its default value.
0x2A	AWG	R/W	12	S	0	Phase A Watt Gain Register. The range of the watt calculation can be adjusted by writing to this register. It has an adjustment range of ±50% with a resolution of 0.0244%/LSB.
0x2B	BWG	R/W	12	S	0	Phase B Watt Gain Register.
0x2C	CWG	R/W	12	S	0	Phase C Watt Gain Register.
0x2D	AVARG	R/W	12	S	0	Phase A VAR Gain Register. The range of the VAR calculation can be adjusted by writing to this register. It has an adjustment range of ±50% with a resolution of 0.0244%/LSB.
0x2E	BVARG	R/W	12	S	0	Phase B VAR Gain Register.
0x2F	CVARG	R/W	12	S	0	Phase C VAR Gain Register.
0x30	AVAG	R/W	12	S	0	Phase A VA Gain Register. The range of the VA calculation can be adjusted by writing to this register. It has an adjustment range of ±50% with a resolution of 0.0244%/LSB.
0x31	BVAG	R/W	12	S	0	Phase B VA Gain Register.
0x32	CVAG	R/W	12	S	0	Phase C VA Gain Register.
0x33	AVRMSOS	R/W	12	S	0	Phase A Voltage RMS Offset Correction Register.
0x34	BVRMSOS	R/W	12	S	0	Phase B Voltage RMS Offset Correction Register.
0x35	CVRMSOS	R/W	12	S	0	Phase C Voltage RMS Offset Correction Register.
0x36	AIRMSOS	R/W	12	S	0	Phase A Current RMS Offset Correction Register.
0x37	BIRMSOS	R/W	12	S	0	Phase B Current RMS Offset Correction Register.
0x38	CIRMSOS	R/W	12	S	0	Phase C Current RMS Offset Correction Register.
0x39	AWATTOS	R/W	12	S	0	Phase A Watt Offset Calibration Register.
0x3A	BWATTOS	R/W	12	S	0	Phase B Watt Offset Calibration Register.
0x3B	CWATTOS	R/W	12	S	0	Phase C Watt Offset Calibration Register.
0x3C	AVAROS	R/W	12	S	0	Phase A VAR Offset Calibration Register.
0x3D	BVAROS	R/W	12	S	0	Phase B VAR Offset Calibration Register.
0x3E	CVAROS	R/W	12	S	0	Phase C VAR Offset Calibration Register.
0x3F	APHCAL	R/W	7	S	0	Phase A Phase Calibration Register. The phase relationship between the current and voltage channel can be adjusted by writing to this signed 7-bit register (see the Phase Compensation section).
0x40	BPHCAL	R/W	7	S	0	Phase B Phase Calibration Register.
0x41	CPHCAL	R/W	7	S	0	Phase C Phase Calibration Register.
0x42	WDIV	R/W	8	U	0	Active Energy Register Divider.
0x43	VARDIV	R/W	8	U	0	Reactive Energy Register Divider.
0x44	VADIV	R/W	8	U	0	Apparent Energy Register Divider.

Address [A6:A0]	Name	R/W[1]	Length	Type[2]	Default Value	Description
0x45	APCFNUM	R/W	16	U	0	Active Power CF Scaling Numerator Register. The content of thisregister is used in the numerator of the APCF output scaling calculation. Bits [15:13] indicate reverse polarity active power measurement for Phase A, Phase B, and Phase C in order; that is, Bit 15 is Phase A, Bit 14 is Phase B, and so on.
0x46	APCFDEN	R/W	12	U	0x3F	Active Power CF Scaling Denominator Register. The content of this register is used in the denominator of the APCF output scaling.
0x47	VARCFNUM	R/W	16	U	0	Reactive Power CF Scaling Numerator Register. The content of this register is used in the numerator of the VARCF output scaling. Bits [15:13] indicate reverse polarity reactive power measurement for Phase A, Phase B, and Phase C in order; that is, Bit 15 is Phase A, Bit 14 is Phase B, and so on.
0x48	VARCFDEN	R/W	12	U	0x3F	Reactive Power CF Scaling Denominator Register. The content of this register is used in the denominator of the VARCF output scaling.
0x49 to 0x7D	Reserved	–	–	–	–	Reserved.
0x7E	CHKSUM	R	8	U	–	Checksum Register. The content of this register represents the sum of all the ones in the last register read from the SPI port.
0x7F	Version	R	8	U	–	Version of the Die.

[1] This column specifies the read/write capability of the register. R = Read only register. R/W = Register that can be both read and written.
[2] Type decoder: U = unsigned; S = signed.

OPERATIONAL MODE REGISTER (0x13)

The general configuration of the ADE7758 is defined by writing to the OPMODE register. Table 18 summarizes the functionality of each bit in the OPMODE register.

Table 18. OPMODE Register

Bit Location	Bit Mnemonic	Default Value	Description
0	DISHPF	0	The HPFs in all current channel inputs are disabled when this bit is set.
1	DISLPF	0	The LPFs after the watt and VAR multipliers are disabled when this bit is set.
2	DISCF	1	The frequency outputs APCF and VARCF are disabled when this bit is set.
3 to 5	DISMOD	0	By setting these bits, the ADE7758 ADCs can be turned off. In normal operation, these bits should be left at Logic 0.

DISMOD[2:0]			Description
0	0	0	Normal operation.
1	0	0	Redirect the voltage inputs to the signal paths for the current channels and the current inputs to the signal paths for the voltage channels.
0	0	1	Switch off only the current channel ADCs.
1	0	1	Switch off current channel ADCs and redirect the current input signals to the voltage channel signal paths.
0	1	0	Switch off only the voltage channel ADCs.
1	1	0	Switch off voltage channel ADCs and redirect the voltage input signals to the current channel signal paths.
0	1	1	Put the ADE7758 in sleep mode.
1	1	1	Put the ADE7758 in power-down mode (reduces AI_{DD} to 1 mA typ).

6	SWRST	0	Software Chip Reset. A data transfer to the ADE7758 should not take place for at least 166 µs after a software reset.
7	Reserved	0	This should be left at 0.

MEASUREMENT MODE REGISTER (0x14)

The configuration of the PERIOD and peak measurements made by the ADE7758 is defined by writing to the MMODE register. Table 19 summarizes the functionality of each bit in the MMODE register.

Table 19. MMODE Register

Bit Location	Bit Mnemonic	Default Value	Description
0 to 1	FREQSEL	0	These bits are used to select the source of the measurement of the voltage line frequency.

FREQSEL1	FREQSEL0	Source
0	0	Phase A
0	1	Phase B
1	0	Phase C
1	1	Reserved

2 to 4	PEAKSEL	7	These bits select the phases used for the voltage and current peak registers. Setting Bit 2 switches the IPEAK and VPEAK registers to hold the absolute values of the largest current and voltage waveform (over a fixed number of half-line cycles) from Phase A. The number of half-line cycles is determined by the content of the LINECYC register. At the end of the LINECYC number of half-line cycles, the content of the registers is replaced with the new peak values. Similarly, setting Bit 3 turns on the peak detection for Phase B, and Bit 4 for Phase C. Note that if more than one bit is set, the VPEAK and IPEAK registers can hold values from two different phases, that is, the voltage and current peak are independently processed (see the Peak Current Detection section).
5 to 7	PKIRQSEL	7	These bits select the phases used for the peak interrupt detection. Setting Bit 5 switches on the monitoring of the absolute current and voltage waveform to Phase A. Similarly, setting Bit 6 turns on the waveform detection for Phase B, and Bit 7 for Phase C. Note that more than one bit can be set for detection on multiple phases. If the absolute values of the voltage or current waveform samples in the selected phases exceeds the preset level specified in the VPINTLVL or IPINTLVL registers the corresponding bit(s) in the STATUS registers are set (see the Peak Current Detection section).

WAVEFORM MODE REGISTER (0x15)

The waveform sampling mode of the ADE7758 is defined by writing to the WAVMODE register. Table 20 summarizes the functionality of each bit in the WAVMODE register.

Table 20. WAVMODE Register

Bit Location	Bit Mnemonic	Default Value	Description
0 to 1	PHSEL	0	These bits are used to select the phase of the waveform sample.
			PHSEL[1:0] / **Source**
			0 0 / Phase A
			0 1 / Phase B
			1 0 / Phase C
			1 1 / Reserved
2 to 4	WAVSEL	0	These bits are used to select the type of waveform.
			WAVSEL[2:0] / **Source**
			0 0 0 / Current
			0 0 1 / Voltage
			0 1 0 / Active Power Multiplier Output
			0 1 1 / Reactive Power Multiplier Output
			1 0 0 / VA Multiplier Output
			Others- / Reserved
5 to 6	DTRT	0	These bits are used to select the data rate.
			DTRT[1:0] / **Update Rate**
			0 0 / 26.04 kSPS (CLKIN/3/128)
			0 1 / 13.02 kSPS (CLKIN/3/256)
			1 0 / 6.51 kSPS (CLKIN/3/512)
			1 1 / 3.25 kSPS (CLKIN/3/1024)
7	VACF	0	Setting this bit to Logic 1 switches the VARCF output pin to an output frequency that is proportional to the total apparent power (VA). In the default state, Logic 0, the VARCF pin outputs a frequency proportional to the total reactive power (VAR).

COMPUTATIONAL MODE REGISTER (0x16)

The computational method of the ADE7758 is defined by writing to the COMPMODE register. Table 21 summarizes the functionality of each bit in the COMPMODE register.

Table 21. COMPMODE Register

Bit Location	Bit Mnemonic	Default Value	Description
0 to 1	CONSEL	0	These bits are used to select the input to the energy accumulation registers. CONSEL[1:0] = 11 is reserved. $\overline{IA}$, $\overline{IB}$, and $\overline{IC}$ are IA, IB, and IC phase shifted by −90°, respectively.
			Registers / **CONSEL[1, 0] = 00** / **CONSEL[1, 0] = 01** / **CONSEL[1, 0] = 10**
			AWATTHR / VA × IA / VA × (IA − IB) / VA × (IA–IB)
			BWATTHR / VB × IB / 0 / 0
			CWATTHR / VC × IC / VC × (IC − IB) / VC × IC
			AVARHR / VA × $\overline{IA}$ / VA × ($\overline{IA}$ − $\overline{IB}$) / VA × (IA–IB)
			BVARHR / VB × $\overline{IB}$ / 0 / 0
			CVARHR / VC × $\overline{IC}$ / VC × ($\overline{IC}$ − $\overline{IB}$) / VC × IC
			AVAHR / $VA_{RMS} \times IA_{RMS}$ / $VA_{RMS} \times IA_{RMS}$ / $VA_{RMS} \times A_{RMS}$
			BVAHR / $VB_{RMS} \times IB_{RMS}$ / $(VA_{RMS} + VC_{RMS})/2 \times IB_{RMS}$ / $VA_{RMS} \times IB_{RMS}$
			CVAHR / $VC_{RMS} \times IC_{RMS}$ / $VC_{RMS} \times IC_{RMS}$ / $VC_{RMS} \times IC_{RMS}$
2 to 4	TERMSEL	7	These bits are used to select the phases to be included in the APCF and VARCF pulse outputs. Setting Bit 2 selects Phase A (the inputs to AWATTHR and AVARHR registers) to be included. Bit 3 and Bit 4 are for Phase B and Phase C, respectively. Setting all three bits enables the sum of all three phases to be included in the frequency outputs (see the Active Power Frequency Output and the Reactive Power Frequency Output sections).
5	ABS	0	Setting this bit places the APCF output pin in absolute only mode. Namely, the APCF output frequency is proportional to the sum of the absolute values of the watt-hour accumulation registers (AWATTHR, BWATTHR, and CWATTHR). Note that this bit only affects the APCF pin and has no effect on the content of the corresponding registers.
6	SAVAR	0	Setting this bit places the VARCF output pin in the signed adjusted mode. Namely, the VARCF output frequency is proportional to the sign-adjusted sum of the VAR-hour accumulation registers (AVARHR, BVARHR, and CVARHR). The sign of the VAR is determined from the sign of the watt calculation from the corresponding phase, that is, the sign of the VAR is flipped if the sign of the watt is negative, and if the watt is positive, there is no change to the sign of the VAR. Note that this bit only affects the VARCF pin and has no effect on the content of the corresponding registers.
7	NOLOAD	0	Setting this bit activates the no-load threshold in the ADE7758.

LINE CYCLE ACCUMULATION MODE REGISTER (0x17)

The functionalities involved the line-cycle accumulation mode in the ADE7758 are defined by writing to the LCYCMODE register. Table 22 summarizes the functionality of each bit in the LCYCMODE register.

Table 22. LCYCMODE Register

Bit Location	Bit Mnemonic	Default Value	Description
0	LWATT	0	Setting this bit places the watt-hour accumulation registers (AWATTHR, BWATTHR, and CWATTHR registers) into line-cycle accumulation mode.
1	LVAR	0	Setting this bit places the VAR-hour accumulation registers (AVARHR, BVARHR, and CVARHR registers) into line-cycle accumulation mode.
2	LVA	0	Setting this bit places the VA-hour accumulation registers (AVAHR, BVAHR, and CVAHR registers) into line-cycle accumulation mode.
3 to 5	ZXSEL	7	These bits select the phases used for counting the number of zero crossings in the line-cycle accumulation mode. Bit 3, Bit 4, and Bit 5 select Phase A, Phase B, and Phase C, respectively. More than one phase can be selected for the zero-crossing detection, and the accumulation time is shortened accordingly.
6	RSTREAD	1	Setting this bit enables the read-with-reset for all the WATTHR, VARHR, and VAHR registers for all three phases, that is, a read to those registers resets the registers to 0 after the content of the registers have been read. This bit should be set to Logic 0 when the LWATT, LVAR, or LVA bits are set to Logic 1.
7	FREQSEL	0	Setting this bit causes the FREQ (0x10) register to display the period, instead of the frequency of the line input.

INTERRUPT MASK REGISTER (0x18)

When an interrupt event occurs in the ADE7758, the $\overline{\text{IRQ}}$ logic output goes active low if the mask bit for this event is Logic 1 in the MASK register. The $\overline{\text{IRQ}}$ logic output is reset to its default collector open state when the RSTATUS register is read. Table 23 describes the function of each bit in the interrupt mask register.

Table 23. Function of Each Bit in the Interrupt Mask Register

Bit Location	Interrupt Flag	Default Value	Description
0	AEHF	0	Enables an interrupt when there is a change in Bit 14 of any one of the three WATTHR registers, that is, the WATTHR register is half full.
1	REHF	0	Enables an interrupt when there is a change in Bit 14 of any one of the three VARHR registers, that is, the VARHR register is half full.
2	VAEHF	0	Enables an interrupt when there is a 0 to 1 transition in the MSB of any one of the three VAHR registers, that is, the VAHR register is half full.
3	SAGA	0	Enables an interrupt when there is a SAG on the line voltage of the Phase A.
4	SAGB	0	Enables an interrupt when there is a SAG on the line voltage of the Phase B.
5	SAGC	0	Enables an interrupt when there is a SAG on the line voltage of the Phase C.
6	ZXTOA	0	Enables an interrupt when there is a zero-crossing timeout detection on Phase A.
7	ZXTOB	0	Enables an interrupt when there is a zero-crossing timeout detection on Phase B.
8	ZXTOC	0	Enables an interrupt when there is a zero-crossing timeout detection on Phase C.
9	ZXA	0	Enables an interrupt when there is a zero crossing in the voltage channel of Phase A (see the Zero-Crossing Detection section).
10	ZXB	0	Enables an interrupt when there is a zero crossing in the voltage channel of Phase B (see the Zero-Crossing Detection section).
11	ZXC	0	Enables an interrupt when there is a zero crossing in the voltage channel of Phase C (see the Zero-Crossing Detection section).
12	LENERGY	0	Enables an interrupt when the energy accumulations over LINECYC are finished.
13	Reserved	0	Reserved.
14	PKV	0	Enables an interrupt when the voltage input selected in the MMODE register is above the value in the VPINTLVL register.
15	PKI	0	Enables an interrupt when the current input selected in the MMODE register is above the value in the IPINTLVL register.
16	WFSM	0	Enables an interrupt when data is present in the WAVEMODE register.
17	REVPAP	0	Enables an interrupt when there is a sign change in the watt calculation among any one of the phases specified by the TERMSEL bits in the COMPMODE register.
18	REVPRP	0	Enables an interrupt when there is a sign change in the VAR calculation among any one of the phases specified by the TERMSEL bits in the COMPMODE register.
19	SEQERR	0	Enables an interrupt when the zero crossing from Phase A is followed not by the zero crossing of Phase C but with that of Phase B.

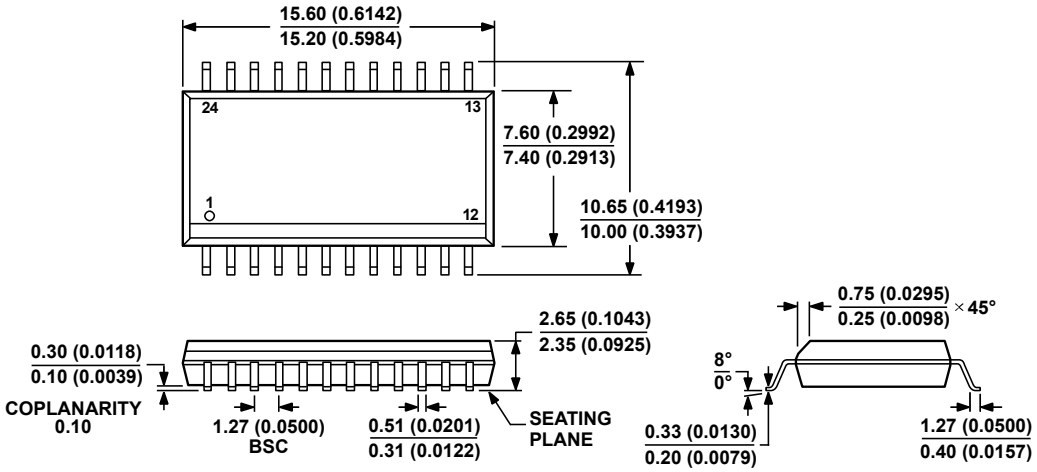
INTERRUPT STATUS REGISTER (0x19)/RESET INTERRUPT STATUS REGISTER (0x1A)

The interrupt status register is used to determine the source of an interrupt event. When an interrupt event occurs in the ADE7758, the corresponding flag in the interrupt status register is set. The $\overline{\text{IRQ}}$ pin goes active low if the corresponding bit in the interrupt mask register is set. When the MCU services the interrupt, it must first carry out a read from the interrupt status register to determine the source of the interrupt. All the interrupts in the interrupt status register stay at their logic high state after an event occurs. The state of the interrupt bit in the interrupt status register is reset to its default value once the reset interrupt status register is read.

Table 24. Interrupt Status Register

Bit Location	Interrupt Flag	Default Value	Event Description
0	AEHF	0	Indicates that an interrupt was caused by a change in Bit 14 among any one of the three WATTHR registers, that is, the WATTHR register is half full.
1	REHF	0	Indicates that an interrupt was caused by a change in Bit 14 among any one of the three VARHR registers, that is, the VARHR register is half full.
2	VAEHF	0	Indicates that an interrupt was caused by a 0 to 1 transition in Bit 15 among any one of the three VAHR registers, that is, the VAHR register is half full.
3	SAGA	0	Indicates that an interrupt was caused by a SAG on the line voltage of the Phase A.
4	SAGB	0	Indicates that an interrupt was caused by a SAG on the line voltage of the Phase B.
5	SAGC	0	Indicates that an interrupt was caused by a SAG on the line voltage of the Phase C.
6	ZXTOA	0	Indicates that an interrupt was caused by a missing zero crossing on the line voltage of the Phase A.
7	ZXTOB	0	Indicates that an interrupt was caused by a missing zero crossing on the line voltage of the Phase B.
8	ZXTOC	0	Indicates that an interrupt was caused by a missing zero crossing on the line voltage of the Phase C.
9	ZXA	0	Indicates a detection of a rising edge zero crossing in the voltage channel of Phase A.
10	ZXB	0	Indicates a detection of a rising edge zero crossing in the voltage channel of Phase B.
11	ZXC	0	Indicates a detection of a rising edge zero crossing in the voltage channel of Phase C.
12	LENERGY	0	In line energy accumulation, indicates the end of an integration over an integer number of half-line cycles (LINECYC). See the Calibration section.
13	Reset	1	After Bit 6 (SWRST) in OPMODE register is set to 1, the ADE7758 enters software reset. This bit becomes 1 after 166 µsec, indicating the reset process has ended and the registers are set to their default values. It stays 1 until the reset interrupt status register is read and then becomes 0.
14	PKV	0	Indicates that an interrupt was caused when the selected voltage input is above the value in the VPINTLVL register.
15	PKI	0	Indicates that an interrupt was caused when the selected current input is above the value in the IPINTLVL register.
16	WFSM	0	Indicates that new data is present in the waveform register.
17	REVPAP	0	Indicates that an interrupt was caused by a sign change in the watt calculation among any one of the phases specified by the TERMSEL bits in the COMPMODE register.
18	REVPRP	0	Indicates that an interrupt was caused by a sign change in the VAR calculation among any one of the phases specified by the TERMSEL bits in the COMPMODE register.
19	SEQERR	0	Indicates that an interrupt was caused by a zero crossing from Phase A followed not by the zero crossing of Phase C but by that of Phase B.

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MS-013-AD
CONTROLLING DIMENSIONS ARE IN MILLIMETERS; INCH DIMENSIONS
(IN PARENTHESES) ARE ROUNDED-OFF MILLIMETER EQUIVALENTS FOR
REFERENCE ONLY AND ARE NOT APPROPRIATE FOR USE IN DESIGN.

12-09-2010-A

Figure 95. 24-Lead Standard Small Outline Package [SOIC_W]
Wide Body (RW-24)
Dimensions shown in millimeters and (inches)

ORDERING GUIDE

Model[1]	Temperature Range	Package Description	Package Option
ADE7758ARWZ	−40°C to + 85°C	24-Lead Wide Body SOIC_W	RW-24
ADE7758ARWZRL	−40°C to + 85°C	24-Lead Wide Body SOIC_W	RW-24
EVAL-ADE7758ZEB		Evaluation Board	

[1] Z = RoHS Compliant Part.

NOTES

NOTES

D04443-0-10/11(E)

Introduction to SPI Interface

By **Piyu Dhaker**

Share on

Serial peripheral interface (SPI) is one of the most widely used interfaces between microcontroller and peripheral ICs such as sensors, ADCs, DACs, shift registers, SRAM, and others. This article provides a brief description of the SPI interface followed by an introduction to Analog Devices' SPI enabled switches and muxes, and how they help reduce the number of digital GPIOs in system board design.

SPI is a synchronous, full duplex master-slave-based interface. The data from the master or the slave is synchronized on the rising or falling clock edge. Both master and slave can transmit data at the same time. The SPI interface can be either 3-wire or 4-wire. This article focuses on the popular 4-wire SPI interface.

Interface

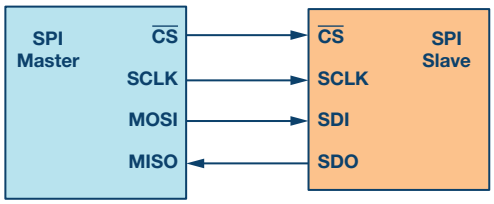


Figure 1. SPI configuration with master and a slave.

4-wire SPI devices have four signals:

- Clock (SPI CLK, SCLK)
- Chip select (CS)
- Master out, slave in (MOSI)
- Master in, slave out (MISO)

The device that generates the clock signal is called the master. Data transmitted between the master and the slave is synchronized to the clock generated by the master. SPI devices support much higher clock frequencies compared to I²C interfaces. Users should consult the product data sheet for the clock frequency specification of the SPI interface.

SPI interfaces can have only one master and can have one or multiple slaves. Figure 1 shows the SPI connection between the master and the slave.

The chip select signal from the master is used to select the slave. This is normally an active low signal and is pulled high to disconnect the slave from the SPI bus. When multiple slaves are used, an individual chip select signal for each slave is required from the master. In this article, the chip select signal is always an active low signal.

MOSI and MISO are the data lines. MOSI transmits data from the master to the slave and MISO transmits data from the slave to the master.

Data Transmission

To begin SPI communication, the master must send the clock signal and select the slave by enabling the CS signal. Usually chip select is an active low signal; hence, the master must send a logic 0 on this signal to select the slave. SPI is a full-duplex interface; both master and slave can send data at the same time via the MOSI and MISO lines respectively. During SPI communication, the data is simultaneously transmitted (shifted out serially onto the MOSI/SDO bus) and received (the data on the bus (MISO/SDI) is sampled or read in). The serial clock edge synchronizes the shifting and sampling of the data. The SPI interface provides the user with flexibility to select the rising or falling edge of the clock to sample and/or shift the data. Please refer to the device data sheet to determine the number of data bits transmitted using the SPI interface.

Clock Polarity and Clock Phase

In SPI, the master can select the clock polarity and clock phase. The CPOL bit sets the polarity of the clock signal during the idle state. The idle state is defined as the period when CS is high and transitioning to low at the start of the transmission and when CS is low and transitioning to high at the end of the transmission. The CPHA bit selects the clock phase. Depending on the CPHA bit, the rising or falling clock edge is used to sample and/or shift the data. The master must select the clock polarity and clock phase, as per the requirement of the slave. Depending on the CPOL and CPHA bit selection, four SPI modes are available. Table 1 shows the four SPI modes.

Table 1. SPI Modes with CPOL and CPHA

SPI Mode	CPOL	CPHA	Clock Polarity in Idle State	Clock Phase Used to Sample and/or Shift the Data
0	0	0	Logic low	Data sampled on rising edge and shifted out on the falling edge
1	0	1	Logic low	Data sampled on the falling edge and shifted out on the rising edge
2	1	1	Logic high	Data sampled on the falling edge and shifted out on the rising edge
3	1	0	Logic high	Data sampled on the rising edge and shifted out on the falling edge

Figure 2 through Figure 5 show an example of communication in four SPI modes. In these examples, the data is shown on the MOSI and MISO line. The start and end of transmission is indicated by the dotted green line, the sampling edge is indicated in orange, and the shifting edge is indicated in blue. Please note these figures are for illustration purpose only. For successful SPI communications, users must refer to the product data sheet and ensure that the timing specifications for the part are met.

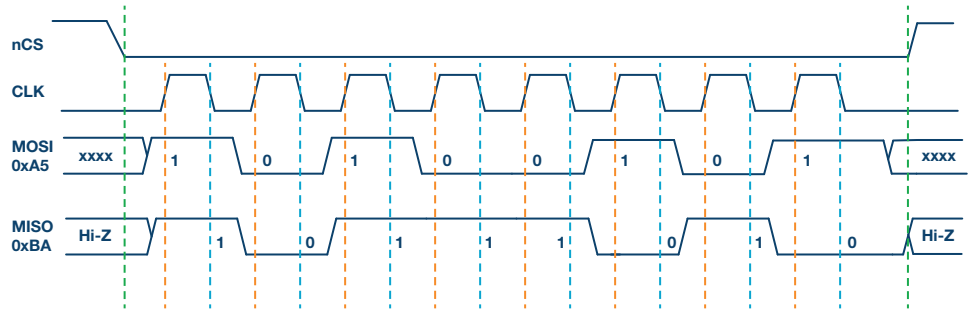


Figure 2. SPI Mode 0, CPOL = 0, CPHA = 0: CLK idle state = low, data sampled on rising edge and shifted on falling edge.

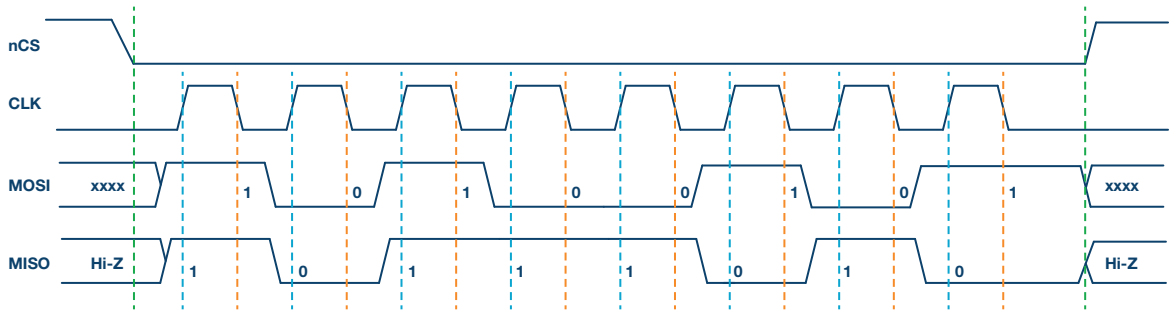


Figure 3. SPI Mode 1, CPOL = 0, CPHA = 1: CLK idle state = low, data sampled on the falling edge and shifted on the rising edge.

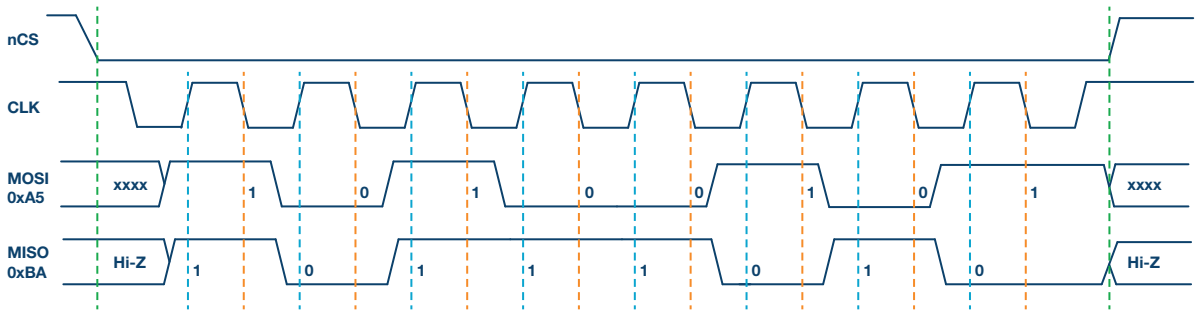


Figure 4. SPI Mode 2, CPOL = 1, CPHA = 1: CLK idle state = high, data sampled on the falling edge and shifted on the rising edge.

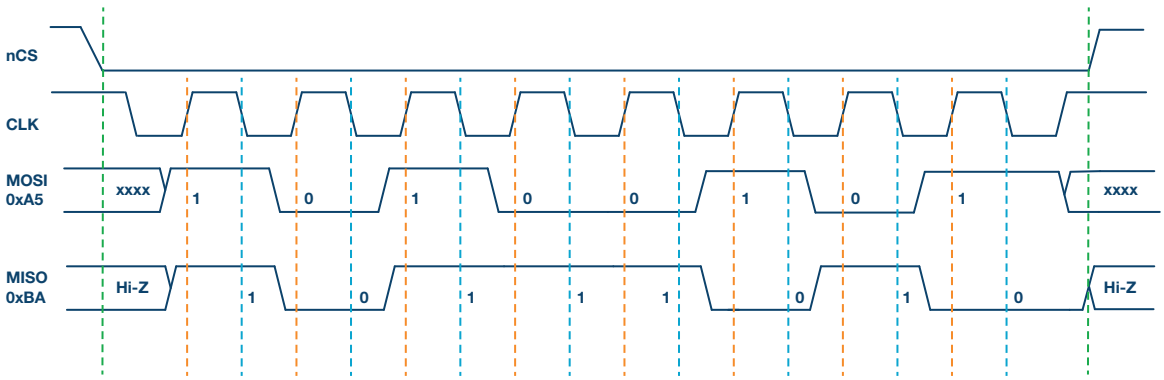


Figure 5. SPI Mode 3, CPOL = 1, CPHA = 0: CLK idle state = high, data sampled on the rising edge and shifted on the falling edge.

Figure 3 shows the timing diagram for SPI Mode 1. In this mode, clock polarity is 0, which indicates that the idle state of the clock signal is low. The clock phase in this mode is 1, which indicates that the data is sampled on the falling edge (shown by the orange dotted line) and the data is shifted on the rising edge (shown by the dotted blue line) of the clock signal.

Figure 4 shows the timing diagram for SPI Mode 2. In this mode, the clock polarity is 1, which indicates that the idle state of the clock signal is high. The clock phase in this mode is 1, which indicates that the data is sampled on the falling edge (shown by the orange dotted line) and the data is shifted on the rising edge (shown by the dotted blue line) of the clock signal.

Figure 5 shows the timing diagram for SPI Mode 3. In this mode, the clock polarity is 1, which indicates that the idle state of the clock signal is high. The clock phase in this mode is 0, which indicates that the data is sampled on the rising edge (shown by the orange dotted line) and the data is shifted on the falling edge (shown by the dotted blue line) of the clock signal.

Multislave Configuration

Multiple slaves can be used with a single SPI master. The slaves can be connected in regular mode or daisy-chain mode.

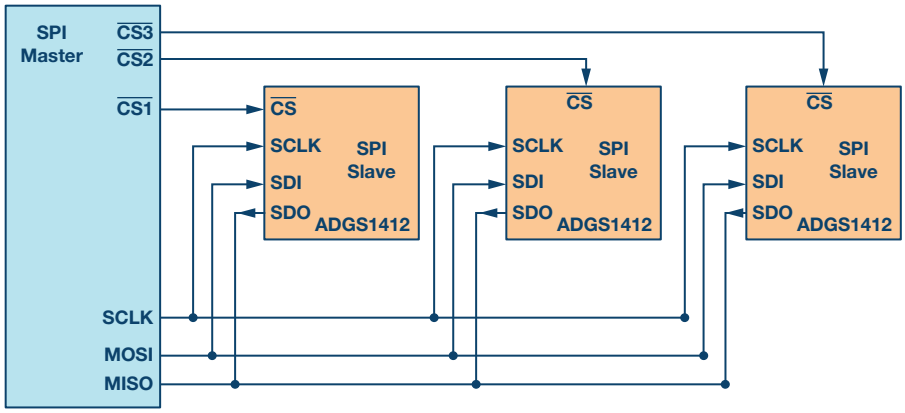


Figure 6. Multislave SPI configuration.

Regular SPI Mode:

In regular mode, an individual chip select for each slave is required from the master. Once the chip select signal is enabled (pulled low) by the master, the clock and data on the MOSI/MISO lines are available for the selected slave. If multiple chip select signals are enabled, the data on the MISO line is corrupted, as there is no way for the master to identify which slave is transmitting the data.

As can be seen from Figure 6, as the number of slaves increases, the number of chip select lines from the master increases. This can quickly add to the number of inputs and outputs needed from the master and limit the number of slaves that can be used. There are different techniques that can be used to increase the number of slaves in regular mode; for example, using a mux to generate a chip select signal.

Daisy-Chain Method:

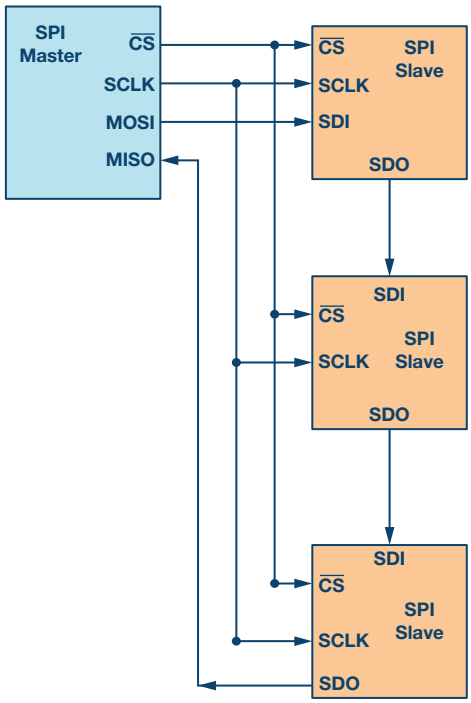


Figure 7. Multislave SPI daisy-chain configuration.

In daisy-chain mode, the slaves are configured such that the chip select signal for all slaves is tied together and data propagates from one slave to the next. In this configuration, all slaves receive the same SPI clock at the same time. The data from the master is directly connected to the first slave and that slave provides data to the next slave and so on.

In this method, as data is propagated from one slave to the next, the number of clock cycles required to transmit data is proportional to the slave position in the daisy chain. For example, in Figure 7, in an 8-bit system, 24 clock pulses are required for the data to be available on the 3rd slave, compared to only eight clock pulses in regular SPI mode. Figure 8 shows the clock cycles and data propagating through the daisy chain. Daisy-chain mode is not necessarily supported by all SPI devices. Please refer to the product data sheet to confirm if daisy chain is available.

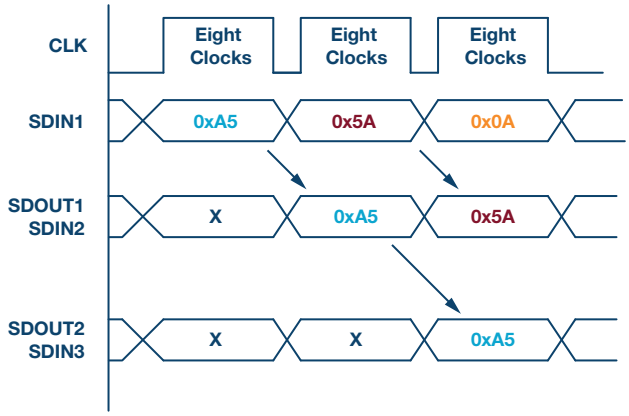


Figure 8. Daisy-chain configuration: data propagation.

Analog Devices SPI Enabled Switches and Muxes

The newest generation of ADI SPI enabled switches offer significant space saving without compromise to the precision switch performance. This section of the article discusses a case study of how SPI enabled switches or muxes can significantly simplify the system-level design and reduce the number of GPIOs required.

The ADG1412 is a quad, single-pole, single-throw (SPST) switch, which requires four GPIOs connected to the control input of each switch. Figure 9 shows the connection between the microcontroller and one ADG1412.

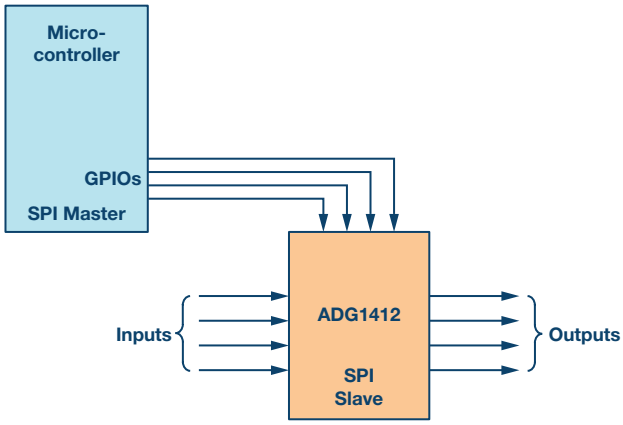


Figure 9. Microcontroller GPIO as control signals for the switch.

As the number of switches on the board increases, the number of required GPIOs increases significantly. For example, when designing a test instrumentation system and a large number of switches are used to increase the number of channels in the system. In a 4 × 4 cross-point matrix configuration, four ADG1412s are used. This system would require 16 GPIOs, limiting the available GPIOs in a standard microcontroller. Figure 10 shows the connection of four ADG1412s using the 16 GPIOs of the microcontroller.

One approach to reduce the number of GPIOs is to use a serial-to-parallel converter, as shown in Figure 11. This device outputs parallel signals that can be connected to the switch control inputs and the device can be configured by serial interface SPI. The drawback of this method is an increase in the bill of material by introducing an additional component.

An alternative method is to use SPI controlled switches. This method provides the benefit of reducing the number of GPIOs required and also eliminates the overhead of additional serial-to-parallel converter. As shown in Figure 12, instead of 16 microcontroller GPIOS, only seven microcontroller GPIOs are needed to provide the SPI signals to the four ADGS1412s.

The switches can be configured in daisy-chain configuration to further optimize the GPIO count. In daisy-chain configuration, irrespective of the number of switches used in the system, only four GPIOs are used from the master (microcontroller).

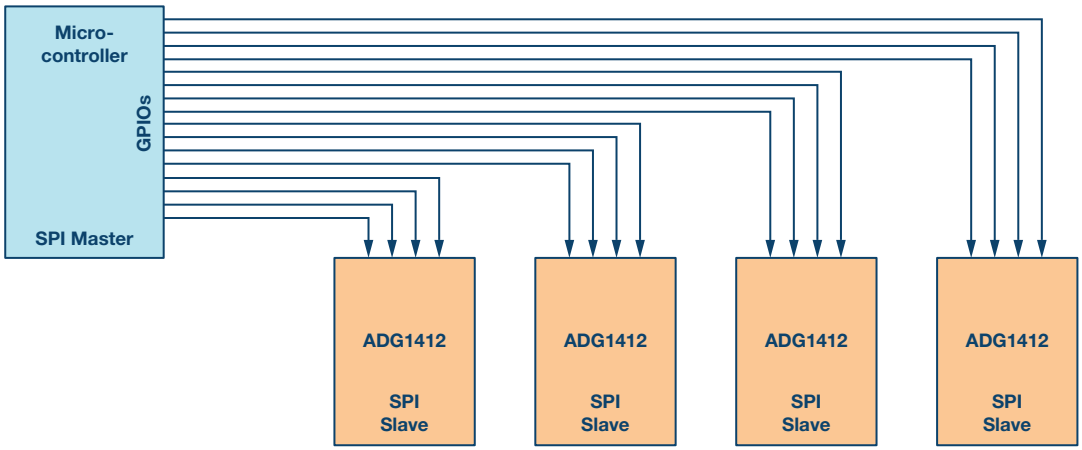


Figure 10. In a multislave configuration, the number of GPIOs needed increases tremendously.

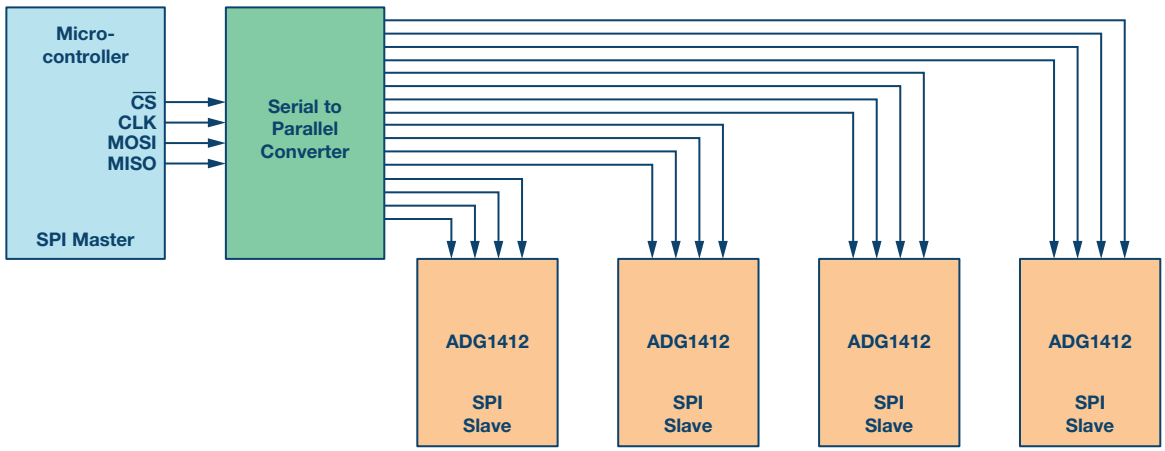


Figure 11. Multislave switches using a serial-to-parallel converter.

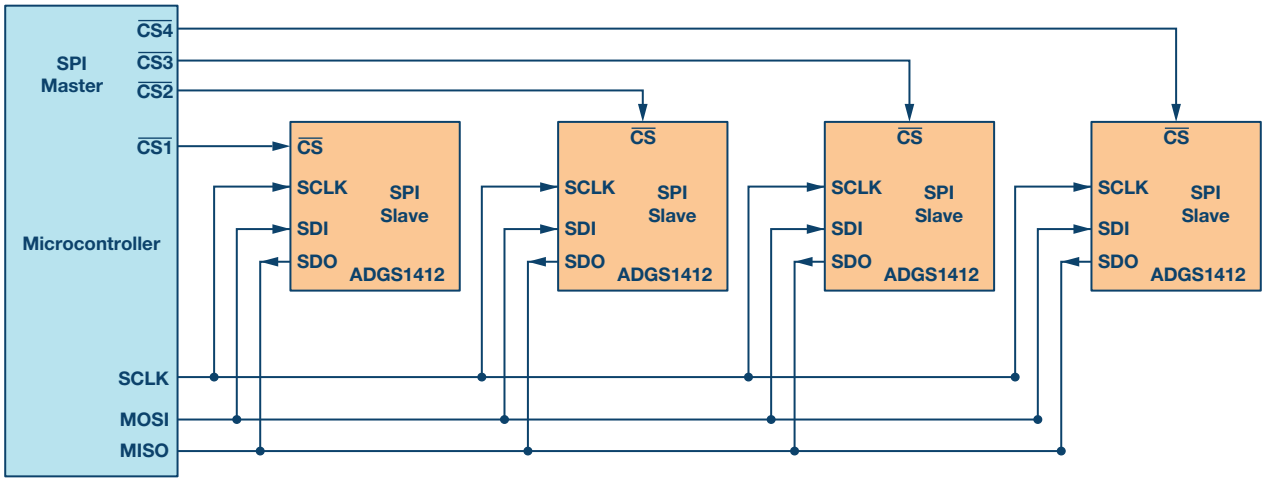


Figure 12. SPI enabled switches save up microcontroller GPIOs.

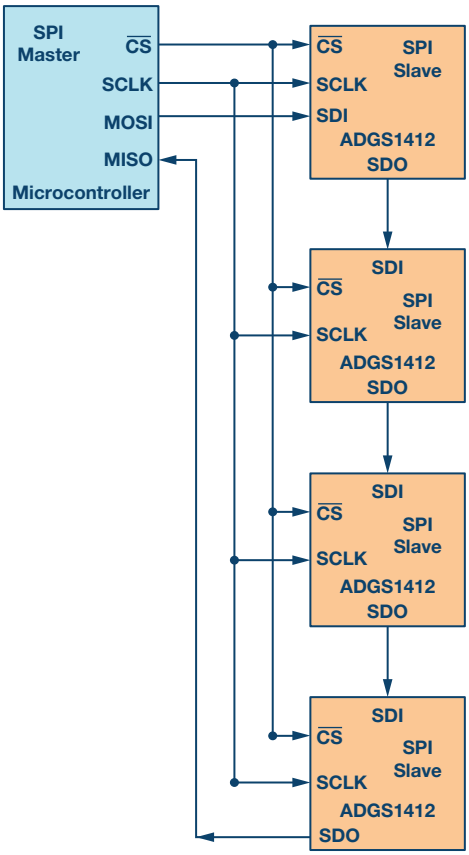


Figure 13. SPI enabled switches configured in a daisy chain to further optimize the GPIOs.

Figure 13 is for illustration purposes. The ADGS1412 data sheet recommends a pull-up resistor on the SDO pin. Please refer to the ADGS1412 data sheet for further details on daisy-chain mode. For the sake of simplicity, four switches have been used in this example. As the number of switches increase in a system, the benefits of board simplicity and space saving is significant. The ADI SPI enabled switches provide a 20% overall board space reduction in a 4 × 8 crosspoint configuration with eight quad SPST switches on a 6-layer board. The article "Precision SPI Switch Configuration Increases Channel Density" provides detail on how precision SPI switch configuration increases channel density.

Analog Devices offers several SPI enabled switches and multiplexers. For more information visit here.

References

ADuCM3029 data sheet. Analog Devices, Inc., March 2017.

Nugent, Stephen. "Precision SPI Switch Configuration Increases Channel Density." *Analog Dialogue*, May 2017.

Usach, Miguel. AN-1248 Application Note: *SPI Interface*. Analog Devices, Inc., September 2015.

Piyu Dhaker [piyu.dhaker@analog.com] is an applications engineer in the North America Central Applications Group of Analog Devices. She graduated from San Jose State University in 2007 with a master's degree in electrical engineering. Piyu joined the North America Central Applications Group in June 2017. She also previously worked in the Automotive Power Train Group and Power Management Group within ADI.

Piyu Dhaker

United States Patent [19]

Miesterfeld et al.

[11] **Patent Number: 4,739,324**

[45] **Date of Patent: Apr. 19, 1988**

[54] **METHOD FOR SERIAL PERIPHERAL INTERFACE (SPI) IN A SERIAL DATA BUS**

[75] Inventors: **Frederick O. R. Miesterfeld**, Troy; **John M. McCambridge**, Northville; **Ronald E. Fassnacht**, Rochester; **Jerry M. Nasiadka**, Warren, all of Mich.

[73] Assignee: **Chrysler Motors Corporation**, Highland Park, Mich.

[21] Appl. No.: **866,630**

[22] Filed: **May 22, 1986**

[51] **Int. Cl.**[4] **H04Q 9/00**; H04J 6/00

[52] **U.S. Cl.** **340/825.5**; 370/85

[58] **Field of Search** 340/825.5, 825.57; 370/85

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,281,380	7/1981	Demesa, III et al.	370/94
4,409,592	10/1983	Hunt	340/825.5
4,429,384	1/1984	Kaplinsky	370/85
4,434,421	2/1984	Baker et al.	340/825.51
4,470,110	9/1984	Chiarottino et al.	370/85
4,472,712	9/1984	Ault et al.	340/825.5
4,584,575	4/1986	Ryckeboer	340/825.5
4,628,311	12/1986	Milling	340/825.5

OTHER PUBLICATIONS

IEEE Publication, "Automotive Applications of Microprocessors", 1984; paper No. CH2072-7/84/00-00-0083, A Data Link for Agriculture and off Highway Communications, by Boyd Nichols, Vijay Dharia and Kanaparty Rao.

SAE Information Report, "J1567 Collision Detection Serial Data Communications Multiplex Bus", Fredrick O. R. Miesterfeld, May 23, 1986.

A. Bozzini et al, "Serial Bus Structures for Automotive Applications", *SAE Technical Paper Series,* 830536, Feb. 28–Mar. 4, 1983.

F. Miestefeld, "Chrysler Collison Detection (C²D)–A Revolutionary Vehicle Network", *SAE Technical Paper Series,* 860389, Feb. 24–28, 1986.

F. Phail et al, "In Vehicle Networking–Serial Communication Requirements and Directions", *SAE Technical Paper Series,* 860390, Feb. 24–28, 1986.

R. Mitchell, "A Small Area Network for Cars", 840317, *SAE Technical Paper Series,* 1984.

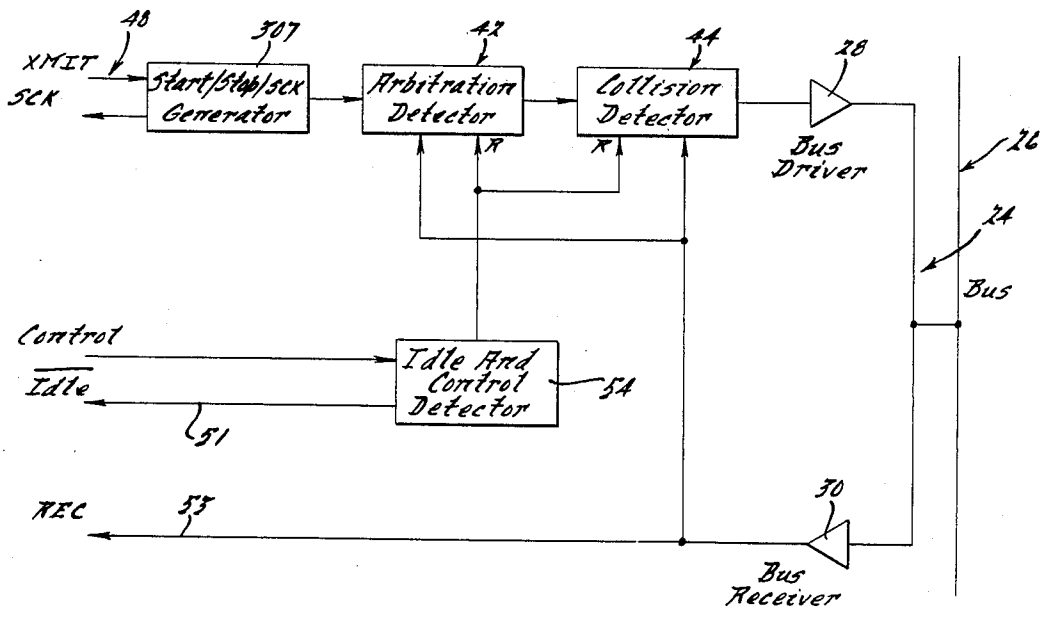
Primary Examiner—Donald J. Yusko

Attorney, Agent, or Firm—Mark P. Calcaterra

[57] **ABSTRACT**

In a communication system for the transmission of messages through a data bus between one or more user microprocessors coupled to the data bus, the user microprocessors having either a serial communications interface (SCI) port or a serial peripheral interface (SPI) port along with a clock port and an input/output port, the user microprocessors being coupled to the data bus by a bus interface integrated circuit, a method to transmit and receive data in an SPI mode of operation in conjunction with a method of arbitrating data on the data bus.

2 Claims, 4 Drawing Sheets



Unbuffered SPI

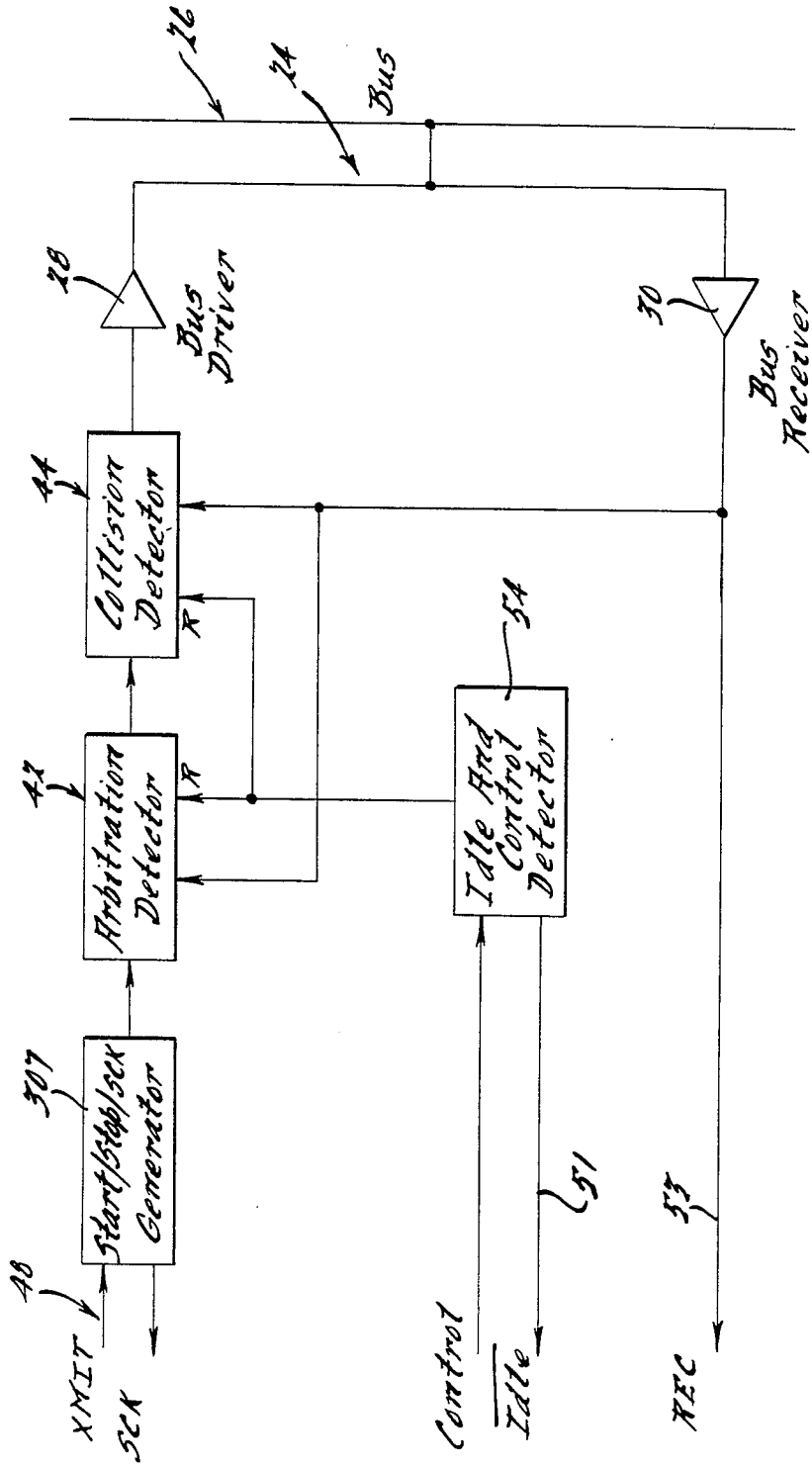

26
24
Bus
28
Bus Driver
30
Bus Receiver
44
Collision Detector
42
Arbitration Detector
54
Idle And Control Detector
50
Start/Stop/Sck Generator
48
XMIT
SCK
Control
Idle
51
52
REC
Unbuffered SPI

Fig. 1.

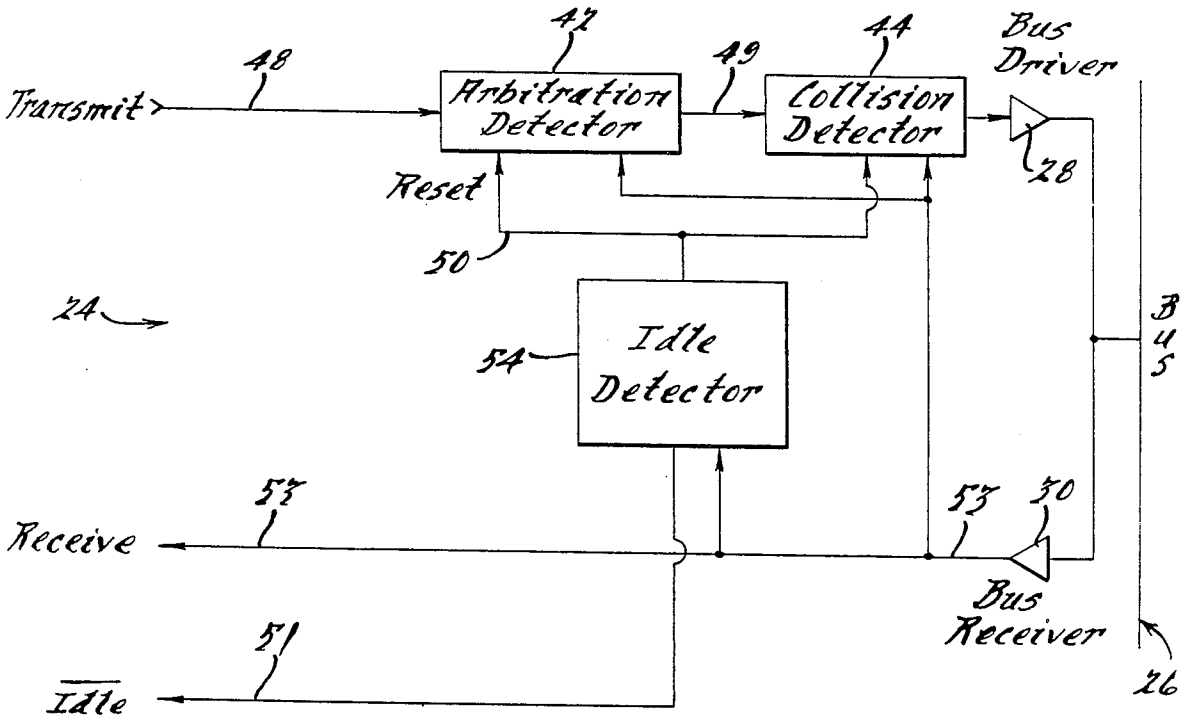

47
44
Bus Driver
48
49
Transmit
Arbitration Detector
Collision Detector
28
Reset
50
24
Idle Detector
54
B
U
S
53
53
30
Receive
Bus Receiver
51
Idle
26

Fig. 2.

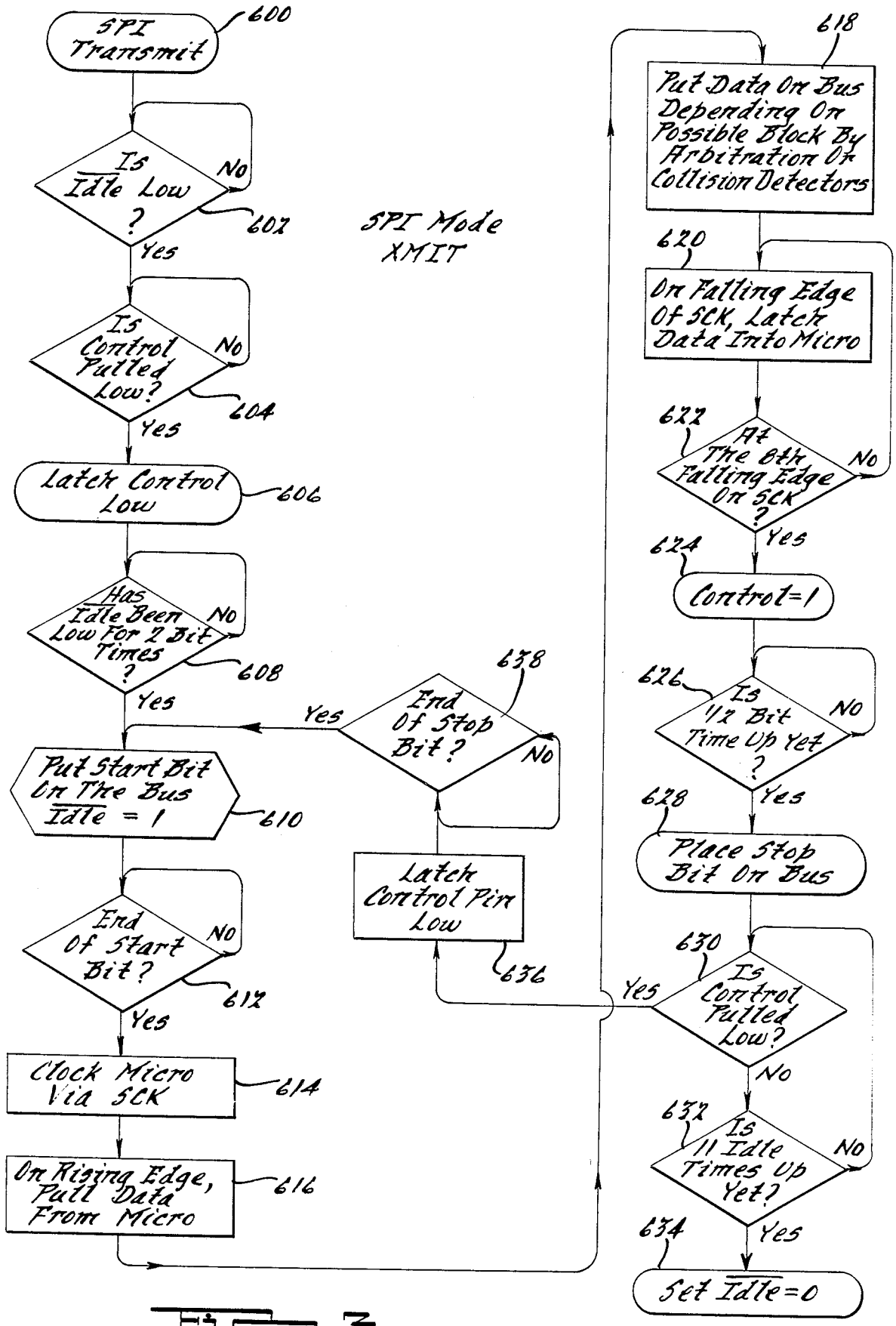
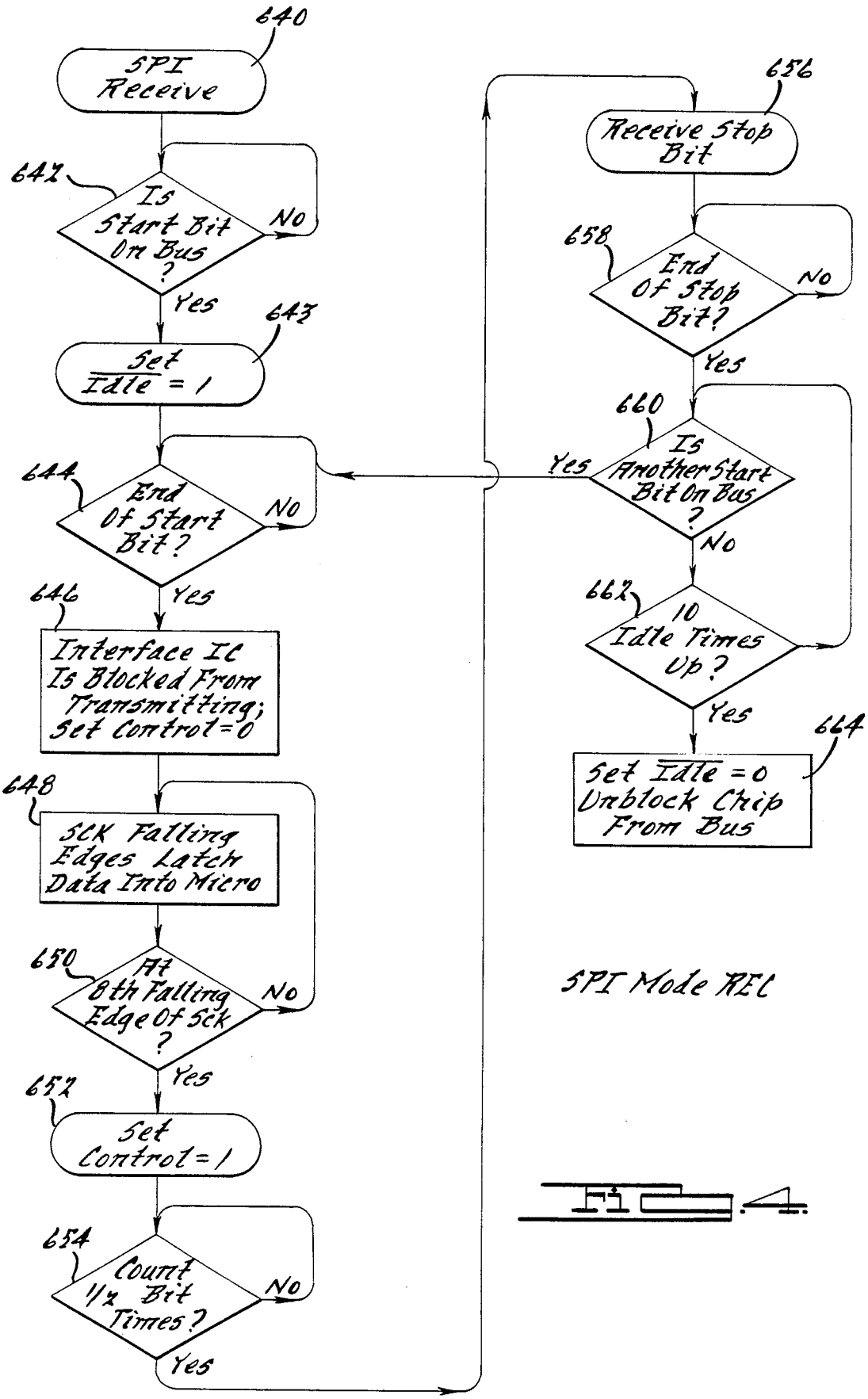

SPI Transmit
600
Is Idle Low ?
No
607
Yes
SPI Mode XMIT
Is Control Pulled Low?
No
604
Yes
Latch Control Low
606
Has Idle Been Low For 2 Bit Times ?
No
608
Yes
Put Start Bit On The Bus Idle = 1
610
End Of Start Bit?
No
612
Yes
Clock Micro Via SCK
614
On Rising Edge, Pull Data From Micro
616
End Of Stop Bit?
638
Yes
No
Latch Control Pin Low
636
Put Data On Bus Depending On Possible Block By Arbitration Or Collision Detectors
618
On Falling Edge Of SCK, Latch Data Into Micro
620
At The 8th Falling Edge On SCK ?
622
No
Yes
Control=1
624
Is 1/2 Bit Time Up Yet ?
626
No
Yes
Place Stop Bit On Bus
628
Is Control Pulled Low?
630
Yes
No
Is 11 Idle Times Up Yet?
632
No
Yes
Set Idle=0
634

Fig. 3.


640
SPI Receive
642
Is Start Bit On Bus ?
No
Yes
643
Set Idle = 1
644
End Of Start Bit?
No
Yes
646
Interface IC Is Blocked From Transmitting; Set Control=0
648
SCK Falling Edges Latch Data Into Micro
650
At 8th Falling Edge Of SCK ?
No
Yes
652
Set Control=1
654
Count 1/2 Bit Times?
No
Yes
656
Receive Stop Bit
658
End Of Stop Bit?
No
Yes
660
Is Another Start Bit On Bus ?
Yes
No
662
10 Idle Times Up?
Yes
664
Set Idle = 0 Unblock Chip From Bus
SPI Mode REC

METHOD FOR SERIAL PERIPHERAL INTERFACE (SPI) IN A SERIAL DATA BUS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The purpose of the disclosed method for serial peripheral interface (SPI) in a serial data bus is to provide a method of handling an SPI port as part of the serial data bus interface IC circuit described herein.

2. Description of the Prior Art

Data communications between microprocessors or microcomputers need to communicate with each other in many applications.

Local area networks (LAN) link such microprocessors or microcomputers, allowing one of the microcomputers to seize control of the serial data channel commonly linked to all other microprocessors on the LAN and transmit data to any other unit. The protocols, controllers and software needed in a LAN are very complex, especially in large systems.

An automotive environment is a smaller application and, thus, does not require the complex performance capabilities available in a LAN.

Digital data buses have been designed to handle the above-described data communications link in a small area. Such a system is described in SAE Paper No. 840317, by Ronald L. Mitchell entitled "A Small Area Network For Cars." This document is hereby expressly incorporated by reference. Also descriptive of such a digital data bus is U.S. Pat. No. 4,429,384 to Kaplinsky entitled "Communication System Having An Information Bus And Circuits Therefor."

Also descriptive of developments in this field is SAE Paper No. 860390 by Frederick H. Phail and David J. Arnett entitled "In Vehicle Networking - Serial Communications Requirements and Directions." This document is also hereby expressly incorporated by reference.

The subject invention differs from the art noted above by use of a constant speed, the lack of use of an acknowledgement bit and the lack of requirements for a tight link between the transmitting station and the receiving stations. Also important in the subject invention is the communication link between the message transmitter and receiver.

Generally, the following U.S. patents discuss collision detection in data communications systems: U.S. Pat. No. 4,281,380 of DeMesa III et al. entitled "Bus Collision Avoidance System For Distributed Network Data Processing Communications System" dated July 28, 1981; U.S. Pat. No. 4,409,592 of V. Bruce Hunt entitled "Multipoint Packet Data Communication System Using Random Access And Collision Detection Techniques" dated Oct. 11, 1983; U.S. Pat. No. 4,434,421 of Baker et al. entitled "Method For Digital Data Transmission With Bit-Echoed Arbitration" dated Feb. 28, 1984; U.S. Pat. No. 4,470,110 of Chiarottino et al. entitled "System For Distributed Priority Arbitration Among Several Processing Units Competing For Access To A Common Data Channel" dated Sept. 4, 1984; and U.S. Pat. No. 4,472,712 of Ault et al. entitled "Multipoint Data Communication System With Local Arbitration" dated Sept. 18, 1984.

The U.S. Pat. No. 4,434,421 patent to Baker et al. deals with a method to reduce the number of collisions. This is done by reducing the number of slave stations attempting bus access until there is one master and one slave station in communication. This differs from the subject invention in that a broadcast method is employed whereby several users can receive the same message.

The U.S. Pat. No. 4,470,110 to Chiarottino et al. discloses a system to exchange messages including an interface. In addition, U.S. Pat. No. 4,470,110 assigns a priority to an address bit of a particular logical level.

Also of interest is an article in an IEEE publication "Automotive Applications of Microprocessors," 1984; Paper No. CH2072-7/84/0000-0083 entitled "A Data Link For Agricultural And Off Highway Communications" by Boyd Nichols, Vijay Dharia and Kanaparty Rao.

Of paramount importance in the subject invention is the inclusion of the capability to communicate with a serial communication interface (SCI) port, a serial peripheral interface (SPI) port and a buffered serial peripheral interface (BSPI) port.

SUMMARY OF THE INVENTION

The purpose of the serial data bus system disclosed herein, also known as Chrysler Collision Detection (C^2D) bus, is to allow multiple microprocessors to easily communicate with each other over a common pair of wires or bus using a scheme similar to a telephone party line. All microprocessors connected to the bus are able to receive all messages transmitted on the bus. Any microprocessor with a message to transmit on the bus waits until any current user is finished before attempting to use it.

Whenever the bus is available, its use is allocated on a first-come first-serve basis. That is, whichever microprocessor begins transmitting its message on the bus, after any previous message finishes, gets the use of the bus. If, however, multiple microprocessors attempt to begin transmitting their messages on the bus at exactly the same time, then the message with the highest priority wins the use of the bus. All messages have unique message priority values and each message is transmitted by only one microprocessor.

The invention disclosed herein is further summarized in two co-pending patent applications on related material. Both applications were filed in the U.S. Patent & Trademark Office on Feb. 24, 1986, and are commonly owned with the subject patent application. They are: "Serial Data Bus For Intermodule Data Communications," U.S. Ser. No. 06/832,908; and "Method Of Data Arbitration and Collision Detection On A Data Bus," U.S. Ser. No. 06/832,909. Both of these applications are hereby expressly incorporated by reference.

Also hereby incorporated by reference is SAE Information Report entitled "J1567 Collision Detection Serial Data Communications Multiplex Bus" to be presented to the SAE Multiplexing Committee on May 23, 1986.

Attention is invited to the above-described applications for further explanation of the summaries of some of the basics of the invention described in the subject application.

It is an object of the subject invention to provide an SCI port, an SPI port and a buffered SPI port as part of the serial data interface integrated circuit described herein. This allows communication with any device configured with any one of these three ports all on the same bus. The inclusion of the ports augments the simplification of the serial data communication described in

the previously filed patent applications on the related subject matter.

DESCRIPTION OF THE DRAWINGS

Other objects, features and advantages of the present invention will become more fully apparent from the detailed description of the preferred embodiment, the appended claims and the accompanying drawings in which:

FIG. 1 illustrates in block diagram form hardware used in the SPI mode of a serial data bus interface integrated circuit (IC);

FIG. 2 illustrates another block diagram showing the serial data bus in an SCI mode;

FIG. 3 is a flowchart showing the SPI mode methods under a transmit condition; and

FIG. 4 shows the flowchart of the SPI mode in a receive condition.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

This application is one of three filed on the same day and having related specifications and drawings. The other cases are commonly owned with the same inventors are Ser. Nos. 06/866628 and 06/866629 and are entitled "Method For Serial Peripheral Interface In A Serial Data Bus" and "Method For A Buffered Serial Peripheral Interface In A Serial Data Bus." Both of these cases are hereby expressly incorporated by reference.

Further documents hereby expressly incorporated by reference include U.S. Pat. No. 4,429,384 issued to Kaplinsky and entitled "Communication System Having An Information Bus And Circuits Therefor"; SAE Technical Paper No. 830536 entitled "Serial Bus Structures For Automotive Applications" by Anthony J. Bozzini and Alex Goldberger dated Feb. 28, 1983; SAE Paper No. 840317 by Ronald L. Mitchell entitled "A Small Area Network For Cars"; SAE Paper No. 860390 by Frederick H. Phail and David J. Arnett entitled "In-Vehicle Networking - Serial Data Communication Requirements And Directions"; and SAE Paper No. 860389 by Frederick O. R. Miesterfeld entitled "Chrysler Collision Detection (C^2D) A Revolutionary Vehicle Network."

Attention is invited to the previously filed patent applications and the concurrently filed patent applications on the related subject matters for a more complete description of some of the hardware disclosed in FIG. 1 and FIG. 2.

The interaction between the arbitration detector **42**, collision detector **44**, work counter **202**, word flip-flop **203**, start bit detector **200**, firming air detector **204**, idle counter **206**, idle flip-flop **207**, clock divider **201**, digital filter **210**, bus driver made up of OR gate **62** and NAND gate **63**, along with bus receiver **30** in conjunction with current source **34** and current sink **36** as connected to the bus **26**.

An understanding of the above-listed blocks is necessary for understanding the improvements outlined in the subject application. Attention is, therefore, invited to patent applications U.S. Ser. Nos. 06/832,908 and 06/832,909 and the explanations included therein and the drawings which all have been incorporated by reference.

Referring to FIG. 1 and FIG. 2, the hardware of the serial data bus interface IC **24** is shown in two ways. In FIG. 2, the bus interface IC **24** is shown in a serial communications interface (SCI) mode. That is, only the hardware which is used exclusively for SCI and which is common to other modes of operation for the serial data bus, namely SPI and buffered SPI is illustrated.

FIG. 1 augments the diagram of FIG. 2 by adding in the hardware from the bus interface IC **24** which is used in an SPI mode (in an unbuffered condition).

The block diagram of FIG. 2 is adequately described in the documents incorporated by reference and should not be repeated here.

The unbuffered SPI hardware diagram shown in FIG. 1 merely augments the diagram of FIG. 2 with some of the blocks and lines described in other copending patent applications, namely, the start/stop/SCK generator **307** and the SCK and CONTROL lines. Also augmented in this figure is the idle control detector **54** to show that the idle and control detector is comprised of the idle counter **206**, idle flip-flop **207** and the scheduler and controller block **309**, all described in the copending patent application entitled "Serial Data Bus For SCI, SPI and Buffered SPI Modes of Operation."

Turning now to FIG. 3 and to FIG. 4, the SPI method will be described in its transmit and receive condition. This is the heart of the subject invention.

When SPI transmit condition exists, the serial data bus interface IC **24** will utilize an SPI mode to govern the transmitting of the data. This mode is begun in block **600** and the bus interface IC **24** checks to see if the $\overline{\text{IDLE}}$ line is in a low condition in decision block **602**. If the $\overline{\text{IDLE}}$ line is not low, the bus interface IC **24** waits until it is, and then falls through to decision block **604** to check the control line to see if it has been pulled low. If not, the bus interface IC waits until the control line is pulled low and then falls through to block **606** to latch the control line in a low condition.

Next, the bus interface IC checks to see if the $\overline{\text{IDLE}}$ line has been low for two bit times in decision block **608**. If not, it waits for that condition to occur and falls through a decision block **610** to put a start bit on the bus **26** and set the $\overline{\text{IDLE}}$ line to a logical one.

Next, the bus interface IC checks for the end of a start bit and waits until the end of a start bit occurs in block **612** before falling through to block **614** to clock the user microprocessor which is not shown in the subject application, but which is shown and illustrated in the copending patent applications. The clock of the user microprocessor is done via the SCK line which is also more fully explained in the co-pending patent applications.

Next, the bus interface IC falls through to block **616** and, on the rising edge of the SCK signal, pulls the data from the user microprocessor. Next, the user microprocessor puts the data on the bus **26** depending on the possible blocking operation by the arbitration or collision detectors **42** and **44**, respectively. This is done in block **618**.

Next, on the falling edge of an SCK signal, the user microprocessor in block **620** latches the data into the user microprocessor and falls through the decision block **622** where it watches for the eighth falling edge on the SCK signal. If the eighth falling edge has not occurred, the method calls for the bus interface IC to return to block **620** to again latch data into the microprocessor. This continues until the eighth falling edge of the SCK signal is observed in decision block **622** at which time the user microprocessor falls through to block **624** to set the control line to a logical one.

Next, the ½ bit time is checked in block **626** and, once it has been reached, the bus interface IC **24** places a stop bit on the bus **26**. This is shown in block **628**.

In block **630**, the bus interface IC **24** checks to see whether the control line has been pulled low. If it has, the bus interface IC **24** returns to the method beginning with block **610**, after latching the CONTROL line low in block **636** and waiting for the end of the stop bit in block **638**, and puts a start bit on the bus and sets the $\overline{\text{IDLE}}$ line equal to one. If the control line has not been pulled low as checked in block **630**, the routine falls through to block **632** to check to see whether eleven idle times have occurred. If not, the decision block **630** is re-entered to recheck the control line.

Once eleven idle times have occurred, the routine falls through to block **634** to set the $\overline{\text{IDLE}}$ line equal to zero, thus terminating the SPI mode transmission.

Turning now to FIG. **4**, the SPI mode is illustrated in a receive condition.

When the bus interface IC **24** is in the SPI mode, it must receive data. The bus interface IC **24** begins in block **640** and falls through to block **642**, waiting until a start bit appears on the bus **26** before setting $\overline{\text{IDLE}}=1$ in block **643**. The bus interface IC then sets the $\overline{\text{IDLE}}$ line to a logical one.

The user microprocessor then checks in block **644** to watch for the end of the start bit and waits until this occurs before falling through to block **646** to block the bus interface IC from transmitting while setting the control line to a logical zero.

Upon the occurrence of the falling edges of the SCK signal, the user microprocessor in block **648** latches the data into the user microprocessor. The bus interface IC **24** in block **650** looks for the eighth falling edge of the SCK signal and repeats the latching of data into the user microprocessor as called out in block **648** until the eighth falling edge of the SCK signal occurs. At that point, the bus interface IC falls through to block **652** to set the control line to a logical one.

After counting ½ bit times in decision block **64**, the user microprocessor falls through to block **656** to receive a stop bit period. Once the end of the stop bit has been observed in block **658**, the bus interface IC **24** falls through to block **660** to check to see whether there is another start bit on the bus **26**. If there is, the routine returns to block **644** and repeats the above-described procedure. If not, the routine fall through to block **662** to watch for the occurrence of ten idle times. Once this condition occurs, and no other start bit appears on the bus, the routine falls through to block **64** to set the $\overline{\text{IDLE}}$ line to zero thereby unblocking the bus interface IC from accessing the bus **26**.

Thus, the SPI mode in the receive condition is terminated.

While the present invention has been disclosed in connection with the preferred embodiment thereof, it should be understood that there may be other embodiments which fall within the spirit and scope of the invention and that the invention is susceptible to modification, variation and change without departing from the proper scope or fair meaning of the following claims.

We claim:

1. In a communication system for the transmission of messages through a data bus between one or more user microprocessors coupled to the data bus, the user microprocessors having either a serial communications interface (SCI) port or a serial peripheral interface (SPI) port along with a clock port and an input/output port, the user microprocessors being coupled t the data bus by a bus interface integrated circuit, a method of transmitting data in an SP mode of operation comprising:

a bit-wise contention and a deterministic priority access method to (a) resolve contentions among user microprocessors that try to send messages at the same time and to (b) synchronize each data byte and to (c) allow the priority of an ID byte of the message to determine which of a plurality of messages will be sent first in the case of a contention, the determination of which user microprocessor transmits first being made without losing bus time when contention occurs; and

checking to see whether the data bus is idle, waiting if it is not;

waiting for the bus to be idle for 2-bit times;

placing a start bit on the data bus;

waiting until the end of the start bit;

closing the user microprocessor;

pulling data from the user microprocessor;

placing the data from the user microprocessor onto the bus depending on a possible block from bit-wise contention and deterministic priority access method utilized by the bus interface IC;

latching the data back into the user microprocessor;

waiting for the eighth falling edge of a clock signal;

waiting for the occurrence of ½ bit time;

placing a stop bit on the data bus;

checking to see whether the user microprocessor wishes to transmit another byte;

if the user microprocessor does not wish to transmit another byte, waiting for the occurrence of eleven idle times, and making another check to see whether the user microprocessor wishes to put more data on the data bus; and

if the user microprocessor wishes to transmit another byte, return to repeat the previous steps to transmit the data beginning with the step of checking to see whether the data bus is idle.

2. In a communication system for the transmission of messages through a data bus between one or more microprocessors coupled to the data bus, the user microprocessors having either a serial communications interface (SCI) port or a serial peripheral interface (SPI) port along with a clock port and an input/output port, the user microprocessor being coupled to the data bus by a bus interface integrated circuit, a method of receiving data in an SPI mode of operation comprising:

a bit-wise contention and a deterministic priority access method to (a) resolve contentions among user microprocessors that try to send messages at the same time and to (b) synchronize each data byte and to (c) allow the priority of an ID byte of the message to determine which of a plurality of messages will be sent first in the case of a contention, the determination of which user microprocessor transmits first being made without losing bus time when contention occurs; and

watching for a start bit to appear on the data bus;

waiting until the end of the start bit once a start bit appears on the bus;

signaling the bit-wise contention and deterministic priority access method to lock the bus interface IC from transmitting;

latching data into the user microprocessor;

waiting for the eighth falling edge of the clock signal and then counting 2-bit times;

receiving a stop bit on the data bus and waiting for the end of the stop bit to occur;

checking for the occurrence of another start bit on the data bus, repeating the previous steps to latch in more data if another start bit is on the data bus;

waiting ten idle times if there is no other start bit on the data bus and signaling the bit-wise contention and deterministic priority access method to unlock the bus interface IC from transmitting data.

* * * * *