

industriales
etsii UPCT

Departamento de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingenieros Industriales

PROYECTO FIN DE CARRERA

*Estudio y construcción de una
plataforma para el control de
maquetas docentes con Arduino.
Comparativa con autómatas
programables S7-200*

Titulación: I.T.I. Electrónica Industrial

Autor: Jorge Muñoz Heredia

Director: Dr. Miguel Almonacid Kroeger

2012

No importa las veces que has caído sino las que te has levantado.

El proyecto está dedicado a mis padres, Pedro e Isabel, a mi hermano Pedro José y mis abuelos.

Reflexiones

Antes de comenzar con la exposición de este proyecto fin de carrera, me gustaría expresar algunas sensaciones que tengo una vez concluida ésta ilustre etapa en mi vida.

Después de un año empleado en terminar este trabajo, he convivido con experiencias y situaciones a cada cual su complejidad pero llegando a un fin muy satisfactorio.

La realización del proyecto específico lleva una trayectoria en la cual el proyectante se siente solo en multitud de ocasiones, teniendo muchos retos para poder cumplir con el objetivo inicial.

Sin embargo, a pesar de todos esos obstáculos que se encuentran en el camino, ya que si no me hubiese sido imposible adquirir ese conocimiento práctico de la carrera, tratar y aprender de profesores, colaboradores de departamentos, al fin y al cabo, son PERSONAS que ofrecen un apoyo que se necesita en momentos críticos y difíciles como es en esta situación... todo ello que de ninguna otra forma podría haber vivido.

Como conclusión, señalar que si volviese atrás en el tiempo y tuviese la oportunidad de elegir, volvería a escoger la realización del proyecto fin de carrera específico, tal y como ha ocurrido.

Agradecimientos

En primer lugar quiero dar las gracias al Dr. Miguel Almonacid Kroeger, tutor que me dio la oportunidad de iniciar esa aventura que siempre quise, el proyecto fin de carrera. Por ser mucho más que un maestro y confiar en mí, dándome ánimos en cada momento, prestándome su ayuda absoluta y ser mi guía en cada situación comprometida. Todo se resume en, GRACIAS POR TODO MIGUEL.

Agradecer el apoyo y facilidades en cada momento de los componentes del departamento de sistemas y automática, desde el técnico de laboratorio Pablo A. Martínez Ruiz que a pesar de su infinita tarea, me ha dedicado parte de su valioso tiempo en mí, al Dr. Julio José Ibarrola Lacalle por su ánimo y seguimiento del proyecto. Y sin dejarme de lado, del departamento de Tecnología Electrónica al Dr. Ing. José Antonio Villarejo Mañas por su colaboración y dedicación, a todos ellos les doy las gracias por hacerme sentir como un compañero más dentro del departamento y laboratorio.

Por supuesto dar las gracias a Francisco Javier Toledano Moreno, proyectante y compañero de clases, por ofrecerme todo su conocimiento.

Y por último y sin desmerecer lo más mínimo, mi familia, a mis padres, Pedro e Isabel, por haberme dado todo lo que he necesitado en mi vida para llegar hasta aquí, de poder realizar ese sueño, que desde pequeño en el colegio mal escribía: "Jorge quiere ser Ingeniero Electrónico", a pesar de que yo no siempre lo he puesto fácil, y porque son los mejores. Gracias "papis", os quiero.

Gracias a mi hermano, porque simplemente ser un hermano no es nada fácil y por sacarme esas risas que día tras día me hacen falta para desconectar de las dificultades que conlleva la realización del proyecto. Pedro José, te quiero.

Muchas gracias a todos. Jorge.

Índice

1. Motivación y objetivos	1
1.1.Motivación.....	1
1.2.Objetivos y fases del Proyecto	1
2. Equipos del proyecto	3
2.1.Introducción	3
2.2.Tarjeta de control Arduino.....	3
2.2.1. Componentes	4
2.2.2. Software	11
2.2.3. Ventajas e Inconvenientes generales de Arduino.....	20
2.3.Automatas programables	20
2.3.1. Componentes	21
2.3.2. Instalación	27
2.3.3. Puesta a punto	30
2.3.4. Autómata Programable S7-200.CPU 224 AC/DC/Relé.....	31
2.3.5. Comunicación PC/S7-200	33
2.3.6. Software	36
2.3.7. Ventajas e inconvenientes generales del autómata programable	43
2.4. Maqueta docente.....	44
3. Plataforma A-M para la automatización mediante Arduino	51
3.1.Introducción	51
3.2.Diseño y construcción de la plataforma A-M	52
3.3.Conexión de la plataforma A-M	60
3.4.Presupuesto del material.....	61

4. Pruebas experimentales	63
4.1. Introducción	63
4.2. Ejercicio 1: Mando del motor	63
4.2.1. Programación con autómeta y Arduino	64
4.3. Ejercicio 2: Uso del Paro de Emergencia	66
4.3.1. Programación con autómeta y Arduino	67
4.4. Ejercicio 3: Paro por el Sensor Óptico	70
4.4.1. Programación con autómeta y Arduino	70
4.5. Ejercicio 4: Temporizadores	73
4.5.1. Programación con autómeta y Arduino	73
4.6. Ejercicio 5: Uso del Encoder Magnético	76
4.6.1. Programación con autómeta y Arduino	77
5. Conclusiones y Vías futuras	80
5.1. Conclusiones	80
5.2. Vías futuras.....	81
6. Bibliografía.....	82
6.1.Referencias	82
6.2.Direcciones web.....	83
7. Anexos	86
Anexo A: Datasheet utilizados.....	86
7.1.Anexo Micro-Controlador ATMEGA328(Arduino)	86
7.2.Anexo sensor óptico Opto-BERO	87
7.3.Anexo tarjeta de bornes DN-37/DN-37-A	94
7.4.Anexo caja de mando	95
7.5.Anexo fuente de alimentación Omron S8VS	105
7.6. Anexo opto-acoplador 4N25.....	109

7.7. Anexo opto-acoplador 6N137	114
7.8. Anexo circuito integrado LMD 18200 3A, 55V, H-Brigde	119
7.9. Anexo amplificador de aislamiento HCPL-7200	129
7.10. Anexo placas de Arduino	135
7.11. Anexo Relés 5V	149
Anexo B: Programación con Arduino de los ejercicios.....	151
7.12.1. Mando del motor	151
7.12.2. Uso del paro de emergencia	153
7.12.3. Paro por el sensor óptico	156
7.12.4. Temporizadores.....	160
7.12.5. Uso del encoder magnético	163

Capítulo 1:

Motivación y objetivos

1.1. Motivación

Una vez aprobadas todas las asignaturas de Ingeniería Técnica Industrial en la especialidad Electrónica Industrial, decidí realizar el proyecto fin de carrera específico para poder poner en práctica todos los conocimientos teóricos adquiridos en las materias Automáticas y Electrónica durante la presente diplomatura, de esta forma tener una experiencia nueva antes de salir al mundo laboral.

Elegí este tipo de proyecto al acabar mi carrera ya que introducía todos los campos: automática, electrónica, programación e incluyendo a la buena la impresión y propuesta que me dio Miguel, mi director del proyecto, tomé esta decisión.

Además, el estudio y desarrollo de dicho Proyecto Fin de Carrera concluirá, una vez aprobado éste, con la obtención del título en Ingeniería Técnica Industrial, especialidad en Electrónica Industrial por la Universidad Politécnica de Cartagena.

1.2. Objetivos y fases del proyecto

El objetivo principal del proyecto es el estudio e implementación del control de diferentes maquetas de laboratorio de Automatización del Departamento de Ingeniería de Sistemas y Automática a partir una tarjeta de control de bajo costo y software libre: Arduino. Para de esta forma poder tener una alternativa de automatización a los autómatas programables.

La realización de dicho proyecto se espera que contribuya y se exporte posteriormente a la docencia de los estudiantes de la Universidad Politécnica de Cartagena para la ejecución de prácticas en el Laboratorio de Automática y Robótica del Departamento de Ingeniería de Sistemas y Automática para tener una variante a los Autómatas Siemens S7-200 para complementar la docencia.

Una vez evaluado el margen de trabajo que consta el sistema y sus características técnicas, describiré las distintas fases que consta este proyecto, a destacar:

1. Estudio de las limitaciones de hardware y programación que contiene Arduino.
2. Estudio de las necesidades mecánicas y eléctricas para poder poner en funcionamiento las maquetas docentes.
3. Diseño y construcción de etapa de ampliación y aislamiento para controlar el sistema completo de forma segura y eficaz.
4. Familiarización con los sistemas y lenguajes de programación.
5. Pruebas experimentales realizadas con la maqueta a automatizar ordenados por dificultad.
6. Conclusiones de la comparativa.

Capítulo 2:

Equipos del proyecto

2.1. Introducción

El proyecto a realizar trata de automatizar las maquetas docentes con la tarjeta Arduino de igual forma que con el Autómata Programable S7-200, ofreciendo una alternativa con ventajas y desventajas respecto este, para ello se tendrá que diseñar y construir una plataforma que englobe todas las necesidades electrónicas que necesita Arduino para adaptarse a todas las maquetas existentes del laboratorio, ya que sus rangos de tensión son de 24V las digitales y una señal analógica variable de 10V, ya que Arduino solo podrá manejar valores de hasta 5V. Además como prevención, aislaremos entradas y salidas de las maquetas respecto a Arduino.

2.2. Tarjeta Arduino

El proyecto está basado en la utilización del hardware y software de Arduino, es una plataforma de prototipos electrónica de **código abierto**, en concreto utilizaremos Arduino Uno SMD Edition, integra un conjunto sofisticado de circuitos que a continuación describiremos, todos ensamblados en una pequeña placa de circuito impreso (PCB) y una configuración de **software abierta** a cualquier usuario para la creación de prototipos de desarrollo rápido para entusiastas o estudiantes de ingeniería que quieren llevar a cabo proyectos de automatización.

Lo que lo hace tan especial es que crea una capa de abstracción que evita que aprendas tantos por menores de cada integrado o micro-controlador, y te deja enfocarte realmente en la lógica o en lo que quieres realizar. La finalidad es que tu código pueda portarse a otro micro más moderno o que otra persona lo pueda tomar para incorporarlo a su propio proyecto sin problemas de compatibilidad. El diseño está totalmente realizado en **hardware libre**, es decir lo puedes armar en tu casa o puedes comprarlo a un **precio económico**.

Además, hay una comunidad muy grande desarrollando en esta plataforma que te brindan consejos y asesoría, además de ejemplos. El Arduino Uno SMD Edition es mostrado en la Figura 2.1.

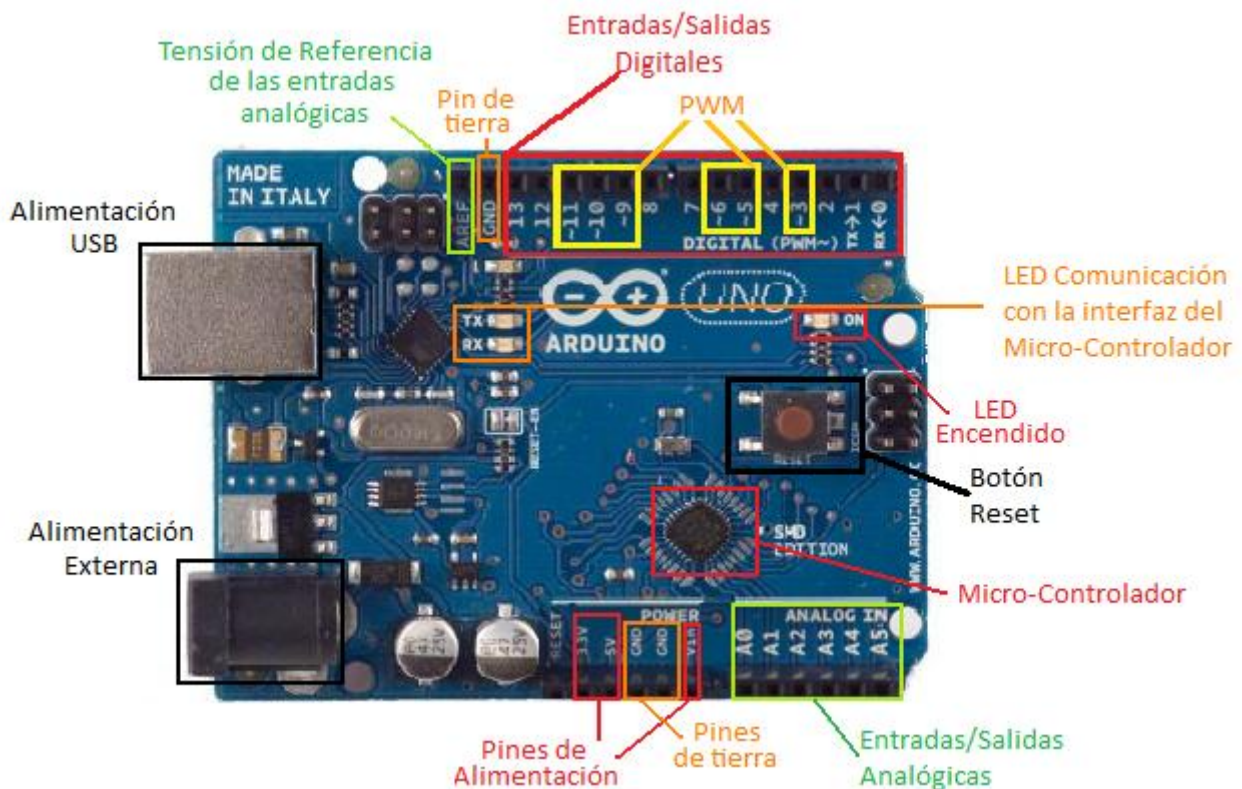


Figura 2.1. Arduino Uno SMD Edition.

2.2.1. Componentes

El Arduino Uno SMD es una versión de la Arduino Uno, pero utiliza una versión de montaje en superficie de **la ATMEGA328P** lugar de la versión a través de hoyos. Esta versión se hizo en respuesta a una escasez en el suministro del orificio pasante ATMEGA328P. El consejo está basado en el ATmega328 (ficha técnica).

Cuenta con **14 entradas / salidas digitales** pines (de las cuales 6 se puede utilizar como salidas PWM), 6 entradas analógicas, un joven de 16 MHz oscilador de cristal, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reset. Contiene todo lo necesario para apoyar el micro-controlador, basta con conectarlo a un ordenador con un cable USB o el poder con un adaptador de CA a CC o batería para empezar.

El Uno se diferencia de todas las anteriores en que no utiliza el chip controlador **FTDI USB-to-serial**. En su lugar, ofrece la Atmega8U2 programado como un convertidor de USB a serie. "Uno" significa uno en italiano y se nombra para conmemorar el próximo lanzamiento de Arduino 1.0. El Uno y la versión 1.0 será la versión de referencia de Arduino, moviéndose hacia adelante. El Uno es el penúltimo de una serie de placas Arduino USB, el modelo más reciente es la placa Leonardo (ver anexo 7.10.placas Arduino) y el modelo de referencia para la plataforma Arduino, para una comparación con las versiones anteriores, consulte el índice de la placa Arduino.

Micro-controlador	ATmega328
Voltaje de operación	5V
Voltaje de Entrada	7-12V
Voltaje de Entrada	6-20V
Digital pines I / O	14 (6 PWM)
Pines de entrada analógica	6
Corriente DC por I / O Pin	40 mA.
Corriente DC de 3,3 V Pin	50 mA.
Memoria Flash	32 KB (0,5 KB gestor de arranque)
SRAM	2 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

La **memoria Flash** (espacio del programa) es donde Arduino almacena el sketch, nombre que usa Arduino para un programa. Es la unidad de código que se sube y ejecuta en la placa. Esta memoria es no volátil, si Arduino deja de ser alimentado eléctricamente los datos que haya en esta memoria permanecerán. Su tamaño no es muy grande, por lo que debemos desarrollar los programas de forma optimizada, usando tipos de variables que requieran menos memoria, al igual que el código fuente de la aplicación para no repetir líneas de código, en medida de lo posible.

La **memoria SRAM** (memoria estática de acceso aleatorio) es de tipo volátil, es el espacio donde los sketches (programas) almacenan y manipulan variables al ejecutarse. La información guardada en esta memoria será eliminada cuando Arduino pierda la alimentación. Esta memoria es de uso exclusivo para el programa en ejecución. Su tamaño es muy pequeño, por lo que debemos optimizar nuestros programas al máximo y no abusar de variables de tipo *char* muy grandes. Hay que tener en cuenta que cada carácter de una variable *char* utiliza un byte.

En el caso que la SRAM se queda sin espacio, el programa fallará de forma imprevista, aunque se compile y se suba correctamente, la aplicación no se ejecutará correctamente e incluso ni se ejecutará. Para optimizar y evitar que consuman la memoria SRAM disponible se aconseja:

El programa se comunica con una aplicación ejecutándose en un ordenador, se puede intentar trasladar los datos o cálculos al ordenador, reduciendo la carga.

El programa usa tablas de referencia u otros arreglos de gran tamaño, es recomendable utilizar el tipo de datos más pequeño que se pueda para almacenar estos datos; un *int* utiliza 2 bytes, mientras que un *byte* utiliza solo 1 byte.

Si no se necesita modificar las cadenas o datos mientras el programa se ejecuta, se pueden almacenar en la Flash en vez de la SRAM utilizando el keyword *PROGMEM*.

La **memoria EEPROM**, espacio de memoria que puede ser utilizado por los programadores para almacenar información a largo plazo. Este tipo de memoria es no volátil, por lo que los datos guardados en ella permanecerán aunque Arduino pierda la alimentación. Esta memoria puede ser usada para guardar y leer valores si es necesario.

Tenemos la posibilidad de escribir en la memoria EEPROM, mostramos un sencillo ejemplo para escribir 1024 valores en la memoria EEPROM:

```
#include

void setup () {

//usamos un bucle que se ejecutará 1024 veces en la posición i de la memoria EEPROM y guardaremos el valor de i

    for (int i = 0; i < 1024; i++)

        if (i <= 255) {

            EEPROM.write (i, i);

        } else {

            EEPROM.write (i, i - 255)

        }

}

void loop () {}
```

El ejemplo anterior se ejecutará una sola vez (no hemos usado "loop"). El bucle *for* se ejecutará 1024 veces y guardará en cada posición de la memoria el valor actual de *i*, teniendo en cuenta que el valor máximo que se puede guardar en una posición de memoria es de 255, por ello cuando llegamos a 255 guardamos el valor de *i* menos 255.

Incluso podremos leer valores de la memoria EEPROM, un sencillo ejemplo que lee y envía por el puerto serie de Arduino todos los valores guardados en la memoria EEPROM:

```
#include

int posicionActual = 0;

int valorLeido;

void setup() {

Serial.begin(9600);

}

void loop () {

/*obtenemos el valor de la posición "posicionActual" de la EEPROM, valorLeido = EEPROM.read (posicionActual);
enviamos por el puerto serie la posición leída */

Serial.print (posicionActual);

Serial.print ("");

Serial.print (valorLeido); //enviamos por el puerto serie el valor leído de la posición

Serial.println ();

/*incrementamos la posición actual, puesto que el programa se ejecuta indefinidamente leeremos todas las
posiciones de memoria EEPROM*/

posicionActual = posicionActual + 1;

/*puesto que el programa se ejecuta indefinidamente para evitar que dé error cuando nos excedamos de la última
posición de la EEPROM, cuando llegemos al máximo 1024 empezaremos de nuevo*/

if (posicionActual == 1024) {

posicionActual = 0;

delay (1000); //esperamos un segundo

}

}
```

En ambos casos usamos la librería *EEPROM.h*, necesaria para acceso a la memoria EEPROM de Arduino.

Tenemos la posibilidad de ampliar el almacenamiento de Arduino, una opción para ampliar la capacidad de memoria es usar memoria EEPROM externa, adquiriendo este tipo de memoria (que no es muy cara) podremos disponer de más memoria EEPROM para nuestro programa. Este tipo de memoria es no volátil. Por ejemplo, una EEPROM 24LC64 puede costar unos 4 euros y tiene 64K de tamaño. Obviamente, este tipo de memorias EEPROM no tienen un gran tamaño, por ello, no serán útiles para proyectos con grandes requerimientos de espacio. Por supuesto existen librerías para trabajar con estas tarjetas en Arduino, Figura 2.2:

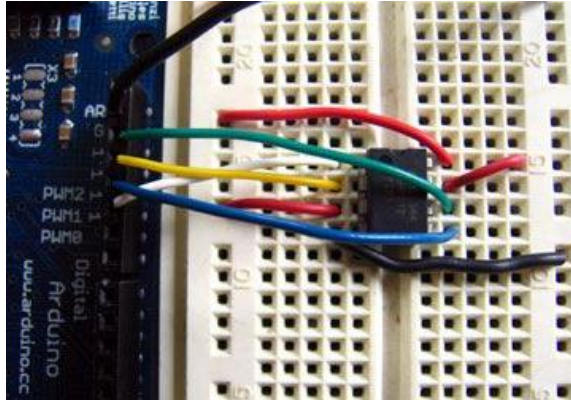


Figura 2.2. Ampliación con EEPROM 24LC64

Ampliar memoria con SD Card, si vamos a desarrollar un proyecto hardware con Arduino que tenga grandes requerimientos de espacio (Megas, Gigas), la mejor opción es utilizar algún tipo de medio de almacenamiento removible, a través de las tarjetas flash (SD, microSD), permiten almacenar varios GB de datos en un espacio muy reducido. Por supuesto existen librerías para trabajar con estas tarjetas en Arduino. En la Figura 2.3. mostramos un módulo SD Card para Arduino, acoplable directamente a nuestro Arduino:



Figura 2.3. Módulo SD Card para Arduino

En este otro ejemplo mostramos en la Figura 2.4. una SD Card para Arduino que tenemos que conectar manualmente, como se muestra en la imagen:

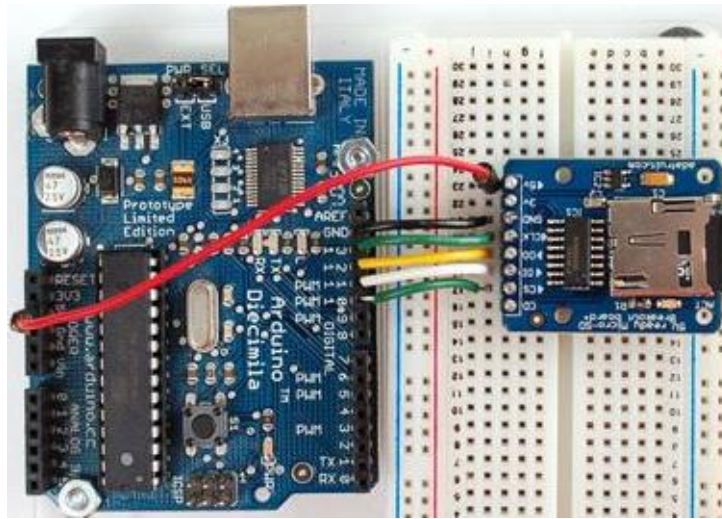


Figura 2.4. Modulo SD Card para Arduino.

La Comunicación del micro ATmega328 ofrece UART TTL (5V) de comunicación en serie disponible en los pines digitales 0 (RX) y 1 (TX). Un ATmega8U2 en los canales de comunicación bordo de esta serie a través de USB y aparece como un puerto COM virtual con el software en el ordenador. El '8 U2 firmware utiliza el estándar de los controladores USB, COM, y no hay ningún controlador externo es necesario. Sin embargo, en Windows, un archivo.inf es necesario. El software de Arduino incluye un monitor de serie que permite simples datos de texto que se envían y reciben la placa Arduino. El RX y TX LED en el tablero parpadea cuando se están transmitiendo datos a través del chip USB a serie y conexión USB al ordenador (pero no para la comunicación de serie en los pines 0 y 1).

Una biblioteca SoftwareSerial permite la comunicación serie con los pines digitales de la ONU.

El ATmega328 también es compatible con I2C de comunicación (TWI) y SPI. El software de Arduino incluye una librería Wire para simplificar el uso del I2C bus, ver la documentación para más detalles. Para la comunicación SPI, el uso de la biblioteca de SPI.

También cuenta con entradas y salidas de propósito general, puertos e/s, con 14 entradas digitales / salidas (de los cuales 6 pueden ser utilizados como salidas PWM), 6 entradas analógicas, un pequeño oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Todo lo necesario para apoyar al micro-controlador.

La **alimentación de la placa** a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente:

La alimentación externa, no USB, puede venir de un adaptador AC-DC con el 2.1mm plug-positivo en toma de alimentación de la placa.

Por batería que se puede insertar en el Gnd y Vin encabezados pin del conector de alimentación, el rango recomendado es 7-12V, si se proporcionan menos de 7V el pin de 5V puede proporcionar menos de 5V y de la junta puede ser inestable, si utiliza más de 12V el regulador de voltaje se puede calentar y dañar la placa. Sin embargo puede funcionar 6-20V pero sin garantías a que funciona correctamente fuera de los valores dichos.

Además contiene pines de alimentación son las siguientes:

VIN. La tensión de entrada a la placa Arduino cuando se utiliza una fuente de alimentación externa (en lugar de 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Puede tensión de alimentación a través de este pin, o, si el suministro de voltaje a través del conector de alimentación, el acceso a través de este pin.

5V. La fuente de alimentación regulada utilizada para alimentar el micro-controlador y otros componentes en el tablero. Esto puede venir ya sea de VIN a través de un regulador a bordo, o se suministra a través de USB o de otra fuente de 5V regulados.

3V3. Salida de 3,3V generados por el regulador a bordo. Con un máximo de 50 mA.

GND. Pines a tierra.

Contiene **14 pines digitales** en el Uno se puede utilizar como una entrada o salida, con pinMode () , digitalWrite () , y digitalWrite (). Cada pin puede proporcionar o recibir un máximo de 40 mA., 5 voltios y tiene una resistencia de pull-up (desconectado por defecto) de 20 a 50 kOhm. Además, algunos pines tienen funciones especializadas:

Serie: 0 (RX) y 1 (TX) Se utiliza para recibir (RX) y transmisión (TX) datos serie TTL. Estos pines están conectados a los pines correspondientes de la ATmega8U2 USB-a-chip de serie TTL.

Las interrupciones externas: 2 y 3 Estos pines pueden ser configurados para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor. Ver el attachInterrupt () funciones para más detalles.

PWM: 3, 5, 6, 9, 10 y 11 ofrecen 8 bits de salida PWM con la función analogWrite ().

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK), Estos apoyo pins comunicación SPI utilizando la biblioteca de SPI.

LED: 13. Hay un built-in LED conectado al pin digital 13. Cuando el pin es de alto valor, el LED está encendido, cuando el pasador está bajo, es apagado.

Integra también **6 entradas analógicas**, con la etiqueta A0 a A5, cada una de las cuales ofrecen 10 bits de resolución (1.024 valores diferentes). Por defecto, medida desde masa a 5V, aunque es posible cambiar el extremo superior de su rango con el pin AREF y el analogReference () función. Además, algunos pines tienen funciones especializadas:

I²C: A4 (SDA) y A5 (SCL). Apoyo I²C (TWI) usando la comunicación librería Wire.

Hay un par de pines en la placa:

AREF. Tensión de referencia para las entradas analógicas. Se utiliza con analogReference ().

Reset. Lleva esta línea LOW para reiniciar el micro-controlador.

2.2.2. Software

La Programación de la placa con un software de Arduino de descarga libre. El ATmega328 en el Arduino Uno viene pre-quemado con un gestor de arranque (bootloader) que le permite cargar nuevo código sin el uso de un programador de hardware externo, se comunica con el original STK500 protocolo (de referencia ,archivos de cabecera C).

También se puede pasar por alto el gestor de arranque y programar el micro-controlador a través de la ICSP (In-Circuit Serial Programming, método de acceso a toda la memoria de programa de un procesador Atmel), el programa básico que escucha al puerto serie y así poder descargar programas desde el IDE, consume 1K de la memoria de programa, pero si no tuviéramos un bootloader, nos haría falta un dispositivo externo, programador para poder descargar el programa en el chip a través del ICSP.

El ATmega8U2 tiene un código fuente disponible del firmware, esta interfaz se carga con DFU, un cargador de arranque que puede ser activado mediante la conexión del puente de soldadura en la parte posterior de la placa (cerca del mapa de Italia) y luego reiniciar el 8U2. Y a continuación, puede utilizar el software FLIP de Atmel (Windows) o el programador DFU (Mac OS X y Linux) para cargar un nuevo firmware. O puede utilizar el encabezado de ISP con un programador externo (sobrescribir el gestor de arranque DFU).

Contiene un **Reset Automático** en lugar de presionar el botón de reset antes de subir, diseñado de que le permite poner a cero por software que se ejecuta en un ordenador conectado.

Una de las líneas de control de flujo por hardware (DTR) de la ATmega8U2 está conectada a la de reset del ATmega328 mediante un condensador de 100nF. Cuando esta línea se afirma (tomado bajo), la línea de reset cae lo suficiente para restablecer el chip. El software de Arduino utiliza esta capacidad que le permite cargar el código con sólo pulsar el botón de subida en el entorno Arduino. Esto significa que el gestor de arranque puede tener un tiempo de espera más cortos, como la reducción de DTR puede ser bien coordinada con el inicio de la subida.

El Arduino Uno contiene una traza que se puede cortar para desactivar el auto-reset. Las almohadillas a ambos lados de la traza se pueden soldar juntas para volver a habilitarlo. Es llamado "RESET-ES". También puede ser capaz de desactivar el reinicio automático al conectar una resistencia de 110 ohm de 5V a la línea de reset.

Incluye también una **protección de USB**, un poli-fusible reseteable que protege a los puertos USB de su computadora de cortos y sobre-corriente. Aunque la mayoría de las computadoras ofrecen su protección interna, el fusible proporciona una capa adicional de protección. Si hay más de 500 mA. se aplica al puerto USB, el fusible automáticamente se rompe la conexión hasta que la corta o se elimina la sobrecarga.

Una vez adquirida la placa Arduino Uno, la primera toma de contacto se identifica cada componente para empezar a configurar, crear pequeños programas y ponerlos en práctica. Para ello, se el software libre de Arduino desde la página oficial del propio micro-controlador www.Arduino.cc y ver los ejemplos incluidos en el programa, para saber cómo se programa.

Entorno de programación

Arduino utiliza para escribir el software lo que denomina "sketch" (*programa*). Estos programas son escritos en el editor de texto. Existe la posibilidad de cortar/pegar y buscar/remplazar texto. En el área de mensajes se muestra información mientras se cargan los programas y también muestra errores. La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones. La barra de herramientas permite verificar el proceso de carga, creación, apertura y guardado de programas, y la monitorización serie, Figura 2.5.:

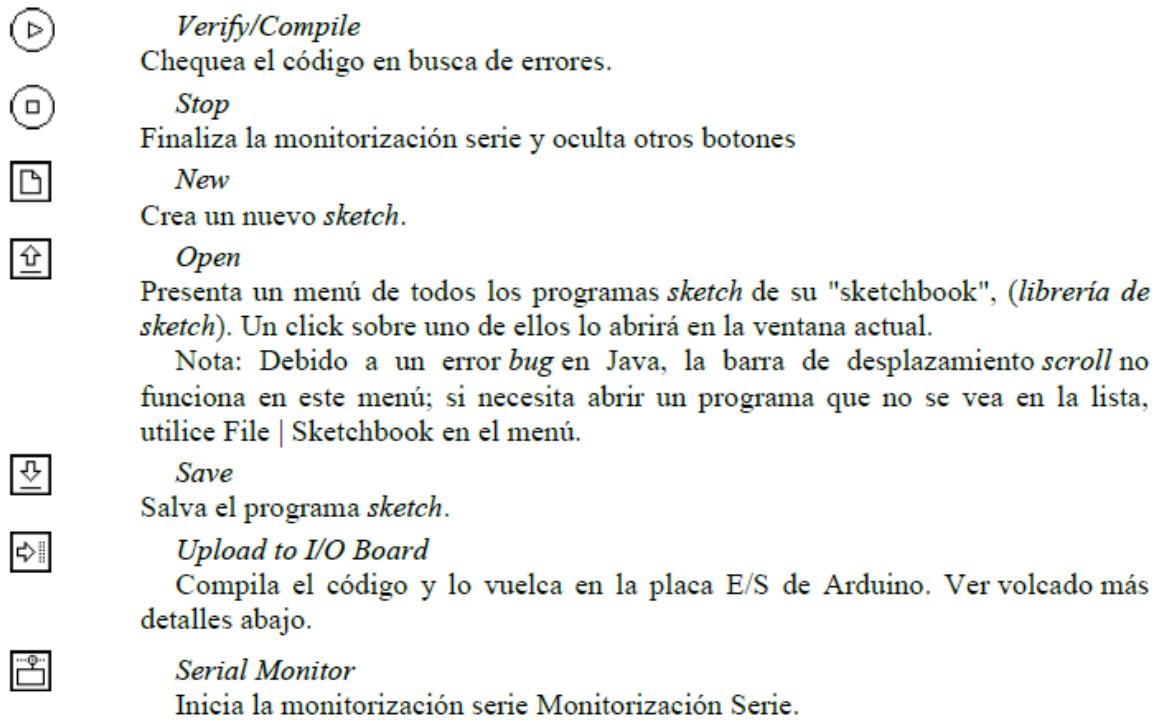


Figura 2.5. Iconos del entorno de programación.

Encontrará otros comandos en los cinco menús: File, Edit, Sketch, Tools, Help. Los menús son sensibles al contexto, lo que significa que estarán disponibles sólo los elementos relevantes para la tarea que esté realizando en ese momento:

-Edit

Copy for Discourse: Copia el código de su sketch en el portapapeles para con el formato adecuado para publicarlo en un foro, incluyendo la sintaxis coloreada.

Copy as HTML: Copia el código de un programa (sketch) al portapapeles en formato HTML, adecuándolo para incrustarlo en una página web.

-Sketch

Verify/Compile: Verifica los errores de su programa (sketch).

Import Library :Añade una librería a su programa (sketch) insertando la sentencia `#include` en el código.

Show Sketch Folder: Abre la carpeta de programas (sketch) en el escritorio.

Add File: Añade un fichero fuente al programa (se incluirá desde su ubicación actual). El fichero aparece en una nueva pestaña en la ventana del programa. Los ficheros pueden ser quitados del programa (sketch) utilizando el menú "tab".

-Tools

Auto Format: Da formato al código proporcionando estética: por ejemplo realiza tabulaciones entre la apertura y cierre de llaves, y las sentencias que tengan que ser tabuladas lo estarán.

-Board

Selecciona la placa que estás usando. Ver más abajo en Descripción de las placas.

Serial Port: Este menú contiene todos los dispositivos de serie (reales o virtuales) de su equipo. Se refrescará automáticamente cada vez que abras el menú tools.

Burn Bootloader: Este elemento del menú le permite grabar un gestor de arranque (bootloader) dentro del micro-controlador de la placa Arduino. Aunque no es un requisito para el normal funcionamiento de la placa Arduino, le será útil si compra un nuevo ATmega (el cual viene normalmente sin gestor de arranque). Asegúrese que ha seleccionado la placa correcta en el menú Boards antes de grabar el bootloader. Cuando use AVR ISP, tendrá que seleccionar en el menú Serial Port el puerto correspondiente.

-Sketchbook (Librería de Sketch)

El entorno de Arduino incluye el concepto de "sketchbook": que es el lugar estándar para el almacenamiento de sus programas (o "sketch"). Los "sketches" dentro de su "sketchbook" pueden abrirse desde el menú File > Sketchbook o desde el botón de la barra de herramientas Open. La primera vez que arranque el software Arduino, se creará un directorio para su "sketchbook". Puede visualizar o cambiar su localización dentro de "sketchbook location" desde el apartado Preferences.

-Tabs, Multiple Files, and Compilation (Pestañas, Ficheros múltiples y compilación)

Permite manejar "sketches" con más de un fichero (cada uno de los cuales aparece en su pestaña). Pueden ser normalmente ficheros de código Arduino (no extensiones), ficheros C (extensiones .c), ficheros c++ (.cpp), o ficheros de cabecera (.h).

-Uploading (Volcado)

Antes de volcar su "sketch", necesitará seleccionar los elementos correspondientes desde los menús Tools > Board y Tools > Serial Port. Las boards (placas) están descritas abajo.

En los Mac, el puerto serie será probablemente algo como /dev/tty.usbserial-1B1 (para una placa USB), o /dev/tty.USA19QW1b1P1.1 (para una placa serie conectada con un adaptador Keyspan USB-to-Serial).

En Windows, probablemente sea COM1 o COM2 (para una placa serie) o COM4, COM5, COM7, o superior (para una placa USB)- para encontrarlos, debes buscar los dispositivos serie USB en la sección de puertos del Administrador de Dispositivos de Windows.

En Linux, debería ser /dev/ttyUSB0, /dev/ttyUSB1 o similar.

Una vez que ha seleccionado el puerto serie y la placa, presione el botón de volcado en la barra de herramientas o seleccione Upload to I/O Board desde el menú File.

Las actuales placas de Arduino se resetearán automáticamente y comenzará el volcado. Como las placas antiguas carecen de auto-reset, necesitará presionar el botón de reset en la placa, justo antes de iniciar el volcado. En muchas placas verá el led RX y TX parpadeando cuando el "sketch" está actualizándose. El entorno de Arduino mostrará un mensaje cuando el volcado esté completado, o mostrará un error.

Cuando se vuelca un "sketch", está utilizando el "bootloader" de Arduino, un pequeño programa que ha sido cargado en el micro-controlador en su placa. Permite el volcado del código sin utilizar hardware adicional. El "bootloader" está activo durante unos segundos cuando la placa es reseteada; después se inicia el "sketch" que más recientemente se hubiera actualizado en el micro-controlador. El "bootloader" produce un parpadeo en el LED de la placa (pin 13) cuando se inicia (p.e. cuando las placas son reseteadas)

-Third-Party Hardware (Hardware de terceros)

Se puede agregar soporte para hardware de terceros en el directorio hardware del directorio "sketchbook". Las plataformas instaladas aquí pueden incluir la definición de las placas (que aparecen en el menú board), librerías del núcleo, "bootloaders", y definiciones de programador. Para instalarla, cree un directorio hardware, y en él descomprima la plataforma de terceros con su directorio. (No utilices "Arduino" como nombre del subdirectorio o sobre-escribirás la plataforma Arduino).

Para desinstalarlo, simplemente borre ese directorio. Y más detalles sobre la creación de paquetes de hardware de terceros visita Páginas de Plataformas en la web de Google Code developers.

-Serial Monitor (Monitor Serie)

Muestra los datos enviados desde la placa Arduino (placa USB o serie). Para enviar datos a la placa, teclee el texto y pulsa el botón "send" o enter. Seleccione la velocidad (baud rate) en el menú desplegable que coincida con el configurado enSerial.begin dentro de su "sketch".

Advertir que en Mac o Linux, la placa Arduino se resetea (su "sketch" es reiniciado desde el principio) cuando conecta con el monitor serie.

Se puede también comunicar con la placa desde Processing, Flash, MaxMSP, etc. (ver Web de interface ('interfacing page') para más detalles)

-Preferences (Preferencias)

Pueden configurarse otras preferencias en el apartado preference (lo podrás encontrar bajo el menú Arduino para los Mac, o en File para Windows y Linux). El resto de opciones puede ser localizado en el fichero de preferencias, que se podrá encontrar dentro del mismo apartado preference.

-Boards (Placas)


La selección de placa tiene dos efectos: los parámetros utilizados cuando compila (por ejemplo, CPU usada y velocidad (baud rate)) y vuelcan los "sketches"; y el fichero y configuración utilizados por el gestor de arranque (*bootloader*) al ser cargado. Algunas de las definiciones de las placas difieren sólo en lo segundo, incluso si ha sido cargado satisfactoriamente con una particular selección, usted podrá comprobarlo antes de grabar el "bootloader".


Para ello este proceso será necesario siempre que queramos utilizar nuestro Arduino, nos referimos a la configuración de la conexión al PC con USB:

- 1) Se selecciona la placa con la que trabaje ==> En menú Tools > Board—Arduino Uno.
- 2) Se busca el puerto que se encuentra el Arduino, en Windows ==> Inicio > Panel de Control > Sistema y mantenimiento > Administrador de dispositivos> Puertos (COM y LPT)>Arduino UNO (COM X).
- 3) Se selecciona el puerto que ha reconocido el PC (COM X) en el software ==> En menú Tools > Serial Port—COM X.

Entonces ahora podrá escribir el código de instrucciones a realizar o controlar por medio del micro-controlador.

De igual forma que la configuración de conexión es esencial, los siguientes pasos también:

- 1) La compilación o verificación del código, se pulsa , en unos segundos en la barra de estado aparecerá un mensaje: Done Compiling y abajo en el panel negro informará si ha sido exitosa la compilación y si ha tenido errores, si no hay errores, en este caso se pasa al siguiente paso, sino tendrá que rectificar el código.

- 2) Cargar o subir el código compilado correctamente al micro-controlador una vez configurado bien las conexiones, se pulsa  en unos pocos segundos (los LEDs Rx y Tx de la placa se iluminarán). Si la carga es correcta, el mensaje que aparecerá en la barra de estado: Done Uploading.

En el caso que crea que el dispositivo no funciona correctamente, se podrá hacer uso del programa facilitado por Arduino, el Blink ==> File > Sketchbook > Examples > Digital > Blink. Éste enciende el led del pin13 durante un segundo y lo apaga otro segundo así repetidamente:

```
Blink // El led situado en el pin 13 se encenderá un segundo y se apagará otro
void setup () { // inicialice el pin digital como una salida.
  pinMode (13, OUTPUT);
}
void loop () { // función del programa
  digitalWrite (13, HIGH); // LED encendido
  delay (1000); // espera un segundo
  digitalWrite (13, LOW); // LED apagado
  delay (1000); // espera un segundo
}
```

Si realmente hace lo que nuestro código pretendía, significa que tanto la configuración de las conexiones, la compilación y carga del programa son correctos.

Arduino admite la mayor parte de librerías usadas comúnmente en C++ y todas sus estructuras básicas, pero además la tarjeta de control nos proporciona funciones específicas para Arduino que son de gran ayuda al programador. A continuación se comenta la estructura de programa y las funciones específicas más relevantes en nuestro caso.

-setup () (Configuración)

La función setup () se llama cuando se inicia un programa. Se usa para inicializar las variables, los modos de contactos, comenzar a usar las bibliotecas, etc. La función de configuración sólo se ejecutará una vez, después de cada momento del encendido o reinicio de la placa Arduino.

-loop () (Bucle)

Después de crear una función `setup ()`, que inicializa y establece los valores iniciales, el bucle `()` hace precisamente lo que sugiere su nombre, y los bucles de forma consecutiva, permitiendo que su programa para cambiar y responder. Se usa para controlar activamente la placa Arduino.

-pinMode ()

Configura un pin como entrada o salida. Para utilizarla, le pasas el número del pin que vas a configurar y la constante `INPUT` u `OUTPUT`. Es decir configura el pin especificado para comportarse como una entrada o una salida. Se usa dentro de la función `setup ()`. Sintaxis `->pinMode (pin, modo)`.

-digitalWrite ()

Escribe o envía un valor `HIGH` o `LOW` hacia un pin digital. Por ejemplo, la línea: `digitalWrite (ledPin, HIGH)`;

-digitalRead ()

Lee el valor de un pin digital especificado, `HIGH` o `LOW`.

Sintaxis-> `digitalRead (pin)`.

Parámetros->pin: el número de pin digital que quieres leer (int).

Devuelve `HIGH` o `LOW`.

-delay ()

Pausa el programa por un tiempo determinado (en milisegundos) especificado por un parámetro. Hace a Arduino esperar por el número especificado de milisegundos antes de continuar con la siguiente línea. Hay 1000 milisegundos en un segundo, por lo que la línea siguiente crea un retraso de un segundo: `delay (1000)`.

- attachInterrupt (interrupción, función, modo)

Especifica la función a la que invocar cuando se produce una interrupción externa. Reemplaza cualquier función previa que estuviera enlazada a la interrupción. La mayoría de las placas Arduino tienen dos interrupciones externas: Las número 0 (en el pin digital 2) y la 1 (en el pin digital 3).

Parámetros:

Interrupción: el número de la interrupción (int).

Función: la función a la que invocar cuando la interrupción tiene lugar; esta función no debe tener parámetros ni devolver nada. Esta función es a veces referenciada como rutina de interrupción de servicio.

Modo define cuando la interrupción debe ser disparada. Hay cuatro constantes predefinidas como valores válidos:

LOW para disparar la interrupción en cualquier momento que el pin se encuentre a valor bajo (LOW).

CHANGE para disparar la interrupción en cualquier momento que el pin cambie de valor.

RISING para disparar la interrupción cuando el pin pase de valor bajo (LOW) a alto (HIGH).

FALLING para cuando el pin cambie de valor alto (HIGH) a bajo (LOW).

-detachInterrupt (interrupt)

Apaga la interrupción dada. Interrupt: el número de interrupción a invalidar (0 o 1).

-noInterrupts ()

Desactiva las interrupciones (pueden reactivarse usando interrupts ()). Las interrupciones permiten que las operaciones importantes se realicen de forma transparente y están activadas por defecto. Algunas funciones no funcionarán y los datos que se reciban serán ignorados mientras que las interrupciones estén desactivadas. Las interrupciones pueden perturbar ligeramente el tiempo de temporizado, sin embargo puede que sea necesario desactivarlas para alguna parte crítica del código.

- interrupts ()

Activa las interrupciones (después de haberlas desactivado con noInterrupts ()). Las interrupciones permiten que se ejecuten ciertas tareas en segundo plano que están activadas por defecto. Algunas funciones no funcionarán correctamente mientras las interrupciones estén desactivadas y la comunicación entrante puede ser ignorada. Las interrupciones pueden perturbar ligeramente la temporización en el código y deben ser desactivadas sólo para partes particularmente críticas del código.

2.2.3. Ventajas e inconvenientes generales de Arduino

En Arduino se programa en el lenguaje C++, resulta más complicado, pero a su vez es mucho más versátil, se puede programar casi cualquier funcionamiento que se desee.

Entre las ventajas que tiene Arduino podríamos nombrar:

La única conexión para funcionar, es PC-Arduino a través de un USB convencional.

El hardware puede ser construido por uno mismo.

El software es libre y de continua actualización gratuita.

El precio económico de la compra de la tarjeta Arduino.

Sus dimensiones son pequeñas.

Mantenimiento económico.

Posibilidad de añadir escudos para funciones específicas.

No necesita puesta a punto ni instalación.

Y sobre las desventajas:

El control del sistema no es en tiempo real.

No es un robusto necesita circuito acondicionamiento de tensión y corriente.

2.3. Autómatas programables

Se trata de un sistema con un hardware estándar, con capacidad de conexión directa a las señales de campo (niveles de tensión y corriente industriales, transductores periféricos electrónicos) y programable por el usuario que hace de unidad de control incluyendo total o parcialmente las interfaces con las señales de proceso.

Otra definición más simple, entiende por controlador lógico programable (PLC), o autómatas programables, a toda máquina electrónica diseñada para controlar en tiempo real y en medio industrial procesos secuenciales.

Un autómatas programables suele emplearse en procesos industriales que tengan una o varias de las siguientes necesidades:

Espacio reducido.

En ambientes exigentes o agresivos.

Procesos de producción cambiantes.

Procesos secuenciales.

Maquinaria de procesos variables.

Instalaciones de procesos complejos.

Chequeo de programación centralizada de las partes del proceso.

Aplicaciones generales:

Maniobra de máquinas e instalaciones.

Señalización y control.

Esto se refiere a los autómatas programables industriales, dejando de lado los pequeños autómatas para uso más personal (que se pueden emplear, incluso, para automatizar procesos en el hogar, la puerta de una cochera, las luces de la casa).

2.3.1. Componentes

Diferenciando la parte externa con la interna del autómatas programables tendríamos:

Estructura Externa

En cuanto a su estructura, todos los autómatas programables se clasifican en:

- Compactos: en un solo bloque están todos los elementos.
- Modulares: separan por unidades las distintas partes operativas.

Y, en este caso, se distingue entre dos estructuras:

- Americana: separa las E/S del resto del autómata.
- Europea: cada módulo es una función (fuente alimentación, CPU, E/S...)

Exteriormente se encuentra con cajas que contienen una de estas estructuras, las cuales poseen indicadores y conectores en función del modelo y fabricante.

Para el caso de una estructura modular se dispone de la posibilidad de fijar los distintos módulos en raíles (o *racks*) normalizados, para que el conjunto sea compacto y resistente.

Los micro-autómatas pueden servirse sin caja, en formato kit, ya que su empleo no es determinado y se suele incluir dentro de un conjunto más grande de control o dentro de la misma maquinaria que se debe controlar.

Estructura Interna

Los elementos esenciales de un autómata programable son los que muestra la Figura 2.6:

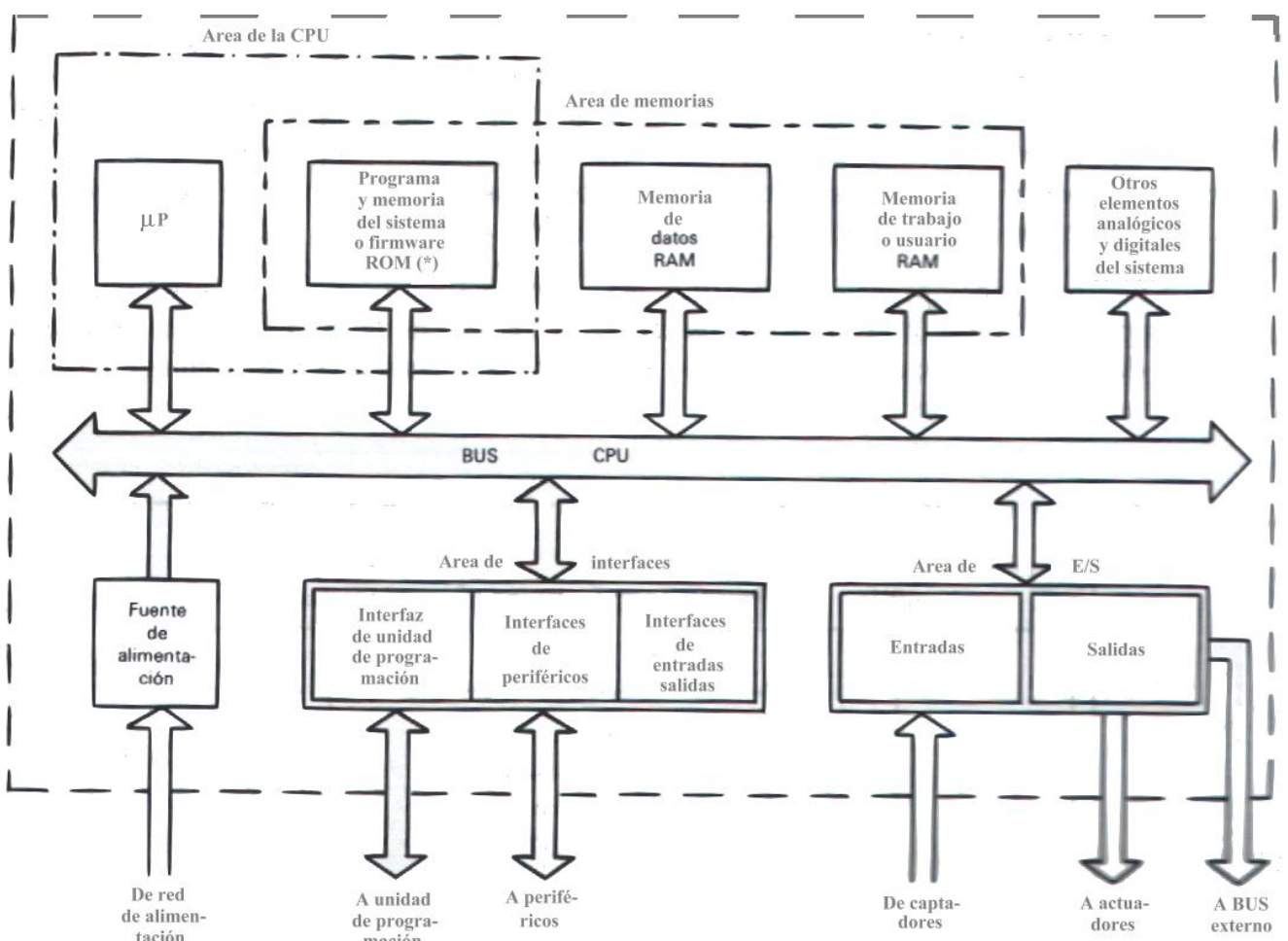


Figura 2.6. Estructura interna del autómata.

- En lo que respecta a **las Entradas/Salidas del Propósito General**:

Sección de entradas: se trata de líneas de entrada que se le conectará los sensores o captadores y su información es enviada a la CPU para ser procesada de acuerdo al programa. Se pueden diferenciar dos tipos de captadores conectables al módulo de entradas:

Captadores Pasivos, aquellos que cambian su estado lógico, activado/no activado por medio de una acción mecánica (interruptores, pulsadores, finales de carrera...).

Captadores Activos, dispositivos electrónicos que necesitan ser alimentados por una tensión para que varíen su estado lógico, hay diferentes tipos de detectores (Inductivos, Capacitivos, Fotoeléctricos). Muchos de estos pueden ser alimentados por la propia fuente de alimentación del autómata.

Sección de salidas: una serie de líneas que se le conectará actuadores como por ejemplo:

Relés, usados en circuitos de corriente continua y alterna. Están basados en la conmutación mecánica, por la bobina del relé, de un contacto eléctrico normalmente abierto, como se muestra en la figura 2.7.:

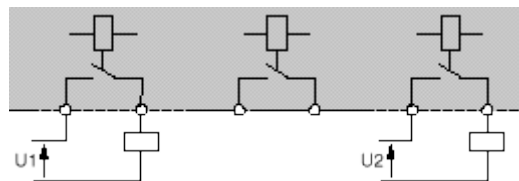


Figura 2.7. Relés.

Triacs, se utilizan en circuitos de corriente continua y alterna que necesiten maniobras de conmutación muy rápidas. Un ejemplo en la figura 2.8.:

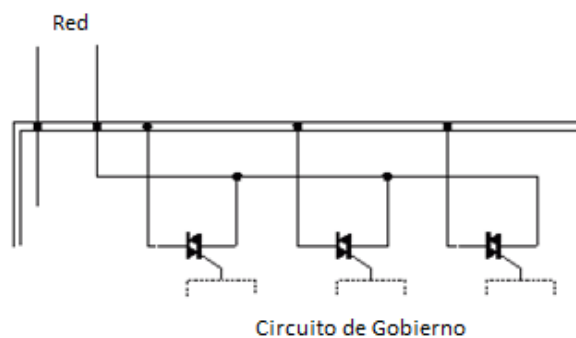


Figura 2.8. Triac.

Transistores a colector abierto, el uso de este tipo de módulos es exclusivo de los circuitos de corriente continua. Igual que en los Triacs, es utilizado en circuitos que necesiten maniobras de conexión/desconexión muy rápidas, como se muestra en la Figura 2.9.:

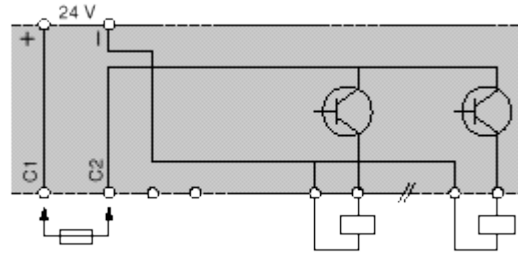


Figura 2.9. Transistores

En ambas secciones dispone de dos tipos:

Digital. Las E/S digitales se basan en el principio de todo-nada (tipo relé o diodo), se manejan a nivel de bit dentro del programa de usuario.

Analógica. Las E/S analógicas pueden poseer cualquier valor dentro del rango especificado por el fabricante. Se basan en conversores A/D-D/A aislados de la CPU. Estas señales se manejan a nivel de byte o palabra (8/16 bits) dentro del programa de usuario.

Las E/S son leídas y escritas dependiendo del modelo y fabricante, pueden estar incluidas sus imágenes dentro del área de memoria o ser manejadas a través de instrucciones específicas.

- **Unidad central de proceso (CPU):** es el cerebro del sistema, se encarga de procesar e interpretar el programa de usuario ante los cambios detectados por la sección de entrada, una vez procesado, se envía la respuesta al modulo de salida. Para ello, se dispone de diversas zonas de memoria, registros, e instrucciones de programa. Adicionalmente, algunos modelos disponen de funciones ya integradas en la CPU (reguladores PID, control de posición...).

Entre sus funciones a destacar son:

Crear una imagen llegada de las entradas.

Watchdog, vigila el tiempo de ejecución del programa usuario para no exceder el tiempo máximo (tiempo de ciclo máximo) y evitar una detención por un bucle infinito...

Ejecutar el programa de usuario.

Actualizar los contadores y temporizadores programados.

Envía las señales determinadas por el programa a la sección de salida.

Renovar el estado de las salidas en función de la imagen de las mismas obtenida al final del ciclo de ejecución del programa de usuario.

Chequeo del sistema.

Para ello posee un ciclo de trabajo que se ejecuta de forma continua Figura 2.10.:

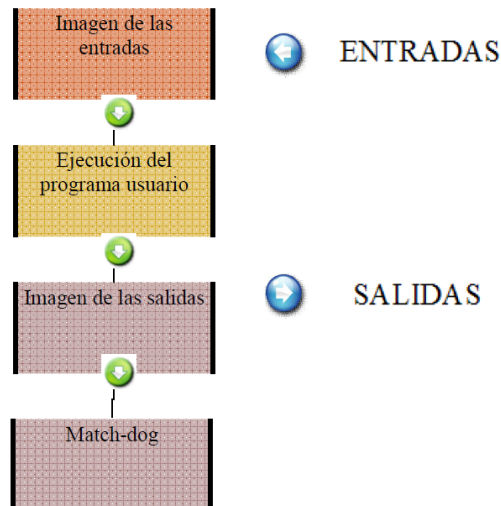


Figura 2.10. Ciclo trabajo Autómata

- **Memoria:** En los autómatas está separada en diversas áreas según su función o datos que debe contener.

Memoria del programa, es la encargada de contener el programa de usuario y de trabajo que se va a ejecutar cíclicamente.

Memoria interna, contiene datos intermedios de los cálculos realizados así como variables internas y una imagen de las entradas y salidas. Tanto las entradas como las salidas están aisladas de la CPU según el tipo de autómata que utilizemos. Normalmente se suelen emplear opto-acopladores en las entradas y relés/opto-acopladores en las salidas.

Memoria de datos: se suele subdividir en zonas según el tipo de datos (como temporizadores, contadores, etc.).

Memoria del sistema: aquí se encuentra el programa en código máquina que monitoriza el sistema (programa del sistema o firmware). Este programa es ejecutado directamente por el micro-procesador/micro-controlador que posea el autómata.

Memoria auxiliar: se trata de memoria externa que empleamos para almacenar el programa de usuario, y en ciertos casos parte de la memoria de la tabla de datos.

La memoria puede ser accesible bit a bit o en palabras de 8-16 bits. Cada autómata divide su memoria de esta forma genérica, haciendo subdivisiones específicas según el modelo y fabricante.

- **Unidad de alimentación** (algunas CPU la llevan incluida) encargada de convertir la tensión de la red, 220V de corriente alterna a baja tensión de corriente continua, normalmente 24V, siendo la tensión de trabajo en los circuitos electrónicos que forma el Autómata.

- **Unidad o consola de programación:** que permite introducir, modificar y supervisar al operario con el programa de usuario. Permiten la transferencia y modificación de programas, verificación de la programación, información del funcionamiento de los procesos. Hoy en día existen terminales de programación portátiles a parte de los compatibles de PC.

A parte de todo ello, añaden lo que se conoce como **bus interno** un conjunto de líneas y conexiones que permiten la unión eléctrica entre la unidad de control, las memorias y las interfaces de entrada y salida.

Un bus se compone de un conjunto de líneas utilizadas para intercambiar datos u órdenes. Permite minimizar el número de conexiones entre subsistemas y el acceso de los dispositivos al mismo es controlado por la unidad de control.

Los tres buses característicos de un sistema digital son:

- Bus de datos, por el que tienen lugar las transferencias de datos del sistema.
- Bus de direcciones, en él se direccionan la memoria y el resto de los periféricos.
- Bus de control, constituido por todas las conexiones destinadas a gobernar los intercambios de información, se reunifican en el autómata en uno sólo, que recibe el nombre de bus interno.

El número de líneas del interno depende de cada fabricante.

Se considera también como bus del autómata cualquier conexión entre bloques o módulos que no necesite de procesadores específicos de comunicaciones en sus extremos, como, por ejemplo, el cable de conexión entre el autómata y una unidad externa de expansión de E/S.

Todo autómata, salvo casos excepcionales, posee la virtud de poder comunicarse con otros dispositivos (como un PC o un MODEM), a esto se le llama **Interfaces**. Lo normal es que posea una E/S serie del tipo RS-232 / RS-422. A través de esta línea se pueden manejar todas las características internas del autómata, incluido la programación del mismo, y suele emplearse para monitorización del proceso en otro lugar separado.

El autómata programable normalmente puede ser ampliable. Las ampliaciones abarcan un abanico de posibilidades, desde **dispositivos periféricos**, redes internas (LAN, buses de campo), módulos auxiliares E/S, memoria adicional...hasta la conexión con otros autómatas del mismo modelo. Cada fabricante facilita las posibilidades de ampliación de sus modelos.

2.3.2. Instalación

Dadas las características constructivas y de diseño de los autómatas programables, su instalación es viable en prácticamente cualquier ambiente industrial siempre que no se sobrepasen las especificaciones dadas por el fabricante. No obstante, existen ciertas recomendaciones para asegurar un correcto funcionamiento del sistema, que atañen a las condiciones de temperatura, humedad y la inmunidad frente a interferencias eléctricas.

En general, se monta en un armario de maniobra de dimensiones adecuadas para poder modificar los componentes del equipo y el resto de elementos cómodamente, y si puede ser de armarios metálicos, ya que minimizan los efectos de la radiación electromagnética generada por equipos de conmutación posibles en las inmediaciones. Para la instalación, se seguirán las normas y reglamentos vigentes de aplicación habitual en cualquier instalación eléctrica de control.

La convección natural es suficiente ya que la mayoría de los fabricantes preparan los autómatas para que trabajen a una temperatura máxima de 60º.

Situación de los Componentes

Los componentes del autómata se montan siguiendo las recomendaciones del fabricante y las siguientes pautas de aplicación general:

- Montaje vertical de los componentes para facilitar la disipación del calor.
- Las fuentes de alimentación deberán ocupar una posición por encima del resto de componentes, ya que son generadores de calor.
- La unidad central ocupará una posición adyacente o por debajo de las fuentes de alimentación, quedando a una altura que facilite su inspección.
- Los racks de E/S estarán dispuestos de la forma más conveniente para el acceso y cableado, en el espacio libre.
- Se dejarán espacios suficientes entre los componentes y entre estos y la envolvente para una adecuada disipación del calor.
- Para el resto de componentes suelen estar en posiciones alejadas del equipo y más si son componentes electromecánicos, para minimizar las interferencias electromagnéticas.

Cableado

Siempre que sea posible se intentará agrupar los módulos por categorías (entradas/salidas, tensión alterna o continua, señales discretas o analógicas).

Esto permite un cableado racional y una necesaria segregación de los cables de señal débil respecto a los que alimentan cargas, y de los de comunicaciones. Se separarán los cables de CC/CA, para minimizar las interferencias producidos por la conmutación de cargas y también los cables de interconexión de racks y de comunicaciones se separan completamente de otros.

Puesta a Tierra

Se seguirá lo especificado en la normativa vigente y las recomendaciones del fabricante, pero hay que recordar que cada una de las estructuras (racks) del autómata, debe estar unida mediante un cable independiente de sección adecuada, a la pletina de tomas de tierra del armario. Nunca compartir circuitos de tierra entre racks o con otros componentes del sistema.

Circuitos de Seguridad

Los dispositivos de parada de emergencia se instalarán con independencia del autómata, para permitir la parada del sistema aún en caso de avería del mismo; en general, deben actuar sobre un contacto de maniobra que corta la alimentación a las cargas de la instalación.

Circuitos y Consideraciones de los Dispositivos de E/S

En general, o por lo menos para los dispositivos de salida, es deseable que exista un contacto de maniobra que permita cortar la alimentación de estos y que hará posible trabajar con seguridad en la puesta a punto o investigación de averías, con el autómata alimentado.

Las consideraciones sobre la instalación de los dispositivos electrónicos de detección, como elementos de entrada, en general, el problema se reduce a que el indicador de entrada se ilumina tenuemente, pero en ocasiones, cuando la corriente residual es elevada, o dependiendo de los umbrales de disparo del circuito de entrada pueden darse señales falsas.

Cuando los dispositivos de entrada trabajan a niveles de señales débiles como TTL, analógicas, termopares...hay que realizar conducciones de cableado separadas para evitar el problema de la inducción, para evitar las interferencias electromagnéticas, se recomienda la instalación mediante cables trenzados y apantallados.

Los circuitos de salida controlan habitualmente cargas inductivas (solenoides), que provocan la aparición de picos de tensión cuando se interrumpe el circuito de alimentación (descarga del circuito inductivo). Estas crestas, que pueden alcanzar varios centenares de voltios, deben ser suprimidas, ya que pueden averiar los circuitos de salida (estáticos) y provocar interferencias en todo el sistema. Los fabricantes suelen incorporar supresores de transitorios en los circuitos de los módulos de salida pero a veces no son suficientes para evitar anomalías.

En general los módulos de salida incorporan circuitos fusibles de protección dimensionados adecuadamente a las características nominales de la salida (transistor, triac); si no es así, hay que instalarlos en el exterior (regleta de bornes) teniendo en cuenta las especificaciones del fabricante ya que no protegerán adecuadamente la salida en caso de sobrecarga si no están bien dimensionados.

Alimentación

El empleo de transformadores separadores de alimentación ya que proporcionan una buena protección frente a interferencias introducidas en las líneas por la conmutación de cargas importantes existentes en la instalación. Además es deseable que los dispositivos de E/ S se alimenten de la misma línea que el autómatas, ya que la fuente de alimentación del mismo posee circuitos de detección de nivel de tensión que provocan la secuencia de parada del equipo en caso de anomalía en la red, y de este modo se evitarán las falsas lecturas de señal de entrada.

Algunos autómatas incorporan una fuente auxiliar de 24Vcc para uso externo de los dispositivos de entrada sobre módulos de entrada a 24Vcc.

Hay que vigilar que no supere la capacidad de esta fuente, particularmente cuando se alimentan de ella dispositivos estáticos y deben seguir las recomendaciones de cableado del fabricante para minimizar la posibilidad de interferencia sobre estos circuito.

En caso de que se prevea la existencia de variaciones de tensión en la línea de alimentación que puedan superar los márgenes de trabajo especificados para el equipo, habrá que instalar transformadores estabilizadores, para evitar frecuentes paradas del sistema; en estas circunstancias es mejor alimentar las salidas del autómatas directamente desde la línea de entrada para descargar el transformador permitiendo que sea de una menor potencia.

2.3.3. Puesta a punto

Una vez montado e instalado el equipo, cargado el programa en la memoria de la Unidad Central, hay que poner en marcha el sistema para comprobar que responde adecuadamente a la descripción de la tarea de control original, y en su caso realizar las correcciones y mejoras oportunas.

Antes de dar alimentación, hay que hacer una serie de comprobaciones rutinarias pero importantes:

1.- Comprobar que todos los componentes del Autómata están en su lugar perfectamente insertados en sus conectores y asegurados.

2.- Comprobar que la línea de alimentación está conectada a los correspondientes terminales de la fuente de alimentación del equipo, y que se distribuye adecuadamente a los módulos de entrada y salida (si procede).

3.- Verificar que los cables de interconexión entre racks están correctamente instalados.

4.- Verificar que los cables de conexión a periféricos están correctamente instalados.

5.- Verificar que las conexiones de los bornes de E/S están firmes y corresponden al esquema de cableado.

6.- Verificar que las conexiones a los módulos de E / S están firmes y corresponden al esquema de conexiones.

Previo al ensayo de funcionamiento según lo programado, hay que comprobar que los dispositivos de E/S funcionan correctamente,

a) Con el equipo en PARO (STOP, HALT, DISABLE, TEST, etc. dependiendo del modelo) aplicar tensión al sistema.

b) Verificar que los indicadores de diagnóstico de la Unidad Central reflejan una situación correcta.

c) Comprobar que los paros de emergencia actúan correctamente.

d) Accionar los dispositivos de entrada manualmente y verificar que su estado es registrado por el Autómata; el funcionamiento se puede seguir en los indicadores de los módulos y también se puede seguir visualizando la tabla de E/S mediante un equipo de programación.

Para la comprobación de los dispositivos de salida, hay que cortar la alimentación de las cargas que pudieran dar lugar a situaciones peligrosas y verificar con el procesador en MARCHA (RUN) que las salidas se activan. Esta comprobación resulta más fácil si se utiliza un terminal de programación en el modo 'forzado de E/S' para activar o desactivar las salidas una a una.

Una vez finalizadas todas las comprobaciones anteriores, hay que introducir el programa en la memoria de la Unidad Central y dar alimentación al sistema. Se recomienda que siempre que sea posible, las pruebas de funcionamiento se hagan por áreas, particularmente si se trata de sistemas grandes, dejando fuera de servicio los componentes de las áreas que no se prueban; esto puede realizarse cortando la alimentación de campo de los racks de E/S o inhibiendo su funcionamiento, incluyendo las oportunas instrucciones en el programa (MCR) que se eliminarán una vez concluidas las pruebas.

Verificadas y corregidas las distintas secuencias, el sistema puede arrancar en automático debiendo funcionar correctamente si todas las comprobaciones se han efectuado con éxito. Las correcciones efectuadas, tanto en la instalación como en el programa deben ser documentadas inmediatamente, y se obtendrán copias del programa definitivo (copia, en disco o cinta) tan pronto como sea posible.

2.3.4. Autómata Programable S7-200. CPU 224 AC/DC/Relé

Para el proyecto se ha utilizado un autómata programable S7-200, de modelo CPU 224. Los autómatas programables pertenecientes a la serie S7-200 son PLCs de gama baja. Gracias a su diseño compacto, su capacidad de ampliación y su amplio juego de operaciones están especialmente indicados para solucionar tareas de automatización sencillas. En la Figura 2.11. se muestra una CPU de la serie S7-200.

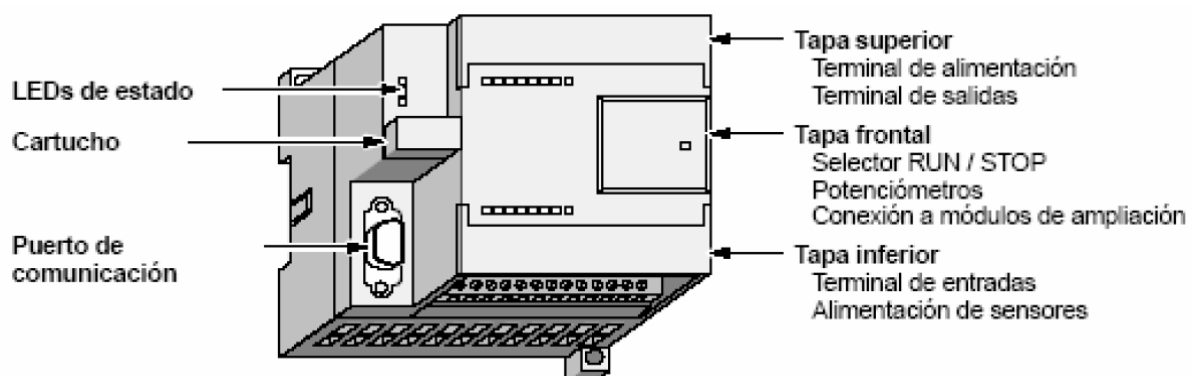


Figura 2.11. Vista general de una CPU224.

La CPU 224 AC/DC/Relé se alimenta a tensión alterna de 100 a 230 V, proporciona tensión continua 24V y 5V y sus salidas son de tipo Relé (de ahí la terminología AC/DC/Relé). Consta de los siguientes elementos:

- Una unidad central de procesamiento o CPU.
- Una fuente de alimentación integrada con salida 24VDC para sensores y actuadores y 5VDC para alimentación de módulos de ampliación.
- LEDs de estado. Se contemplan tres estados:
 - RUN
 - STOP
 - FALLO
- Selector de estado. Tres posiciones:
 - RUN: ejecución del programa de usuario.
 - STOP: la CPU no procesa ningún programa de usuario.
 - TERM: posición para controlar el estado de la CPU por software.
- Ranura para cartucho de memoria.
- Puerto de comunicación integrado.
- 14 entradas digitales a 24VDC.
- 10 salidas digitales tipo relé a 24VDC/24 a 230 VAC.
- 2 potenciómetros analógicos con resolución 8bits asociados a dos áreas de memoria internas

En cuanto a características técnicas de esta CPU hay que destacar los siguientes puntos importantes para este proyecto:

- Comunicación integrada. Puerto de comunicación integrado para comunicaciones PPI/MPI/Freeport. En este proyecto se utilizan la comunicación modo MPI y la comunicación modo Freeport.
- Mapa de memoria. Número de entradas, salidas, contadores, temporizadores, marcas, bloques de datos, funciones integradas. son datos relevantes a la hora de programar una CPU 224.

• Marcas especiales. Las marcas especiales permiten intercambiar datos entre la CPU y el programa. Dichas marcas se puede utilizar para seleccionar y controlar algunas funciones especiales de la CPU 224. En este proyecto se utilizan:

- SM0.0. – Marca en estado “1” siempre.
- SM0.1. – Marca que pasa a estado “1” sólo en el primer ciclo de la CPU.
- SMB30. – Registros de control modo Freeport.
- SMB87, SMB88, SMB92, SMB94. – Control de recepción de mensajes.
- SMB226, SMB228. – Datos del estado módulo EM 277.

Para el control de la unidad funcional el uso de una consola S7-200 de SIMATIC-SIEMENS como la que se muestra en la figura 2.3.4.b) compuesta por un PLC S7-200 de SIEMENS, las entradas y salidas al PLC, el fusible de 2A y el interruptor de encendido. Observar en la siguiente Figura 2.12. que el módulo del PLC presenta el cableado de las entradas y salidas.

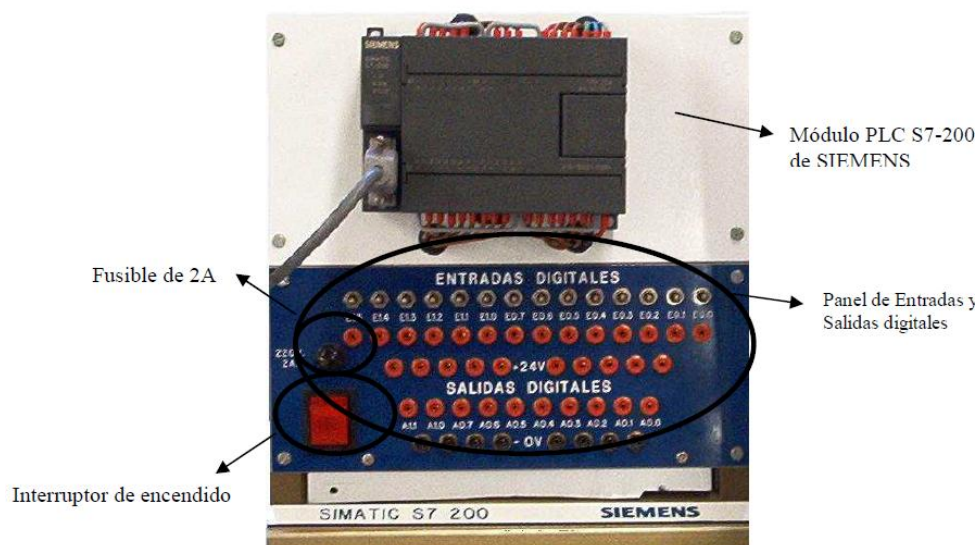


Figura 2.12. Simatic S7-200

2.3.5. Comunicación PC/S7-200

La comunicación PPI sólo es posible con un automático SIMATIC S7-200. Una conexión PPI es un enlace punto a punto. El panel de operador es el maestro y el automático SIMATIC S7-200 actúa de esclavo.

Al panel de operador puede conectarse como máximo un autómata SIMATIC S7-200. El panel de operador se conecta a través del puerto serie de la CPU. A un autómata SIMATIC S7-200 pueden conectarse varios paneles de operador. No obstante, el autómata SIMATIC S7-200 sólo puede establecer un enlace a la vez, y al ser esclavos responden a las peticiones de los maestros.

Para la comunicación PPI se utiliza el cable de la Figura 2.13.



Figura 2.13. Cable de comunicación PI/PPI.

Es el cable de comunicación para la serie S7-200. Transforma RS 485 en RS 232, es decir, transforma el interface PPI (Point to Point Interface) del autómata en RS 232 para comunicación con un PC, impresora, MODEM, etc.

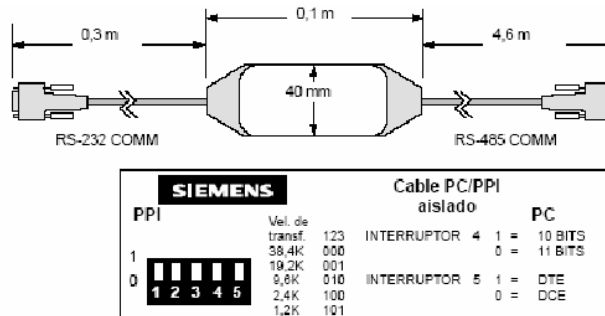


Figura 2.14. Cable PC/PPI Dil

Las principales características para este tipo de cable, en relación con este proyecto son:

- Velocidades de transferencia desde 1.2 Kbit/s hasta 38.4 Kbit/s.
- Posibilidad de conexión en modo DTE ó modo DCE.
- Posibilidad de conexión a MODEM 10/11 bits.
- No soporta conexión MPI/PROFIBUS, solo puede utilizarse con CPUs de S7 200.

Es muy importante darle al ordenador y al autómata direcciones unívocas, en este caso el PC tiene dirección 1 y el autómata dirección 2.

Para pasar del autómatas al MicroWin hay que configurar los parámetros de comunicación, por un lado hay que acceder al panel de control y hacer doble clic en “ajustar interface PG/PC”, y nos aparecerá una ventana como la de la Figura 2.15., aquí se le da la dirección al autómatas, se configura la velocidad de transferencia y se define el punto de acceso, que para los autómatas de Siemens siempre hay que poner el S7ONLINE, también hay que definir la interface de comunicación que para el S7-200 siempre tiene que ser PPI.

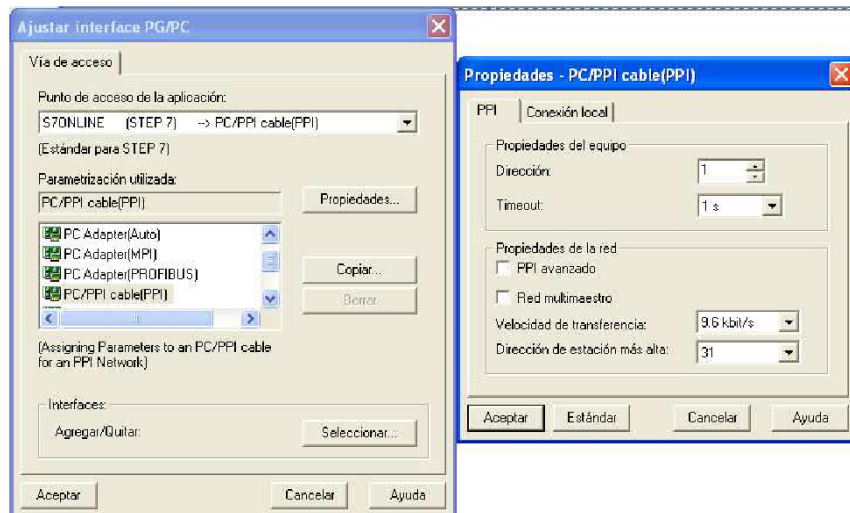


Figura 2.15. Ajuste de interfaz.

Para comunicar el S7-200 se utiliza el cable PC/PPI como la Figura 2.16., hay que tener cuidado porque a ese cable tiene un interruptor DIP, el cual dependiendo de cómo se encuentre, el cable tendrá una velocidad de transferencia u otra, en nuestro caso tiene que estar a 9.600 bit/s.

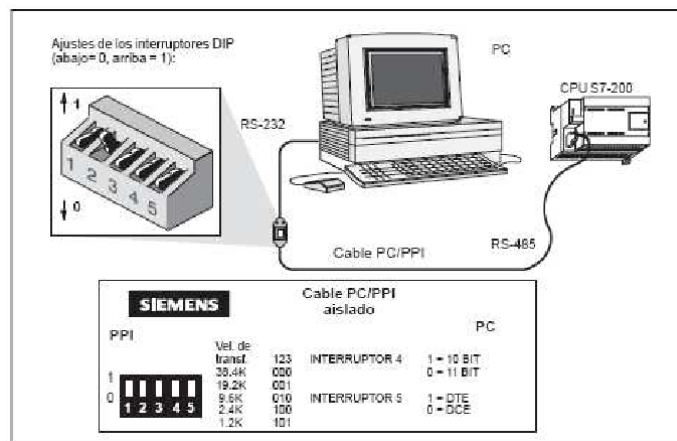


Figura 2.16. Comunicación PPI entre PC y autómatas.

2.3.6. Software

El autómata debe disponer de alguna forma de **programación**, la cual en su día se encargaron que la comunicación hombre-máquina debiera de ser similar a la utilizada hasta ese momento, ser interpretado con facilidad por los técnicos y han ido evolucionando con el tiempo. Para ello se suele realizar empleando alguno de los siguientes elementos como hemos nombrado anteriormente en él su parte hardware:

- **Unidad de programación:** suele ser en forma de calculadora, es la forma más simple de programar el autómata y se suele reservar para pequeñas modificaciones del programa o la lectura de datos en el lugar de colocación del autómata.
- **Consola de programación:** es un terminal a modo PC que proporciona una forma cómoda de realizar el programa de usuario y observar parámetros internos del autómata. Su alto coste y la ubicuidad del PC portátil han relegado su uso.
- **PC:** es el modo más potente y empleado en la actualidad. Permite programar desde un ordenador personal estándar, con todo lo que ello supone: herramientas más potentes, posibilidad de almacenamiento en soporte magnético, impresión, transferencia de datos, monitorización mediante software SCADA, etc.

Para cada caso el fabricante proporciona lo necesario, bien el equipo o el software/cables adecuados. Cada equipo puede poseer una conexión a uno o varios de los elementos anteriores. En el caso de los micro-PLC se escoge la programación por PC o por unidad de programación integrada en la propia CPU a través de lenguajes de programación, los más significativos son:

Lenguaje a contactos. (LD), es el más parecido al utilizado por un electricista al elaborar cuadros de automatismos. Muchos autómatas incluyen módulos especiales de software para poder programar gráficamente de esta forma, como el ejemplo de la Figura 2.17.:

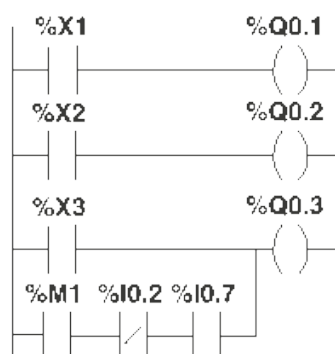


Figura 2.17. Lenguaje de Contactos

Lenguaje por Lista de Instrucciones. (IL), en los autómatas de gama baja, es el único modo de programación. Consiste en elaborar una lista de instrucciones o nemónicos que se asocian a los símbolos y su combinación en un circuito eléctrico a contactos. También decir, que este tipo de lenguaje es, en algunos los casos, la forma más rápida de programación e incluso la más potente. En la siguiente Figura 2.18. se muestra un ejemplo:

```

000 LD %I0.1 Bp. inicio ciclo
    AND %I0.0 Dp. presencia vehículo
    AND %M3 Bit autorización reloj calendario
    AND %I0.5 Fc. alto rodillo
    AND %I0.4 Fc. detrás pórtico
005 S %M0 Memo inicio ciclo
    LD %M2
    AND %I0.5
    OR %I0.2 Bp. parada ciclo
    R %M0
010 LD %M0
    ST %Q0.0 Piloto ciclo
    
```

Figura 2.18.Lenguaje por Lista de Instrucciones.

El **texto estructurado (structured text o ST)**, un lenguaje de alto nivel estructurado por bloques que posee una sintaxis parecida al PASCAL. El ST puede ser empleado para realizar rápidamente sentencias complejas que manejen variables con un amplio rango de diferentes tipos de datos, incluyendo valores analógicos y digitales. También se especifica tipos de datos para el manejo de horas, fechas y temporizaciones, algo importante en procesos industriales. El lenguaje posee soporte para bucles iterantes como REPEAT UNTIL, ejecuciones condicionales empleando sentencias IF-THEN-ELSE y funciones como SQRT () y SIN (), muy parecido a Arduino.

GRAFSET. (SFC) Gráfico de Orden Etapa Transición. Ha sido diseñado para resolver problemas de automatismos secuenciales. Las acciones son asociadas a las etapas y las condiciones a cumplir a las transiciones. Este lenguaje resulta sencillo de interpretar por operarios sin conocimientos de automatismos eléctricos, en la Figura 2.3.6.c) se muestra un ejemplo.

Muchos de los autómatas que existen en el mercado permiten la programación en GRAFCET, tanto en modo gráfico o como por lista de instrucciones.

También podemos utilizarlo para resolver problemas de automatización de forma teórica y posteriormente convertirlo a plano de contactos.

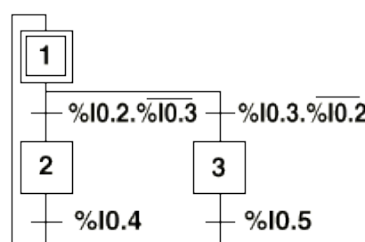


Figura 2.19.Ejemplo

Plano de Funciones. (FBD), el plano de funciones lógicas, resulta especialmente cómodo de utilizar, a técnicos habituados a trabajar con circuitos de puertas lógicas, ya que la simbología usada en ambos es equivalente. Figura 2.20.:

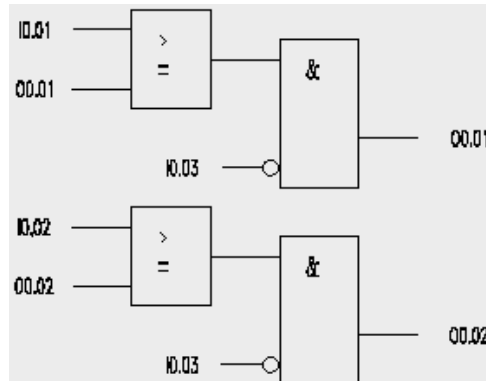


Figura 2.20. Plano de Funciones.

Funcionamiento

Los autómatas programables son máquinas de funcionamiento secuencial que ejecutan las instrucciones de programa que se les introduce, una detrás de otra y continuamente mientras el autómata está funcionando. Utilizan en la ejecución del programa las señales de entrada al PLC y generan unas señales de salida para el control de la planta una vez procesado por el programa.

Las señales de entrada son copiadas a una memoria intermedia de manera que el programa que se ejecuta no accede nunca directamente a las entradas. Se crea una imagen de la entrada. Lo mismo se hace con las salidas pero esta vez la CPU escribirá la memoria con las salidas (imagen de la salida) en la salida.

Estas dos acciones se ejecutan de una sola vez para ahorrar tiempo y ser más eficiente. Además, un autómata tiene varios modos de funcionamiento como son:

RUN: Ejecuta el programa de usuario, hace funcionar contadores y temporizadores evolucionando normalmente.

STOP: Está parado y listo para ser programado o para labores de mantenimiento.

ERROR: Ante una situación de mal funcionamiento, éste se detiene y bloquea a la espera de atención por el programador o salida del error que causó la detención.

El paso entre estos modos de funcionamiento se hace desde consolas o botones. El modo actual suele estar indicado mediante LEDs en la carcasa del PLC.

Existe un área de memoria no volátil en la que puede configurarse el modo de arranque del autómata. Puede hacerlo en modo STOP o RUN.

En cuanto al modo de funcionamiento RUN, cuando se arranca un PLC, se ejecutan una serie de comprobaciones del hardware y se hace un borrado de ciertas partes de la memoria del PLC antes de pasar al ciclo que se repite continuamente durante el funcionamiento del aparato.

Esta primera fase suele ser muy rápida (menos de 1 segundo). Luego, la fase del ciclo de operación se divide en tres partes:

Fase de procesos comunes, se comprueba periódicamente que el programa de usuario no tarde más de un tiempo en ejecutarse (watchdog), que no existan errores de tensión en conexiones, errores de sintaxis del programa y se mantengan los datos.

Fase de ejecución de programa, se atienden las entradas leyendo su contenido y se actualizan las salidas escribiendo los nuevos valores al finalizar la fase. También se ejecuta el programa guardado en su memoria.

El tiempo que tarde el PLC en ejecutarlo depende del procesador utilizado, de la longitud del programa, del tipo de instrucciones usadas en el programa y del número y ubicación de las interfaces de E/S. Este tiempo suele ser del orden de 5ms para un autómatas compacto con interfaces locales.

Fase de atención a periféricos, se atiende si existen peticiones de intercambio de información pendientes con los periféricos o procesadores auxiliares conectados a la CPU.

El autómatas contiene rutinas cíclicas de auto chequeo y de arranque en ROM con las que comprueban el programa de usuario y el propio hardware y en caso de detectar algún problema, registrarlo y, si es grave, detener la ejecución e informar del mismo mediante LEDs o un número de error en un registro.

Resumiendo se compone de funciones como:

Detección, lectura de la señal de los captadores distribuidos por el sistema.

Manda, elabora y envía acciones al sistema mediante pre-accionadores.

Dialogo hombre-máquina, mantiene un diálogo con los operarios, obedeciendo sus consignas e informando del estado del proceso.

Programación, introduce, elabora y cambia el programa de aplicación del autómatas.

Redes de comunicación, las redes industriales permiten establecer comunicación e intercambio de datos con otras partes de control a tiempo real.

Sistemas de supervisión al comunicarse (red industrial o serie) con ordenadores provistos de programas de supervisión industrial.

Control de procesos continuos, dispone de módulos E/S analógicos y la posibilidad de ejecutar reguladores PID programados en el autómatas para poder ser vigilados.

Entradas/Salidas distribuidas y se comunican con la CPU mediante un cable de red.

Buses de Campo, mediante un solo cable de comunicación se pueden conectar al bus captadores y accionadores, reemplazando al cableado tradicional. El autómatas consulta cíclicamente el estado de los captadores y actualiza el estado de los accionadores.

Configuración software STEP7 MicroWin

Dentro del programa de MicroWin, accediendo a comunicaciones, saldrán las direcciones del autómatas y el PC, como indica la Figura 2.21 y si detecta la CPU, en este caso la CPU224, es que la comunicación entre PC y autómatas se realizaron satisfactoriamente.

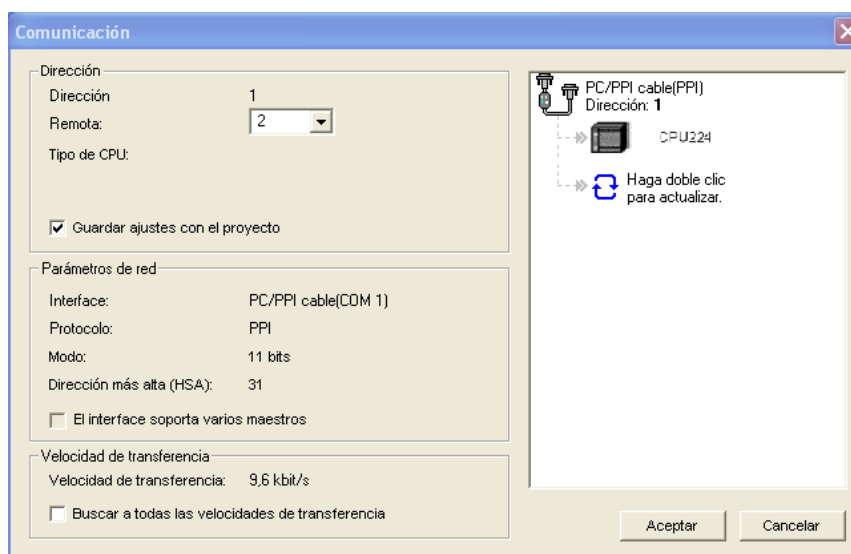


Figura 2.21. Comunicación MicroWin.

Una vez que hemos establecido correctamente la comunicación, podremos disponernos a programar en Step7 MicroWin.

Una vez abierta la ventana el programa, nos aparece una ventana como la siguiente Figura 2.22.:

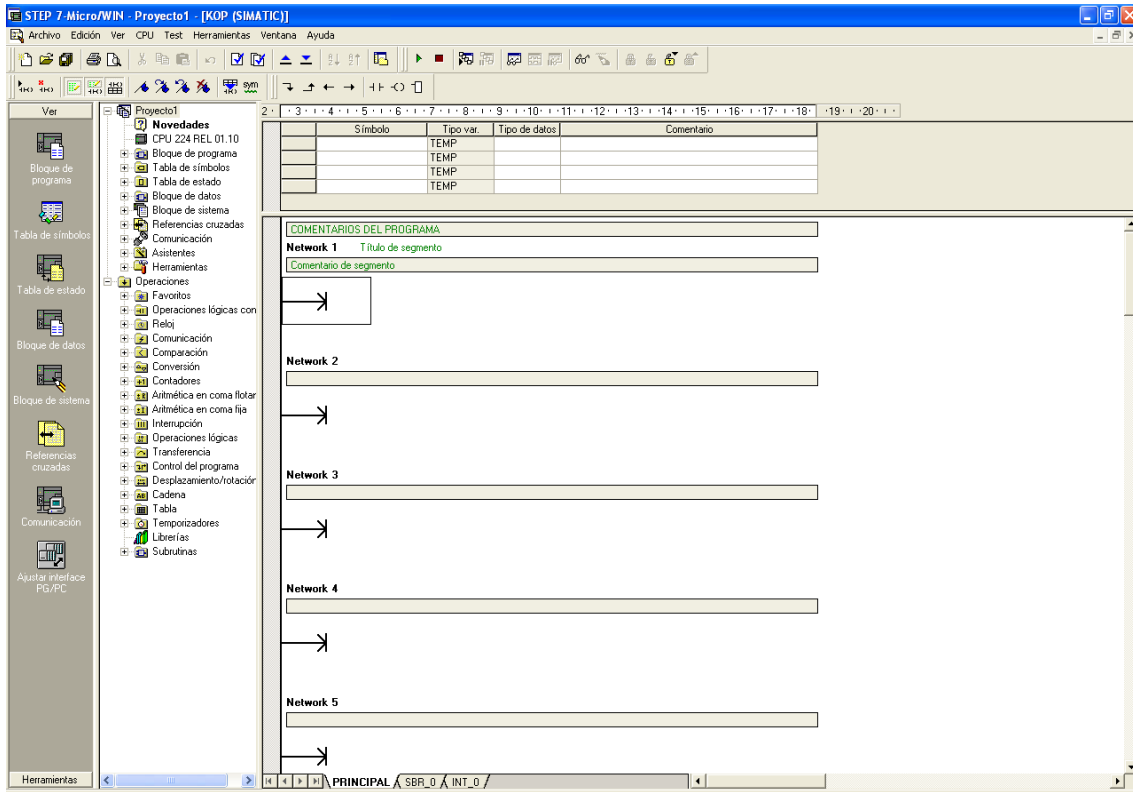


Figura 2.22. Ventana inicial Step7 MicroWin.

Hay una barra de menús, una barra de herramientas, en las cuales nos aparecen diferentes iconos que serán los que utilizaremos para una vez establecido nuestra tabla de símbolos, con las variables asignadas de entrada y salidas, estas variables deben coincidir con la configuración que hemos puesto con nuestras entradas y salidas digitales del S7-200 para poder construir nuestro código de programación. Figura 2.23.:



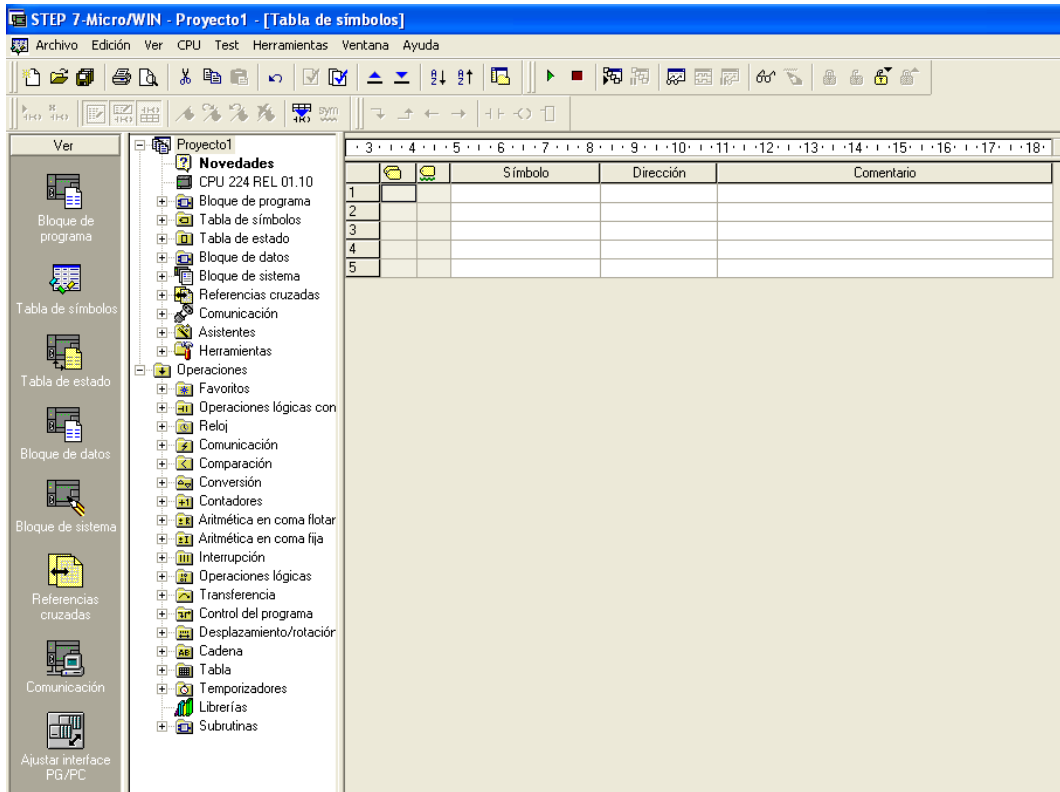


Figura 2.23. Herramientas de programación.

Una vez construido nuestro programa, lo cargaremos en la CPU, siguiendo los siguientes pasos: barra de menús, archivos, cargar en la CPU, como muestra la Figura 2.24.:

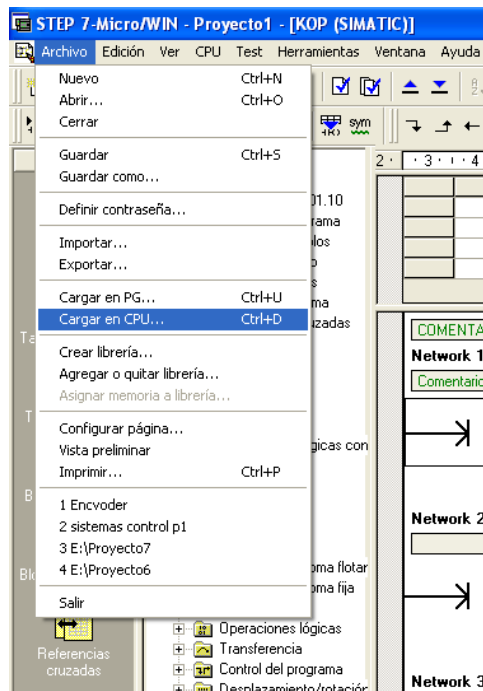






Figura 2.24. Cargar programa en CPU.

Y ya por último tendremos que poner en marcha nuestro S7-200, (RUN), para ello podremos hacerlo de dos formas:

Una tan simple como pulsando el icono de play  y pasara a RUN y estará dispuesto a su automatización. En el caso que queramos parar pulsando el icono stop, .

La otra formas, pulsando al estado de programa, , pasará al modo RUN y en la pantalla del PC podremos ver en tiempo real como se encuentran las señales de cada dispositivo, para parar tan simple como pulsar el icono de al lado, .

2.3.7. Ventajas e inconvenientes general del autómatas programable

Entre las ventajas de los autómatas frente a los sistemas cableados:

Menor tiempo de elaboración de proyectos.

Posibilidad de añadir modificaciones sin coste añadido en otros componentes.

Mínimo espacio de ocupación.

Menor costo de mano de obra.

Mantenimiento económico.

Posibilidad de gobernar varias máquinas con el mismo autómatas.

Menor tiempo de puesta en funcionamiento.

Si el autómatas queda pequeño para el proceso industrial puede seguir siendo de utilidad en otras máquinas o sistemas de producción.

Y entre los inconvenientes:

Adiestramiento de técnicos.

Coste.

A día de hoy, los inconvenientes se han minimizado, ya que la formación previa del personal suele incluir la automatización como una de sus asignaturas. En cuanto al costo tampoco hay problema, ya que hay autómatas para todas las necesidades y a precios ajustados (se tiene desde pequeños autómatas por unos 100€ hasta PLCs que alcanzan cifras exorbitantes).

2.4. Maqueta docente

De las maquetas disponibles, automatizaremos la cinta transportadora lineal, a pesar de que la plataforma será diseñada y construida para todas las maquetas. Se trata de una cinta transportadora regulable en velocidad y dirección de sentido, cuyo funcionamiento se basa en marcha-paro (emergencia). La maqueta se muestra en la siguiente figura 2.25.:

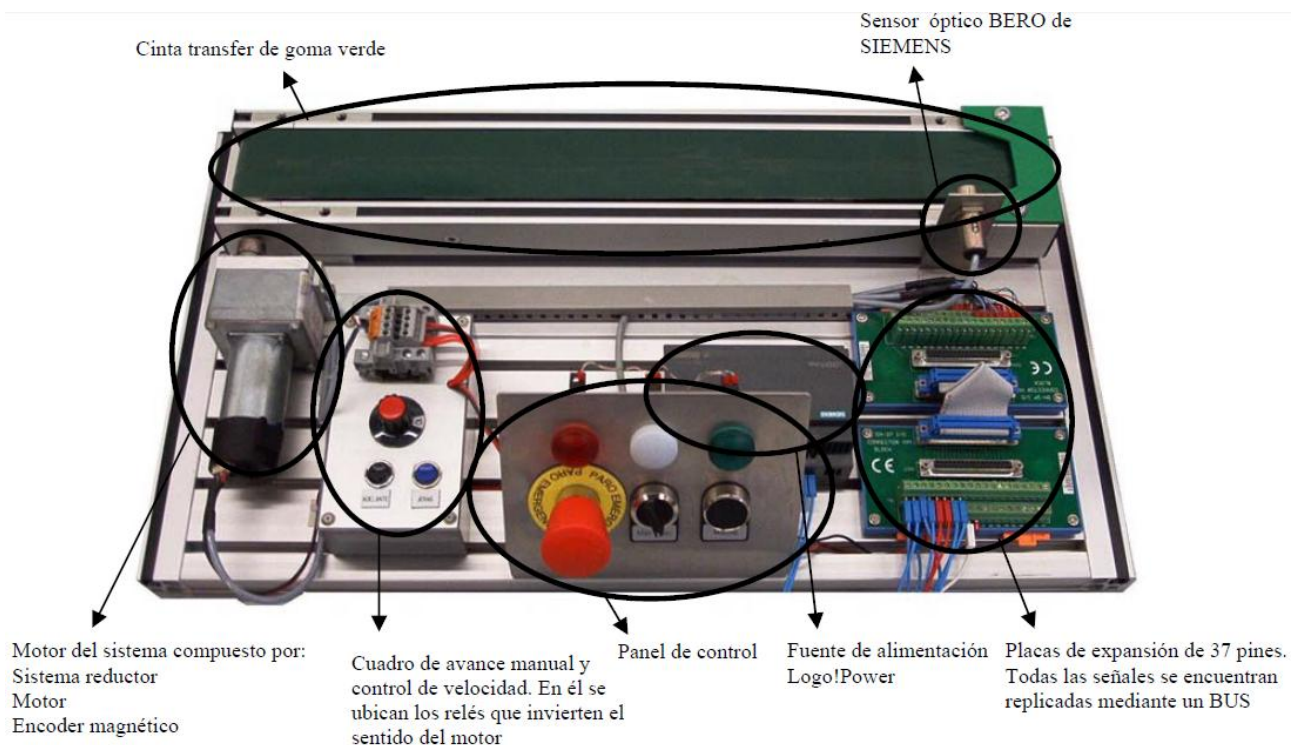
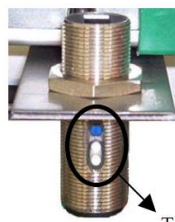


Figura 2.25. Cintra transportadora lineal.

La cinta transportadora lineal consta de una cinta transfer de goma verde movida por un motor de corriente continua de 24V y 1.8W de potencia, capaz de obtener un par de 4Nm.

El motor dispone de un sistema reductor y de un encoder magnético que proporciona un pulso de 24V por revolución del eje del motor. El motor cambia el sentido de giro gracias a la acción de un sistema de relés que invierten la polaridad de la alimentación entre 0 – 24V.

Dispone de un panel de control de selección Manual / Automático, un pulsador de marcha, una seta de emergencia con luz y dos lámparas de iluminación de colores verde y blanca. Al final del recorrido se ubica un sensor óptico BERO de Siemens (Figura 2.26.) que se regulan con los tornillos la distancia de detección.



Tornillos de regulación

Figura 2.26. Sensor óptico Bero de Siemens.

Todos los módulos disponen de unas placas de expansión de 37 pines en las cuales todas las señales del módulo se encuentran replicadas mediante un BUS de 37 conexiones para ser accesibles tanto vía conector DB-37 (para conectar a una tarjeta A/D) como mediante la bornera eléctrica de tornillos (para ser utilizada con el PLC).

El módulo didáctico dispone de todas las señales replicadas sobre una tarjeta de expansión DN-37. La numeración de los pines de la tarjeta coincide con los utilizados en los planos eléctricos. La bornera de tornillos dispone de 38 zócalos (el nº 38 está inutilizado) que permiten la conexión directa con el PLC.

El conector DB-37 (Figura 2.27.) accesible de la tarjeta dispone de las señales replicadas para ser utilizadas mediante el cable paralelo que acompaña al equipo en una conexión directa sobre una tarjeta A/D.

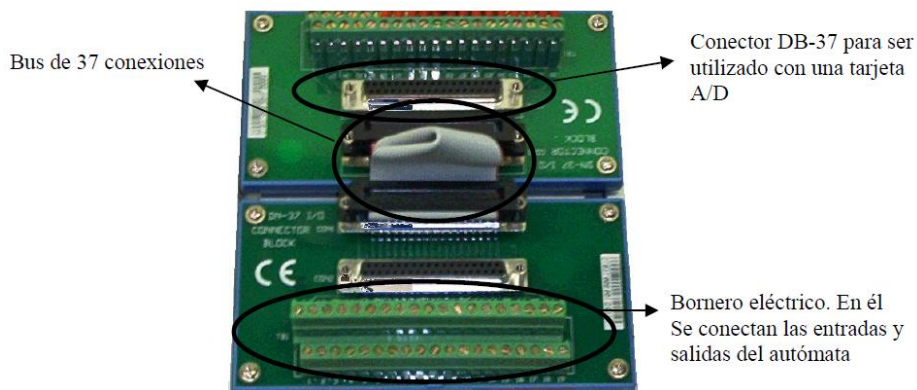


Figura 2.27. Tarjeta de expansión DN-37.

El motor de 24V DC está controlado mediante una serie de relés que invierten la polaridad de la tensión de alimentación. Los relés pueden ser accionados mediante los pines 24 y 25 o mediante los botones de la caja de control. La electrónica varía la tensión de alimentación de 0 – 24V mediante un potenciómetro escalado permitiendo la regulación de velocidad del motor. Incluye también un paro de emergencia accionada por los pines 2 o mediante la seta de emergencia de la caja de control mostrada en la Figura 2.28.:

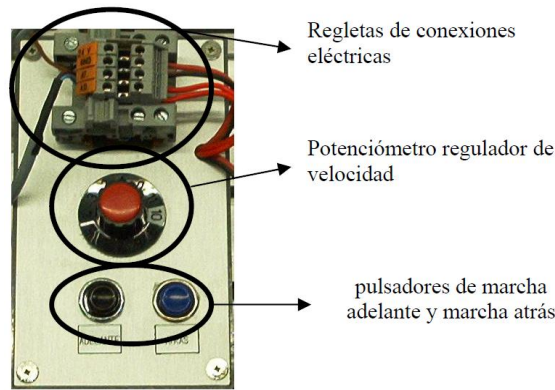


Figura 2.28. Módulo de control del motor.

El esquema eléctrico (Figura 2.29.) de cambio de sentido mediante relés se reduce al esquema siguiente. En él se ven los pines 24 y 25 los cuales pueden ser accionados en el bornero eléctrico ó mediante los pulsadores del módulo de control (I=Marcha atrás, D=Marcha adelante).

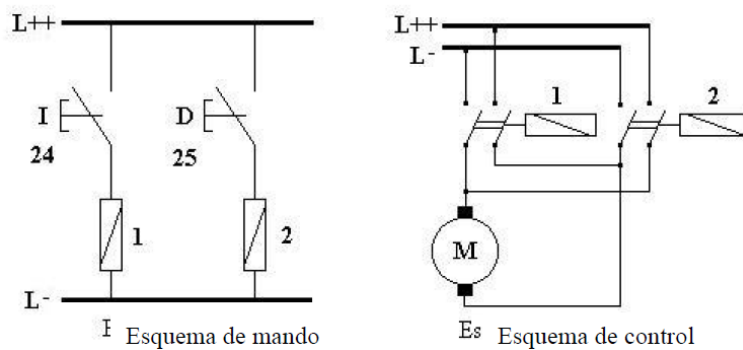


Figura 2.29. Esquema de mando y esquema de control del motor.

En el mapa de conexiones (Figura 2.30.) se distingue la fuente de alimentación y el paro de emergencia. Los pines 1, 2 y 10 son los que corresponden al bornero eléctrico.

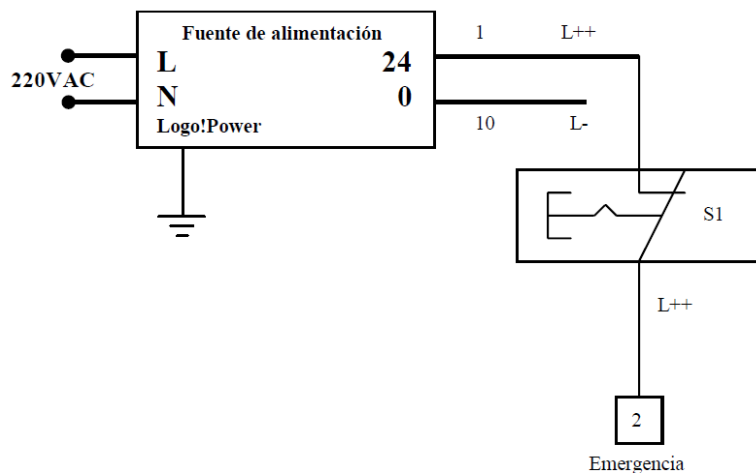


Figura 2.30. Plano eléctrico del paro de emergencia.

También podríamos ver con el siguiente esquema (Figura 2.31) los pulsadores de marcha y manu/auto del panel de control (S2 y S3). Los pines corresponden con el bornero eléctrico.

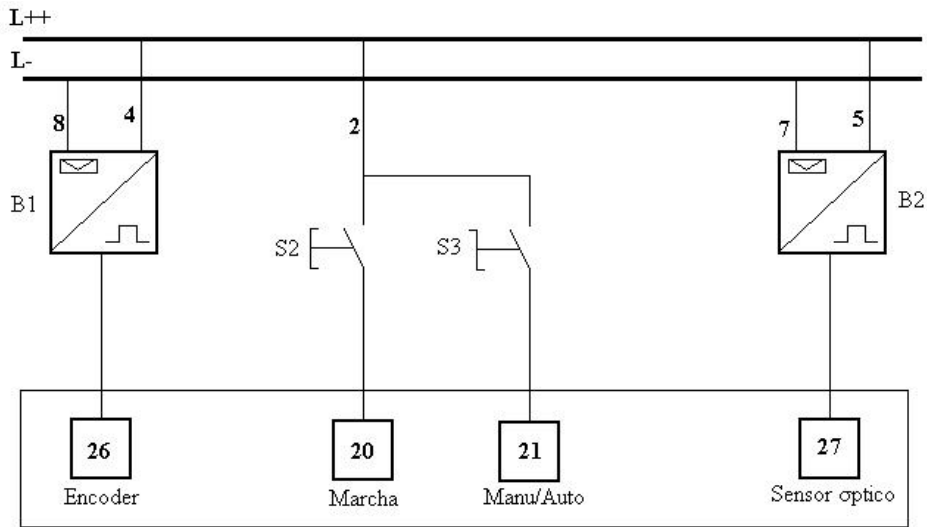


Figura 2.31. Plano eléctrico de las entradas de la cinta transportadora.

En la Figura 2.32.se ven las salidas del motor izquierda y derecha que entran a su vez en el módulo de control en el cual y mediante el sistema de relés se invierte el sentido del motor.

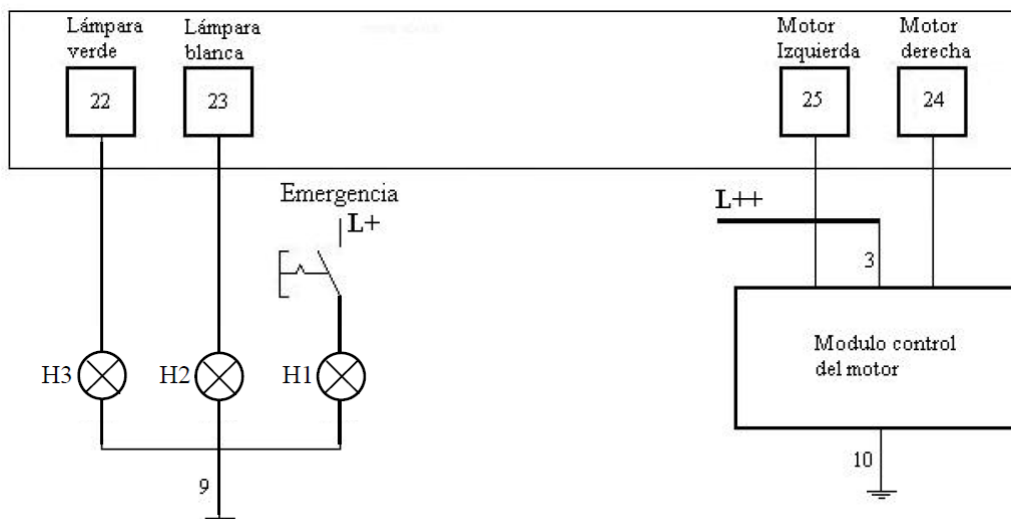


Figura 2.32. Plano eléctrico de las salidas de la cinta transportadora.

Encoder magnético incremental

El sistema codificador puede ajustarse fácilmente al eje de salida posterior de cada motor (este puede dar un impulso por revolución del eje del motor). Este puede emplearse con una unidad de realimentación de tacómetro digital para crear una señal análoga proporcional a la velocidad del motor, indicado en la siguiente Figura 2.33.:

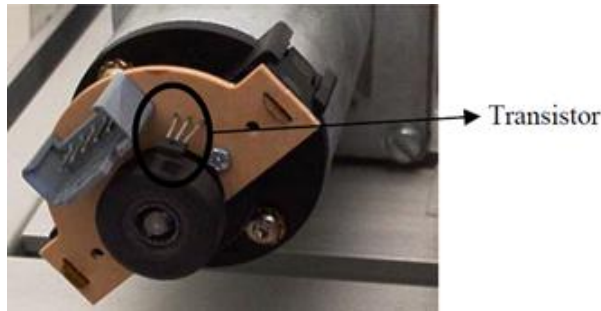


Figura 2.33. Encoder magnético incremental.

Hacer especial mención en el transistor de la Figura 2.34. el cual se encarga de suministrar la corriente en forma de impulsos.

Características técnicas				
	Sin carga	Unidades	Con carga	Unidades
Velocidad	3100	rpm	2000	rpm
Corriente	0.2	A	1.2	A
Par			75	Nm
Potencia			16	W

Figura 2.34. Características técnicas del encoder incremental.

Diagrama de conexiones del encoder Figura 2.34.:

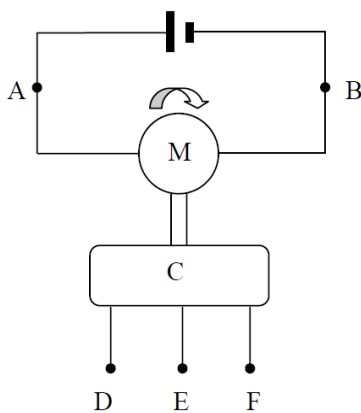


Diagrama de conexiones	
	Patilla del encoder
A	Patilla 1
B	Patilla 5
C	Generador de impulsos
D	Patilla 2 Vcc
E	Patilla 3, Masa
F	Patilla 4, Salida

Figura 2.34. Diagrama de conexiones del encoder magnético incremental.

Diagrama lógico del encoder incremental y diagrama de conexiones Figura 2.35.:

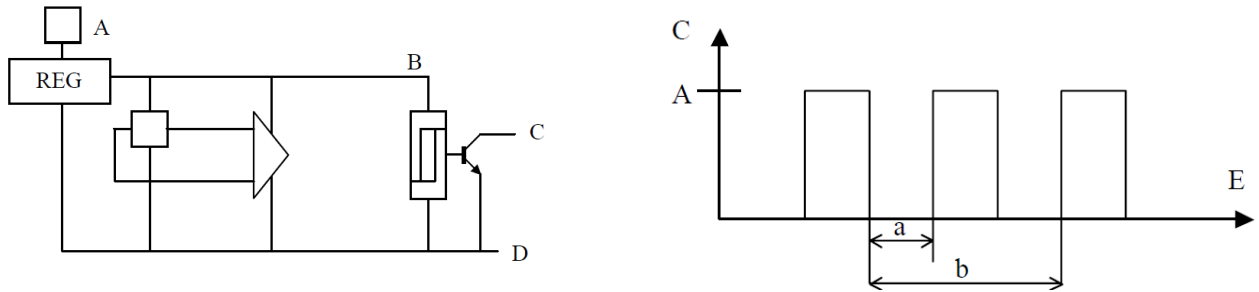


Diagrama de conexiones	
A	Vcc
B	Vcc=4.5 a 18V salida en corriente
C	Salida
D	Masa
E	1 impulso por rev. a/b=50 ± 20%

Figura 2.35. Diagrama lógico y de conexiones del encoder incremental.

Nota: Es importante comentar que, la maqueta actual tiene un transformador nuevo Omron S8VS-03024 de uso general de alimentación con una entrada de 100 a 240 VAC 30 W, salida de 24V 1.3.



Mapa de conexiones de la maqueta:

Pin	Descripción	Color de cable
1	Alimentación 24V	azul
2	24V Señal de emergencia	Verde
3	24V para motor	Negro
4	24V para encoder	Marrón
5	24V para sensor	Marrón
6		
7	0V para sensor	Azul
8	0V para encoder	Negro
9	0V para lámparas	Blanco/verde
10	0V para motor	Marrón
⋮		
20	Pulsador marcha	Marrón
21	Interruptor Manu/Auto	Naranja
22	Lámpara verde	Blanco/azul
23	Lámpara blanca	Blanco/naranja
24	Motor derecha	Azul
25	Motor izquierda	Marrón
26	Señal encoder	Azul
27	Señal sensor	negro

Nota: Las flechas representadas en el cuadro se refieren a los puentes existentes en el bornero eléctrico de la maqueta.

Capítulo 3:

Plataforma A-M para la automatización mediante Arduino

3.1. Introducción

La construcción y diseño de ésta plataforma electrónica se debe a las necesidades de la tarjeta Arduino para poder comandar las maquetas docentes del Laboratorio de Automatización Industrial, todas estas maquetas funcionan con señales digitales y analógicas, cuyos valores de intensidad son 2A y una tensión de 24Vdc para señales digitales o de 10Vdc para señales analógicas variables. Arduino como hemos descrito en las características técnicas, solo puede enviar y recibir señales de un rango de 5Vdc con una intensidad de tan solo 40mA. Por lo tanto, debe ir acompañado de un circuito que pueda elevar esos valores de salida y reducir esos valores de entrada hacia la tarjeta controladora.

También tendremos que tener en cuenta una de las desventajas que tiene Arduino, es una tarjeta que en cuanto sobrepases los niveles de tensión, intensidad e incluso ruidos y rebotes no deseados, no obtendrás el resultado que estas buscando. Por ello se ha tenido en cuenta ésta característica y nuestra nueva maqueta su circuito electrónico deberá de estar aislada las salidas de las maquetas docentes a la entrada de Arduino y viceversa.

3.2. Diseño y construcción de la plataforma A-M

Con las características necesarias, entradas y salidas que disponen las maquetas docentes, se compra una placa perforadora cuyas dimensiones abarquen todos nuestros componentes y sean de fácil acceso para posible sustitución de componentes, Figura 3.1.:

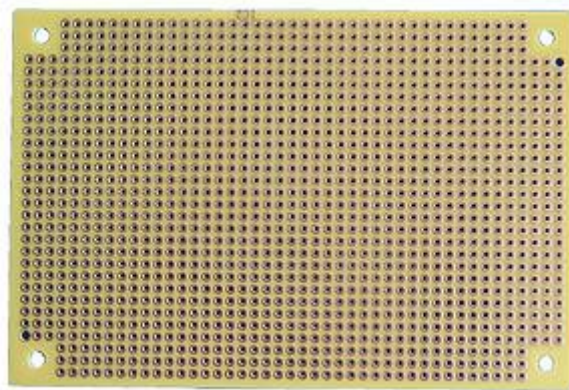
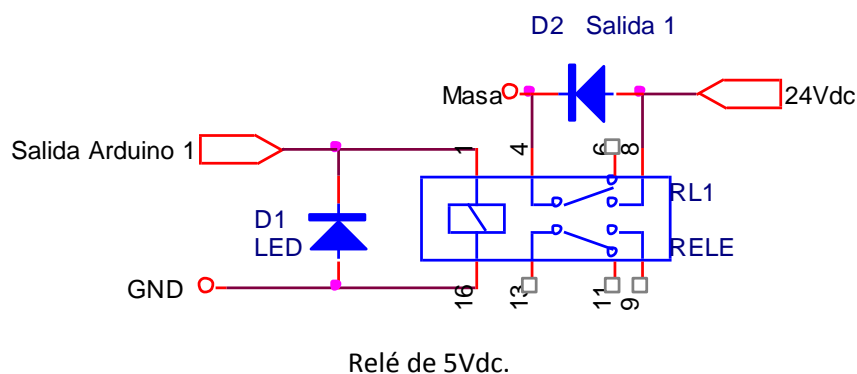


Figura 3.1. Placa perforada.

Ocho Relés de 5Vdc, Finder 43.41.7.005.200, y **un Relé de 5Vdc**, Ralux Mod.H, accionados por los pines configurados de salida del Arduino Uno para activar las salidas de la maqueta luces, control del motor...En ellos se ha incluido un diodo led para la corriente inversa producida por la conmutación de la bobina no se introduzca al pin del micro-controlador. Todo ello lo podemos ver en la Figura 3.2.:



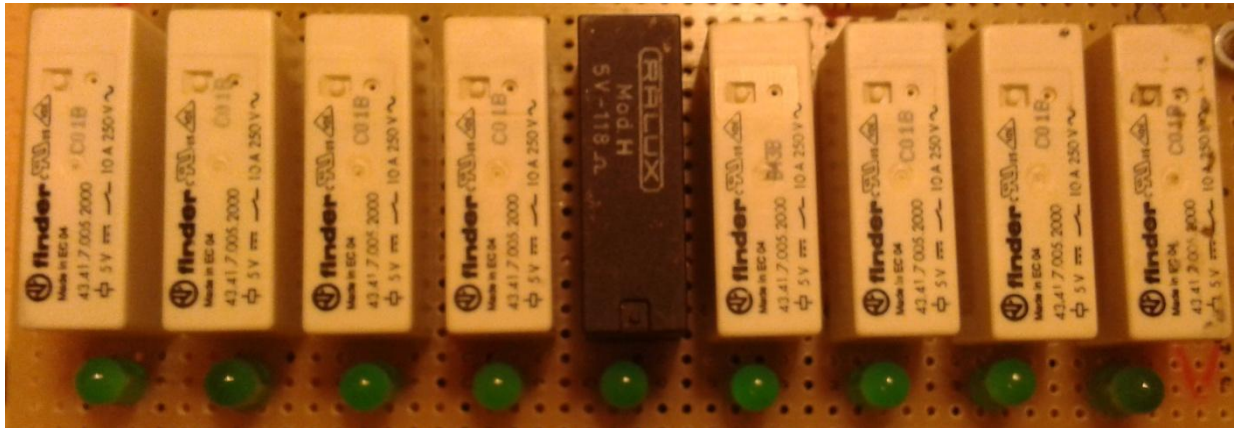


Figura 3.2. Relés y diodo led de protección.

Seis Opto-acopladores, 4N25, con resistencias de 2k2Ω a su entrada limitando la corriente procedente de los accionadores de la maqueta (pulsadores, encoder, sensores...) y la salida del opto-acoplador se ha alimentado con 5Vdc procedentes del micro-controlador que se le añade en paralelo un condensador de 100nF para filtrar la señal. La salida del emisor que es la salida hacia el pin configurado como entrada en Arduino, se le coloca una resistencia de 10KΩ en paralelo para polarizar y limitar la corriente colector-emisor del fototransistor.

El diodo de entrada tiene una $I_f = 10\text{mA}$ y tenemos una entrada de 24Vdc:

$$R = \frac{V}{I} = \frac{24}{10 \cdot 10^{-3}} = 2400\Omega$$

Como las resistencias normalizadas, sus valores cercanos a ésta cifra serían, 1k8Ω, 2k2Ω y 2k7Ω, el valor más cercanos sería 2k2Ω y como a la entrada son 24Vdc pero al consumir más accionadores de ésta fuente su valor se encuentra por debajo 23.7Vdc, por lo tanto éste valor es el más adecuado.

Respecto a los valores de la resistencia de 10kΩ y del condensador de 100nF, son valores no relevantes, tan solo evitamos que haya una caída y se filtre la tensión y circule correctamente la corriente.

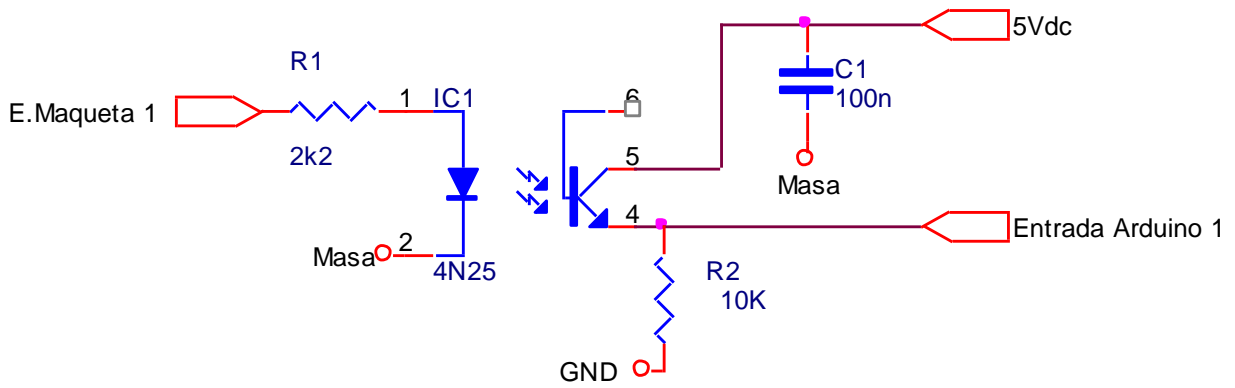
Un Amplificador de Aislamiento, HCPL 7800-A, con un divisor de tensión compuesto por dos resistencias de 100kΩ a la entradas positivas y negativas, de esta forma obtengo de (0-10) V la mitad (0-5) V e inversamente a la negativa para obtener (0-(-5)) V. alimentado con 5V obtenidos del circuito de control del motor independiente, donde se encuentra LMD 18200 y opto-acopladores. Sus salidas positiva y negativa se introducen directamente a los pines configurados como entrada en Arduino y alimentado por éste mismo 5Vdc. Figura 3.3.:

El divisor de tensión sería:

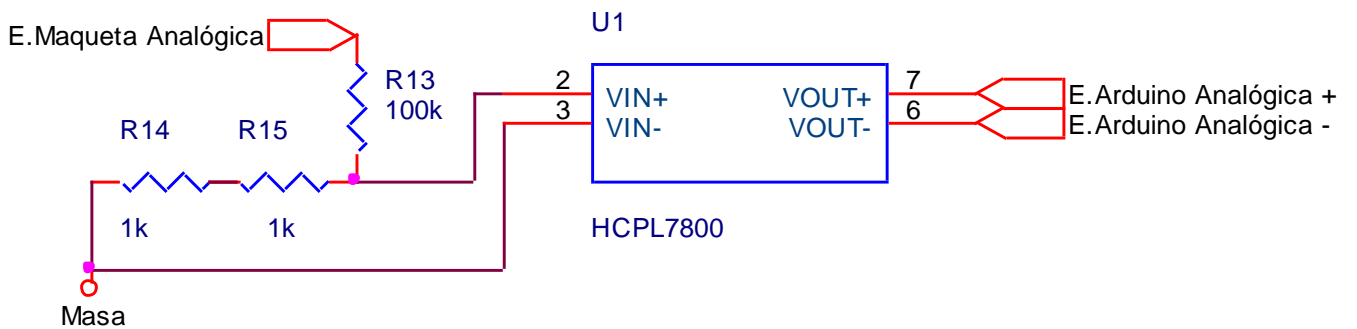
$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

$$5 = 10 \cdot \frac{x_2}{x + x_2} \Rightarrow \frac{5}{10} = \frac{1}{2} = \frac{x_2}{x + x_2}$$

$$\text{Si } x_2 = 100k\Omega \Rightarrow x = 100k\Omega$$



Opto-Acoplador 4N25.



Amplificador de Aislamiento HCPL7800.

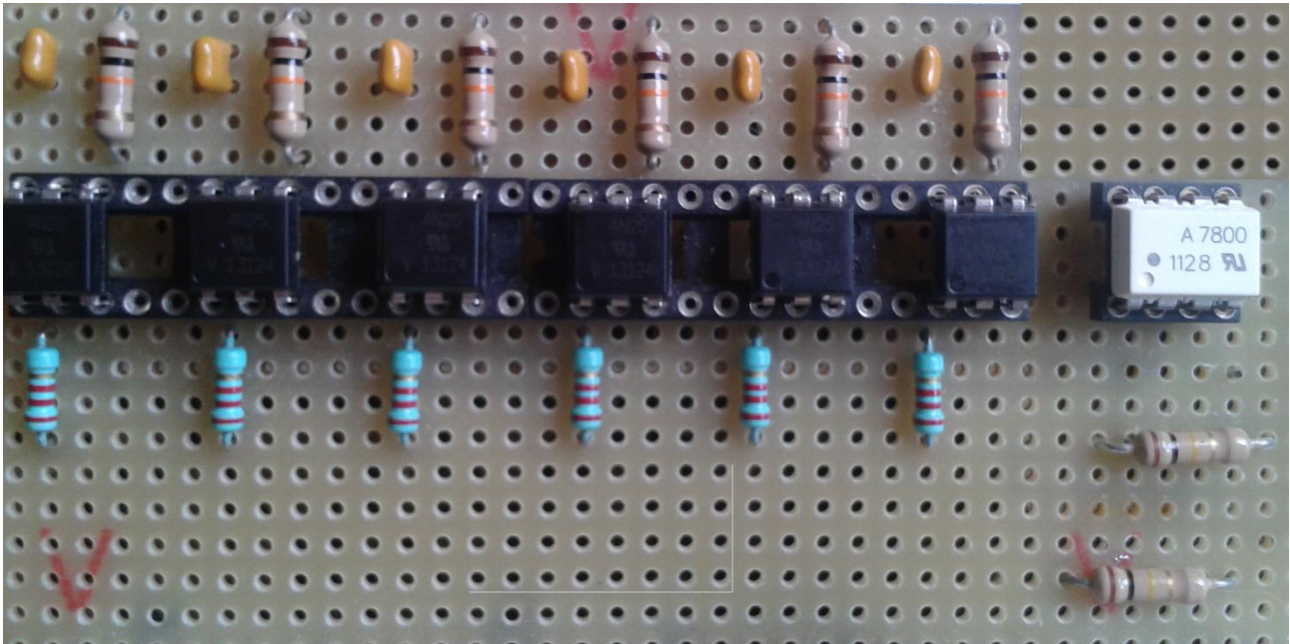


Figura 3.3. Opto-acopladores 4N25 y un amplificador de aislamiento HCPL 7800.

La placa adicional, Figura 3.4., se trata del control del motor de forma aislada:

Circuito integrado, LMD 18200, que es capaz de soportar grandes voltajes 55Vdc y hasta 3A a sus salidas dos salidas, de tal forma que puede conmutarlas ambas.

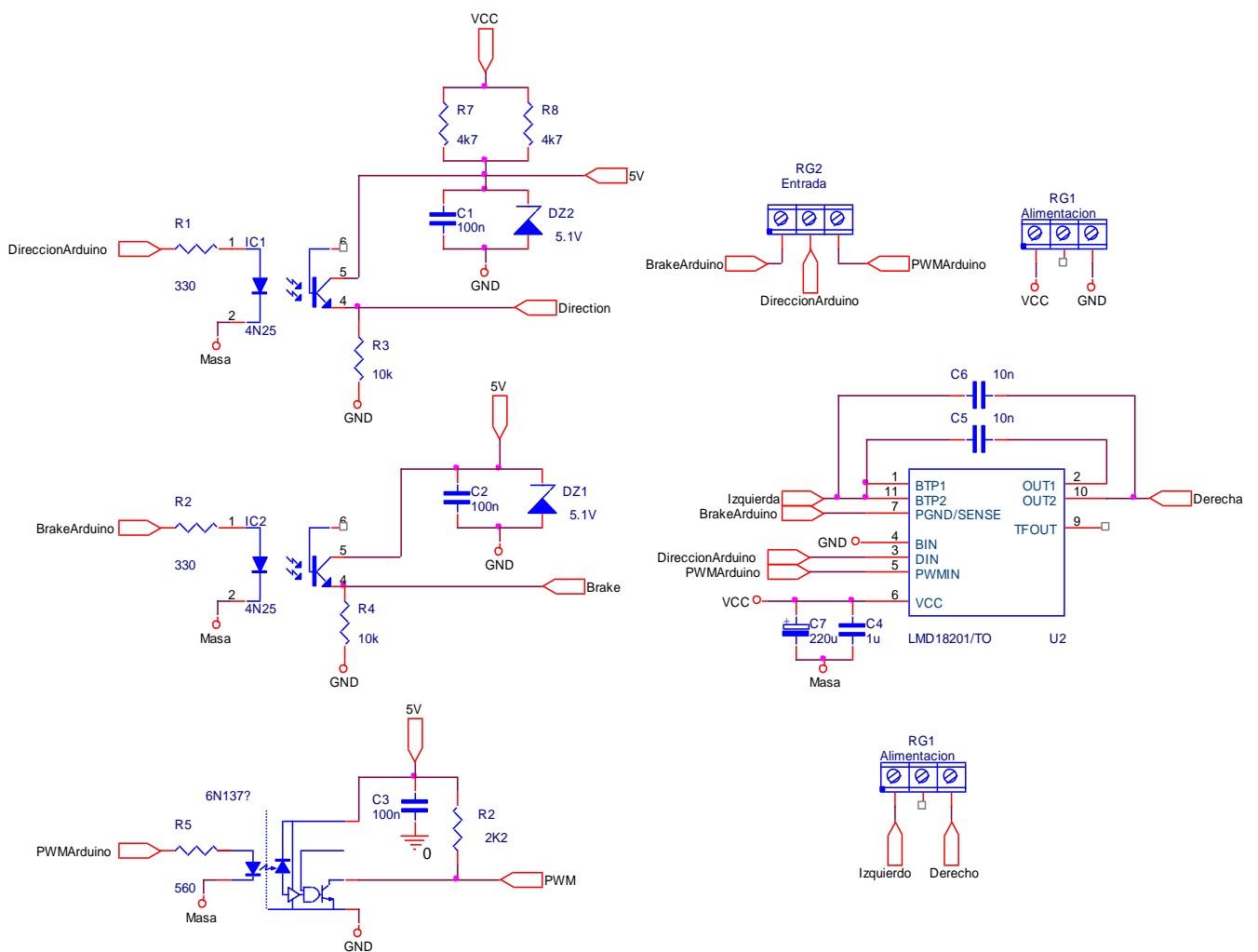
En este caso, se le alimenta con 24Vdc de la fuente de alimentación que incorpora la maqueta que le incluiremos un par de condensadores para filtrar la señal de interferencias, un electrolítico de 220 μ F y un cerámico de 1 μ F, a la salidas dos condensadores conectados desde los bootstraps de 10nF para tener unas señales más limpias y no tener pérdidas de señal. El control del LMD 18200 es a través de las señales de Dirección, PWM y Brake procedentes del Arduino:

Las señales Brake y Dirección son introducidas por Arduino y para estar aisladas del micro-.controlador contiene: **dos Opto-acopladores, 4N25**, con una resistencia de 330 Ω a las entradas de estos y los alimentamos con 5V reducidos por dos resistencias en paralelo 4k7 Ω para disipar la menos energía y reducir la corriente, un condensador cerámico de 100nF para filtrar la señal y conservar la energía.

Además se añade un diodo Zener 5.1V para contra-restar la energía llegada de las dos resistencia en paralelo obteniendo 4.9-5V de los 24Vdc que obtenemos de la maqueta. La señal de salida como los demás optos colocamos una resistencia en paralelo de 10k Ω para obtener una corriente y polarización del transistor adecuada y ésta señal se introduce al pin configurado como entrada al LMD 18200 como Brake y Direction.

La señal de PWM utilizaremos un **Opto-Acoplador más rápido, 6N136**, a su entrada limitamos corriente con una resistencia de 560Ω y a la salida del componente lo alimentamos aprovechando los 4,9-5V procedentes de la alimentación de los optos 4N25 de la señal Brake y Direction, se incluye un condensador $100nF$ y una resistencia en paralelo a la salida de $2k2\Omega$ conectada entre colector y la alimentación (base), la salida se introduce al pin configurado como entrada al LMD 18200 como PWM.

Nota: La señal PWM su opto-acoplador es de funcionamiento negado, por lo tanto mandando un High por la tarjeta Arduino el LMD18200 leerá un Low y viceversa.



Circuito de Control de Motor Independiente.

Brake	Direction	PWM	
High	High	Low	Source 1 y Sink 2
High	Low	Low	Sink 1 y Source 2
Low	X	Low	Source 1 y Source 2
High	High	High	Source 1 y Source 2
High	Low	High	Sink 1 y 2
Low	X	High	None

Tabla de la verdad del LMD 18200.



Figura 3.4. Control del Motor por LMD 18200.

Todas las entradas y salidas tanto de maqueta, como de Arduino, como entradas de la señales Brake Dirección y PWM y ambas salidas del control del motor independiente, están conectadas a sus correspondientes regletas y de igual forma las masas, GND y alimentación de Arduino, alimentación a la placa a través de los 24Vdc de la maqueta, Figura 3.5..

Todos los componentes que han sido detallados se podrán ver en el esquemático siguiente con sus detalles Figura 3.6.

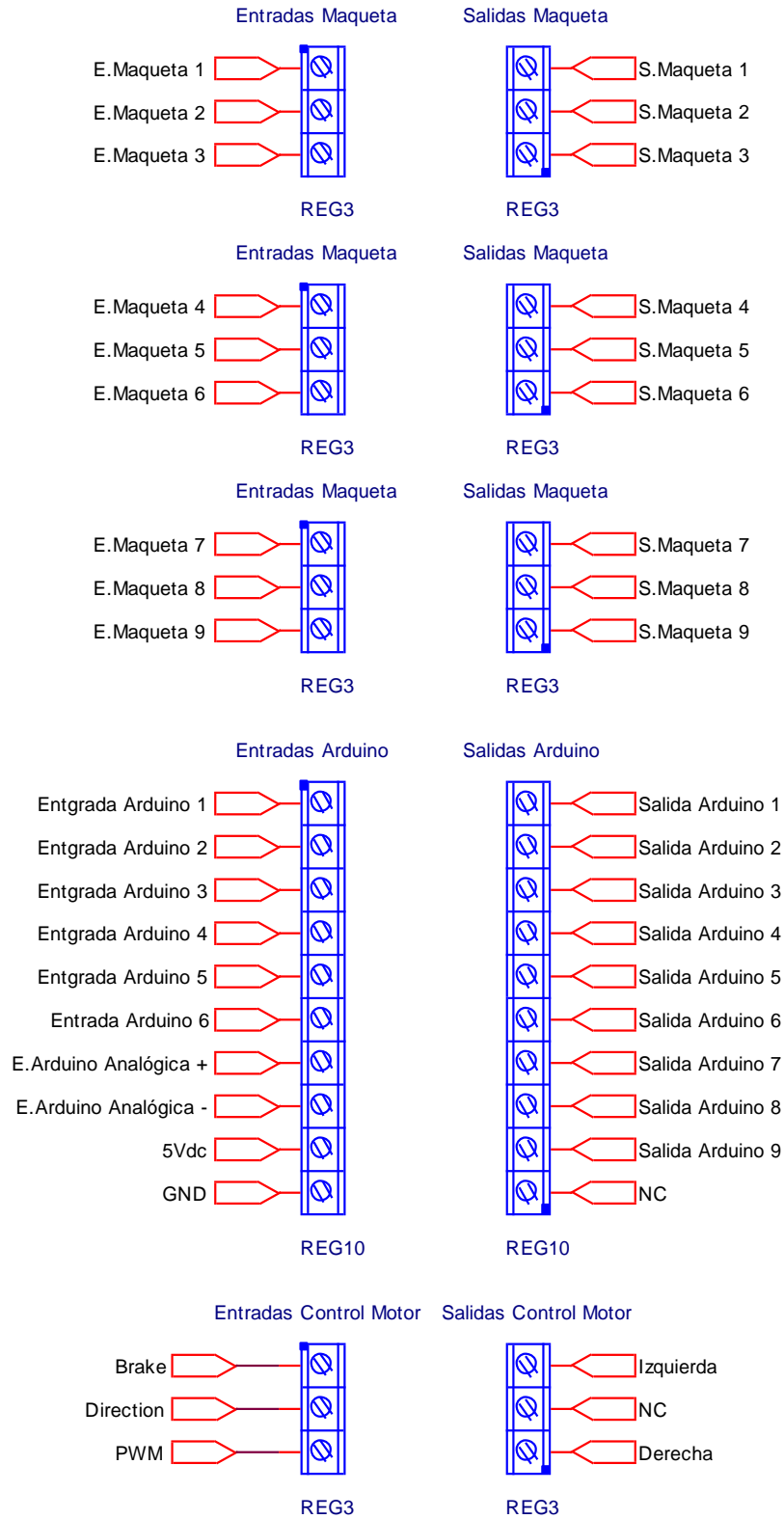
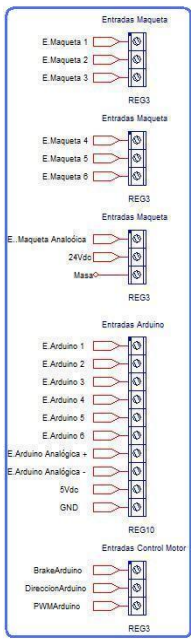
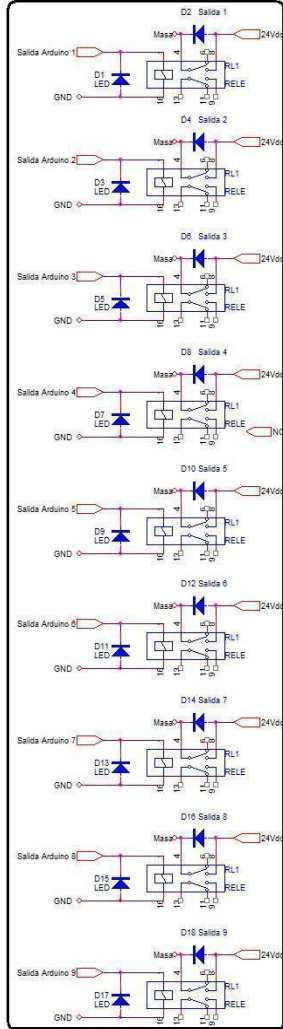


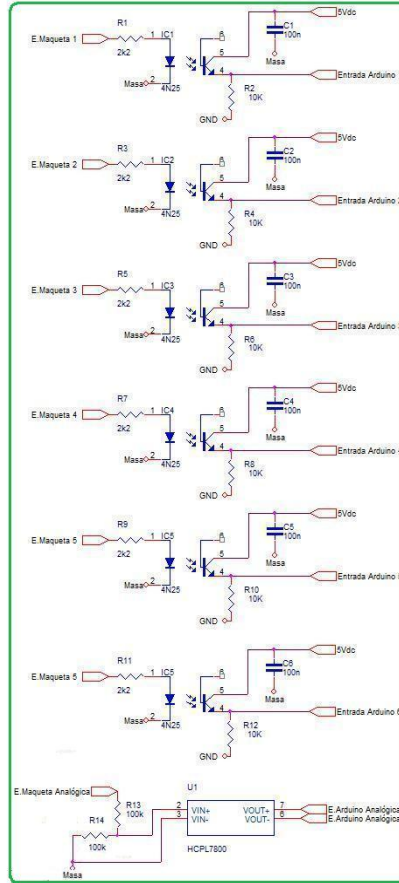
Figura 3.5. Conexiones de entradas, salidas y de control.



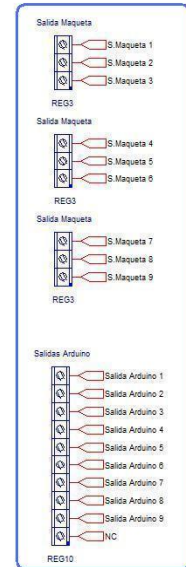
Ver Figura 3.5.



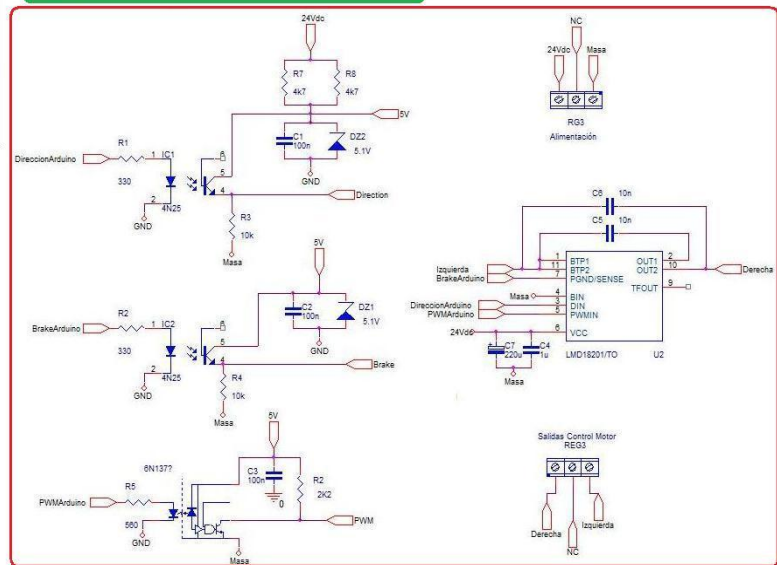
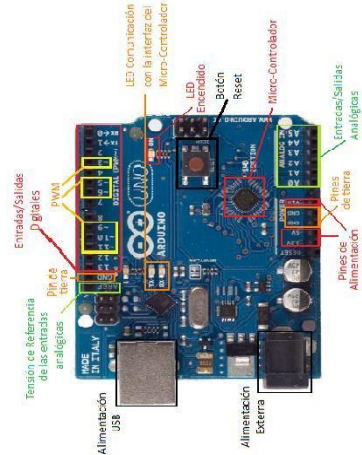
Ver Figura 3.2.



Ver Figura 3.3.



Ver Figura 3.5.



Ver Figura 3.4.

Figura 3.6. Esquemático para la automatización con Arduino.

El software utilizado para el control del sistema es el definido anteriormente, Arduino 1.0.1 y sus actualizaciones, utilizado para todos los tipos de tarjetas que existen en Arduino.

3.3. Conexión de la plataforma A-M

Para usar el circuito debemos seguir los siguientes pasos antes comenzar a programar:

1. Conectar Arduino-PC a través del cable USB.
2. Comprobar si ha sido reconocido el puerto COM correspondiente de la tarjeta Arduino. (ver Capítulo 2, Apartado 2.2.4, Entorno de programación).
3. Conectar alimentación de 24Vdc de la maqueta a la plataforma electrónica.
4. Establecer una tabla de asignación de variables, tabla 3.4.:

Las variables de entradas son aquellas señales que salen con un valor de 24Vdc desde la maqueta docente y al conectarlo al circuito se reducirá al valor de tensión de Arduino (5Vdc) y por el pin configurado será obtenido su valor de alto o bajo.

En el caso de las variables de salidas son aquellos pines configurados desde Arduino como salida, cuyo valor es el de Arduino (5Vdc) y son amplificados por el circuito a 24Vdc que es la tensión con la que funcionan las maquetas docentes. En nuestro caso es la maqueta “cinta transportadora lineal”.

Variables de Entrada			Variables de Salida		
Arduino	Maqueta	Plataforma	Arduino	Maqueta	Plataforma
Int inicio = 7	Pin 20	Pin 1	int verde = 9	Pin 22	Pin 1
Int manuauto = 4	Pin 21	Pin 2	int blanca = 10	Pin 23	Pin 2
Int sensor = 8	Pin 28	Pin 3	int derecha = 6	Pin 24	Pin 3
Int encoder = 12	Pin 26	Pin 4	int izquierda = 5	Pin 25	Pin 4

Tabla 3.1. Asignación de variables.

5. Una vez conectados todos los cables deberemos pasar a escribir nuestro programa.
6. Compilaremos y cargaremos nuestro programa en nuestra tarjeta Arduino (ver ver Capítulo 2, Apartado 2.2.4, Entorno de programación).

3.4. Presupuesto del material

En la siguiente tabla 3.1. viene recogido el coste estimado del material de construcción de la plataforma electrónica, mostrada en la figura 3.7., que acondiciona el control con Arduino de las maquetas docentes:

CONCEPTO	CANTIDAD	PRECIO UNIDAD (IVA INCLUIDO)	TOTAL EN EUROS
Arduino Uno SMD Edition	1	30,77	30,77
Cable unifilar (2m)	1	0,50	0,50
Placa perforada de Fibra de Vidrio (pequeña)	1	4,00	4,00
Placa perforada de Fibra de Vidrio (grande)	1	10,00	10,00
Diodo LED 5mm	9	0,20	1,80
Relés de 5Vdc	9	2,00	18,00
Opto-acopladores 4N25	8	0,58	4,64
Opto-acoplador 6N137	1	1,64	1,64
Regletas de conexión 3 pines	9	0,38	3,42
Regletas de conexión de 10 pines	2	1,90	3,80
Conectores de 8 pines hembra	10	0,46	4,60
Conectores de 5 pines macho	4	0,80	3,20
Amplificador Operacional HCPL 7800	1	10,29	10,29
LMD 18200	1	18,31	18,31
Condensadores Cerámicos de 10nf,100nf y 1uf	13	0,20	2,60
Condensador Electrolítico 220uF	1	0,34	0,34
Resistencias 1/4W	23	1,30	29,9
Diodo Zener 5,1V	1	0,15	0,15
Tuercas	12	0,18	2,16
Tornillos	14	0,18	2,52
Estaño	1	1,00	1,00
Papel pegatina	1	1,00	1,00
			154,64

Tabla 3.7. Total del presupuesto material del circuito electrónico.

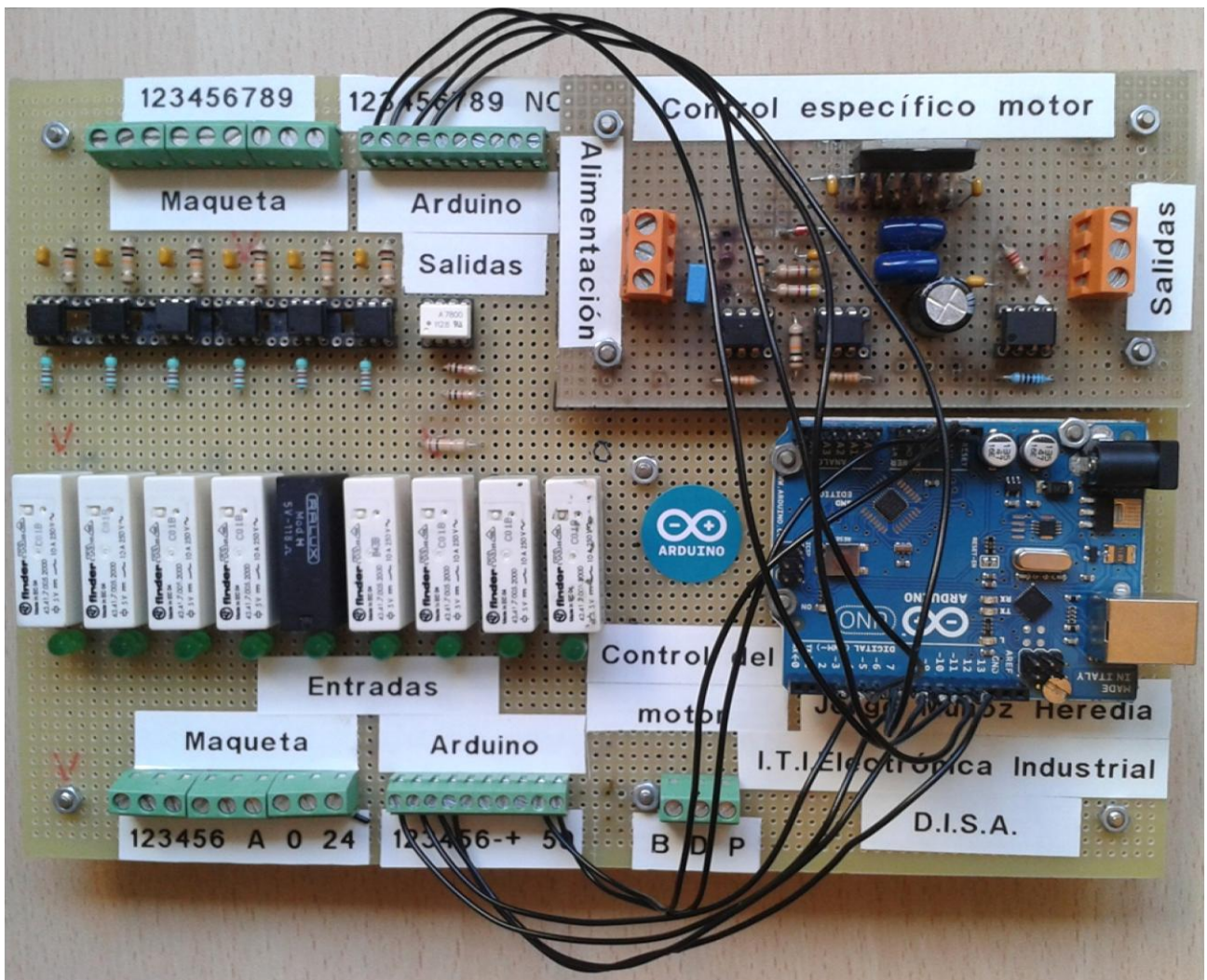


Figura 3.7. Aspecto final de la plataforma electrónica.

Capítulo 4:

Pruebas experimentales

4.1. Introducción

El sistema a automatizar se trata de una cinta transportadora lineal que consta de cinco ejercicios ordenados por dificultad (Mando del motor; Uso del paro de emergencia; Paro por sensor óptico; Temporizadores; Uso del encoder magnético), en cada una de ellos se explica la arquitectura e indicaciones más importantes para abarcar los ejercicios con los software de Arduino (Arduino 1.0.1 y actualizaciones) y de autómatas programable (Step7 MicroWin) y obtener una comparativa entre ellos.

Para la utilización del Arduino a parte de su hardware se necesita el uso de la plataforma A-M para automatizar la maqueta y sin embargo con el autómatas programable tan solo el S7-200.

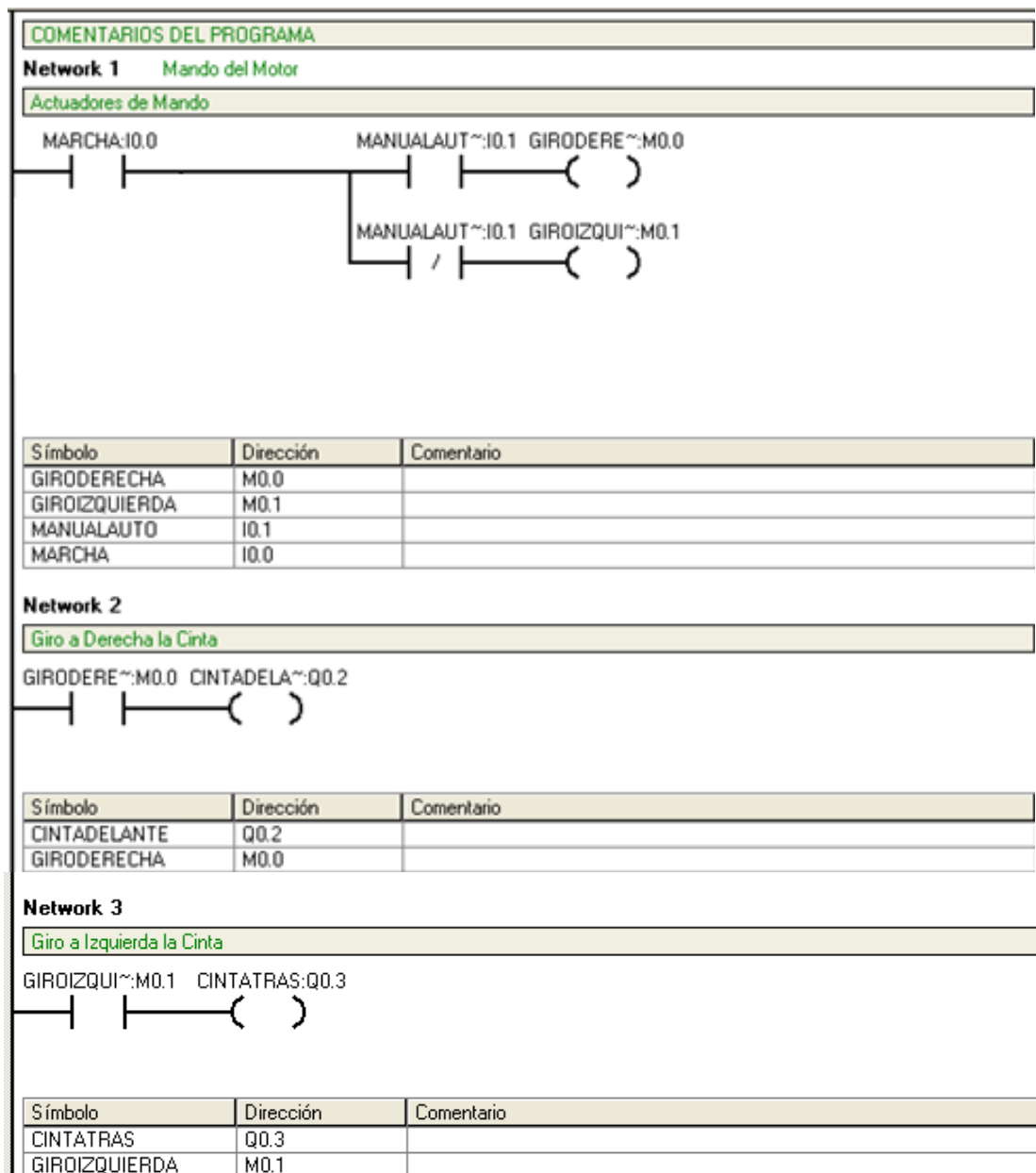
4.2. Ejercicios 1: Mando del motor

La cinta transportadora tiene que moverse mientras se esté accionando el pulsador de marcha), a la vez que el interruptor Manual / Automático permanece en la posición Manual, alejándose del sensor óptico (hacia atrás) y en la posición del interruptor Automático en dirección al sensor óptico (hacia adelante).

4.2.1. Programación con autómatas y Arduino

Automatización con Autómata:

El control a través del autómatas debemos establecer una señal interna por cada dirección del motor (izquierda y derecha) que serán mandadas por el interruptor manu/auto siempre que se encuentre el pulsador marcha activo. Su programación mediante Step7, sería:



Automatización con Arduino:

El problema que nos enfrentamos es sencillo, en el cual solo debemos configurar las entradas y salidas correspondientes, crear varias funciones respecto al movimiento que tiene la cinta transportadora lineal, una de marcha, una de retroceso, ambas se efectuarán con el pulsador llamado inicio mantenido pulsado y con el interruptor de manu/auto daremos la dirección que tomará la cinta (motor). Además tendremos una adicional, llamada paro, es el estado inicial o nivel bajo del pulsador inicio (en maqueta se llama marcha).

```
//declaración de variables

int izquierda =5;

int derecha =6;

int inicio =7;

int sensor =8;

int manuauto = 4;

//configuración e inicialización de variables

void setup () {

  Serial.begin (9600);

  pinMode (izquierda, OUTPUT);

  pinMode (derecha, OUTPUT) ;}

// Configuración del funcionamiento:

void loop () {

  int estadosentido = digitalRead(manuauto);

  int estadomarcha = digitalRead(inicio);

  if ( estadomarcha == HIGH) {

    if ( estadosentido == LOW) {

      avance ();

    } else {

      retroceso ();

    }

  }

}
```

```

    } } else {
    paro (); }}
// Funciones:
void avance () {
    digitalWrite (derecha, HIGH);
    digitalWrite (izquierda, LOW);    }
void retroceso () {
    digitalWrite (derecha, LOW);
    digitalWrite (izquierda, HIGH);  }
void paro () {
    digitalWrite (derecha, LOW);
    digitalWrite (izquierda, LOW);   }

```

Conclusión:

En este ejercicio ambos software tenemos que inicializar y configurar las variables, ya sean de entrada o salida. En Arduino una vez que ha cumplido la condición exigida llama y actúa una función definida, en cambio, el autómatas una vez cumplida la condición se activa el actuador.

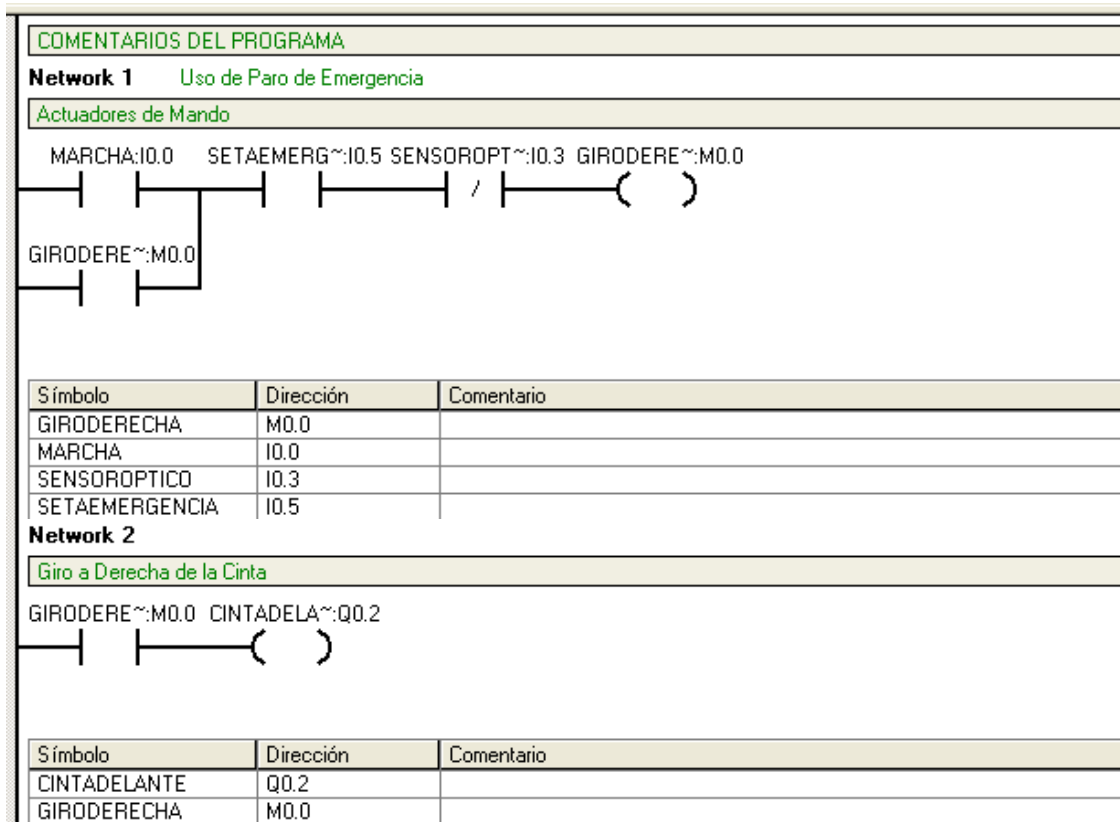
4.3. Ejercicios 2: Uso del paro de emergencia

La cinta transportadora se mueve hacia delante cuando se presiona el pulsador de marcha. Otra condición para funcionar es que no se encuentre ninguna pieza en el tope de final de cinta o sensor óptico. La cinta transportadora sólo se detiene cuando se acciona brevemente el pulsador de emergencia (ATENCIÓN: Contacto normalmente cerrado). Nótese que en el anterior ejercicio, el pulsador de emergencia detiene el proceso, pero no está implementado, o sea, si volvemos a quitar la seta de emergencia el proceso sigue donde se había quedado.

4.3.1. Programación con autómata y Arduino

Automatización con Autómata:

Para el control de uso de paro de emergencia con autómatas programables, tendríamos que añadir dos variables, la seta de emergencia y el sensor óptico, que significarían dos interruptores, normalmente cerrado (seta emergencia) y normalmente abierto, quedando así.



Automatización con Arduino:

En este caso respecto al anterior ejercicio, tendríamos las declaraciones de variables anteriores y tan solo la función avance (). Pero tenemos la variante que la cinta al ser pulsado inicio (en maqueta se llama marcha) éste quedará enclavado y se pondrá en movimiento hacia derecha la cinta, siempre y cuando no haya ninguna pieza en el final de la cinta. Se parará al pulsar la seta de emergencia en el control de mando de la maqueta docente.

Tenemos dos posibilidades para realizar el ejercicio, mediante interrupciones, éste método se ha obviado porque tan solo podríamos utilizar dos interrupciones 0 y 1 que serían los pines 2 y 3 en la tarjeta Arduino y la activación/desactivación de las interrupciones se hace mediante funciones concretas que tiene Arduino, lo que conlleva más un límite de interrupciones posibles para usar y más líneas de código. Y otra vía es enclavar el pulsador mediante el almacenamiento de variables del estado del actuador, siendo la más aceptada para el usuario.

Además hay que tener en cuenta que al ser pulsada la seta de emergencia manda un ruido de señal por el sensor. Por lo tanto debemos establecer un tiempo mínimo en el cual la señal es aceptada y desechar la señal instantánea que manda ese rebote o ruido.

Se configuraría el pulsador para actuar como un pulsador con enclavamiento:

//Tendríamos que declarar los estados del pulsador y del sensor para cuando se produzcan cambios a través de ellos poder actuar sobre ellos.

```
int inicio =7;
```

```
int sensor =8;
```

```
int salida =0;
```

```
int marcha =0;
```

```
int estadomarcha;
```

```
int previomarcha =0;
```

```
int pieza =0;
```

```
int estadosensor ;
```

```
int previosensor =0;
```

```
long time =0;
```

```
long espera =20; //tiempo que debe estar activado el sensor
```

//En void setup tendríamos que hacer la salida del sensor como resistencia pull-up de la siguiente forma:

```
digitalWrite (sensor, HIGH); //resistencia pull-up
```

//Configuración del sensor para no tener rebotes:

```
int optico;
```

```
estadosensor=digitalRead (sensor);
```

```
if (estadosensor!=previosensor) { //al cambiar de estado LOW => HIGH
```

```

    time=millis ();                //cuenta el tiempo en que está HIGH
}

if ((millis ()-time)>espera) {    // si es superior a 20ms
    optico=estadosensor;        // Es decir pasa HIGH
    if (optico==HIGH) {         //Si está activado hay pieza y sino no.
        pieza=HIGH;
    } else {
        pieza=LOW;
    } }

previosensor = estadosensor;

//Configuración del pulsador como interruptor sin interrupciones:

int value;

estadomarcha=digitalRead (inicio);

if (estadomarcha!=previomarcha) { //al cambiar de estado LOW => HIGH
    value=estadomarcha;         // Es decir pasa HIGH
    if (value==HIGH && pieza==LOW) { //Si está activado si hay pieza y sino desactivado
        marcha=HIGH;
    } }

previomarcha = estadomarcha;

```

Conclusión:

En la automatización con Arduino, la tarjeta al ser sensible a cualquier señal debemos configurar que la lectura de la señal solo sea reconocida en un periodo de tiempo que el sensor está activado, en este caso el autómata al ser más robusto esa señal no la concibe como buena.

La configuración correcta del pulsador, Arduino para enclavar debemos de almacenar variables de estado al accionar el pulsador para poder mantener activa la cinta, lo que conlleva más líneas de códigos y de dificultad que el autómata, éste tan solo se almacena el estado activo en una variable interna y se realimenta el circuito.

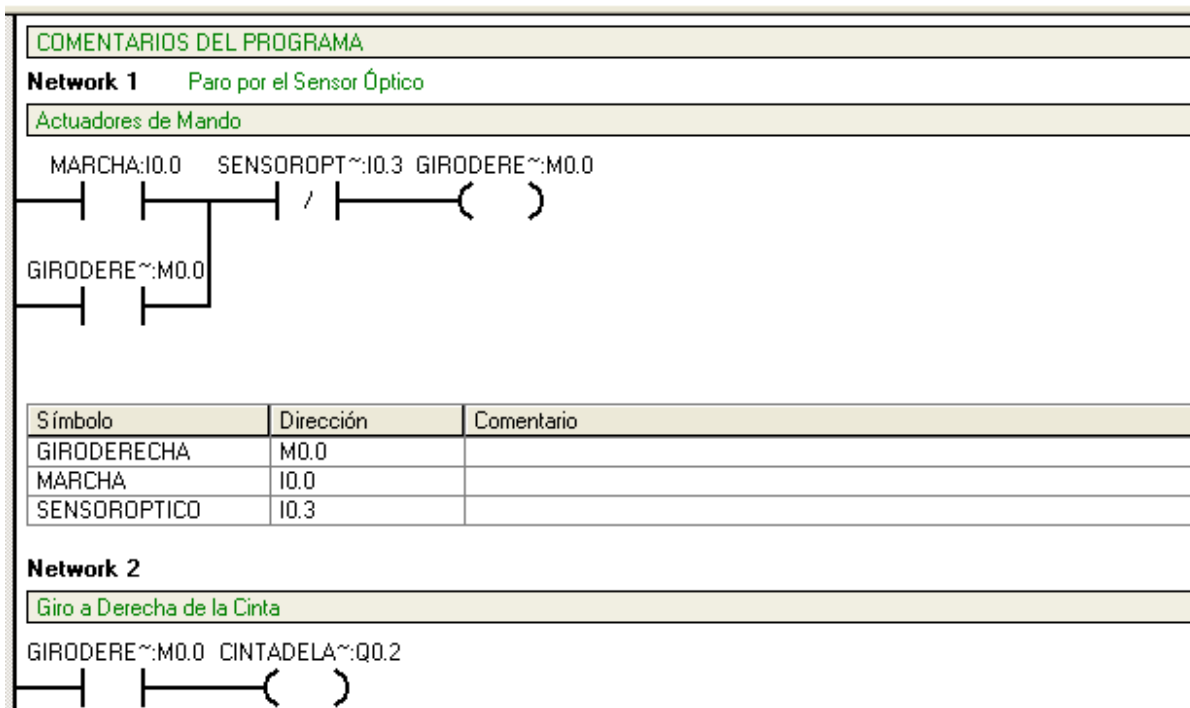
4.4. Ejercicios 3: Paro por sensor óptico

La cinta transportadora se mueve hacia delante cuando se presiona el pulsador de marcha y mientras no exista ninguna pieza al final del tope de la cinta. La cinta se detiene automáticamente cuando una pieza llega al final del recorrido.

4.4.1. Programación con autómata y Arduino

Automatización con Autómata:

En éste caso sería más simple que el anterior, tan solo con la señales del sensor óptico:



Automatización con Arduino:

Es idéntico al anterior la diferencia es mínima, se para la cinta al haber una pieza al final de la cinta transportadora lineal, pero lo hemos estructurado de otra forma pensando en problemas más complejos, a través de estados, hemos declarado unos estados y en cada etapa se activa un estado y se desactiva otro estado, la condición de que lea el resto del código es que se vayan cumpliendo los estados. Entonces el código una vez declarado los estados necesarios, funciones y configurar las variables, quedaría de la siguiente forma:

```
void loop () {

//Configuración del sensor para no tener rebotes:

int optico;

estadosensor=digitalRead (sensor);

if (estadosensor!=previosensor) { //al cambiar de estado LOW => HIGH
  time=millis ();} //cuenta el tiempo en que está HIGH
if ((millis ()-time)>espera) { // si es superior a 20ms
  optico=estadosensor; // Es decir pasa HIGH
  if (optico==HIGH) { //Si está activado hay pieza y sino no.
    pieza=HIGH;
    E2=1;
  } else {
    pieza=LOW;
  }
}

previosensor = estadosensor;

//Configuración del pulsador como interruptor sin interrupciones:

int value;

estadomarcha=digitalRead (inicio);

if (estadomarcha!=previomarcha) { //al cambiar de estado LOW => HIGH
```

```

value=estadomarcha;      // Es decir pasa HIGH
if (value==HIGH) {      //Si está activado hay pieza y sino no.
    E2=0;
    marcha=HIGH;
}
}

previomarcha = estadomarcha;

// Configuración del funcionamiento:
if (E2==0 && E1==0) {
    E0=1; }
if (E0==1 && marcha==HIGH && pieza==LOW) {
    E1=1;          //Si está E0=1 && se pulsa marcha && no hay pieza
    E0=0; }        //activa E1 y desactiva E0
if (E1==1) {
    avance (); }
if (E1==1 && pieza==HIGH) {
    E2=1;          //Si está E1=1 && hay pieza
    E1=0;          //activa E2 y desactiva E1
    paro (); }
if (E2==1) {      // El temporizador ha llegado a 5000ms paro
    paro ();
} }

```

Conclusión:

Se trata de un ejercicio parecido al anterior luego no tenemos nada novedoso, tan solo la estructura por etapas o estados, el programa es más interactivo y se puede resolver mejor los problemas que pueden surgir.

4.5. Ejercicios 4: Temporizadores

La cinta transportadora se mueve hacia delante cuando se presiona el pulsador de marcha (S3). Otra condición para que se ponga en marcha es que no se encuentre ninguna pieza en el tope del final de cinta. Al llegar una pieza al tope, el sensor óptico indica su presencia y es entonces cuando la cinta se detiene y se mueve 5 segundos en dirección contraria.

Las luces de las lámparas indican el sentido de giro del motor de la siguiente manera:

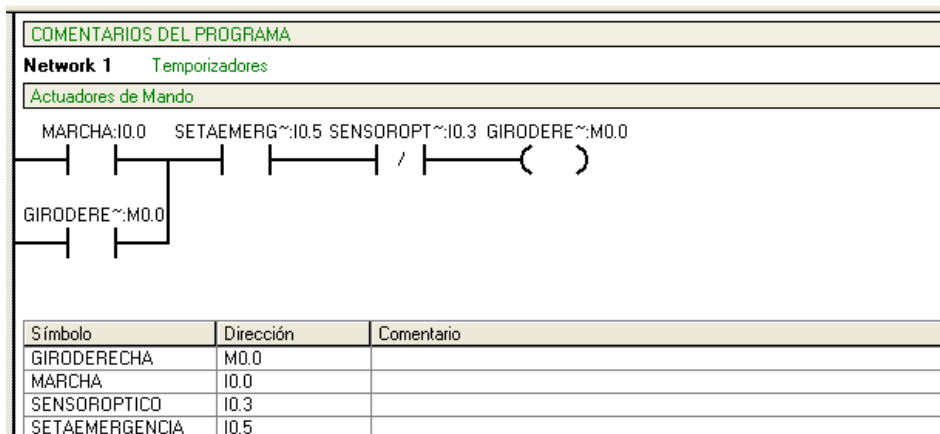
Luz verde: La cinta se mueve hacia delante.

Luz blanca: La cinta se mueve hacia atrás.

4.5.1. Programación con autómeta y Arduino

Automatización con Autómata:

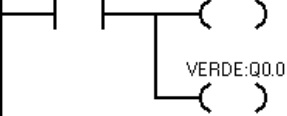
La novedad es la introducción de la activación de las luces al girar a derecha, es decir dar una salida más aparte de que la cinta gire hacía delante, al sensor óptico le añadimos una variable interna para realimentar la presencia de un objeto y al detectarlo, la cinta gira a izquierdas, se enciende la luz blanca y el temporizador que introducimos comienza su cuenta hasta 5 segundos, que resetea la señal interna de objeto y se apagará a su vez la luz y la cinta.



Network 2

Giro a Derecha de la Cinta

GIRODERE~:M0.0 CINTADELA~:Q0.2

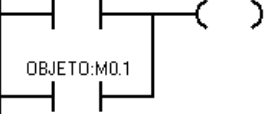


Símbolo	Dirección	Comentario
CINTADELANTE	Q0.2	
GIRODERECHA	M0.0	
VERDE	Q0.0	

Network 3

Sensor

SENSOROPT~:I0.3 OBJETO:M0.1

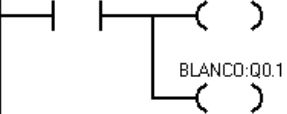


Símbolo	Dirección	Comentario
OBJETO	M0.1	
SENSOROPTICO	I0.3	

Network 4

Detección de un Objeto

OBJETO:M0.1 CINTATRAS:Q0.3

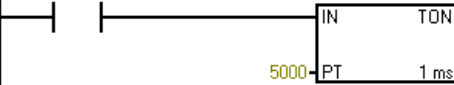


Símbolo	Dirección	Comentario
BLANCO	Q0.1	
CINTATRAS	Q0.3	
OBJETO	M0.1	

Network 5

Temporización

OBJETO:M0.1

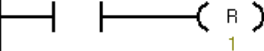


Símbolo	Dirección	Comentario
OBJETO	M0.1	

Network 6

Paro de la Cinta

T32 OBJETO:M0.1



Símbolo	Dirección	Comentario
OBJETO	M0.1	

Automatización con Arduino:

Un vez visto el funcionamiento de los estados o etapas, la novedad en éste ejercicio es que añadimos un temporizador que para la cinta cuando gira hacia izquierda el motor. Entonces declaramos el intervalo de tiempo que queremos que se encuentre activa la cinta y el temporizador, cuando el temporizador sea igual o mayor que el intervalo de tiempo establecido la cinta se parará. Tendríamos que añadir estas líneas de código:

```

long temporizador =0;

long intervalo =5000;

// Configuración del funcionamiento:

if (E2==0 && E1==0 && E3==0) {

    E0=1;    }

if (E0==1 && marcha==HIGH && pieza==LOW) {

    E1=1;                //Si está E0=1 && se pulsa marcha && no hay pieza

    E0=0;    }          //activa E1 y desactiva E0

if (E1==1) {

    avance ();    }

if (E1==1 && pieza==HIGH) {

    E2=1;                //Si está E1=1 && hay pieza

    E1=0;                //activa E2 y desactiva E1

    temporizador=millis ();    }          //comienza el temporizador

if (E2==1) {                // Si está activada la E2 retroceso

    retroceso ();    }

if (E2==1 && (millis ()-temporizador)>intervalo) {

    E3=1;                // Si está E2=1 y se cumple el temporizador

    E2=0;    }          // desactiva E2

if (E3==1) {                // El temporizador ha llegado a 5000ms paro

    paro ();    }    }

```

Conclusiones:

La diferencia entre un código y otro es más clara que los ejercicios efectuados anteriormente, el número de variables a inicializar y configurar por parte de Arduino es mayor debido a que la temporización se hará a través de la comparación entre el intervalo de tiempo que nos pedía el enunciado y el tiempo que lleva transcurrido desde que llamamos al inicio del giro a izquierdas de la cinta a la función de tiempo millis (), en caso que sea superior pararíamos la cinta. Sin embargo, para el autómata tan solo eligiendo uno de los temporizadores que nos facilita el software, introducimos la señal por el que va a ser iniciado y el tiempo que queremos y parará la cinta.

En Arduino debemos de ingeniárnosla para conseguir en forma de código un temporizador y en autómatas nos facilita diferentes tipos de temporizadores, con su información, para elegir el adecuado.

4.6. Ejercicios 5: Uso del encoder magnético

El ejercicio se basa en mover la cinta 50 cm en la dirección contraria al tope de la cinta al accionar el pulsador de marcha. El encendido de la lámpara verde indica que se ha alcanzado la posición requerida.

Para ello se configura un contador que cuenta los impulsos que le llegan desde el encoder magnético. Éste encoder magnético de un único canal proporciona un pulso de 24V por cada revolución del eje del motor. Obsérvese que lo que pasa en la práctica anterior de si la cinta se mueve más rápida, el objeto se cae de la cinta, no pasa ahora en esta aplicación porque la cinta recorre 50 cm independientemente que ésta de vueltas más rápidas ó más despacio.

Experimentalmente se obtiene que cada impulso del encoder, la cinta recorra aproximadamente unos 5mm. El contador se programa a 98 impulsos que son los suficientes para alcanzar los 50 cm deseados.

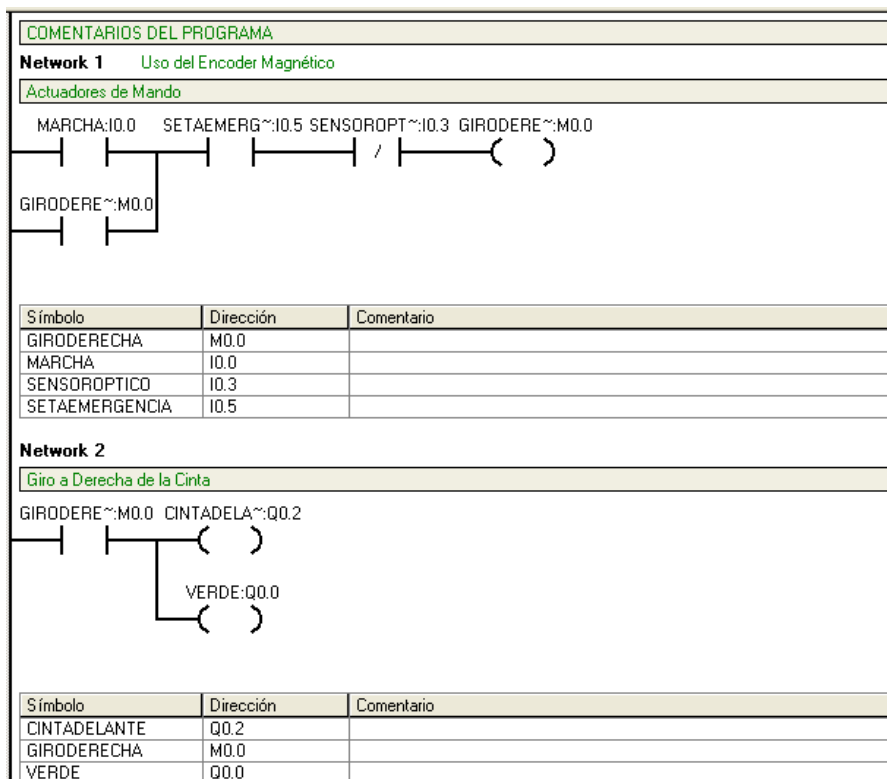
Las luces de las lámparas indican el sentido de giro del motor de la siguiente manera:

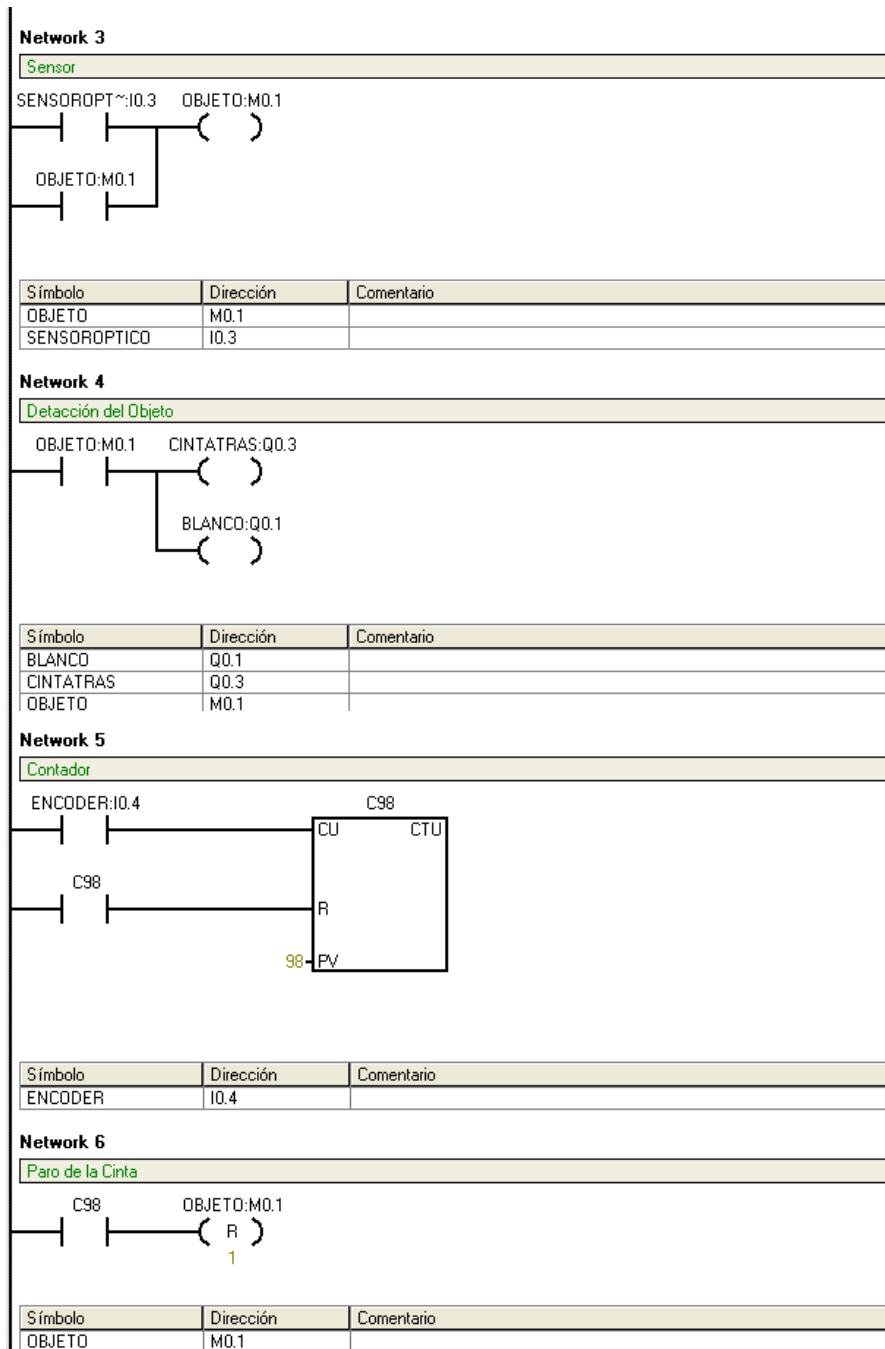
- Luz verde: La cinta se mueve hacia delante.
- Luz blanca: La cinta se mueve hacia atrás.

4.6.1. Programación con autómeta y Arduino

Automatización con Autómata:

Respecto a la anterior situación debemos sustituir el temporizador por el contador de pulsos ascendente, configurar la variable encoder con su conexión y al llegar a los 98 pulsos el encoder reseteará la señal objeto que a su vez, apaga la luz y deja de girar la cinta hacia atrás.





Automatización con Arduino:

Este caso es de estructura igual, a través de estados o etapas pero se le sustituye el temporizador con un contador de pulsos del encoder magnético del motor, de ésta forma evitaríamos que si la velocidad del motor varía no tiremos la pieza fuera de la cinta transportadora, Añadiríamos las declaraciones de la variable del encoder y añadimos el contador cuando la cinta se mueva de derecha a izquierdas:

```

int vuelta =0;

int contador =0;

void setup () {

void loop () {

// Configuración del funcionamiento:

if (E2==1) {           // Si está activada la E2 retroceso

    retroceso ();

//Configuración del contador

    if (digitalRead(encoder) == HIGH ) {

        if (digitalRead(encoder) == LOW ){

            contador++;

            delay (10);

        } }

if (contador==5) {

    E3=1;           // Si está E2=1 y se cumple el temporizador

    E2=0;           // desactiva E2

}} if (E3==1) {     // El temporizador ha llegado a 5000ms paro

    paro ();

    contador=0;

} }

```

Conclusión:

Viene a ser lo mismo que en el anterior ejercicio. En Arduino realizamos un contador mediante código, con la señal de marcha atrás inicializamos el contador y una vez cumplido ese conteo pararíamos la cinta. Sin embargo, para el autómatas tan solo eligiendo uno de los contadores que nos facilita el software, introducimos la señal por el que va a ser iniciado y resetea la señal que para la cinta al llegar al número de conteo deseado.

Capítulo 5:

Conclusiones y vías futuras

5.1. Conclusiones

En este proyecto se ha llevado a cabo el diseño y la construcción de una plataforma para poder realizar la automatización de las maquetas del laboratorio mediante Arduino. Esto ha llevado a las siguientes conclusiones:

- A nivel de software, si bien la programación de los autómatas es más de tipo gráfico y más cercana al usuario inexperto en programación, en Arduino es necesario un conocimiento mínimo de lenguajes de programación, tales como C o C++. A parte, la depuración del programa en tiempo real con esta versión de Arduino no es tan directa como con los autómatas. Esto supone un inconveniente para el aprendizaje. Sin embargo, este tipo de tarjetas de bajo coste empiezan a ser muy conocidas y con muchas aplicaciones desarrolladas, foros, blogs lo que supone un atractivo adicional para el alumnado.

- A nivel de hardware, para la conexión de las maquetas docentes con Arduino es necesario la adaptación de señales para lo que se ha construido la plataforma descrita en el proyecto. Esto supondría un coste adicional descrito en el apartado de presupuesto. Para su uso a nivel docente sería necesario incorporar también un panel como el de los autómatas para facilitar al usuario las conexiones y la modificación manual de las entradas.

- A nivel práctico, se han resuelto varios ejercicios sobre la maqueta describiendo con detalle cada uno de ellos y su comparativa respecto al autómata programable S7-200. También se ha visto que Arduino al trabajar con señales de menor voltaje es más sensible a pequeños ruidos lo que llevado a varios inconvenientes descritos en los ejercicios prácticos.

5.2. Vías futuras

Se proponen como posibles futuras líneas de trabajo las siguientes:

1. Manejo específico de velocidad motores a través del LMD18200 y de las restantes maquetas del laboratorio de prácticas de la asignatura Automatización Industrial, de tal forma sirva como docencia.
2. Automatización a través de SCADA Web con Arduino. Posteriormente de exportar dicho a SCADA, hacerlo accesible vía Internet para de esta forma, poder monitorizar y manejar un proceso desde cualquier parte del mundo y con cualquier equipo. Incluso desde el propio móvil del alumno.
3. Automatizar las maquetas docentes mediante un pc y un Arduino maestro, a través del sistema maestro-esclavo, con posibilidad de manejar vía Ethernet o Bluetooth.
4. Simulación y visualización del proceso en tiempo real a través del PC.

Capítulo 6:

Bibliografía

6.1. Referencias

1. Libro de Electrónica: Teoría de circuitos y dispositivos electrónicos.

Robert L. Boylestad y Louis Nashlky, Traducido Carlos Mendoza Barraza y Revisado por M. en C. Agustín Suárez Fernández.

Pearson, Prentice Hall.

La 8ª Edición, año 2003.

ISBN: 970-26-0436-2

2. Temario y Guía de Prácticas de la asignatura Automatización Industrial.

Antonio Guerrero González y Miguel Almonacid Kroeger

Automatización Industrial, 2º I.T.I. Electrónica Industrial

Universidad Politécnica de Cartagena

3. Ethernet Industrial entre autómatas S7-300 y S7-200 con implementación SCADA basado en HMI y en Web

José Luís Capilla Maldonado

Proyecto Fin de Carrera, UPCT

Año 2011

4. Diseño y construcción de una maqueta para el control semafórico con Arduino

Francisco Javier Toledano Moreno

Proyecto Fin de Carrera, UPCT

Año 2012

5. Temario de Fundamentos de Informática

Pedro Sánchez Palma, Dolores Cano Gil, Carlos Fernández Andrés y Pedro J. Navarro Lorente

Fundamentos de Informática, 1º I.T.I. Electrónica Industrial

Universidad Politécnica de Cartagena

6.2. Direcciones Web

Micro-Controladores:

<http://www.robotshop.com/content/PDF/what's-a-microcontroller-1.1-spanish-28122.pdf>

<http://www.neoteo.com/microcontroladores>

<http://perso.wanadoo.es/pictob/microcr.htm>

<http://www.mikroe.com/eng/chapters/view/79/capitulo-1-el-mundo-de-los-microcontroladores/>

<http://www.monografias.com/trabajos34/microcontroladores-genericos/microcontroladores-genericos.shtml>

Atmel:

<http://www.atmel.com/devices/atmega328.aspx?tab=parameters>

Arduino:

<http://www.aipdsoft.com/modules.php?name=News&file=article&sid=571>

<http://www.neoteo.com/comparativa-arduino-arduino-vs-el-resto-15399>

Autómatas Programables:

<http://olmo.pntic.mec.es/jmarti50/automatas/auto4.htm#inicio>

http://www.grupo-maser.com/PAG_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/PLC/plc.htm

<http://www.monografias.com/trabajos58/automatas-programables/automatas-programables.shtml>

http://es.wikipedia.org/wiki/Aut%C3%B3mata_programable

Electrónica General

<http://www.autoplcs.com/>

Sensor Óptico Bero

http://www.ifm.com/ifmar/web/pselect2!1_40_10_30.html

http://www.nortecnica.com.ar/pdf/teoria_opticos_2_2.pdf

<http://www.elecerrano.com.ar/siemens/sensores/opticos/index.php>

<http://www.slideshare.net/josueacerov/sensor-optico>

Tarjeta de Bornes DN-37/DN-37-A

http://www.icpdas.com/products/DAQ/screw_terminal/dn_37.htm

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/piso-dio/manual/piso_p16r16_series_manual.pdf

Caja de Mando

Relés:

<http://www.farnell.com/datasheets/64041.pdf>

http://www.omron.com/ecb/products/pry/121/g2r_2.html#Dimensions

Fuente Regulada:

<http://www.comunidadelectronicos.com/proyectos/fuente3.htm>

<http://www.national.com/mpf/LM/LM350.html#Overview>

<http://www.datasheetcatalog.org/datasheet2/2/0425f3e9h3yjottti27ttog8qify.pdf>

Fuente de Alimentación

<http://www.valinonline.com/products/S8VS-03024>

<http://www.ia.omron.com/product/item/s8vs0161f/index.html>

Capítulo 7:

Anexos

Anexo A: Datasheet utilizados

7.1. Anexo Micro-Controlador ATMEGA328 (Arduino)

El Atmel de alto rendimiento de 8-bits AVR RISC micro-controlador combina ISP 32KB de memoria flash con capacidades de lectura y escritura al mismo tiempo-, 1KB EEPROM, SRAM 2 KB, de 23 años de uso general de E / S de líneas, 32 registros de propósito general de trabajo, tres temporizador flexible / contadores con comparar modos, alarmas internas y externas, USART programable de serie, un byte por byte de 2 cables de interfaz serial, puerto serial SPI, 6 canales de 10-bit A / D (8 canales en TQFP y paquetes de QFN / FM) , temporizador watchdog con oscilador interno, y cinco modos seleccionables por el software de ahorro de energía. El dispositivo opera entre 1.8-5.5 voltios.

Mediante la ejecución de las instrucciones de gran alcance en un solo ciclo de reloj, el dispositivo logra rendimientos que se acercan 1 MIPS por MHz, equilibrando el consumo de energía y velocidad de procesamiento.

Los parámetros clave del Micro-Controlador son:

Flash (Kbytes): 32 Kbytes

Hardware QTOUCH de adquisición: No

Patillas: 32

Máximo de pines I / O: 23

Max. Frecuencia de operación: 20 MHz

Ext.Interrupciones:24

CPU: 8-bits AVR

Transceptor USB: 0

N ° de Canales Touch: 16

Canales cuadratura decodificador: 0

Velocidad de USB: No	Temperatura Sensor: Sí
Interfaz USB: No	Crypto del motor: No
SPI: 2	SRAM (Kbytes): 2
TWI (I2C): 1	EEPROM (bytes): 1024
UART: 1	Memoria de programa Automática: SI
CAN: 0	Interfaz de bus externo: 0
LIN: 0	Memoria DRAM y memoria: No
SSC: 0	NAND de la Interfaz: No
Ethernet: 0	Alimentación de Pico: No
SD / eMMC: 0	Temperatura Rango (grados C): -40 A 85
Segmento de LCD: 0	E/S Clase de alimentación: 1,8-5,5V
Pantalla gráfica LCD: No	Tensión de funcionamiento (Vcc): 1,8-5,5V
Video Decoder: No	FPU: No
Cámara de la Interfaz: No	REG/MMU: no/no
ADC canales: 8	Contadores de tiempo: 3
ADC Resolución (bits): 10	Comparar salida canales: 6
ADC Velocidad (Kbps): 15	Los canales de entrada de captura: 1
Los comparadores analógicos: 1	PWM Canales: 6
Pantalla táctil resistiva: No	32kHz RTC: Sí
Canales del CAD: 0	Oscilador RC calibrado: Sí
Resolución del CAD (bits): 0	

7.2. Anexo Sensor Óptico Opto-BERO

Se trata de un sistema de reflexión directa, en este caso, el tipo de detector es fotoeléctricos de proximidad Opto-BERO reaccionan frente a cambios en la cantidad de luz recibida para la detección directa de objetos.

La amplia serie de Opto-BERO trabaja con luz infrarroja, luz roja o luz de láser y cubre alcances desde 3 cm hasta 50 m. Debido los diferentes principios físicos en que se basan estos sistemas, para las mismas condiciones externas las barreras fotoeléctricas directas permiten alcanzar una zona de detección mayor que las barreras fotoeléctricas por reflexión. Los sensores difusos permiten reaccionar también en presencia de objetos que reflejen la luz de forma difusa. Por ello tienen una zona de detección menor que las barreras fotoeléctricas por reflexión.

Los Opto-BERO pueden ajustarse de forma rápida y cómoda a través de un potenciómetro regulado a través de los tornillos.

De las gamas de Opto-Bero se les denomina como Barreras fotoeléctricas directas o tipo T al componente utilizado en ésta maqueta, el rayo de luz emitido por el diodo emisor es reflejado por el objeto a detectar, evaluando la reflexión del rayo de luz como detección de un objeto. En el mismo componente se encuentra el emisor y el receptor como podemos ver en la Figura 7.1.:

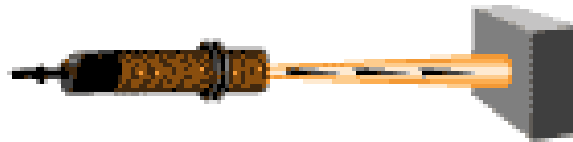


Figura 7.1. Opto-Bero

Campos de Aplicación

Los Opto-BERO se aplican, preferentemente en las áreas siguientes:

- Transporte y manutención.
- Sistemas de embalaje.
- Maquinaria.
- Papel, textil y transformación de plásticos.
- Artes gráficas.
- Control de accesos.

Estos sensores fotoeléctricos reconocen todos los objetos con independencia de su naturaleza, ya sea metal, madera o plástico. Para detectar objetos transparentes existen versiones especiales en la forma K 20 en cuerpo miniatura, así como C 40.

Con las variantes especiales, como el sensor de color o el lector de marcas impresas, es posible detectar también diferencias de color o de contraste. La versión con láser analógico permite mediciones exactas de distancias y controles de posición.

Construcción

Estos aparatos se pueden montar en cualquier posición. Si la situación lo permite, se recomienda montarlos en lugares que no estén expuestos a los efectos de la suciedad. Los accesorios disponibles permiten montar los aparatos con toda rapidez y facilidad.

Nota: Los detectores de proximidad no deberán afectarse mutuamente. Por ello es necesario respetar una distancia mínima (a) entre dos detectores. Los datos que figuran en la tabla tienen carácter meramente orientativo. Están referidos a la sensibilidad máxima.

Ajuste de Distancia de Trabajo

La sensibilidad se regula con el potenciómetro multi-vuelta integrado, aumenta girándolo en sentido horario, este no puede dañarse por rebasar el límite de giro (no hay topes).

En las barreras fotoeléctricas por reflexión y barreras fotoeléctricas directas, por lo general el potenciómetro deberá girarse siempre a la posición de máxima sensibilidad (giro en sentido horario). Con ello se disfruta del máximo exceso de ganancia. La reducción de la sensibilidad puede resultar necesaria únicamente para captar objetos transparentes.

El sensor tiene unos parámetros a tener en cuenta para poder trabajar con él que son distinguidos en la Figura 7.2. :

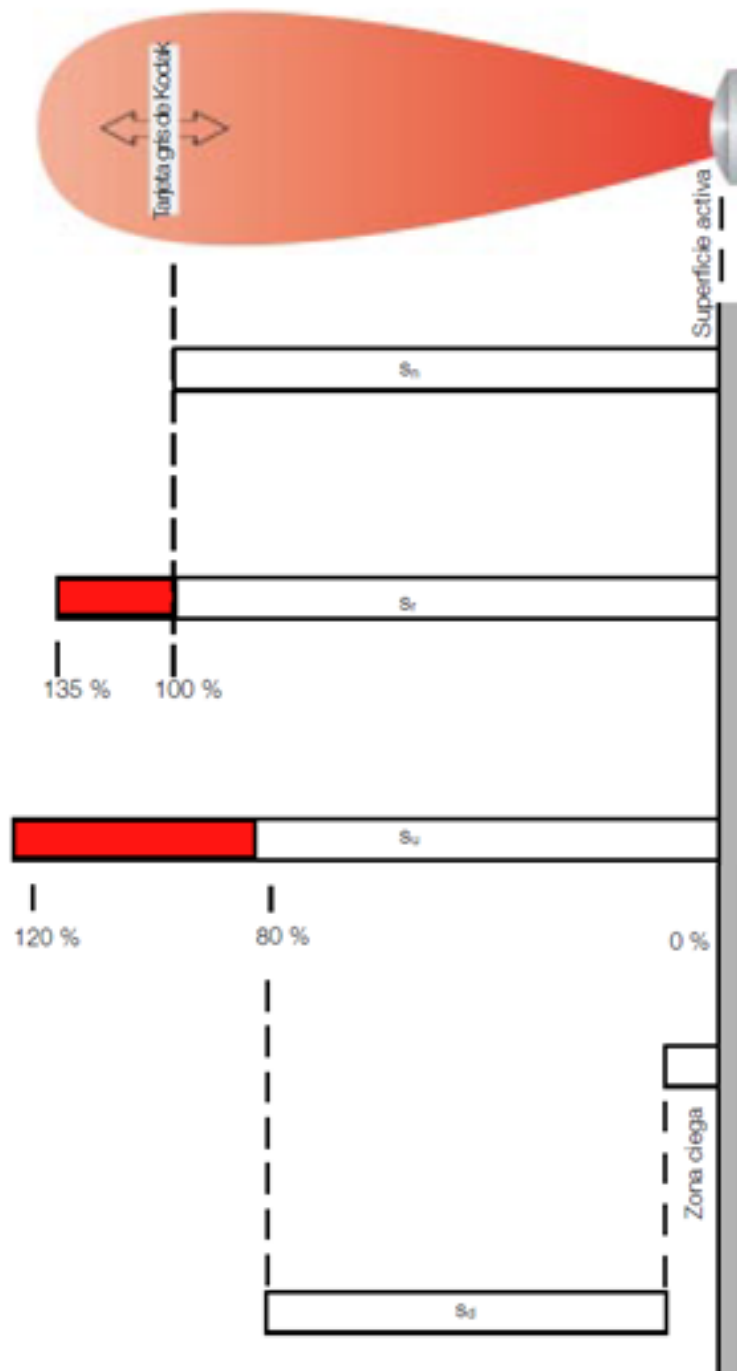
-Distancia de actuación la que se denomina entre la placa normalizada y la superficie activa del detector fotoeléctrico en caso de cambio de señal.

-Distancia asignada de actuación, aquella que sin tener en cuenta las tolerancias de fabricación, las dispersiones entre ejemplares e influencia externas. La distancia de actuación real, es decir, la que de actuación con una tensión asignada U_s teniendo en cuenta las tolerancias de fabricación a temperatura ambiente ($T=23^{\circ}\text{C}\pm 0.5$)

-Distancia de actuación útil, es la admisible dentro de unos límites de tensión y de temperatura determinados ($0.80S_n \leq U_s \leq 1.20S_n$).

-Zona ciega, se encuentra entre la superficie activa y la distancia de actuación mínima en la cual no puede detectarse objeto alguno.

-Rango de detección S_d , la que puede ajustarse la distancia de actuación de un detector fotoeléctrico respecto a la placa normalizada.



Datos técnicos, generalidades

	Barrera fotoeléctrica de reflexión		
	2 m	4 m	8 m
Distancia asignada de actuación s_n	2 m	4 m	8 m
Distancia de actuación real (en % de s_n)	150	150	150
Histéresis de conmutación (en %)	≤ 10	≤ 10	≤ 10
Ø típico del haz en $s_n/2$ (mm)	50	100	150
Ø de la zona activa (mm)			

Figura 7.2. Datos técnicos.

Funcionamiento

El emisor irradia luz que es reflejada por el objeto a reconocer y recibida por el receptor. Con este sistema no se evalúa la interrupción del haz de luz, sino la reflexión de la luz por un objeto. Importante: la función de conmutación se invierte, en relación a conmutación de luz/oscuridad. La luz emitida está también sincronizada en este sistema.

La luz del diodo emisor se concentra con una lente y se orienta hacia un reflector a través de un filtro de polarización (principio del espejo triple).

Filtros de polarización

Una parte de la luz de emisor de barreras fotoeléctricas de reflexión es reflejada directamente al receptor por objetos a detectar con superficies brillantes. Las barreras fotoeléctricas de reflexión sencillas no pueden por tanto distinguir con seguridad entre una “luz de objeto” y una “luz de reflector” reflejadas. Por tanto no pueden excluirse detecciones incorrectas.

Por esto motivo, las barreras fotoeléctricas de reflexión Balluff están equipadas alternativamente con filtros de polarización, los cuales forman junto con un reflector Balluff, un espejo tipo prisma “ópticamente activo”, en cierta medida una barrera selectiva contra la “luz de objeto” reflejada, pero dejan pasar la “luz de reflector”. Figura 7.3.:

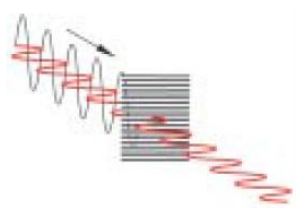


Figura 7.3. Diferencia de luces.

La luz está formada por un gran número de “haces individuales” que oscilan todos de forma senoidal alrededor de sus ejes de propagación. Sin embargo, sus planos de oscilación son independientes entre sí y pueden adoptar cualquier posición angular. Si hacen impacto sobre un filtro de polarización (retículo lineal fino) entonces solo pueden pasar los haces oscilantes paralelamente al plano reticular, sin embargo, los haces oscilantes perpendicularmente al plano se eliminan por completo. De todos los demás planos de oscilación solo se permite el paso respectivamente a la parte que corresponda a la componente paralela. Figura 7.4.:

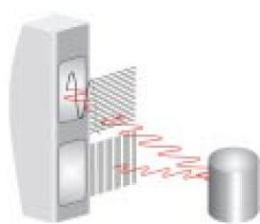


Figura 7.4. Haz del objeto

Para la supresión de reflexiones por espejos, detrás del filtro la luz oscila solo paralelamente al plano de polarización. Por esta luz el siguiente filtro de polarización girando 90° constituye una barrera impenetrable. Con un filtro de polarización girado 90° en cada caso delante de la óptica de emisor y de receptor de una barrera fotoeléctrica de reflexión puede evitarse así que la luz reflejada por un objeto a detectar reflectante clasifique la señal de foto-receptor.

Para la detección segura de objetos a detectar reflectantes. Por el contrario, la luz reflejada por el espejo triple, cuyo plano de polarización, como se ha descrito anteriormente, esta girado 90° , puede atravesar sin obstáculos este filtro.

De modo que el receptor de la barrera fotoeléctrica de reflexión se oscurece por completo al atravesar un objeto, refleja la señal y detecta el objeto con seguridad. Figura 7.5.:

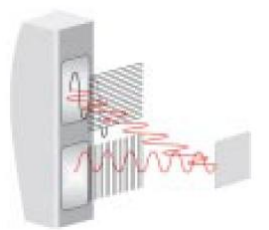


Figura 7.5. Captación del objeto con filtro.

Espejos triples ópticamente activos.

El principio bidimensional descrito anteriormente de la retro-reflexión puede trasladarse a un sistema tridimensional con tres espejos dispuestos perpendicularmente entre sí (un vértice de un cubo apoyado sobre la punta). Un haz luminoso se refleja totalmente en las tres caras y vuelve a salir paralelamente al haz incidente.

A estos espejos triples se les denomina “ópticamente activos”, porque giran adicionalmente 90° el plano de polarización del haz luminoso reflejado. Esta propiedad hace posible detectar con seguridad junto con un filtro de polarización objetos a detectar reflectantes con barreras fotoeléctricas de reflexión.

Los espejos triples se agrupan de seis en seis en hexágonos y se disponen unos junto a otros en forma de panel. Su alineación respecto al haz luminoso no resulta problemática. Figura 7.6.:

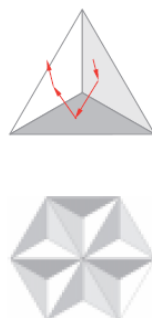


Figura 7.6. Situación de los espejos triples

Estos espejos se inyectan generalmente en placas o se prensan en láminas flexibles a partir de plásticos con una elevada densidad óptica.

Una parte de la luz reflejada llega al receptor a través de otro filtro de polarización. Los filtros se han elegido y colocado de tal manera que sólo la luz reflejada por el reflector llega al receptor pero no la luz de otros objetos que se encuentren en el radio de alcance del haz.

La salida se activa cuando un objeto se interpone en el camino del rayo desde el emisor hasta el receptor a través del reflector.

Estos sensores fotoeléctricos reconocen todos los objetos con independencia de su naturaleza, ya sea metal, madera o plástico. Para detectar objetos transparentes existen versiones especiales en la forma K 20 en cuerpo miniatura, así como C 40.

Con las variantes especiales, como el sensor de color o el lector de marcas impresas, es posible detectar también diferencias de color o de contraste. La versión con láser analógico permite mediciones exactas de distancias y controles de posición.

Ventajas y Desventajas

Ventajas

Los sensores ópticos, presentan importantes ventajas cuando lo que se desea es determinar propiedades físicas o químicas.

Es un método no destructivo y no invasivo.

Ofrece posibilidades de integración en unos sistemas más complejos.

Bajo coste y tecnología bien establecida.

Posibilidades de control a distancia de lugares poco accesibles físicamente.

Capacidad de conformar redes espaciales de sensores para el control de parámetros en grandes superficies.

Desventajas

Distancia de detección corta.

Son muy sensibles a factores ambientales como la humedad.

No selecciona el objeto a detectar.

7.3. Anexo Tarjeta de Bornes DN-37/DN-37-A

Conector E / S de bloques de montaje en carril DIN

Dos 37-pin D-sub (Uno para la expansión)

Pin a pin terminal de tornillo para E / S conectados

Pitch: 5,08 m / m

Dimensiones: 148mm x 76mm. Figura 7.7.:

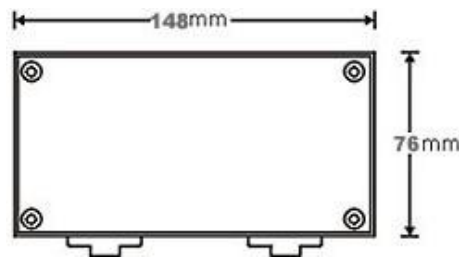


Figura 7.7. Dimensiones de la placa.

DN-37	Conector E/S de bloques (Pitch=5,08 mm) con la norma DIN-Rail Mounting and macho-macho D-sub Cable 1 M (45°) Incluye: CA-3710 (37-pin D-sub cable 1.0m)
DN-37-A	Conector E/S de bloques (Pitch=5,08 mm) con montaje en carril DIN y DN-37 macho-hembra D-sub Cable 1M (45°)

La configuración de pines viene en la siguiente Figura 7.8.:

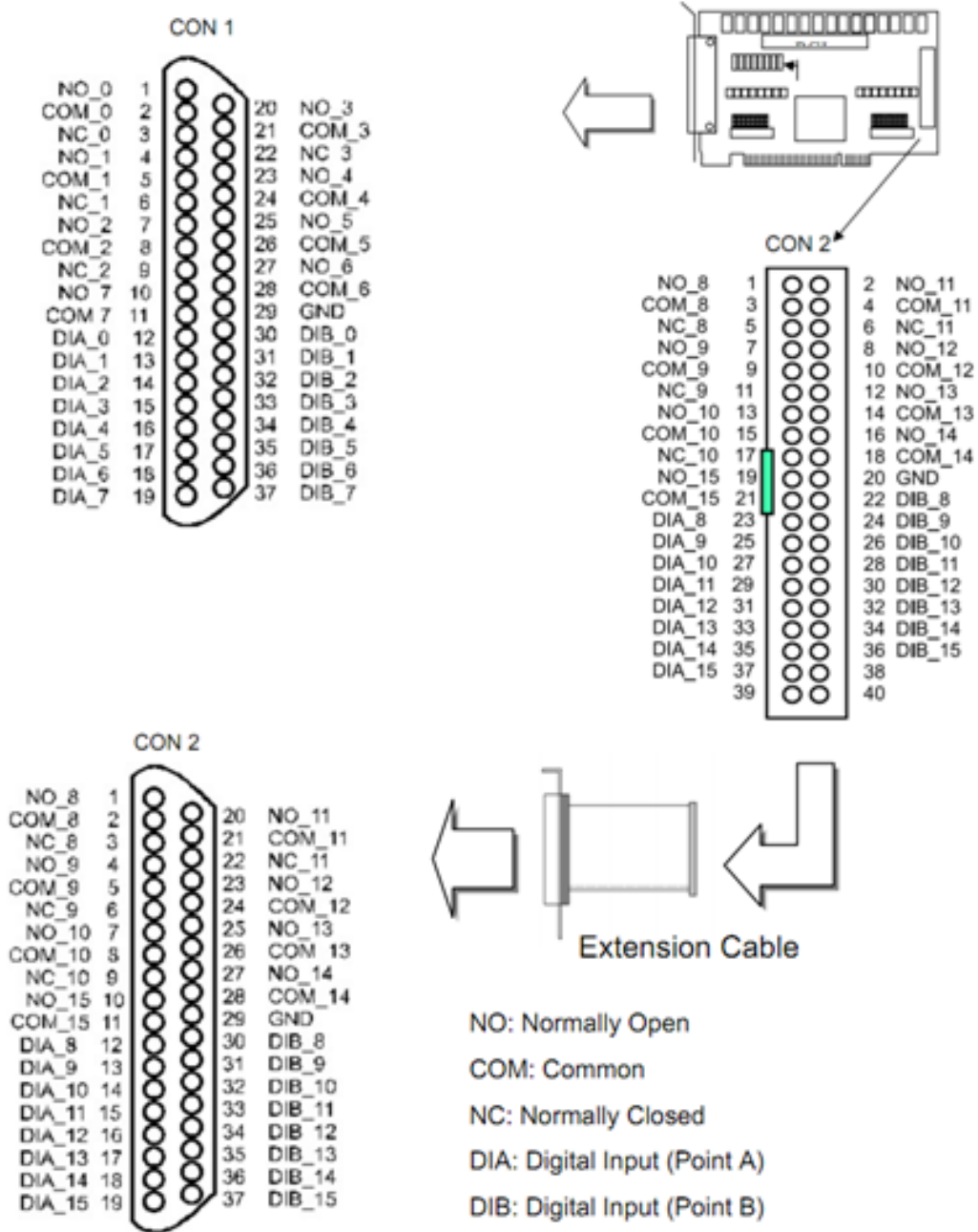


Figura 7.8. Conexiones de Pines.

7.4. Anexo Caja de Mando

En el interior de la caja de control del motor se compone por componentes como relés con sus contactos y zócalos adecuados. Y fuente regulada de tensión para el motor.

Relés

Dos zócalos para relés Finder type 95.75:

Relay type	40.31		40.51, 40.52, 40.61			
Colour	BLUE	BLACK	BLUE	BLACK	BLUE	BLACK
Clamp terminal socket: panel or 35 mm rail (EN 50022) mount retaining clip 095.71 supplied with socket packaging code SMA	95.63	95.63.0	95.65	95.65.0	95.75	95.75.0
Metal retaining clip	095.71					
8-way jumper link for 95.63 and 95.75 sockets	095.08	095.08.0	095.08	095.08.0	095.08	095.08.0
Modules (see table below)	99.01		—		99.01	

Cuyas dimensiones y características vienen dadas en la siguiente Figura 7.9.:

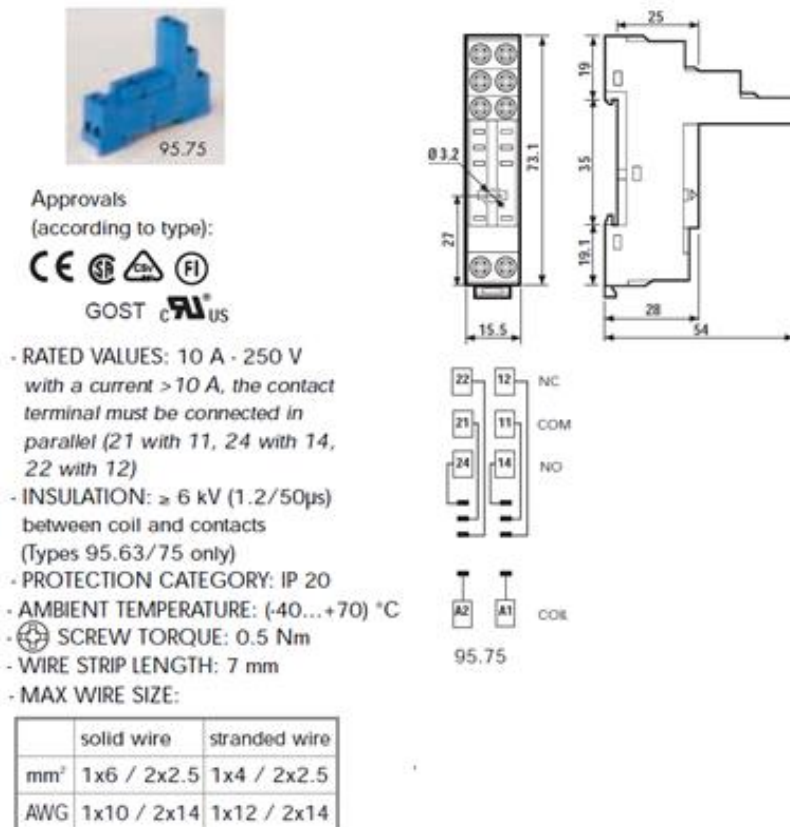


Figura 7.9. Dimensiones y Características Finder 95.75.

Contactos del Relé en la siguiente figura 7.10.:

Clasificación		De envolvente	Valores nominales	Formulario de contacto	
				SPST-NO	DPDT
PCB terminal de	De propósito general	Flujo de la protección	AC / DC	G2R-2A	G2R-2
		Totalmente sellado		G2R-2A4	G2R-24
	De alta sensibilidad	Flujo de la protección	DC	G2R-2A-H	G2R-2-H
	Doble bobina de enganche	Flujo de la protección		G2RK-2A	G2RK-2
Plug-in de la terminal	De propósito general	Sin sellar	AC / DC	---	G2R-2-S
	LED indicador de			---	G2R-2-SN
	Indicador LED con botón de prueba			---	G2R-2-SNI
	Diodo		DC	---	G2R-2-SD
	Indicador LED y diodo			---	G2R-2-SND
	Indicador LED y diodo con botón de prueba			---	G2R-2-SNDI

Nota: 1. Al hacer el pedido, la tensión nominal de bobina para el número de modelo.

. Ejemplo: G2R-1A $\overset{12VDC}{\text{---}}$ Rated coil voltage

OMRON ha preparado también por encima de los relés con contactos AgSnIn, que son más tolerantes a las corrientes de entrada grande y el movimiento físico en

2.comparación con los relés con contactos estándar. Al hacer el pedido, "-ASI" para el número de modelo.

Ejemplo: G2R-1A-ASI

Estándar, sin relés tipo de contacto son 3 TV, productos de clase, de acuerdo con los estándares de televisión de la UL / CSA. Modelos con contactos AgSnIn son 5 TV-productos de la clase.

3.Ejemplo: G2R-1A-ASI

Al pedir una TV-8 modelo de clases, insertar "-TV8" en el número de modelo de la siguiente manera:

Ejemplo: G2R-1A-TV8-ASI

Los modelos con CTI250 materiales también están disponibles.

4. Comuníquese con su representante de OMRON para obtener más detalles.

Accesorios (pedido por separado)

Conexión Sockets

Número de polos	Modelo de relé aplicable	Pista / montaje en superficie Zócalo	Back-Zócalo de fijación	
			Terminales	Modelo
2 polos	G2R-2-S (N) (D) (ND) (NI) (NDI)	P2RF-08-E -08 P2RF	PCB terminales	P2R-08P, 087P-P2R
			Terminales de soldadura	P2R-08A

Nota: Consulte Dimensiones para obtener detalles sobre el tamaño de socket.

Para montaje en carril

Base aplicable	Descripción	Modelo
Seguimiento de la conexión socket	Montaje en carril DIN	50 cm (ℓ) x 7,3 mm (t): PFP-50N 1 m (ℓ) x 7,3 mm (t): PFP-100N 1 m (ℓ) x 16 mm (t): PFP-100N2
	La placa final	PFP-M
	Espaciador	PFP-S
Volver conexión socket	Placa de montaje	P2R-P *

* Se utiliza para montar varias P2R-05A y 08A del lado P2R conectar enchufes al lado del otro.

Figura 7.10. Características de los Contactos del relé.

Prestando atención a las características técnicas nominales de la bobina de la figura 7.11.:

Tensión nominal	24 VDC	
Corriente nominal (50/60 Hz)	21,8 mA	
Resistencia de la bobina	1100 Ω	
Inductancia de la bobina (H) (ref. valor)	Armadura OFF	4.27
	Inducido en	8.55
Debe funcionar de tensión	70% max. de la tensión nominal	
Hay que liberar la tensión	15% min. de la tensión nominal	
Max. voltaje	170% del voltaje nominal (a 23 ° C)	
Consumo de energía	Aprox. 0,53 W	

Nota: Tensión nominal del tipo de contacto bifurcado travesaño: 12 VDC, 24 VDC

Figura 7.1. Características de la bobina.

Y teniendo en cuenta su contacto, PCB/Flujo de Protección ilustrada en la figura 7.12.:

Artículo	De propósito general, terminales de conexión rápida	
Número de polos	2 polos	
Carga	Carga resistiva (cos = 1)	Carga inductiva (cos = 0,4; L / R = 7 ms)
La carga nominal	5 A a 250 VAC, 5 A a 30 VDC	2 A a 250 VAC, 3 A a 30 VDC
Corriente nominal	5 A	
Max. tensión de conmutación	380 VAC VDC, 125	
Max. corriente de conmutación	5 A	
Max. de alimentación de conmutación	1250 VA, 150 W	500 VA, 90 W
La tasa de fallas (valor de referencia)	10 mA a 5 VCC	

Nota: 1.P nivel: λ 60 = 0,1 x 10⁻⁶ / operación

2.(.): Tipo de contacto bifurcado travesaño.

Figura 7.12. Características del contacto de la bobina.

Características de los relés estándar, figura 7.13.:

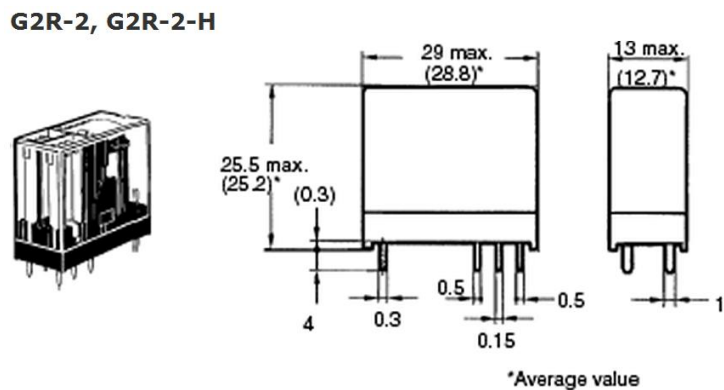
Artículo	2 polos
Resistencia de los contactos	50 mW máximo.
Operar (set)	15 ms máx.
Liberación (reset)	AC: 10 ms máximo; DC: . 5 ms máx. (W / built-in de diodos: . 20 ms máximo)
Max. frecuencia de operación	Mecánica: 18.000 operaciones / hr Eléctrica: 1.800 operaciones / hr (a carga nominal)
Resistencia de aislamiento	1.000 mW min. (A 500 VDC)
Rigidez dieléctrica	5000 VCA, 50/60 Hz durante 1 minuto entre bobina y contactos *, 3000 VAC, 50/60 Hz durante 1 minuto entre contactos de diferente polaridad 1000 VCA, 50/60 Hz durante 1 minuto entre los contactos de la misma polaridad
Resistencia a las vibraciones	Dstrucción: 10 a 55 a 10 Hz, amplitud de 0,75 mm (1,5 mm de amplitud pp) Fallo de funcionamiento: 10 a 55 a 10 Hz, amplitud de 0,75 mm (1,5 mm de amplitud pp)
Resistencia a los golpes	Dstrucción: 1.000 m / s ² Fallo de funcionamiento: 200 m / s ² cuando se activa, 100 m / s ² cuando no hay energía
Resistencia	Mecánica: Bobina de CA: 10.000.000 operaciones mín.; DC de la bobina: 20.000.000 operaciones min. (A 18.000 operaciones / hr) Eléctrica: 100.000 operaciones min. (A 1.800 operaciones / hr a carga nominal)
Temperatura ambiente	De funcionamiento: -40 ° C a 70 ° C (sin hielo)
Humedad ambiental	Funcionamiento: de 5% a 85%
Peso	Aprox. 17 g (plug-in de la terminal: . Aprox. 20 g)

Los valores de la tabla de arriba son los valores iniciales.

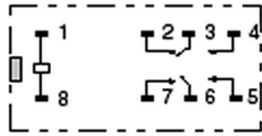
Nota:* 2000 VCA, 50/60 Hz durante 1 minuto cuando la toma de P2R-05A o 08A-P2R se monta.

Figura 7.13. Características estándar del relé

Dimensiones mostradas en la Figura 7.14.:

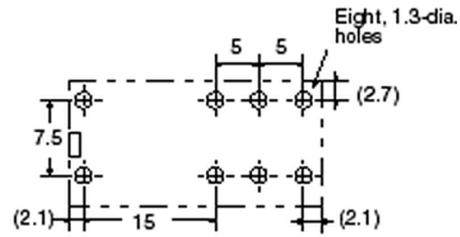


Disposición de terminales /
Conexiones internas
(Vista inferior)



Los agujeros de montaje
(vista inferior)

Tolerancia: ± 0.1



(No polaridad de la bobina)

Figura 7.14. Dimensiones de Contactos

Fuente Regulada

Como fuente regulada para la velocidad del motor se ha utilizado un circuito con el integrado LM350K (encapsulado metálico TO-3) el cual permite variar la tensión de salida entre 1.2-33V con corrientes hasta 3A.

La única precaución que se debe tomar, es montar **IC1** en un buen disipador térmico debidamente aislado. Y de ésta forma sería compuesto el circuito, como la Figura 7.15.:

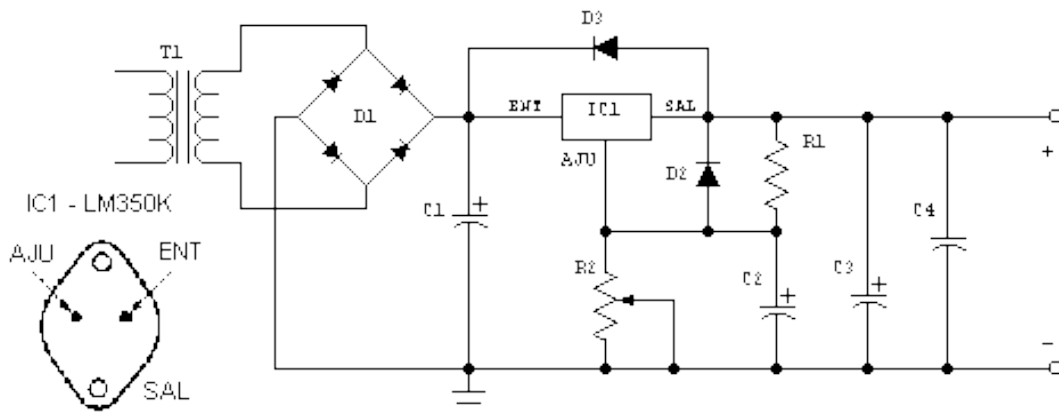


Figura 7.15. Fuente Regulada.

T1 - Transformador con primario adecuado para la red eléctrica (110 o 220V) y secundario de 24V (12+12) 3A.

IC1 - Circuito Integrado LM350K

D1 - Puente rectificador o similar. Pueden usarse también 4 diodos rectificadores para 4A y tensiones de 100V o más.

D2 y D3 - Diodos 1N4002 ~ 1N4007 o similar.

C1 - Condensador electrolítico (filtro)

C2 - Condensador electrolítico (filtro)

C3 - Condensador electrolítico (filtro)

C4 - Condensador

R1 - Resistencia

R2 - Potenciómetro lineal (no logarítmico)

El circuito está compuesto por los componentes anteriores para un fin concreto (transformador, puente de diodos, regulador de voltaje integrado LM350K, potenciómetro para regular el voltaje de salida, varios capacitores y un resistor)

El transformador mostrado es de 220VAC/24VAC, pero puede ser también uno de 110/24VAC (depende del país en que se viva). El voltaje del bobinado secundario de 24V en corriente alterna, es suficiente para hacer funcionar adecuadamente el circuito. Este regulador entregará en su salida que va desde 1.2-35V con una corriente máxima de 3A.

El puente de diodos está configurado como rectificador de onda completa.

El circuito no incluye fusible. Se recomienda colocar uno del lado del primario. El valor de éste dependerá del voltaje en el primario (110 o 220V) y hay que tomar en cuenta el valor máximo de corriente que la fuente entregará.

El potenciómetro que regula la tensión que le entrega a los relés que accionan sus contactos y estos al motor. El cambio de dirección se hace a través de los botones derecha e izquierda que invierten la polarizada del motor, por esta razón el circuito contiene dos relés para poder hacer girar el motor en ambos sentidos.

El regulador LM350K debe ir montado sobre un disipador de calor para evitar que se dañe. La corriente mínima en la carga para que la fuente funcione bien es de 5mA.

Características

De salida ajustable hasta 1,2 V y 3A de salida garantizada actual.

Regulación térmica garantizada. Límite de corriente constante con la temperatura.

La salida es a cortocircuito. 86 dB de rechazo onda.

P + Mejora del producto probado.

A parte de las características técnicas anteriormente dichas sobre el regulador LM350K se muestran en la figura 7.16., tiene un mayor rendimiento que los reguladores fijos, la serie LM150 ofrece protección contra la sobrecarga completa disponible únicamente en circuitos integrados. Incluidos en el chip son el límite de corriente, protección térmica y protección de zona segura. Todos los circuitos de protección contra sobrecarga siguen siendo totalmente funcionales incluso si el terminal de ajuste se desconecta accidentalmente.

Normalmente, no se necesitan condensadores menos que el dispositivo se encuentra a más de 6 pulgadas de los condensadores de filtro de entrada, en cuyo caso se necesita un by-pass de entrada. Un condensador de salida se pueden añadir para mejorar la respuesta transitoria, sin pasar por el pasador de ajuste aumentará el rechazo del regulador de onda.

Además de reemplazar los reguladores fijos o diseños discretos, el LM150 es útil en una amplia variedad de otras aplicaciones. Dado que el regulador está "flotando" y no ve más que la entrada a la tensión de salida diferencial, el suministro de varios cientos de voltios se puede regular el tiempo que la entrada máxima de salida de los diferenciales no se supera, es decir, evitar un cortocircuito en la salida.

Al conectar una resistencia fija entre el pasador de ajuste y de salida, el LM150 se puede utilizar como regulador de corriente de precisión. Suministros con cierre electrónico se puede conseguir por la unión de la terminal de ajuste a la tierra que los programas de la salida de 1.2V en la mayoría de las cargas de llamar la poca corriente.

Los números de parte de la serie LM150, que tiene un sufijo K son empaquetados en un estándar de acero A-3 de paquetes, mientras que los de sufijo T se empaquetan en un envase de plástico A-220. El LM150 está clasificado para $-55^{\circ}C \leq T_J \leq 150^{\circ}C$, mientras que el LM350A está clasificado para $-40^{\circ}C \leq T_J \leq 125^{\circ}C$, y el LM350 está clasificado para $0^{\circ}C \leq T_J \leq +125^{\circ}C$.

Especificaciones de tipo de letra estándar de $T_J = 25^{\circ}C$, aquellos con caracteres en negrita se aplican el rango de temperatura para un funcionamiento completo. A menos que se especifique lo contrario, $V_{IN} - V_{OUT} = 5V$, y $I_{OUT} = 10mA$. (Nota 2).

Parameter	Conditions	LM350A			LM350			Units
		Min	Typ	Max	Min	Typ	Max	
Reference Voltage	$I_{OUT} = 10mA, T_J = 25^{\circ}C$	1.238	1.250	1.262				V
	$3V \leq (V_{IN} - V_{OUT}) \leq 35V,$ $10mA \leq I_{OUT} \leq 3A, P \leq 30W$	1.225	1.250	1.270	1.20	1.25	1.30	V
Line Regulation	$3V \leq (V_{IN} - V_{OUT}) \leq 35V$ (Note 3)	0.005	0.01		0.005	0.03		%/V
		0.02	0.05		0.02	0.07		%/V
Load Regulation	$10mA \leq I_{OUT} \leq 3A$ (Note 3)	0.1	0.3		0.1	0.5		%
		0.3	1		0.3	1.5		%
Thermal Regulation	20 ms Pulse	0.002	0.01		0.002	0.03		%/W
Adjustment Pin Current		50	100		50	100		μA
Adjustment Pin Current Change	$10mA \leq I_{OUT} \leq 3A, 3V \leq (V_{IN} - V_{OUT}) \leq 35V$	0.2	5		0.2	5		μA
Temperature Stability	$T_{MIN} \leq T_J \leq T_{MAX}$		1			1		%
Minimum Load Current	$V_{IN} - V_{OUT} = 35V$		3.5	10		3.5	10	mA
Current Limit	$V_{IN} - V_{OUT} \leq 10V$	3.0	4.5		3.0	4.5		A
	$V_{IN} - V_{OUT} = 30V$	0.3	1		0.25	1		A
RMS Output Noise, % of V_{OUT}	$10Hz \leq f \leq 10kHz$		0.001			0.001		%
Ripple Rejection Ratio	$V_{OUT} = 10V, f = 120Hz, C_{ADJ} = 0\mu F$		65			65		dB
	$V_{OUT} = 10V, f = 120Hz, C_{ADJ} = 10\mu F$	66	86		66	86		dB
Long-Term Stability	$T_J = 125^{\circ}C, 1000hrs$		0.25	1		0.25	1	%
Thermal Resistance, Junction to Case	K Package					1.2	1.5	$^{\circ}C/W$
	T Package		3	4		3	4	$^{\circ}C/W$
Thermal Resistance, Junction to Ambient (No Heat Sink)	K Package					35		$^{\circ}C/W$
	T Package		50			50		$^{\circ}C/W$

Figura 7.16. Características LM150K

Aplicaciones cómo suministros eléctricos ajustables, reguladores de intensidad constante, cargadores de batería...

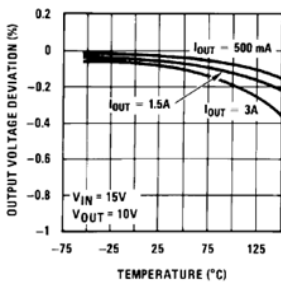
Nota 1: Los rangos máximos absolutos son los límites que indican los daños que puedan ocurrir en el aparato. Los valores de operación indican las condiciones para las que se destina el producto para ser funcional, pero no garantizan los límites de rendimiento específicos. De las especificaciones garantizadas y condiciones de ensayo, ver las características eléctricas.

Nota 2: Estas especificaciones son aplicables a las disipaciones de potencia de hasta 30W de la A-3 (K) y el paquete de 25 W para el A-220 (T) del paquete. La disipación de potencia está garantizada en estos valores hasta 15V de entrada y salida diferencial. Por encima de 15V diferencial, la disipación de potencia estará limitada por los circuitos de protección interna. Todos los límites (es decir, los números en la Min. y Max. columnas) están garantizados para AOQL Nacional (nivel medio de calidad de salida).

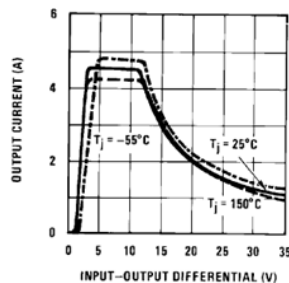
Nota 3: El Reglamento está medido a una temperatura de la unión constante, utilizando la prueba de pulso con un ciclo de baja. Los cambios en la tensión de salida debido a los efectos de calentamiento son cubiertos por las especificaciones para la regulación térmica.

Nota 4: Consulte RETS150K de dibujo para las especificaciones militares de laLM150K.

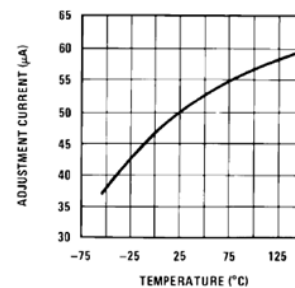
Regulación de carga



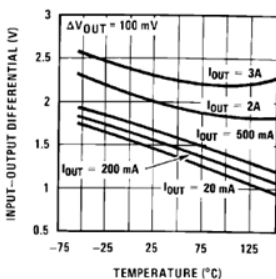
Corriente límite



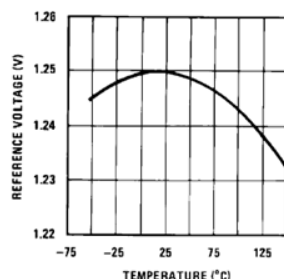
Ajuste de corriente



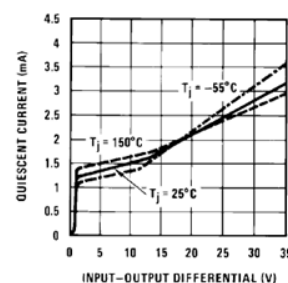
Voltaje Dropout



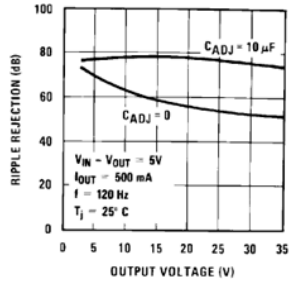
Estabilidad de la temperatura



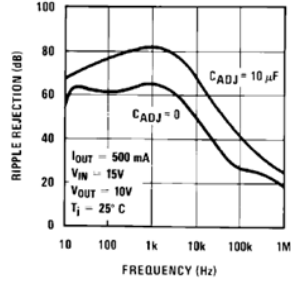
Corriente Mínima de trabajo



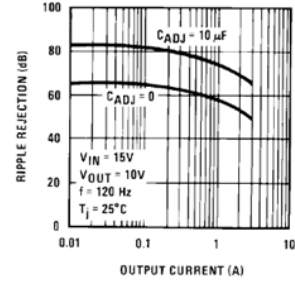
Rechazo al Rizado



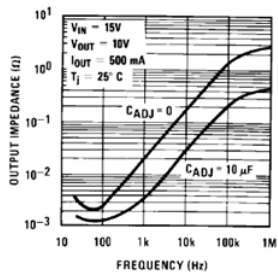
Rechazo al Rizado



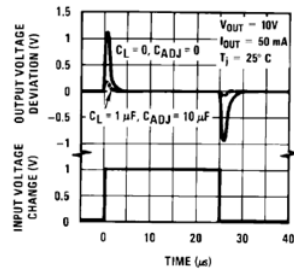
Rechazo al Rizado



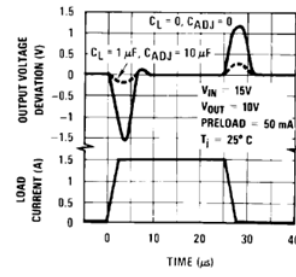
Impedancia de Salida



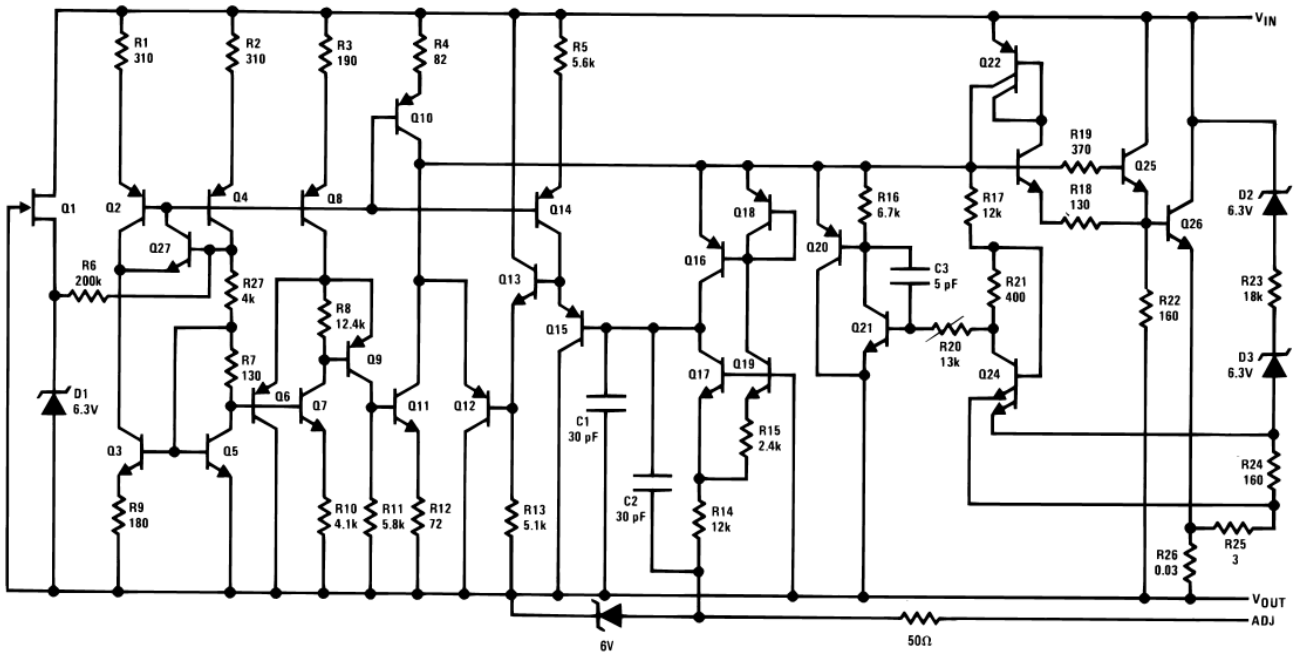
Respuesta Transitoria lineal



Respuesta Transitoria de la Carga



Esquemático:



Dimensiones (mm) mostradas en la siguiente Figura 7.17.:

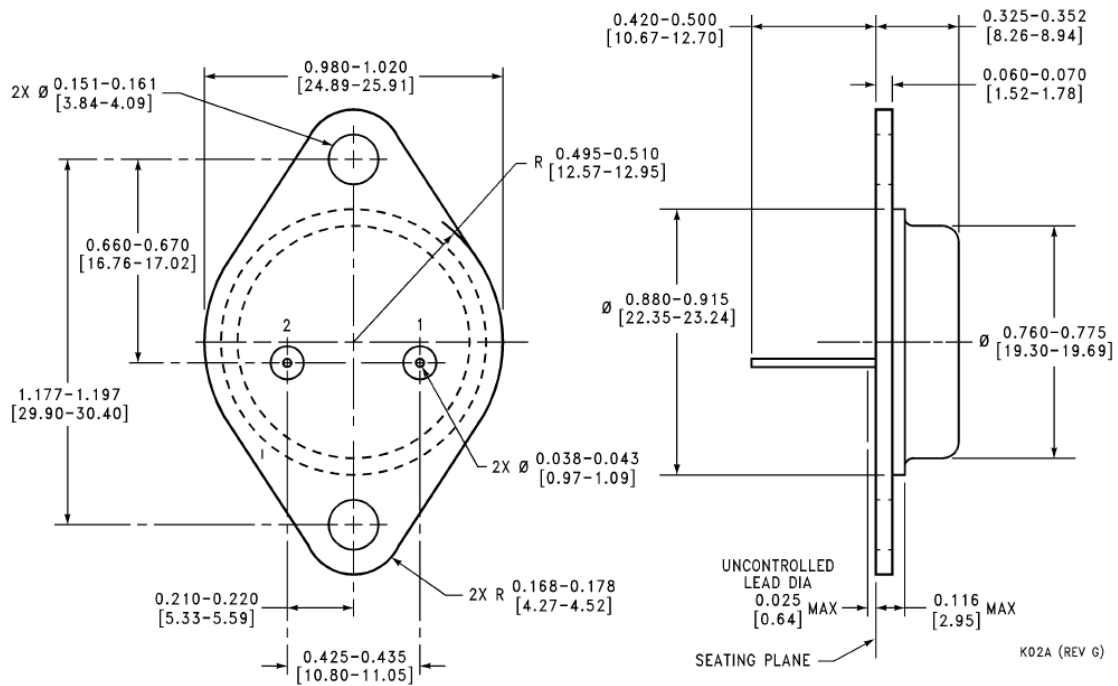


Figura 7.17.Dimensiones.

7.5.Anexo Fuente de Alimentación Omron S8VS

La fuente con la que se alimenta todos los componentes de la maqueta son a través de la fuente Omron S8VS serie de uso general de alimentación, modelo estándar, entrada de 100 a 240 VAC 30 W, salida de 24V 1.3A. Número de pieza: S8VS-03024.

Tipos:

Modelos 60/90/120/180/240/480-W Nuevos modelos con el Monitor de visualización y funciones simples para el sistema de puesta en marcha sencilla:

Nuevo modelo de 90 W con el monitor indica que se ajusten a las normas UL Clase de salida 2. Nuevos modelos con bloques de terminales sin tornillos y controlar indicación. Y estado que se muestra en tres dígitos, display de 7 segmentos.

Las normas de seguridad: UL508/60950-1, CSA C22.2 N ° 14/60950-1 (15-W y los modelos de 30 W), CSA C22.2 No. 107.1/60950-1 (de 60 W a 240 W- y 480-W modelos), EN 50178 (= VDE0160).

Condiciones de entrada: entrada DC también es posible desde 80 hasta 370 VDC (BE sólo en los modelos, no cumple con las Directivas CE y otras normas de seguridad).

Nuestro modelo pertenece al catálogo de los Modelos 15/30-W compacto, delgado suministros de energía que se montan en cualquier lugar de contribuir al control de reducción de personal del Grupo Especial.

Compacto y delgado: 22,5 x 85 x 96,5 mm (W x H x D).

Tres direcciones de montaje (de serie, horizontal, frente a horizontal).

Montaje directamente al panel es posible.

Las normas de seguridad: UL508/60950-1/1604, CUL: CSA C22.2 No. 14/60950-1/213, EN50178 (= VDE0160).

Las características comunes a todos los modelos

Montaje a carril DIN.

Cumple con la SEMI F47-0706 (200 VAC de entrada).

Que cumplen con ROHS

Dimensiones (mm) se muestran en la siguiente Figura 7.18.:

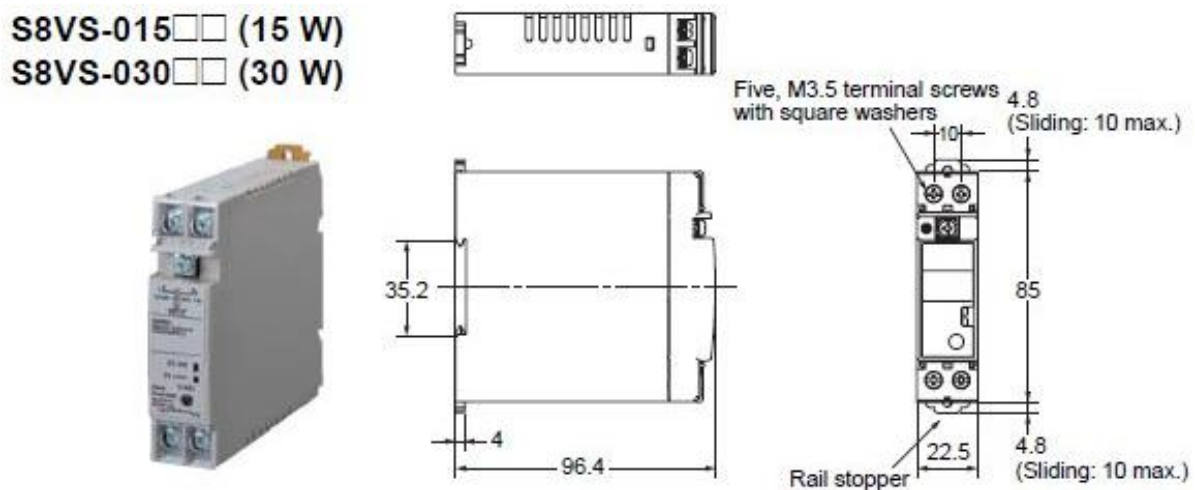


Figura 7.18.Dimensiones.

Potencia del motor	30 W
Eficiencia	80% mínimo.
Tensión nominal de entrada	100 a 240 VCA
Permitida- de entrada de tensión de la variable	85 a 264 VAC
Frecuencia	50 / 60 Hz (47 a 450 Hz de una sola fase)
Corriente nominal de entrada	0.9 A Max. (A 100 VAC de entrada) 0.6 A máx. (A 200 VAC)
Emisiones de corriente armónica	Cumple con EN61000- 3 – 2
La corriente de fuga	0.5 mA. Max. (A 100 VAC) 1 mA. máx. (A 200 VAC)
Corriente de entrada	25 A máx. (A 100 VAC de entrada (para el arranque en frío a 25 CEL)) 50 A máx. (A 200 VAC de entrada (para el arranque en frío a 25 CEL))
Tensión de salida	24 VDC
Tensión de salida variable de rango	- 10 a + 15% (con V. ADJ)
Onda	2% (p- p) Max. (Bajo la calificación de I / O las condiciones.)
Influencia de la variación de entrada estático	0.5% Max. (de 85 a 264 VAC de entrada, 100% de carga)
Influencia de la variación de carga estática	1.5% Max. (tensión de salida: 24 V) (A nominal de entrada, 0 a 100% de carga)
Variación de la temperatura ambiente influyen	0.05% / CEL Max.
Corriente nominal de salida	1.3 A
El tiempo de inicio	1000 ms máx. (En nominal de entrada / salida de tensión, el arranque en frío a 25 CEL)
Tiempo de espera	Min 20 ms. (En nominal de entrada / salida de tensión, el arranque en frío a 25 CEL)
Protección de sobrecarga	105% a 160% de la corriente nominal de salida invertida caída de tensión en L, una operación intermitente, rearme automático
Protección contra sobretensiones	Operado en el 130% Min. de la tensión de salida nominal, Shut- off Restablecer: Reajuste la potencia de volver a (tiempo OFF: 3 min)
Previsión de mantenimiento de salida de monitor	
El tiempo total de ejecución de salida de monitor	No
Salida de alarma de mínima tensión	
Corte de corriente de salida de alarma	

Funcionamiento en paralelo	
Función de detección a distancia	
Funcionamiento en serie	Sí (hasta 2 fuentes de alimentación con diodo externo)
La norma aplicable (UL)	Estándar N°: UL508 (Listado, Clase 2: Por UL1310) , UL60950- 1, UL 1604 (Clase I/División2)
La norma aplicable (CSA)	Estándar N°: CUL: CSA C22. 2 N ° 14 (Clase 2), N ° 213 (clase I / División 2), CUR: CSA C22. 2 N° 60950 – 1
La norma aplicable (EN)	Estándar N ° : EN50178
La norma aplicable (Directiva de la CE (EMI))	Emisión de Caja: Cumple las normas EN61204- 3 Clase B de red de emisión AC: Cumple las normas EN61204- 3 Clase B, FCC Clase Sobre la base de una armónica de corriente: EN61000- 3 - 2 EMS: las normas EN61204- 3 Clase B
La norma aplicable (VDE)	Estándar N ° : VDE0160 Número expediente:
Norma aplicable (SEMI)	F47- 0200 (200 VAC)
Temperatura ambiente	En funcionamiento: - 10 a + 60 CEL de almacenamiento: - 25 a + 65 CEL (sin hielo ni condensación)
Humedad ambiental	En funcionamiento: 25 a 85% de almacenamiento: 25 a 90% (sin hielo ni condensación)
Rigidez dieléctrica	Entre todas las E/S: 3 kVc.a. Durante 1 min (corriente de detección: 20 mA.) Entre todas las entradas y terminales PE: 2 kVc.a. durante 1 min (corriente de detección: 20 mA.) entre todas las salidas y terminales PE: 1 kVc.a. durante 1 min (corriente de detección: 20 mA.)
Resistencia de aislamiento	Entre todas las salidas y las entradas / terminales PE: 100 KOhm Min. (a 500 VDC)
Resistencia a las vibraciones	10 a 55 Hz, 0. 375 mm de amplitud durante 2 h en X, Y, Z.
Resistencia a los golpes	150 m / s * * 2, 3 veces en las direcciones X, Y, Z y direcciones
Indicador de salida	Sí (color: verde)
Calor por radiación	Natural de aire de refrigeración
Construcción	Tipo de cubierta
Tipo de terminal	Terminal de tornillo
Tipo de montaje	Montaje en carril DIN de tipo
Peso	180 g Max.

Nota: No utilice la salida del variador de la fuente de alimentación con variadores de frecuencia de salida de 50/60Hz están disponibles, pero el aumento de la temperatura interna de la fuente de alimentación puede provocar la ignición de este.

Su diagrama de conexión interno es el indicado en la siguiente Figura 7.19.:

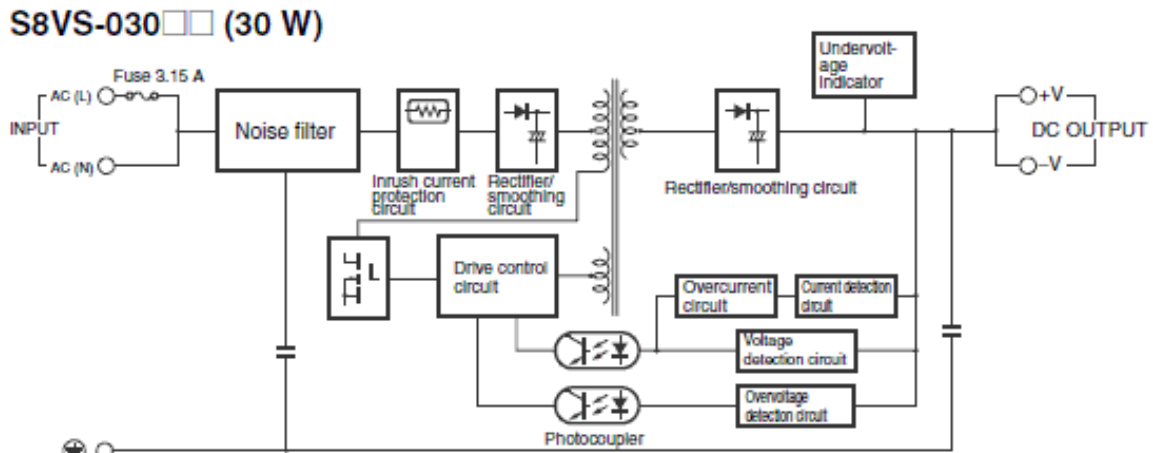


Figura 7.19. Conexiones Internas.

Y la curva de reducción indicada en la figura 7.20.:

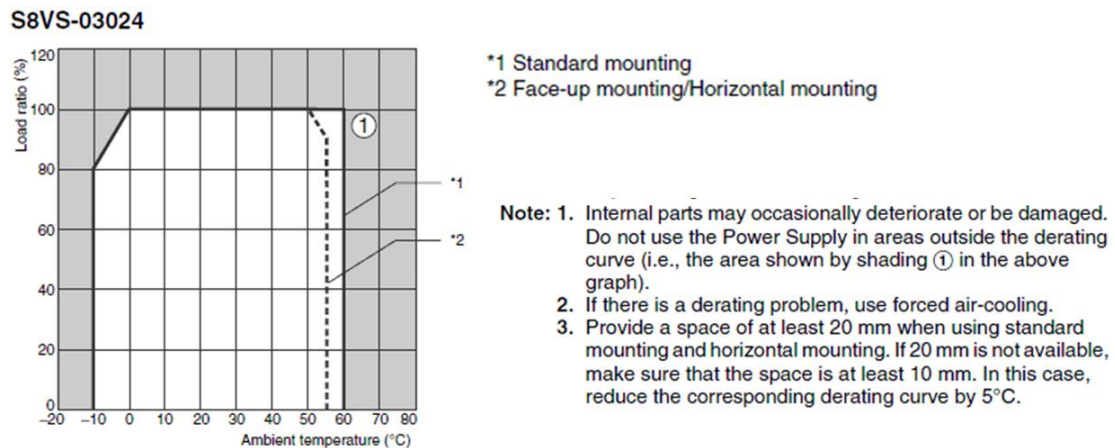


Figura 7.20. Radio de Carga-Temperatura.

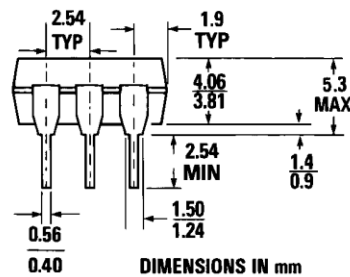
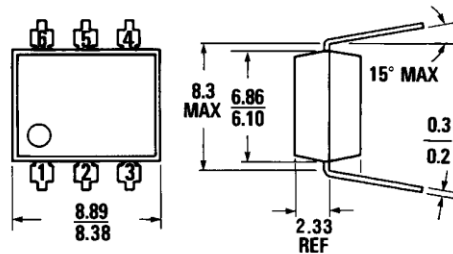
7.6. Anexo Opto-Acoplador 4N25

El 4N25 es un opto-acoplador que tiene un NPN plano de silicio, es fototransistor acoplado que recibe ópticamente la señal a través de un diodo de arseniuro de galio.

Aplicaciones

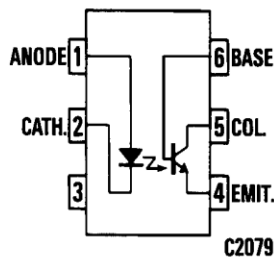
- Aislador de AC línea/lógica digital
- Aislador lógica digital/lógica digital
- Recibidor de línea teléfono/telégrafo
- Receptor de par trenzado de línea
- Control de realimentación de alta frecuencia de alimentación
- Relé de control de monitor
- Fuente de alimentación de monitor
- Tamaño pequeño de package y bajo coste
- Excelente respuesta de frecuencia
- UL reconocido-archivo E90700

Dimensiones del Package



DIMENSIONS IN mm
PACKAGE CODE K

ST1603A



C2079

Características técnicas:

ABSOLUTE MAXIMUM RATINGS	
TOTAL PACKAGE	
*Storage temperature	–55°C to 150°C
*Operating temperature at junction	–55°C to 100°C
*Lead temperature (soldering, 10 sec)	260°C
*Total package power dissipation at 25°C ambient (LED plus detector)	250 mW
*Derate linearly from 25°C	3.3 mW/°C
INPUT DIODE	
*Forward DC current continuous	80 mA
*Reverse voltage	3.0 V
*Peak forward current (300 μs, 2% duty cycle)	3.0 A
*Power dissipation at 25°C ambient	150 mW
*Derate linearly from 25°C	2.0 mW/°C
OUTPUT TRANSISTOR	
*Collector emitter voltage (BV _{CEO})	30 V
*Collector base voltage (BV _{CBO})	70 V
*Emitter collector voltage (BV _{ECO})	7 V
*Power dissipation at 25°C ambient	150 mW
*Derate linearly from 25°C	2.0 mW/°C
*Indicates JEDEC Registered Data.	

ELECTRO-OPTICAL CHARACTERISTICS
(25°C Free Air Temperature Unless Otherwise Specified)

INDIVIDUAL COMPONENT CHARACTERISTICS						
CHARACTERISTICS	SYMBOL	MIN.	TYP.	GUAR. MAX.	UNITS	TEST CONDITIONS
INPUT DIODE						
*Forward voltage	V _F		1.20	1.50	V	I _F = 10 mA
Capacitance	C		150		pF	V _F = 0 V, f = 1 MHz
*Reverse leakage current			.05	100	μA	V _R = 3.0 V, R _L = 1.0 MΩ
DETECTOR						
DC forward current gain	h _{FE}		250			V _{CE} = 5 V, I _C = 500 μA
*Collector to emitter breakdown voltage	BV _{CEO}	30	65		V	I _C = 1.0 mA, I _B = 0
*Collector to base breakdown voltage	BV _{CBO}	70	165		V	I _C = 100 μA, I _E = 0
*Emitter to collector breakdown voltage	BV _{ECO}	7	14		V	I _E = 100 μA, I _B = 0
*Collector to emitter leakage current (4N25, 4N26, 4N27)	I _{CEO}		3.5	50	nA	V _{CE} = 10 V Base Open
*Collector to emitter leakage current (4N28)				100	nA	
*Collector to base leakage current	I _{CBO}		0.1	20	nA	V _{CB} = 10 V Emitter Open

TRANSFER CHARACTERISTICS						
DC CHARACTERISTICS	SYMBOL	MIN.	TYP.	GUAR. MAX.	UNITS	TEST CONDITIONS
*Collector output current (a) (4N25, 4N26) (4N27, 4N28)	I_C	2.0 1.0	5.0 3.0	— —	mA	$V_{CE}=10\text{ V}$, $I_F=10\text{ mA}$, $I_B=0$
*Collector-emitter saturation	$V_{CE(SAT)}$		0.2	0.5	V	$I_C=2.0\text{ mA}$, $I_F=50\text{ mA}$

TRANSFER CHARACTERISTICS				
AC CHARACTERISTICS	SYMBOL	TYP.	UNITS	TEST CONDITIONS
Non-saturated Collector Delay time	t_c	0.5	μS	$R_L=100\ \Omega$, $I_C=2\text{ mA}$, $V_{CC}=10\text{ V}$ (Fig. 10 and 11)
Rise time	t_r	2.5	μS	
Fall time	t_f	2.6	μS	
Non-saturated Collector Delay time	t_d	2.0	μS	$R_L=1\text{ k}\Omega$, $I_C=2\text{ mA}$, $V_{CC}=10\text{ V}$ (Fig. 10 and 11)
Rise time	t_r	15	μS	
Fall time	t_f	15	μS	

*Indicates JEDEC Registered Data.

(a) Pulse Test: Pulse Width=300 μs , Duty Cycle $\leq 2.0\%$

(b) For this test LED pins 1 and 2 are common and Phototransistor pins 4, 5 and 6 are common.

(c) If adjusted to yield $I_C=2\text{ mA}$ and $I_C=0.7\text{ mA RMS}$; Bandwidth referenced to 10 kHz.

ELECTRO-OPTICAL CHARACTERISTICS
(25°C Free Air Temperature Unless Otherwise Specified) (Cont'd)

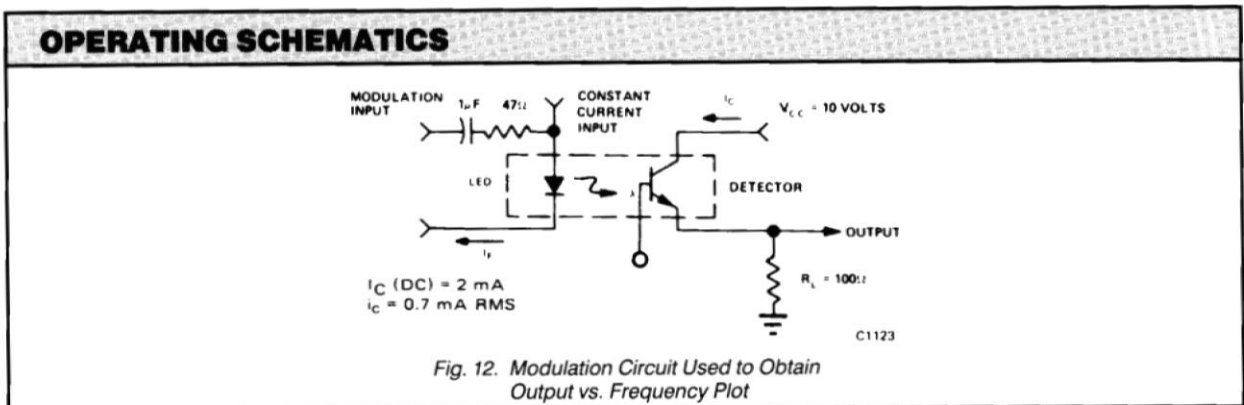
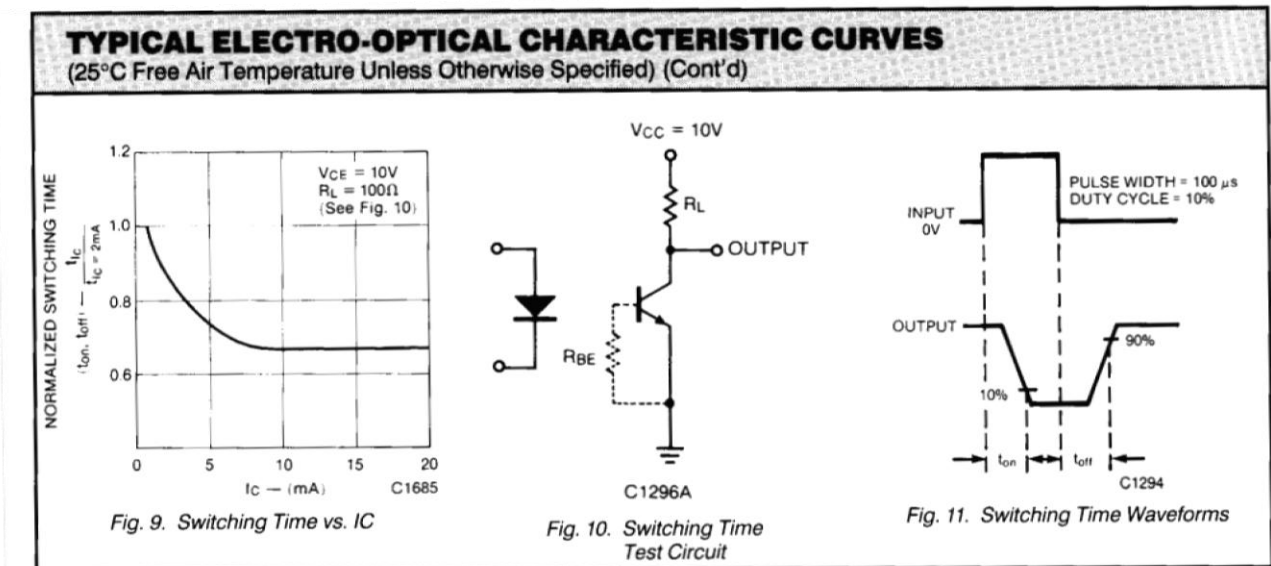
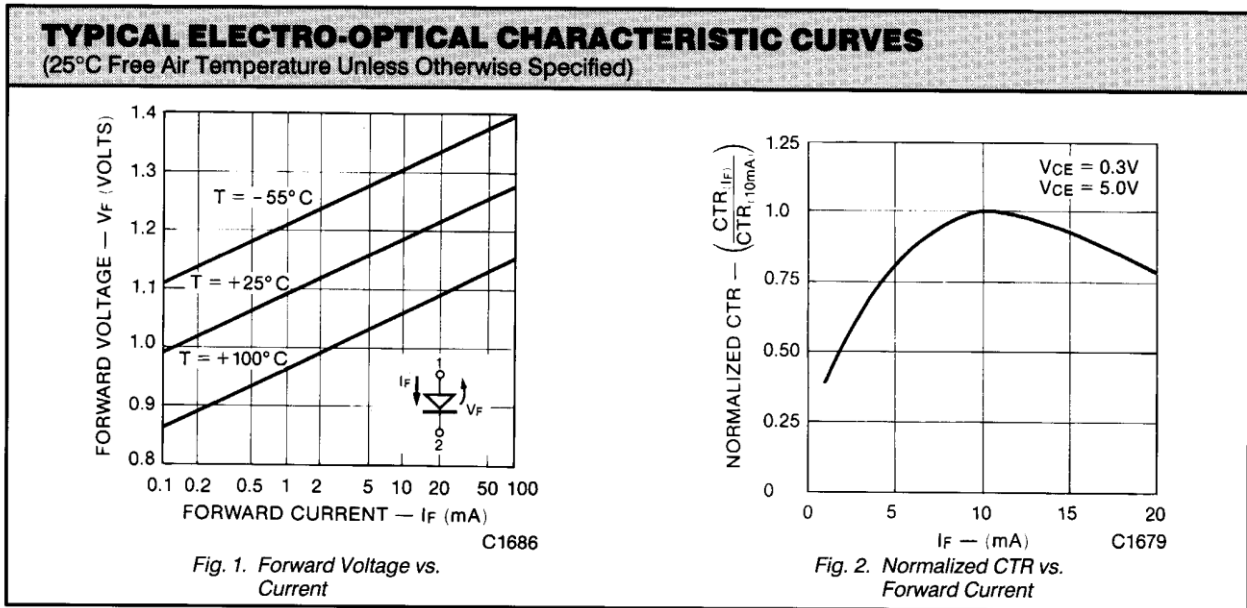
TRANSFER CHARACTERISTICS (Cont'd)						
AC CHARACTERISTICS	SYMBOL	MIN.	TYP.	GUAR. MAX.	UNITS	TEST CONDITIONS
Saturated t_{on} (from 5 V to 0.8 V)	t_{on} (SAT)		5		μS	$R_L=2\text{ k}\Omega$, $I_F=15\text{ mA}$, $V_{CC}=5\text{ V}$
t_{off} (from SAT to 2.0 V)	t_{off} (SAT)		25		μS	$R_B=\text{Open}$ (Fig. 10)
Saturated t_{on} (from 5 V to 0.8 V)	t_{on} (SAT)		5		μS	$R_L=2\text{ k}\Omega$, $I_F=20\text{ mA}$, $V_{CC}=5\text{ V}$
t_{off} (from SAT to 2.0 V)	t_{off} (SAT)		18		μS	$R_B=100\text{ k}\Omega$ (Fig. 10)
Non-saturated Base—Collector photo diode Rise time	t_r		175		ns	$R_L=1\text{ k}\Omega$, $V_{CB}=10\text{ V}$
Fall time	t_f		175		ns	
Isolation voltage (b) (4N25, 4N26, 4N27, 4N28) *(4N26, 4N27) *(4N28)	V_{ISO}	5300 1500 500	— — —	— — —	V V V	$I_{I_0} \leq 1\ \mu\text{A}$ RMS, $t=1\text{ minute}$ Peak Peak
Isolation resistance (b)			10^{11}		Ω	$V=500\text{ VDC}$
Isolation capacitance (b)			1.3		pF	$V=0$, $f=1.0\text{ MHz}$
Bandwidth (c) (also see note 2)	B_w		300		kHz	$I_C=2.0\text{ mA}$, $R_L=100\ \Omega$ (Fig. 12)

*Indicates JEDEC Registered Data.

(a) Pulse Test: Pulse Width=300 μs , Duty Cycle $\leq 2.0\%$

(b) For this test LED pins 1 and 2 are common and Phototransistor pins 4, 5 and 6 are common.

(c) If adjusted to yield $I_C=2\text{ mA}$ and $I_C=0.7\text{ mA RMS}$; Bandwidth referenced to 10 kHz.



La proporción de transferencia corriente (I_c/I_f) es la proporción de la corriente de colector de detector a la corriente de entrada conducida con V_{ce} en 10 voltios.

La frecuencia en la cual I_c está 3dB abajo de 10KHz el valor.

El tiempo de subida es el tiempo requerido para el colector para aumentar del 10 % de su valor final al tiempo de caída del 90% es el tiempo requerido para la corriente de colector para disminuirse del 90 % de su valor inicial al 10 %.

7.7. Anexo Opto-Acoplador 6N137

Descripción:

El 6N137 es un opto-acoplador que funciona a una alta frecuencia, la puerta lógica del foto-detector integrado contiene una salida de stroble. Esta salida cuenta con un colector abierto, lo que permite por cable salidas OR. El acoplado los parámetros están garantizados durante el intervalo de temperatura de -40°C a $+85^{\circ}\text{C}$. A señal de entrada máximo de 5mA proporcionar un sumidero de salida de corriente mínima de 13mA (ventilador de 8).

Un protector de ruido interno proporciona un rechazo superior de modo común de típicamente 10 KV / ms. El HCPL-2601 HCPL y 2631-CMR tiene un mínimo de 5 KV / ms. El HCPL-2611 tiene un mínimo CMR de 10 KV / μS .

Características:

Muy alta velocidad-10 MBit / s.

Superior CMR-10 KV / ms.

Doble voltaje de funcionamiento a 480 V.

Fan-out, de 8 a -40°C a $+85^{\circ}\text{C}$.

Lógica de salida de la puerta.

Salida de Stroble.

O con conexión de cable-colector abierto.

U.L. reconocido (El archivo # E90700).

Aplicaciones:

- Bucle de eliminación de tierra.
- LSTTL a TTL, LSTTL o CMOS de 5 voltios.
- Receptor de Línea, de transmisión de datos.
- Datos multiplexado.
- Conmutación de suministros de energía.
- Reemplazo de transformador de pulso.
- Computadora de interfaz periférica.

TRUTH TABLE
(Positive Logic)

Input	Enable	Output
H	H	L
L	H	H
H	L	H
L	L	H
H	NC	L
L	NC	H

A 0.1 μ F bypass capacitor must be connected between pins 8 and 5.
(See note 1)

Características técnicas:

ABSOLUTE MAXIMUM RATINGS (No derating required up to 85°C)			
Parameter	Symbol	Value	Units
Storage Temperature	T_{STG}	-55 to +125	°C
Operating Temperature	T_{OPR}	-40 to +85	°C
Lead Solder Temperature	T_{SOL}	260 for 10 sec	°C
EMITTER			
DC/Average Forward	I_F	50	mA
Input Current		30	
Enable Input Voltage	V_E	5.5	V
Not to exceed V_{CC} by more than 500 mV			
Reverse Input Voltage	V_R	5.0	V
Power Dissipation	P_I	100	mW
		45	
DETECTOR			
Supply Voltage	V_{CC} (1 minute max)	7.0	V
Output Current	I_O	50	mA
		50	
Output Voltage	V_O	7.0	V
Collector Output	P_O	85	mW
Power Dissipation		60	

RECOMMENDED OPERATING CONDITIONS				
Parameter	Symbol	Min	Max	Units
Input Current, Low Level	I_{FL}	0	250	μA
Input Current, High Level	I_{FH}	*6.3	15	mA
Supply Voltage, Output	V_{CC}	4.5	5.5	V
Enable Voltage, Low Level	V_{EL}	0	0.8	V
Enable Voltage, High Level	V_{EH}	2.0	V_{CC}	V
Low Level Supply Current	T_A	-40	+85	$^{\circ}C$
Fan Out (TTL load)	N		8	

* 6.3 mA is a guard banded value which allows for at least 20 % CTR degradation. Initial input current threshold value is 5.0 mA or less

ELECTRICAL CHARACTERISTICS ($T_A = -40^{\circ}C$ to $+85^{\circ}C$ Unless otherwise specified.)

INDIVIDUAL COMPONENT CHARACTERISTICS

Parameter	Test Conditions	Symbol	Min	Typ**	Max	Unit
EMITTER						
Input Forward Voltage	($I_F = 10$ mA) $T_A = 25^{\circ}C$	V_F		1.4	1.8	V
Input Reverse Breakdown Voltage	($I_R = 10$ μA)	B_{VR}	5.0			V
Input Capacitance	($V_F = 0$, $f = 1$ MHz)	C_{IN}		60		pF
Input Diode Temperature Coefficient	($I_F = 10$ mA)	$\Delta V_F / \Delta T_A$		-1.4		mV/ $^{\circ}C$
DETECTOR						
High Level Supply Current	Single Channel Dual Channel ($V_{CC} = 5.5$ V, $I_F = 0$ mA) ($V_E = 0.5$ V)	I_{CCH}		7	10	mA
Low Level Supply Current				9	13	
Low Level Enable Current	Single Channel Dual Channel ($V_{CC} = 5.5$ V, $V_E = 0.5$ V)	I_{CCL}		15	20	mA
High Level Enable Current	($V_{CC} = 5.5$ V, $I_F = 10$ mA) ($V_E = 0.5$ V)			19	26	
High Level Enable Voltage	($V_{CC} = 5.5$ V, $V_E = 2.0$ V)	I_{EH}	2.0	-0.6	-1.6	mA
Low Level Enable Voltage	($V_{CC} = 5.5$ V, $I_F = 10$ mA) (Note 3)	V_{EL}			0.8	V

SWITCHING CHARACTERISTICS ($T_A = -40^{\circ}C$ to $+85^{\circ}C$, $V_{CC} = 5$ V, $I_F = 7.5$ mA Unless otherwise specified.)

AC Characteristics	Test Conditions	Symbol	Min	Typ**	Max	Unit
Propagation Delay Time to Output High Level	(Note 4) ($T_A = 25^{\circ}C$) ($R_L = 350 \Omega$, $C_L = 15$ pF) (Fig. 12)	T_{PLH}	20	45	75	ns
Propagation Delay Time to Output Low Level	(Note 5) ($T_A = 25^{\circ}C$) ($R_L = 350 \Omega$, $C_L = 15$ pF) (Fig. 12)			25	45	
Pulse Width Distortion	($R_L = 350 \Omega$, $C_L = 15$ pF) (Fig. 12)	$ T_{PHL} - T_{PLH} $		3	35	ns
Output Rise Time (10-90%)	($R_L = 350 \Omega$, $C_L = 15$ pF) (Note 6) (Fig. 12)	t_r		50		ns
Output Fall Time (90-10%)	($R_L = 350 \Omega$, $C_L = 15$ pF) (Note 7) (Fig. 12)	t_f		12		ns
Enable Propagation Delay Time to Output High Level	($I_F = 7.5$ mA, $V_{EH} = 3.5$ V) ($R_L = 350 \Omega$, $C_L = 15$ pF) (Note 8) (Fig. 13)	t_{ELH}		20		ns
Enable Propagation Delay Time to Output Low Level	($I_F = 7.5$ mA, $V_{EH} = 3.5$ V) ($R_L = 350 \Omega$, $C_L = 15$ pF) (Note 9) (Fig. 13)	t_{EHL}		20		ns
Common Mode Transient Immunity (at Output High Level)	($T_A = 25^{\circ}C$) $ V_{CM} = 50$ V, (Peak) ($I_F = 0$ mA, $V_{OH}(\text{Min.}) = 2.0$ V) 6N137, HCPL-2630 ($R_L = 350 \Omega$) (Note 10) HCPL-2601, HCPL-2631 (Fig. 14) HCPL-2611 $ V_{CM} = 400$ V	$ CM_H $	5000	10,000	10,000	V/ μs
Common Mode Transient Immunity (at Output Low Level)	($R_L = 350 \Omega$) ($I_F = 7.5$ mA, $V_{OL}(\text{Max.}) = 0.8$ V) 6N137, HCPL-2630 $ V_{CM} = 50$ V (Peak) HCPL-2601, HCPL-2631 ($T_A = 25^{\circ}C$) (Note 11) (Fig. 14) HCPL-2611 ($T_A = 25^{\circ}C$) $ V_{CM} = 400$ V			10,000	15,000	

TRANSFER CHARACTERISTICS ($T_A = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ Unless otherwise specified.)						
DC Characteristics	Test Conditions	Symbol	Min	Typ**	Max	Unit
High Level Output Current	($V_{CC} = 5.5\text{ V}$, $V_O = 5.5\text{ V}$) ($I_F = 250\ \mu\text{A}$, $V_E = 2.0\text{ V}$) (Note 2)	I_{OH}			100	μA
Low Level Output Current	($V_{CC} = 5.5\text{ V}$, $I_F = 5\text{ mA}$) ($V_E = 2.0\text{ V}$, $I_{CL} = 13\text{ mA}$) (Note 2)	V_{OL}		.35	.06	V
Input Threshold Current	($V_{CC} = 5.5\text{ V}$, $V_O = 0.6\text{ V}$, $V_E = 2.0\text{ V}$, $I_{OL} = 13\text{ mA}$)	I_{FT}		3	5	mA

ISOLATION CHARACTERISTICS ($T_A = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ Unless otherwise specified.)						
Characteristics	Test Conditions	Symbol	Min	Typ**	Max	Unit
Input-Output Insulation Leakage Current	(Relative humidity = 45%) ($T_A = 25^{\circ}\text{C}$, $t = 5\text{ s}$) ($V_{I-O} = 3000\text{ VDC}$) (Note 12)	I_{I-O}			1.0*	μA
Withstand Insulation Test Voltage	($RH < 50\%$, $T_A = 25^{\circ}\text{C}$) (Note 12) ($t = 1\text{ min.}$)	V_{ISO}	2500			V_{RMS}
Resistance (Input to Output)	($V_{I-O} = 500\text{ V}$) (Note 12)	R_{I-O}		10^{12}		Ω
Capacitance (Input to Output)	($f = 1\text{ MHz}$) (Note 12)	C_{I-O}		0.6		pF

** All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25^{\circ}\text{C}$

Notas:

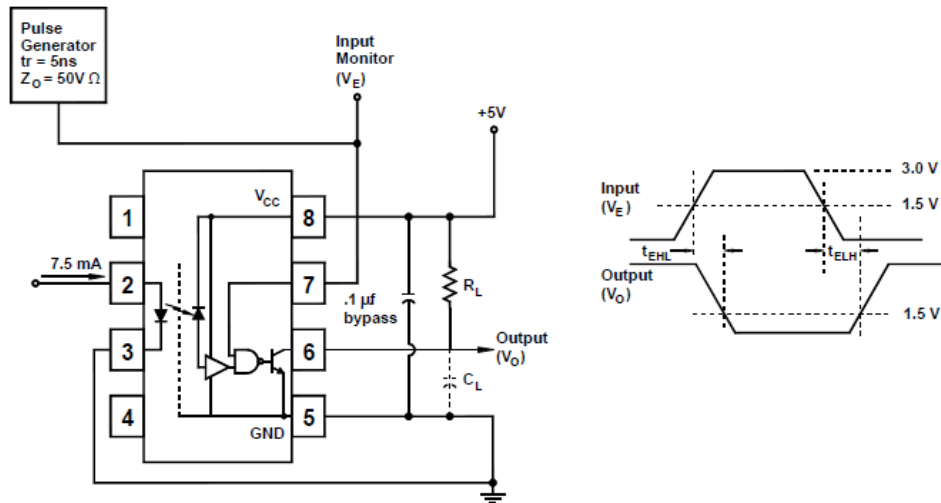
1. Deben evitar el suministro de VCC a cada opto-aislador un 0.1 μ El condensador de f o más grande. Esto puede ser condensador de tántalo de cerámica o sólido con la alta característica de frecuencia buena y debería ser unido como cerca como posible al paquete VCC y los alfileres de GND de cada dispositivo.
2. Cada canal.
3. No Permita la Entrada - Ningún tirón encima de la resistencia requerida como el dispositivo tiene un tirón interno encima de la resistencia.
4. tPLH - el retraso de Propagación es medido del 3.75 nivel de mA. sobre el ALTO a la transición BAJA del pulso de corriente de entrada al 1.5 nivel de V sobre el BAJO A LA ALTA transición del pulso de voltaje de salida.
5. tPHL -el retraso de Propagación es medido del 3.75 nivel de mA. sobre el BAJO A LA ALTA transición del pulso de corriente de entrada al 1.5 nivel de V sobre el ALTO a la transición BAJA del pulso de voltaje de salida.
6. tr -el tiempo de Subida es medido del 90% a los niveles del 10% sobre el BAJO A LA ALTA transición del pulso de salida.
7. tf -el tiempo de Caída es medido del 10% a los niveles del 90% sobre el ALTO a la transición BAJA del pulso de salida.
8. tELH -Permiten el retraso de propagación de entrada es medido del 1.5 nivel de V sobre el ALTO a la transición BAJA del pulso de voltaje de entrada al 1.5 nivel de V sobre el BAJO A LA ALTA transición del pulso de voltaje de salida.
9. tEHL -Permiten el retraso de propagación de entrada es medido del 1.5 nivel de V sobre el BAJO A LA ALTA transición del pulso de voltaje de entrada al 1.5 nivel de V sobre el ALTO a la transición BAJA del pulso de voltaje de salida.

10. CMH -la tarifa máxima tolerable de subida del voltaje de modo común para asegurar la salida permanecerá en el alto estado (p.ej., $V_{OUT} > 2.0V$). Medido en voltios por microsegundo ($V/\mu s$).

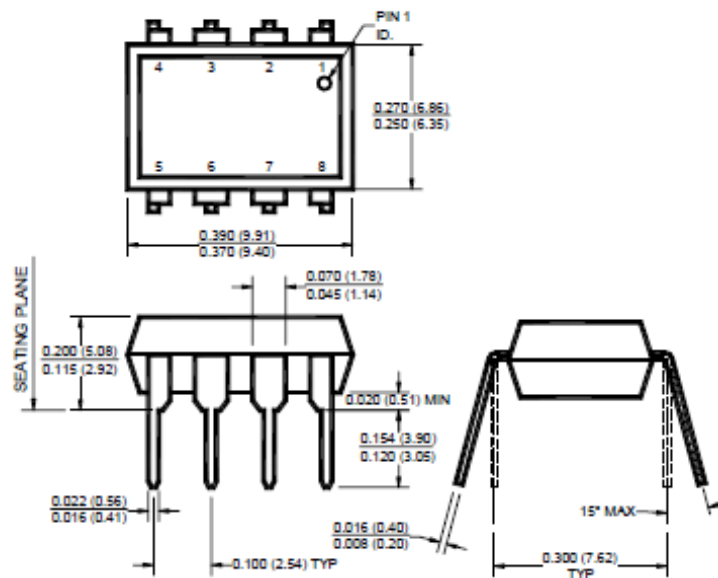
11. CML -la tarifa máxima tolerable de subida del voltaje de modo común para asegurar la salida permanecerá en el estado de salida bajo (p.ej., $V_{OUT} < 0.8V$). Medido en voltios por microsegundo ($V/\mu s$).

12. El dispositivo consideró un dispositivo de dos terminales: Fija 1, 2,3 y 4 shorted juntos, y Fija 5, 6,7 y 8 shorted juntos.

Conexión básica:



Dimensiones:



7.8. Anexo Circuito Integrado LMD18200 3A, 55V, H-Bridge

Descripción:

El LMD18200 es un 3A el Puente de h diseñado para usos de control de movimiento. El dispositivo es construido usando un proceso de multi-tecnología que se combina bipolar y el trazado de circuito de control de CMOS con dispositivos de poder de DMOS sobre la misma estructura monolítica. Ideal para conducir DC y motores paso a paso; el LMD18200 acomoda corrientes de salida máximas hasta 6A. Un circuito innovador que facilita el sentir de pérdida baja de la corriente de salida ha sido puesto en práctica.

Características:

Entrega hasta 3A la salida continua.

Funciona en voltajes de suministro hasta 55V.

Bajo RDS (SOBRE) típicamente 0.3W por interruptor.

TTL y entradas CMOS compatibles.

Salida de bandera de advertencia corriente Termal en 145°C.

A la parada Termal (salidas de) en 170°C diodos de abrazadera Internos.

Shorted la protección de carga.

La bomba de precio Interna con la capacidad de elástico de bota externa.

Aplicaciones:

Para corriente continua y conducción de motores paso a paso.

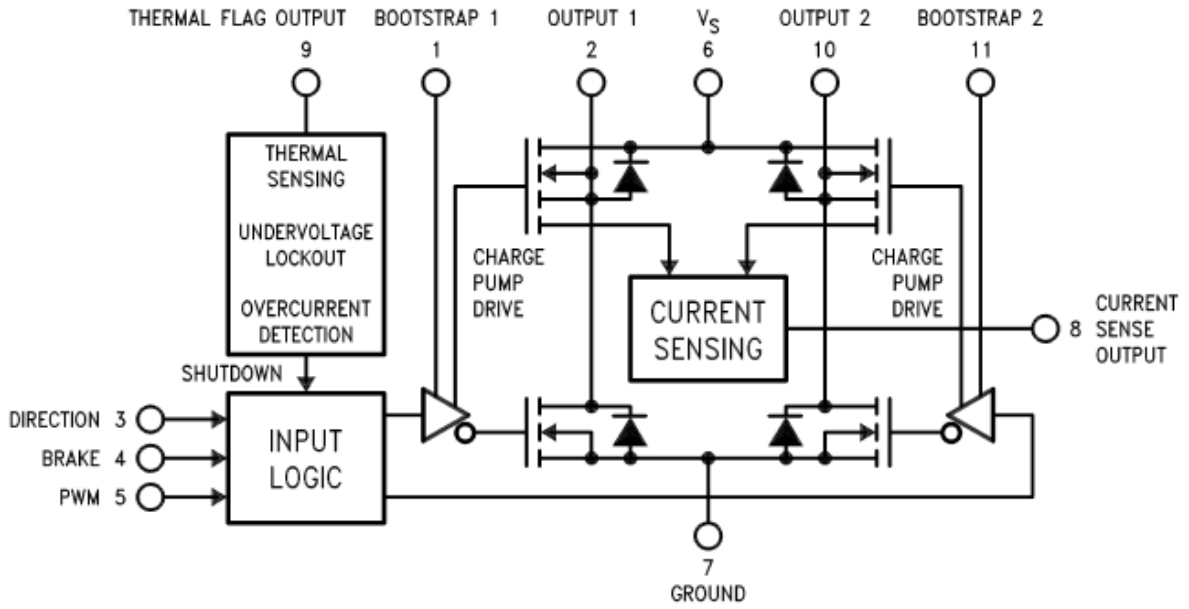
Posición y velocidad de servomecanismos.

Fabricación de automatismo de motores.

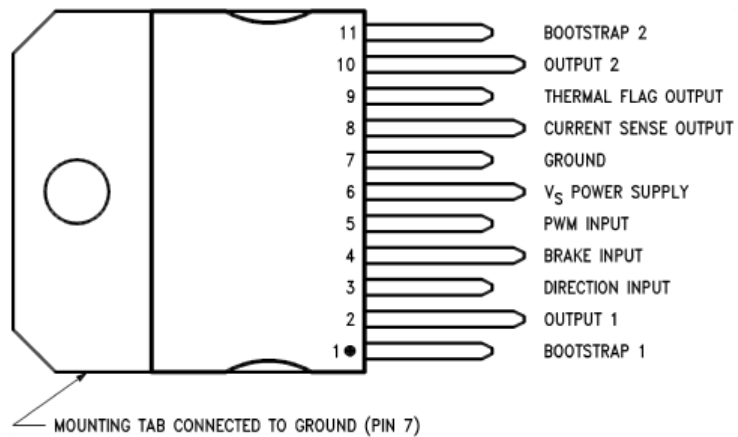
Control de maquinaria numéricamente.

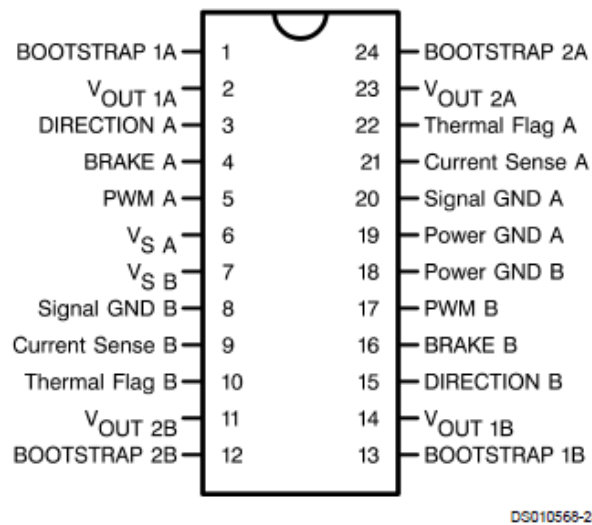
Impresoras de Ordenador y trazadores.

Diagrama Funcional:



Diagramas de Conexión y Ordenamiento de Información:





Rangos Absolutas Máximos (Note 1):

Si requieren dispositivos Militares/Aeroespaciales especificados, por favor se ponen en contacto con la Oficina de venta de Semiconductor Nacional/Distribuidores para la disponibilidad y datos específicos.

El Voltaje de Suministro Total (V_s , Pin 6) 60V.

Voltaje en Pines 3, 4, 5, 8 y 9 12V.

Voltaje en los Pines Bootstraps (Pines 1 y 11) V_{out} +16V.

Corriente de Salida Máxima (200 ms) 6A.

Corriente de Salida Continua (Note 2) 3A.

Disipación de Poder (Note 3) 25W.

Disipación de Poder ($T_A = 25^\circ\text{C}$, Libere el Aire) 3W.

Temperatura de Unión, T_J (máximo) 150°C .

ESD la Sensibilidad (Note 4) 1500V.

Temperatura de Almacenaje, T_{STG} -40°C a $+150^\circ\text{C}$.

Conduce la Temperatura (el Soldar, 10 seg.) 300°C .

Funcionamiento de Posiciones (Note 1):

Temperatura de Unión, T_J -40°C a $+125^\circ\text{C}$...

V_s Voltaje de Suministro +12V a +55V.

Características técnicas:

The following specifications apply for $V_{IS} = 42V$, unless otherwise specified. Boldface limits apply over the entire operating temperature range, $-40^{\circ}C \leq T_J \leq +125^{\circ}C$, all other limits are for $T_A = T_J = 25^{\circ}C$.					
Symbol	Parameter	Conditions	Typ	Limit	Units
$R_{DS(ON)}$	Switch ON Resistance	Output Current = 3A (Note 6)	0.33	0.4/0.6	Ω (max)
$R_{DS(ON)}$	Switch ON Resistance	Output Current = 6A (Note 6)	0.33	0.4/0.6	Ω (max)
V_{CLAMP}	Clamp Diode Forward Drop	Clamp Current = 3A (Note 6)	1.2	1.5	V (max)
V_{IL}	Logic Low Input Voltage	Pins 3, 4, 5		-0.1	V (min)
				0.8	V (max)
I_{IL}	Logic Low Input Current	$V_{IN} = -0.1V$, Pins = 3, 4, 5		-10	μA (max)
V_{IH}	Logic High Input Voltage	Pins 3, 4, 5		2	V (min)
				12	V (max)
I_{IH}	Logic High Input Current	$V_{IN} = 12V$, Pins = 3, 4, 5		10	μA (max)
	Current Sense Output	$I_{OUT} = 1A$ (Note 8)	377	325/300	μA (min)
				425/450	μA (max)
	Current Sense Linearity	$1A \leq I_{OUT} \leq 3A$ (Note 7)	± 6	± 9	%
	Undervoltage Lockout	Outputs turn OFF		9	V (min)
				11	V (max)
T_{JW}	Warning Flag Temperature	Pin 9 $\leq 0.8V$, $I_L = 2 mA$	145		$^{\circ}C$
$V_F(ON)$	Flag Output Saturation Voltage	$T_J = T_{JW}$, $I_L = 2 mA$	0.15		V
$I_F(OFF)$	Flag Output Leakage	$V_F = 12V$	0.2	10	μA (max)
T_{JSD}	Shutdown Temperature	Outputs Turn OFF	170		$^{\circ}C$
I_S	Quiescent Supply Current	All Logic Inputs Low	13	25	mA (max)
t_{Don}	Output Turn-On Delay Time	Sourcing Outputs, $I_{OUT} = 3A$	300		ns
		Sinking Outputs, $I_{OUT} = 3A$	300		ns
t_{on}	Output Turn-On Switching Time	Bootstrap Capacitor = 10 nF			
		Sourcing Outputs, $I_{OUT} = 3A$	100		ns
		Sinking Outputs, $I_{OUT} = 3A$	80		ns
t_{Doff}	Output Turn-Off Delay Times	Sourcing Outputs, $I_{OUT} = 3A$	200		ns
		Sinking Outputs, $I_{OUT} = 3A$	200		ns
t_{off}	Output Turn-Off Switching Times	Bootstrap Capacitor = 10 nF			
		Sourcing Outputs, $I_{OUT} = 3A$	75		ns
		Sinking Outputs, $I_{OUT} = 3A$	70		ns
t_{pw}	Minimum Input Pulse Width	Pins 3, 4 and 5	1		μs
t_{cpr}	Charge Pump Rise Time	No Bootstrap Capacitor	20		μs

Nota 1: Posiciones Absolutas Maximas indican limites mas alla los cuales pueden danar al dispositivo. La corriente continua y la corriente alterna datos especificos electricos no se aplican manejando el dispositivo mas alla de sus condiciones de funcionamiento nominales.

Nota 2: Mirar la Informacion de aplicacion para los detalles en cuanto a la limitacion corriente.

Nota 3: La disipacion de potencia maxima debe ser liberada en temperaturas elevadas y es una funcion de T_J (el maximo), θ_{JA} , y T_A . La disipacion de potencia maxima aceptable en cualquier temperatura es la P_D (el maximo) = $(T_J$ (el maximo) - $T_A)/\theta_{JA}$, o el numero dado en las Posiciones Absolutas, cualquiera es inferior. La resistencia tipica termal de la union al caso (θ_{JC}) es $1.0^{\circ}C/W$ y de la union a ambiental (θ_{JA}) es $30^{\circ}C/W$. Para operacion garantizada T_J (maximo) = $125^{\circ}C$.

Nota 4: Modelo de cuerpo humano, 100pF descargados por una resistencia de 1.5 kws. Excepto los pines Bootstraps (pin 1 y 11) que es protegido a 1000V de ESD.

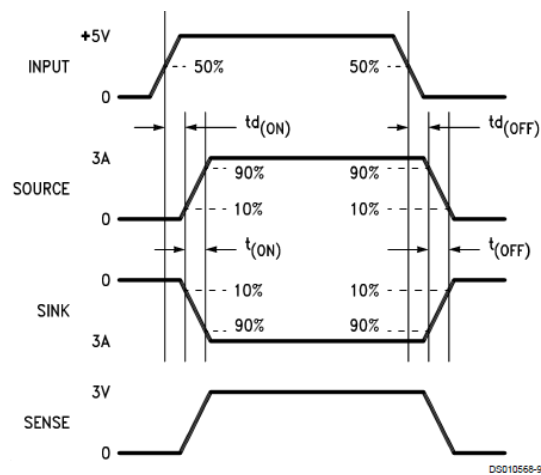
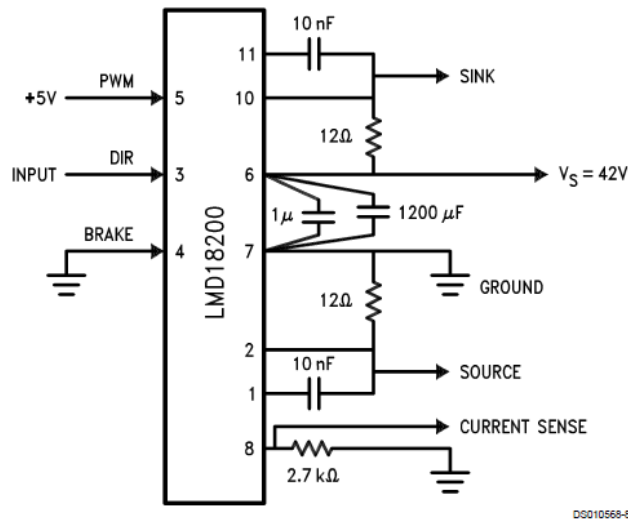
Nota 5: Todos los limites son la produccion del 100 % probada en $25^{\circ}C$. Limites extremos de temperaturas son garantizados via la correlacion que usa SQC aceptado (el Control de calidad Estadstico) metodos. Todos los limites son usados para calcular AOQL, (Hacer un promedio del Nivel de Calidad Saliente).

Nota 6: Las corrientes de salida son pulsadas ($t_W < 2 ms$, Factor de trabajo $< el 5\%$).

Nota 7: La regulacion es calculada en relacion con el valor de salida de sentido corriente con un 1A la carga.

Nota 8: Las selecciones para la tolerancia mas apretada estan disponibles. Fabrica de contacto.

Test del Circuito:



Pin out la Descripción (Mirar el Diagrama de Conexión):

Pin 1, Entrada BootStrap 1: se conecta al BootStrap un condensador para mitad Puentes de h al pin número 1. El condensador recomendado (10 nF) es conectado entre el pin 1 y 2.

Pin 2, Salida 1: Puentes de h de Mitad salida número 1.

Pin 3, Dirección Entrada: Mirar la tabla 1. Esta entrada controla la dirección de flujo corriente entre la salida 1 y la salida 2 (Pines 2 y 10) y, por lo tanto, la dirección de rotación de una carga de motor.

Pin 4, Brake (Freno) Entrada: Mirar la tabla 1. Esta entrada es usada para frenar el motor con eficacia cortando la tensión de sus terminales. Cuando el frenado es deseado, esta entrada es tomada a un nivel alto lógico y es también necesario aplicar la lógica alta a la entrada de PWM, pin 5. Los conductores que cortan el motor es determinado por el nivel lógico en la entrada de dirección (Pin 3): con el Pin 3 de lógica alta, ambos transistores de salida de aprovisionamiento corrientes es ON; con el pin 3 lógica bajo, ambas corrientes que hunde transistores de salida es ON. Todos los transistores de salida pueden ser OFF por aplicando una lógica alta del Pin 4 y una lógica baja de entrada de PWM del Pin 5; en este caso sólo una pequeña corriente de tendencia (aproximadamente-1.5 mA.) existe en cada pin de salida.

Pin 5, PWM la Entrada: Mirar la Mesa 1. Como esta entrada (y la entrada de DIRECCIÓN, Fije 3) es usado es determinado por el formato de la Señal de PWM.

Pin 6, Vs Alimentación.

Pin 7, Conexión a Tierra, este pin es el retorno a tierra, y es internamente conectado a la etiqueta creciente.

Pin 8, Sensor de Corriente de Salida, este pin proviene de la fuente de corriente sensible en la señal de salida, siendo típicamente 377 μ A/A.

Pin 9, Pin de Salida de Temperatura: Este pin proporciona la señal de salida de bandera de advertencia de la temperatura. El Pin 9 se hace activo en bajo en 145°C (la temperatura de unión). Sin embargo el chip no se cerrará hasta que 170°C que sea alcanzado en la unión.

Fije 10, Salida 2: Puente de h de Mitad salida número 2.

Pin 11, Entrada BootStrap 2: se conecta al BootStrap un condensador para mitad Puente de h al pin número 2. El condensador recomendado (10 nF) es conectado entre el pin 10 y 11.

TABLE 1. Logic Truth Table

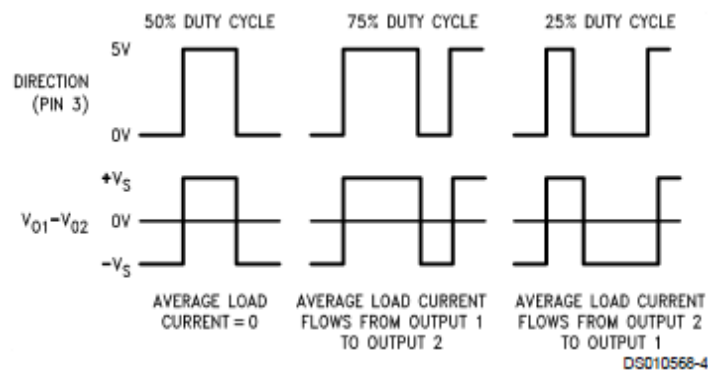
PWM	Dir	Brake	Active Output Drivers
H	H	L	Source 1, Sink 2
H	L	L	Sink 1, Source 2
L	X	L	Source 1, Source 2
H	H	H	Source 1, Source 2
H	L	H	Sink 1, Sink 2
L	X	H	NONE

Información de la aplicación:

Los tipos señal de PWM:

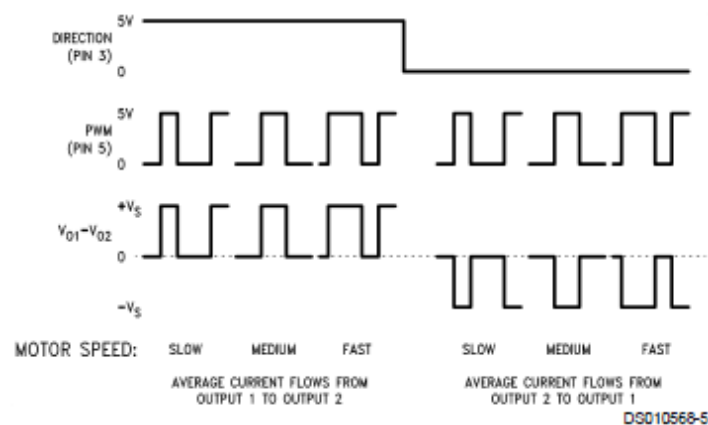
EL LMD18200 fácilmente interfaces con las formas diferentes de señales de PWM. El empleo de la parte con dos de las formas más populares de PWM es descrito en los párrafos siguientes.

La simple anti-fase simple, cerrada PWM consiste en una señal, variable en la cual es codificado tanto dirección como la información de amplitud (mirar la Figura 2). Un factor de trabajo del 50 % PWM la señal representa el paseo cero, desde el valor neto de voltaje (se integró más de un período) entregado a la carga es el cero. Para el LMD18200, la señal de PWM conduce la entrada de dirección (Pin 3) y la entrada de PWM (Pin 5) es atada a la lógica alta.



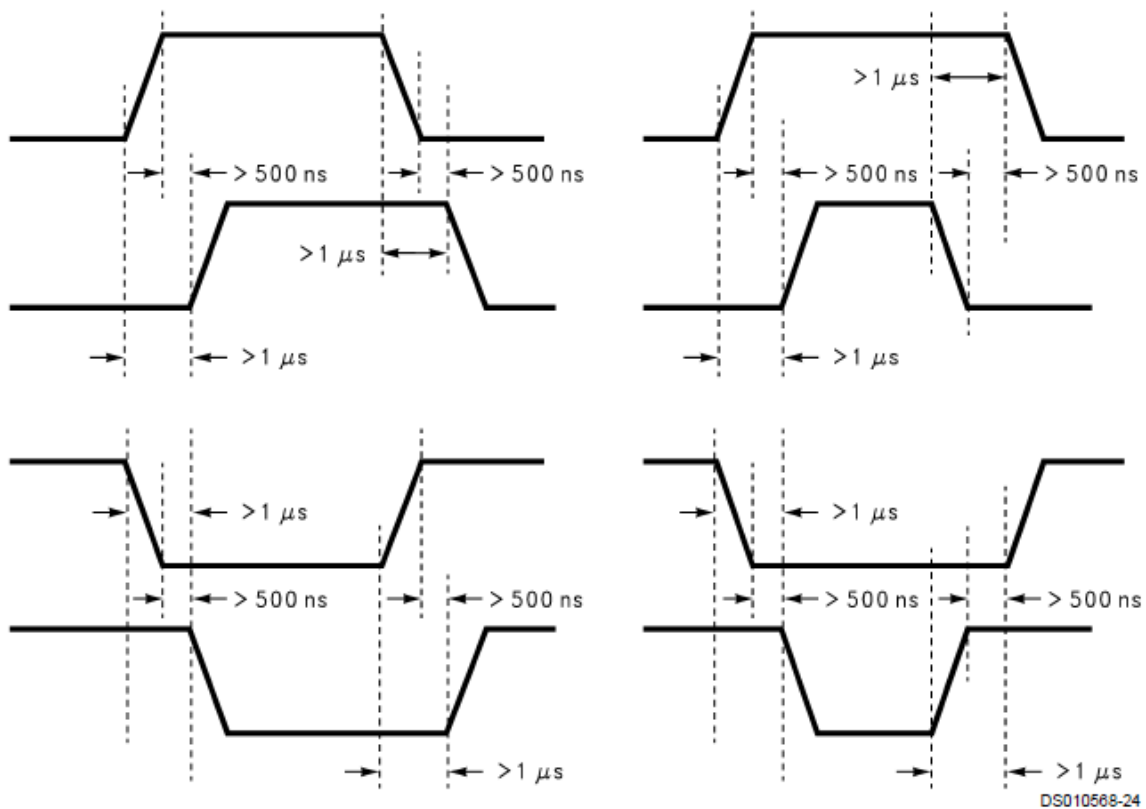
El Signo/magnitud PWM:

Consiste en la separación de señales de dirección (signo) y la amplitud (magnitud) como la figura. La señal de magnitud (absoluta) es el factor de trabajo modulado, y la ausencia de una señal de pulso (una lógica continua de bajo nivel) representa el cero. La corriente entregada a la carga es proporcional para pulsar la anchura. Para el LMD18200, la entrada de dirección (Pin 3) conducen por la señal de signo y la entrada de PWM (Pin 5) conducen por la señal de magnitud.



Exigencias de transición de señal

Para asegurar el funcionamiento apropiado interno lógico, esto es la práctica buena para evitar alinear la caída y los bordes crecientes de señales de entrada. Un retraso de al menos 1 μseg debería ser incorporado entre las transiciones de la Dirección, el Freno, y/o señales de entrada de PWM. Un acercamiento conservador es estar seguro que hay al menos 500ns el retraso entre el final de la primera transición y el principio de la segunda transición. Mirar la Figura 4.



Uso del sensor de salida de corriente:

El sensor de salida de corriente (Pin 8) tiene una sensibilidad de 377 μA por amperio de corriente de salida. Para la exactitud óptima y la linealidad de esta señal, el valor de voltaje que genera la resistencia entre el pin 8 y la tierra debería ser escogido para limitar el voltaje máximo desarrollado en el alfiler 8 a 5V, o menos. El cumplimiento de voltaje máximo es 12V.

Debería ser notado que recirculan corrientes (liberan corrientes wheeling) no son hechos caso por el trazado de circuito de sentido corriente. Por lo tanto, sólo las corrientes en las salidas de aprovisionamiento superiores son sentidas.

Utilización del flag de advertencia de temperatura:

Que la salida de flag de temperatura (fija 9) es un transistor de colector abierto. Esto permite la conexión OR de salidas del flag de advertencia de temperaturas de múltiple LMD18200's, y permite al usuario para poner el nivel alto lógico de la oscilación de señal de salida emparejar exigencias de sistema. Esta salida típicamente conducida la interrupción de entrada de un control de un sistema. El servicio de rutina de interrupción sería diseñada para tomar pasos apropiados, como reducir corrientes de carga o iniciación de una parada de sistema ordenada. El cumplimiento de voltaje máximo sobre el alfiler de bandera es 12V.

Suministros Bypassing

Durante la conmutación de transiciones los niveles de cambios rápidos corrientes experimentado puede causar el voltaje transitorio molesto a través del sistema la inductancia vaga.

Normalmente es necesario evitar el carril de suministro con un condensador/es de alta calidad unido lo más cerca posible al V_s el Suministro de energía (Pin 6) y la tierra (Pin 7). Se recomiendan un condensador cerámico de $1\mu\text{F}$ de alta frecuencia. Se debe tener cuidado el límite para estar por debajo del suministro máximo absoluto del pin del dispositivo por riesgo al transitorio. Manejando el chip en voltajes de suministro encima 40V recomiendan a un supresor de voltaje (transitorio). Típicamente el condensador de cerámica puede ser eliminado en la presencia del supresor de voltaje.

Nota: conduciendo a altas corrientes de carga con mayor cantidad de capacitancia de circulación de suministro (en general al menos $100\mu\text{F}$ por amperio de carga de corriente) se requiere absorber las corrientes de recirculación de las cargas inductivas.

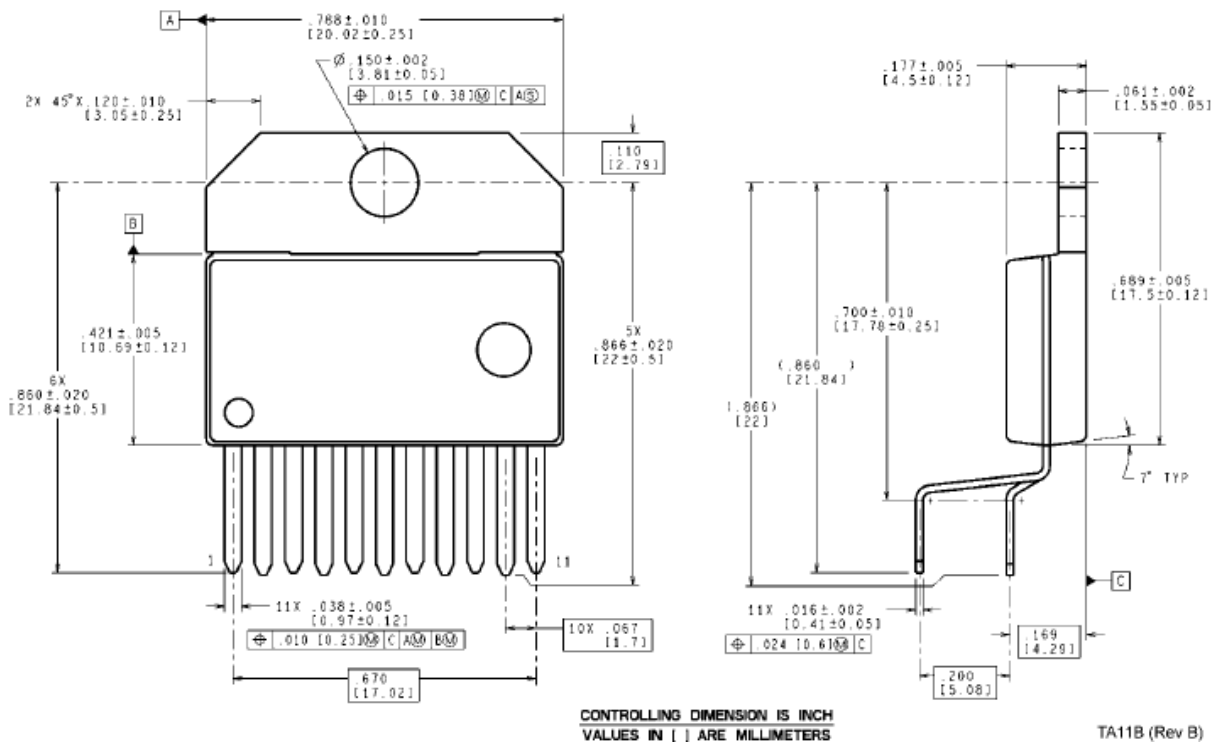
La corriente restrictiva:

El trazado del circuito de protección restrictivo ha sido incorporado en el diseño del LMD18200. Con cualquier dispositivo de potencia es importante considerar los efectos de las corrientes sustanciales por el dispositivo que puede ocurrir como consecuencia de cargas de shorted. El trazado del circuito de protección supervisa este aumento de la corriente (el umbral es puesto a aproximadamente 10A) y cierra el dispositivo de potencia tan rápidamente como posible en caso de una condición de sobrecarga. En el uso típico de conducción de un motor los defectos más comunes son de sobrecarga causados por cuerdas shorted de motor y rotores cerrados. En estas condiciones la inductancia del motor (así como cualquier inductancia de serie en la línea de suministro de V_{cc}) sirve para reducir la magnitud de una corriente a un nivel salvo para el LMD18200.

Una vez que el dispositivo es cerrado, el trazado de circuito de control de vez en cuando tratará de conectar el dispositivo de potencia. Este rango permite a la vuelta inmediata a la operación normal en el caso de que la condición de defecto ha sido quitada. Mientras los restos de defecto, sin embargo, el dispositivo va al ciclo en y de la parada temperatura. Esto puede crear el voltaje transitorio sobre la línea de suministro de Vcc y requieren el suministro apropiado.

La condición más severa para cualquier dispositivo de potencia es una directa, integrada ("el destornillador") el largo plazo corto de una salida para moler. Esta condición puede generar una onda de corriente por el dispositivo de poder sobre la orden de 15A y disiparse hasta 500W de poder para el rato requerido para el trazado de circuito de protección para cerrar el dispositivo de potencia. Esta energía puede ser destructiva, en particular en voltajes más altos de operaciones (> 30V) entonces hay que tener algunas precauciones. El diseño de fuente de calor apropiado es esencial y es normalmente necesario calentar la fuente del pin de suministro de Vcc (Pin 6) con 1 pulgada cuadrada de cobre sobre el PCB.

Dimensiones:



7.9. Anexo Amplificador de Aislamiento HCPL-7200

Características:

15kV/ms de rechazo común a $V_{cm} = 1000V$.

Compacto, Auto-Insertable standard de 8-pin DIP.

4.6 $\mu V/^{\circ}C$ conducción del offset respecto la temperatura.

0.9 mV de entrada de voltaje del offset.

La Amplitud de banda de 85 kHz.

0.1% la No linealidad.

Aprobación Mundial De seguridad: UL 1577 (3750 V rms/1min), VDE 0884 y CSA.

Sigma-delta Avanzado.

Tecnología de convertidor analógico-digital.

Topología de circuito totalmente diferencial.

1 μm CMOS IC tecnología.

Aplicaciones:

Sensor de corriente de fase de un motor.

Sensor de corriente interna.

Supervisión de voltaje de Fuente de potencia de voltaje alto.

Aislamiento de Señal de Suministro de energía de modo pulsante .

Aislamiento de señal analogica.

Aislamiento de Transductor.

Descripción:

El amplificador de aislamiento HCPL-7800 CMR proporciona una combinación única de rasgos inmejorablemente satisfechos para diseñadores de circuito de control de motor.

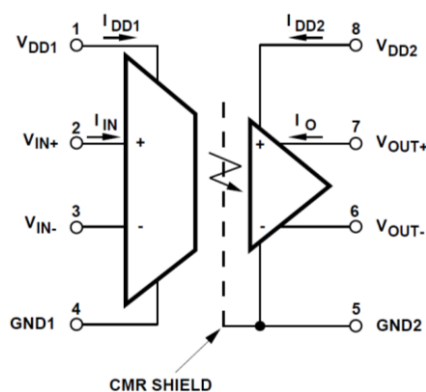
El producto proporciona la precisión y la estabilidad tuvo que con exactitud supervisar la corriente de motor en entornos de control alto ruido de un motor, asegurando el control más liso (menos " la ondulación de momento de rotación ") en varios tipos de usos de control de motor.

Este producto preparado para un camino más pequeño, con más ligereza, más fácil para producir, el alto rechazo ruidoso, es la solución de bajo coste para el sensor de corriente de un motor. El producto también puede ser usado para usos de aislamiento de señal analógica generales que requieren la alta exactitud, la estabilidad y la linealidad en condiciones severas de modo similar ruidosas.

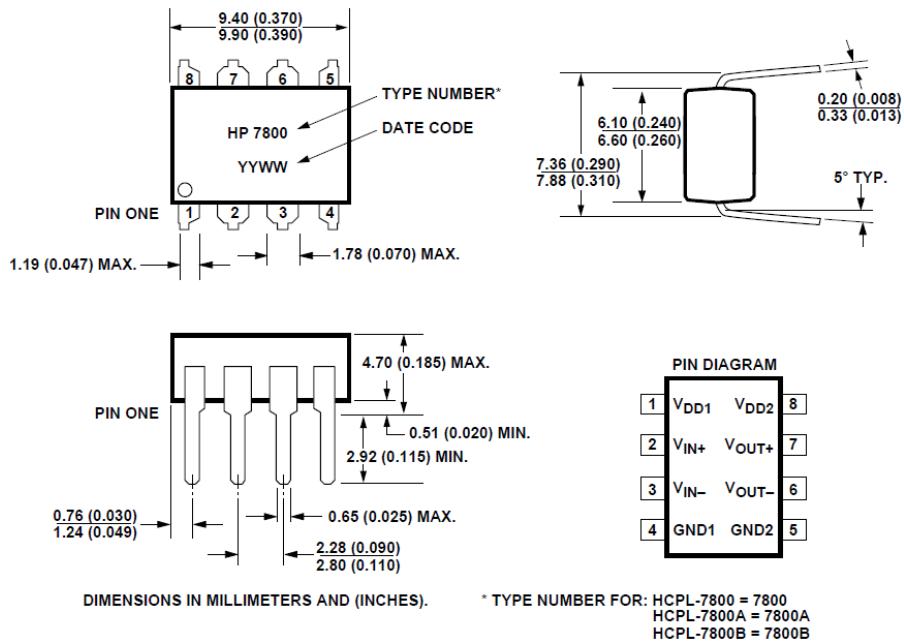
Para usos generales, recomendamos el HCPL-7800 que expone una tolerancia de ganancia de parte-a-parte del $\pm 5\%$. Para usos de precisión, HP ofrecen el HCPL-7800A Y HCPL-7800B, cada uno con las tolerancias de ganancia de parte-a-parte del $\pm 1\%$.

El HCPL-7800 utiliza sigma-delta de la tecnología de convertidor de analógico a digital, estabilizando la interrupción de amplificadores, y una topología de circuito totalmente diferente, fabricada usando HP's $1\mu\text{m}$ de los HP CMOS IC el proceso. La parte también acopla nuestra eficacia en alta velocidad, protegido por ruido que usa nuestro "tubo ligero" patentado opto-acoplador la tecnología de embalaje.

Juntos, estos rasgos entregan el rechazo y aislamiento de ruido, así como la compensación excelente y la exactitud de beneficio y la estabilidad con el tiempo y la temperatura. Este funcionamiento es entregado en un compacto, auto-insertable, el estándar de industria 8-pin de DIP de package que se encuentran las normas por todo el mundo reguladoras de seguridad (la opción de montaje de superficie de ala de gaviota #300 también disponible).



Dimensiones:



Características técnicas:

Insulation and Safety Related Specifications

Parameter	Symbol	Value	Units	Conditions
Min. External Air Gap (External Clearance)	L(IO1)	7.4	mm	Measured from input terminals to output terminals, shortest distance through air
Min. External Tracking Path (External Creepage)	L(IO2)	8.0	mm	Measured from input terminals to output terminals, shortest distance path along body
Min. Internal Plastic Gap (Internal Clearance)		0.5	mm	Through insulation distance, conductor to conductor, usually the direct distance between the photoemitter and photodetector inside the optocoupler cavity
Tracking Resistance (Comparative Tracking Index)	CTI	175	V	DIN IEC 112/VDE 0303 Part 1
Isolation Group		III a		Material Group (DIN VDE 0110, 1/89, Table 1)

Option 300 – surface mount classification is Class A in accordance with CECC 00802.

Description	Symbol	Characteristic	Unit
Installation classification per DIN VDE 0110, Table 1 for rated mains voltage ≤ 300 V rms for rated mains voltage ≤ 600 V rms		I-IV I-III	
Climatic Classification		40/100/21	
Pollution Degree (DIN VDE 0110, Table 1)*		2	
Maximum Working Insulation Voltage	V _{IORM}	848	V _{peak}
Input to Output Test Voltage, Method b** V _{PR} = 1.875 x V _{IORM} , Production test with t _p = 1 sec, Partial discharge < 5 pC	V _{PR}	1591	V _{peak}
Input to Output Test Voltage, Method a** V _{PR} = 1.5 x V _{IORM} , Type and sample test with t _p = 60 sec, Partial discharge < 5 pC	V _{PR}	1273	V _{peak}
Highest Allowable Overvoltage** (Transient Overvoltage t _{TR} = 10 sec)	V _{TR}	6000	V _{peak}
Safety-limiting values (Maximum values allowed in the event of a failure, also see Figure 27) Case Temperature Input Power Output Power	T _S P _{S,Input} P _{S,Output}	175 80 250	°C mW mW
Insulation Resistance at T _S , V _{IO} = 500 V	R _S	≥ 1 x 10 ¹²	Ω

*This part may also be used in Pollution Degree 3 environments where the rated mains voltage is ≤ 300 V rms (per DIN VDE 0110).
**Refer to the front of the optocoupler section of the current catalog for a more detailed description of VDE 0884 and other product safety requirements.

Note: Optocouplers providing safe electrical separation per VDE 0884 do so only within the safety-limiting values to which they are qualified. Protective cut-out switches must be used to ensure that the safety limits are not exceeded.

Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Unit	Note
Storage Temperature	T _S	-55	125	°C	
Ambient Operating Temperature	T _A	-40	100	°C	
Supply Voltages	V _{DD1} , V _{DD2}	0.0	5.5	V	
Steady-State Input Voltage	V _{IN+} , V _{IN-}	-2.0	V _{DD1} + 0.5	V	
Two Second Transient Input Voltage		-6.0			
Output Voltages	V _{OUT+} , V _{OUT-}	-0.5	V _{DD2} + 0.5	V	
Lead Solder Temperature (1.6 mm below seating plane, 10 sec.)	T _{LS}		260	°C	1
Reflow Temperature Profile	See Package Outline Drawings Section				

Recommended Operating Conditions

Parameter	Symbol	Min.	Max.	Unit	Note
Ambient Operating Temperature	T _A	-40	85	°C	2
Supply Voltages	V _{DD1} , V _{DD2}	4.5	5.5	V	3
Input Voltage	V _{IN+} , V _{IN-}	-200	200	mV	4
Output Current	I _O		1	mA	5

DC Electrical Specifications

All specifications and figures are at the nominal operating condition of $V_{IN+} = 0\text{ V}$, $V_{IN-} = 0\text{ V}$, $T_A = 25^\circ\text{C}$, $V_{DD1} = 5.0\text{ V}$, and $V_{DD2} = 5.0\text{ V}$, unless otherwise noted.

Parameter	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	Fig.	Note
Input Offset Voltage	V_{OS}	-1.8	-0.9	0.0	mV		1	
Input Offset Drift vs. Temperature	dV_{OS}/dT		-2.1		$\mu\text{V}/^\circ\text{C}$		1, 2	6
Abs. Value of Input Offset Drift vs. Temperature	$ dV_{OS}/dT $		4.6		$\mu\text{V}/^\circ\text{C}$		1	7
Input Offset Drift vs. V_{DD1}	dV_{OS}/dV_{DD1}		30		$\mu\text{V}/\text{V}$		1, 3	8
Input Offset Drift vs. V_{DD2}	dV_{OS}/dV_{DD2}		-40		$\mu\text{V}/\text{V}$		1, 4	9
Gain ($\pm 5\%$ Tol.)	G	7.61	8.00	8.40		$-200\text{ mV} < V_{IN+} < 200\text{ mV}$	1, 5	10
Gain - A Version ($\pm 1\%$ Tol.)	G_A	7.85	7.93	8.01				
Gain - B Version ($\pm 1\%$ Tol.)	G_B	7.99	8.07	8.15				
Gain Drift vs. Temperature	dG/dT		0.001		$\%/^\circ\text{C}$		5, 6	11
Abs. Value of Gain Drift vs. Temperature	$ dG/dT $		0.001		$\%/^\circ\text{C}$		5	12
Gain Drift vs. V_{DD1}	dG/dV_{DD1}		0.21		$\%/V$		5, 7	13
Gain Drift vs. V_{DD2}	dG/dV_{DD2}		-0.06		$\%/V$		5, 8	14
200 mV Nonlinearity	NL_{200}		0.2	0.35	%		5, 9	15
200 mV Nonlinearity Drift vs. Temperature	dNL_{200}/dT		-0.001		$\% \text{ pts}/^\circ\text{C}$		5, 10	16
200 mV Nonlinearity Drift vs. V_{DD1}	dNL_{200}/dV_{DD1}		-0.005		$\% \text{ pts}/V$		5, 11	17
200 mV Nonlinearity Drift vs. V_{DD2}	dNL_{200}/dV_{DD2}		-0.007		$\% \text{ pts}/V$		5, 12	18
100 mV Nonlinearity	NL_{100}		0.1	0.25	%	$-100\text{ mV} < V_{IN+} < 100\text{ mV}$	5, 13	19
Maximum Input Voltage Before Output Clipping	$ V_{IN+} _{\text{max}}$		300		mV		14	
Average Input Bias Current	I_{IN}		-670		nA		15, 16	20
Input Bias Current Temperature Coefficient	dI_{IN}/dT		3		$\text{nA}/^\circ\text{C}$			
Average Input Resistance	R_{IN}		530		k Ω		15	20
Input Resistance Temperature Coefficient	dR_{IN}/dT		0.38		$\%/^\circ\text{C}$			
Input DC Common-Mode Rejection Ratio	$CMRR_{IN}$		72		dB			21
Output Resistance	R_O		11		Ω			5
Output Resistance Temperature Coefficient	dR_O/dT		0.6		$\%/^\circ\text{C}$			
Output Low Voltage	V_{OL}		1.18		V	$ V_{IN+} = 500\text{ mV}$	14	22
Output High Voltage	V_{OH}		3.61		V	$I_{OUT+} = 0\text{ A}, I_{OUT-} = 0\text{ A}$		
Output Common-Mode Voltage	V_{OCM}	2.20	2.39	2.60	V	$-40^\circ\text{C} < T_A < 85^\circ\text{C}$ $4.5\text{ V} < V_{DD1} < 5.5\text{ V}$	14	
Input Supply Current	I_{DD1}		10.7	15.5	mA		17	23
Output Supply Current	I_{DD2}		11.6	14.5	mA	$V_{IN+} = 200\text{ mV}$, $-40^\circ\text{C} < T_A < 85^\circ\text{C}$ $4.5\text{ V} < V_{DD2} < 5.5\text{ V}$	18	24
Output Short-Circuit Current	$ I_{Osc} $		9.3		mA	$V_{OUT} = 0\text{ V}$ or V_{DD2}		25

AC Electrical Specifications

All specifications and figures are at the nominal operating condition of $V_{IN+} = 0\text{ V}$, $V_{IN-} = 0\text{ V}$, $T_A = 25^\circ\text{C}$, $V_{DD1} = 5.0\text{ V}$, and $V_{DD2} = 5.0\text{ V}$, unless otherwise noted.

Parameter	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	Fig.	Note
Rising Edge Isolation Mode Rejection	IMR_R	10	25		kV/ μs	$V_{IM} = 1\text{ kV}$	19, 20	26
Falling Edge Isolation Mode Rejection	IMR_F	10	15		kV/ μs			
Isolation Mode Rejection Ratio at 60 Hz	IMRR		>140		dB		19	27
Propagation Delay to 10%	t_{PD10}		2.0	3.3	μs	$-40^\circ\text{C} < T_A < 85^\circ\text{C}$	21, 22	
Propagation Delay to 50%	t_{PD50}		3.4	5.6	μs			
Propagation Delay to 90%	t_{PD90}		6.3	9.9	μs			
Rise/Fall Time (10%-90%)	t_{RF}		4.3	6.6	μs			
Bandwidth (-3 dB)	f_{-3dB}	50	85		kHz		23, 24	
Bandwidth (-45°)	f_{-45°		35		kHz			
RMS Input-Referred Noise	V_N		300		$\mu\text{V rms}$	Bandwidth = 100 kHz	25, 26	28
Power Supply Rejection	PSR		5		mV _{p-p}			29

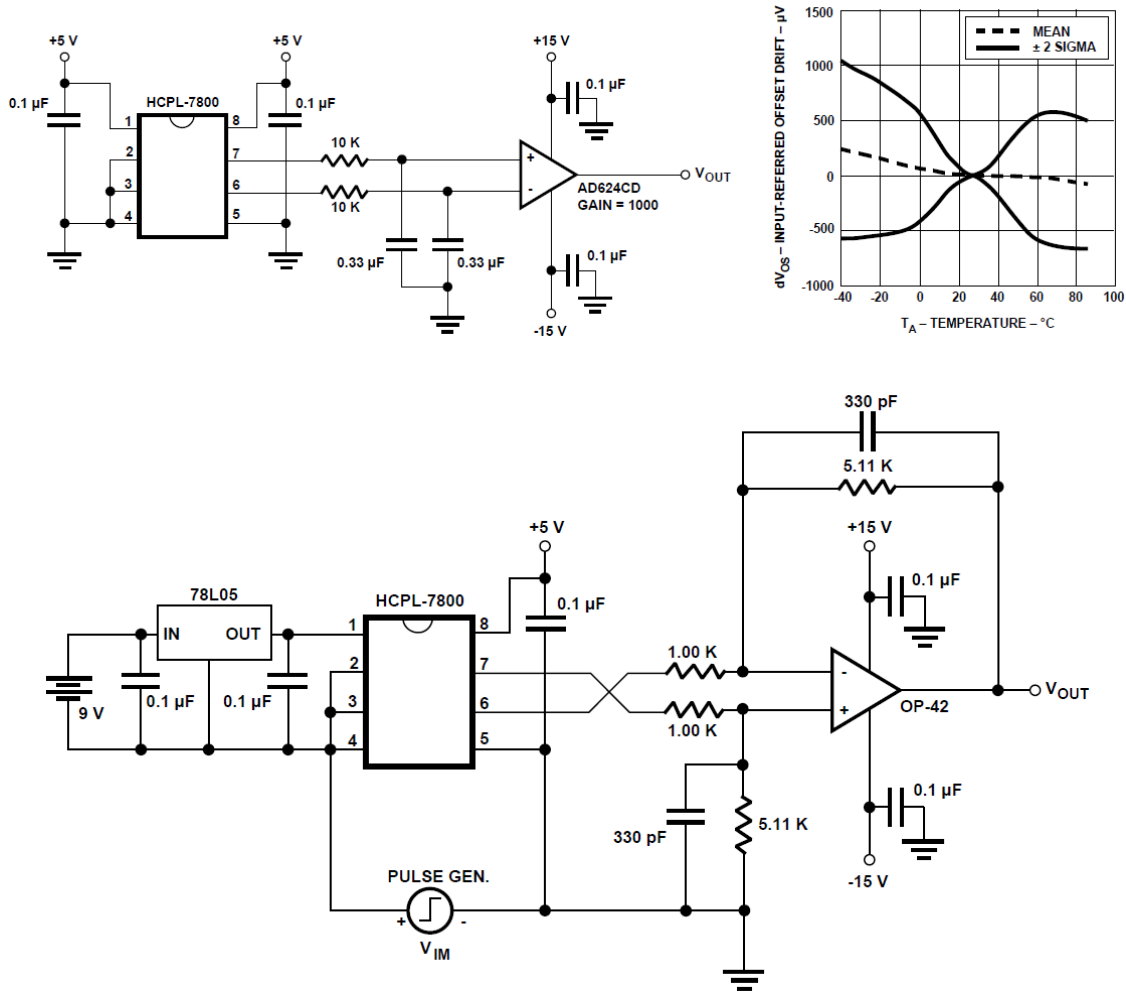
Package Characteristics

All specifications and figures are at the nominal operating condition of $V_{IN+} = 0\text{ V}$, $V_{IN-} = 0\text{ V}$, $T_A = 25^\circ\text{C}$, $V_{DD1} = 5.0\text{ V}$, and $V_{DD2} = 5.0\text{ V}$, unless otherwise noted.

Parameter	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	Fig.	Note
Input-Output Momentary Withstand Voltage*	V_{ISO}	3750			V rms	$t = 1\text{ min.}, RH \leq 50\%$		30, 31
Input-Output Resistance	R_{I-O}	10^{12}	10^{13}		Ω	$T_A = 25^\circ\text{C}$	$V_{I-O} = 500\text{ Vdc}$	30
		10^{11}				$T_A = 100^\circ\text{C}$		
Input-Output Capacitance	C_{I-O}		0.7		pF	$f = 1\text{ MHz}$		30
Input IC Junction-to-Case Thermal Resistance	θ_{JcI}		96		$^\circ\text{C/W}$			32
Output IC Junction-to-Case Thermal Resistance	θ_{JcO}		114		$^\circ\text{C/W}$			

*The Input-Output Momentary Withstand Voltage is a dielectric voltage rating that should not be interpreted as an input-output continuous voltage rating. For the continuous voltage rating refer to the VDE 0884 Insulation Characteristics Table (if applicable), your equipment level safety specification, or HP Application Note 1074, "Optocoupler Input-Output Endurance Voltage."

Circuitos típicos:



7.10. Anexo tarjetas Arduino

Hay multitud de diferentes versiones de placas Arduino, todas ellas se pueden ver en la página oficial de Arduino, www.arduino.cc. La actual placa básica, Leonardo difiere de todas las placas anteriores, en el que el ATmega32u4 se ha incorporado en la comunicación USB, eliminando la necesidad de un procesador secundario, el Duemilanove, usa Atmel ATmega328. La anterior Diecimila, y las primeras unidades de Duemilanove usaban el Atmel ATmega168, mientras que las placas más antiguas usan el ATmega8. El Arduino Mega está basado en el ATmega1280.

A continuación veremos las diferentes versiones de placas Arduino junto con sus características básicas.

Arduino Leonardo



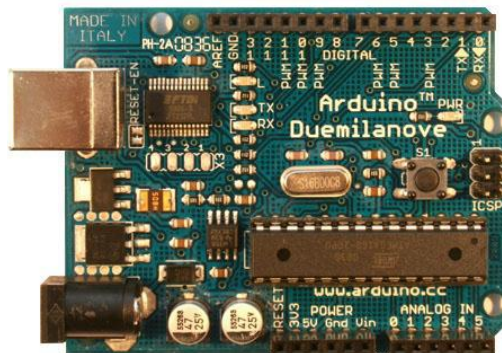
Arduino Leonardo

El Leonardo Arduino es una placa electrónica basada en el ATmega32u4. Tiene 20 entradas / salidas digitales pins del (de los cuales 7 se pueden utilizar como salidas de PWM y 12 como entradas analógicas), un 16 MHz del oscilador de cristal, un micro conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el micro-controlador, basta con conectarlo a un ordenador con un cable USB o el poder con un adaptador de CA a CC o batería para empezar.

El Leonardo difiere de todas las placas anteriores, en el que el ATmega32u4 se ha incorporado en la comunicación USB, eliminando la necesidad de un procesador secundario. Esto permite que el Leonardo a comparecer a un ordenador conectado como un ratón y el teclado, además de una virtual (CDC) de serie / COM puerto. También tiene otras implicaciones para el comportamiento de la junta, las cuales se detallan en la página de instalación inicial .

Microcontroladores	ATmega32u4
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limites)	6-20V
Digital I / O Pins	20
Canales PWM	7
Los canales de entrada analógica	12
Corriente por I DC / O Pin	40 mA
Corriente DC por Pin 3.3V	50 mA
Memoria Flash	32 KB (ATmega32u4), de los cuales 4 KB utilizado por gestor de arranque
SRAM	2,5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Velocidad del reloj	16 MHz

Arduino Duemilanove

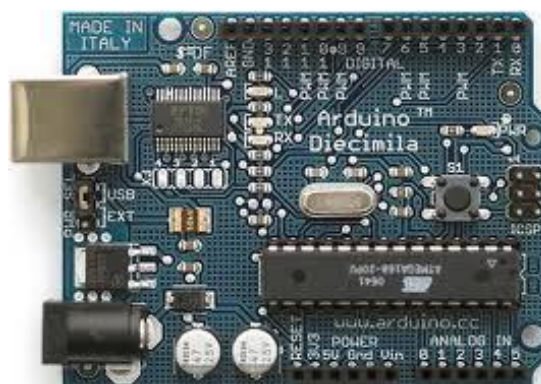


Arduino Duemilanove

El Arduino Duemilanove ("2009") es una placa con micro-controlador basada en el ATmega168 o el ATmega328. Tiene 14 pines con entradas/salidas digitales (6 de las cuales pueden ser usadas como salidas PWM), 6 entradas analógicas, un cristal oscilador a 16Mhz, conexión USB, entrada de alimentación, una cabecera ICSP, y un botón de reset. Contiene todo lo necesario para utilizar el micro-controlador; simplemente conéctalo a tu ordenador a través del cable USB o aliméntalo con un transformador o una batería para empezar a trabajar con él.

Microcontrolador	ATmega368 (ATmega168 en versiones anteriores)
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limite)	6-20V
Pines E/S digitales	14 (6 proporcionan salida PWM)
Pines de entrada analógica	6
Intensidad por pin	40 mA
Intensidad en pin 3.3V	50 mA
Memoria Flash	16 KB (ATmega168) o 32 KB (ATmega328) de las cuales 2 KB las usa el gestor de arranque(bootloader)
SRAM	1 KB (ATmega168) o 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) o 1 KB (ATmega328)
Velocidad de reloj	16 MHz

Arduino Diecimila



Arduino Diecimila

La Arduino Diecimila es una placa micro-controladora basada en el chip ATmega168. Tiene 14 E/S digitales (6 de las cuales se pueden utilizar como salidas PWM), 6 entradas analógicas, un cristal de 16MHz, conexión USB y botón de reseteo. Contiene todo lo necesario para el soporte del micro-controlador; simplemente conectarla a un ordenador con un cable USB o alimentarla con un adaptador AC/DC o una batería y comenzará a funcionar.

Microcontrolador	ATmega168
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12 V
Voltaje de entrada (limites)	6-20 V
Pines E/S Digitales	14 (de ellos 6 son salidas PWM)
Pines de entrada Analógica	6
Intensidad por pin de E/S	40 mA
Intensidad por pin de 3.3V	50 mA
Memoria Flash	16 KB (2 KB reservados para el gestor de arranque)
SRAM	1 KB
EEPROM	512 bytes
Velocidad del reloj	16 MHz

Arduino Nano

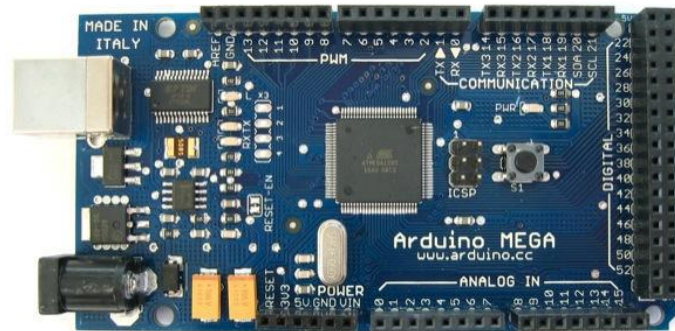


Arduino Nano

El Arduino Nano es una pequeña y completa placa basada en el ATmega328 (Arduino Nano 3.0) o ATmega168 (Arduino Nano 2.x) que se usa conectándola a una protoboard. Tiene más o menos la misma funcionalidad que el Arduino Duemilanove, pero con una presentación diferente. No posee conector para alimentación externa, y funciona con un cable USB Mini-B en vez del cable estándar. El nano fue diseñado y está siendo producido por Gravitech.

Microcontrolador	Atmel ATmega168 o ATmega328
Tensión de Operación (nivel lógico)	5 V
Tensión de Entrada (recomendado)	7-12 V
Tensión de Entrada (límites)	6-20 V
Pines E/S Digitales	14 (de los cuales 6 proveen de salida PWM)
Entradas Analógicas	8
Corriente máx por cada PIN de E/S	40 mA
Memoria Flash	16 KB (ATmega168) o 32 KB (ATmega328) de los cuales 2KB son usados por el bootloader
SRAM	1 KB (ATmega168) o 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) o 1 KB (ATmega328)
Frecuencia de reloj	16 MHz
Dimensiones	18,5mm x 43.2mm

Arduino Mega



Arduino Mega

Es una placa micro-controlador basada ATmega1280. Tiene 54 E/S digitales (14 proporcionan salida PWM), 16 entradas digitales, 4 UARTS (puertos serie por hardware), un cristal oscilador de 16MHz, conexión USB, entrada de corriente, conector ICSP y botón de reset. Contiene todo lo necesario para hacer funcionar el micro-controlador; simplemente conéctalo al ordenador con el cable USB o aliméntalo con un transformador o batería para empezar. El Mega es compatible con la mayoría de shields diseñados para el Arduino Duemilanove o Diecimila.

Microcontrolador	ATmega1280
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limite)	6-20V
Pines E/S digitales	54 (14 proporcionan salida PWM)
Pines de entrada analógica	16
Intensidad por pin	40 mA
Intensidad en pin 3.3V	50 mA
Memoria Flash	128 KB de las cuales 4 KB las usa el gestor de arranque(bootloader)
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

Arduino Bluetooth



Arduino Bluetooth

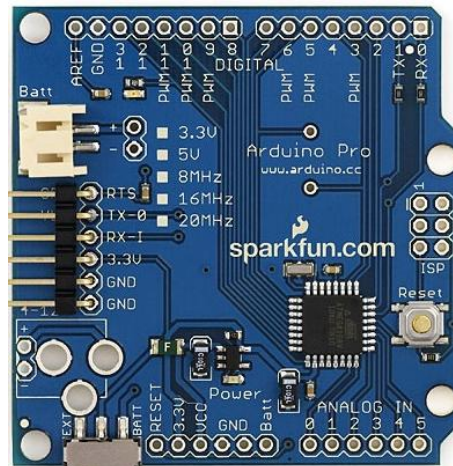
El Arduino BT es una placa Arduino con el módulo Bluetooth incorporado, que permite la comunicación inalámbrica. El módulo Bluetooth utilizado es el Bluegiga WT11, la versión iWrap.

El módulo Bluetooth se puede configurar con comandos enviados a través del puerto serie del ATmega168 (consulta la guía del usuario iWrap para más detalles). Un programa para configurar el nombre y código del módulo bluetooth se ejecuta una vez en cada BT Arduino. El nombre se establece en ARDUINOBT y el código de acceso en 12345.

El ATmega168 viene precargado con un gestor de arranque que te permite subir *sketches* al consejo de administración a través de bluetooth. El código fuente del gestor de arranque está disponible en el repositorio SVN de Arduino.

Microcontroladores	ATmega328
Tensión de funcionamiento	5V
Voltaje de entrada	1.2-5.5 V
Digital I / O Pins	14 (de los cuales 6 proporcionan salida PWM)
Pines de entrada analógica	6
Corriente por I DC / O Pin	40 mA
Corriente DC por Pin 3.3V	50 mA
Memoria Flash	32 KB (de los cuales 2 KB utilizado por gestor de arranque)
SRAM	2 KB
EEPROM	1 KB
Velocidad del reloj	16 MHz

Arduino Pro



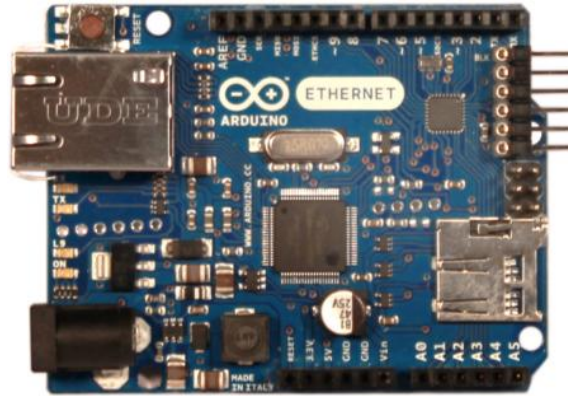
Arduino Pro

El Pro Arduino es una placa electrónica basada en el ATmega168 o ATmega328. El Pro viene en 3,3 V / 8 MHz y 5 V / 16 MHz versiones. Cuenta con 14 entradas / salidas digitales pines (de las cuales 6 se puede utilizar como salidas PWM), 6 entradas analógicas, un conector de alimentación de la batería, un interruptor de encendido, un botón de reinicio, y los agujeros para el montaje de un conector de alimentación, una cabecera ICSP, y pin headers. Un encabezado de seis pines se puede conectar a un cable FTDI o Sparkfun ruptura bordo para proporcionar alimentación USB y la comunicación a la junta.

El Arduino Pro está diseñado para instalación semi-permanente en los objetos o exposiciones. La tarjeta viene sin cabeceras pre-montados, lo que permite el uso de varios tipos de conectores o soldar directamente de los cables. La distribución de los pines es compatible con Arduino escudos. Las versiones de 3,3 V del Pro puede ser alimentado con una batería.

Microcontroladores	ATmega168 o ATmega328
Tensión de funcionamiento	3,3 V o 5 V
Voltaje de entrada	3,35 - 12 V (3,3 V versiones) o 5 - 12 V (versiones 5V)
Digital I / O Pins	14 (de los cuales 6 proporcionan salida PWM)
Pines de entrada analógica	6
Corriente por I DC / O Pin	40 mA
Memoria Flash	16 KB (ATmega168) o 32 KB (ATmega328) de los cuales 2 KB utilizado por gestor de arranque
SRAM	1 KB (ATmega168) o 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) o 1 KB (ATmega328)
Velocidad del reloj	8 MHz (3,3V versiones) ó 16 MHz (5V versiones)

Arduino Ethernet



Arduino Ethernet

El Arduino Ethernet es una placa electrónica basada en el ATmega328 (ficha técnica). Cuenta con 14 entradas / salidas digitales pins, 6 entradas analógicas, un joven de 16 MHz oscilador de cristal, un RJ45 conexión, un conector de alimentación, una cabecera ICSP, y un botón de reset.

NB: Los pasadores 10, 11, 12 y 13 están reservados para la interfaz con el módulo Ethernet y no debe ser utilizado de otra manera. Esto reduce el número de pines disponibles a 9, con 4 disponibles como salidas PWM.

Un Poder sobre Ethernet módulo opcional se puede agregar a la tarjeta también.

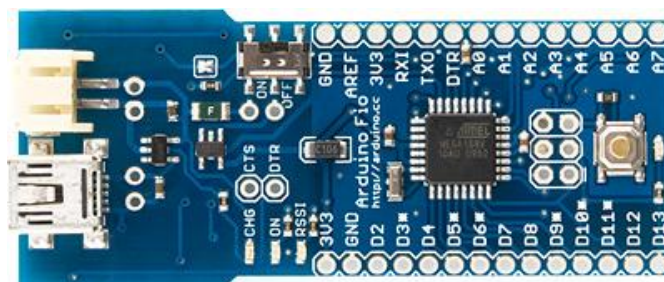
La Ethernet se diferencia de otras placas ya que no tiene a bordo USB-to-serial chip controlador, pero tiene una interfaz Ethernet Wiznet. Esta es la misma interfaz se encuentra en el escudo Ethernet.

Una bordo lector de tarjetas microSD, que puede ser usado para almacenar archivos para servir a través de la red, se puede acceder a través de la Biblioteca SD. Pin 10 se reserva para la interfaz Wiznet, SS para la tarjeta SD está en el pin 4.

El encabezado de 6-pin de programación en serie es compatible con el USB Serial adaptador y también con los cables FTDI USB o con Sparkfun y Adafruit FTDI de estilo básicas USB a serial juntas para grupos. Cuenta con soporte para rearme automático, lo que permite bocetos para cargar sin necesidad de pulsar el botón de reinicio en el tablero. Cuando se conecta a un adaptador de USB a Serial, Ethernet Arduino es alimentado desde el adaptador.

Microcontroladores	ATmega328
Tensión de funcionamiento	5V
Plug de entrada de voltaje (recomendado)	7-12V
Plug de entrada de tensión (límite)	6-20V
Voltaje de entrada PoE (límites)	36-57V
Digital I / O Pins	14 (de los cuales 4 proporcionar una salida PWM)
<i>Clavijas Arduino reservados:</i>	
	<i>10 a 13 usado para SPI</i>
	<i>4 utilizado para tarjeta SD</i>
	<i>2 W5100 interrupción (cuando puenteado)</i>
Pines de entrada analógica	6
Corriente por I DC / O Pin	40 mA
Corriente DC por Pin 3.3V	50 mA
Memoria Flash	32 KB (ATmega328) de los cuales 0,5 KB utilizado por gestor de arranque
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad del reloj	16 MHz
W5100 TCP / IP Embedded Ethernet Controller	
Power Over Ethernet listo magnética Jack	
Micro SD tarjeta, con adaptadores activos	

Arduino Fio



Arduino Fio.

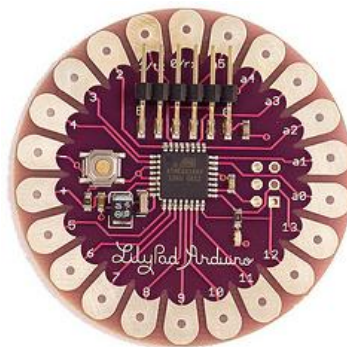
El Arduino Fio es una placa electrónica basada en el ATmega328P (hoja de datos) funciona a 3.3V y 8 MHz. Cuenta con 14 entradas / salidas digitales pines (de las cuales 6 se puede utilizar como salidas PWM), 8 entradas analógicas, un resonador de a bordo, un botón de reinicio, y los agujeros para los conectores macho de montaje. Dispone de conexiones para una batería de polímero de litio e incluye un circuito de carga a través de USB. Un XBee socket está disponible en la parte inferior del tablero.

El Arduino Fio está diseñado para aplicaciones inalámbricas. El usuario puede subir sketches con un cable FTDI o un tablero Sparkfun ruptura. Además, mediante el uso de una versión modificada de USB a XBee adaptador como XBee Explorador de USB, el usuario puede subir bocetos inalámbrica. La tarjeta viene sin cabeceras pre-montados, lo que permite el uso de varios tipos de conectores o soldar directamente de los cables.

El Arduino Fio fue diseñado por Shigeru Kobayashi y SparkFun Electronics, y fabricado por SparkFun Electronics.

Microcontroladores	ATmega328P
Tensión de funcionamiento	3.3V
Voltaje de entrada	3.35 -12 V
Tensión de entrada para carga	3,7 a 7 V
Digital I / O Pins	14 (de los cuales 6 proporcionan salida PWM)
Pines de entrada analógica	8
Corriente por I DC / O Pin	40 mA
Memoria Flash	32 KB (de los cuales 2 KB utilizado por gestor de arranque)
SRAM	2 KB
EEPROM	1 KB
Velocidad del reloj	8 MHz

Arduino LilyPad



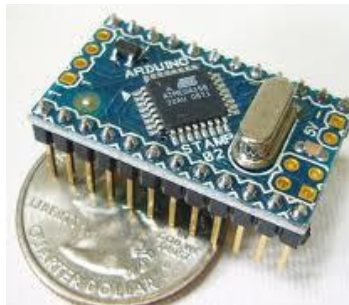
Arduino LilyPad

El LilyPad Arduino es una placa electrónica diseñado para llevar encima y e-textiles. Se puede coser a la tela y montado de manera similar fuentes de alimentación, sensores y actuadores con hilo conductor. El consejo se basa en la ATMEGA168V (la versión de bajo consumo del ATmega168) ([ficha técnica](#)) o el ATmega328V ([ficha técnica](#)). El LilyPad Arduino fue diseñado y desarrollado por Leah Buechley y SparkFun Electronics.

Advertencia: No encienda el LilyPad Arduino con más de 5,5 voltios, o conectar la alimentación al revés: lo vas a matar.

Microcontroladores	ATMEGA168V o ATmega328V
Tensión de funcionamiento	2.7-5.5 V
Voltaje de entrada	2.7-5.5 V
Digital I / O Pins	14 (de los cuales 6 proporcionan salida PWM)
Pines de entrada analógica	6
Corriente por I DC / O Pin	40 mA
Memoria Flash	16 KB (de los cuales 2 KB utilizado por gestor de arranque)
SRAM	1 KB
EEPROM	512 bytes
Velocidad del reloj	8 MHz

Arduino Mini



Arduino Mini

Arduino Mini es una placa con un pequeño micro-controlador basada en el ATmega168, pensada para ser usada en placas de prototipado y donde el espacio es un bien escaso. Puede ser programada con el adaptador Mini USB u otros adaptadores USB o RS232 a TTL serial.

Advertencia: No Alimente el Arduino mini con más de 9 voltios, o conecte la alimentación al revés.

Microcontrolador	ATmega168
Voltaje de funcionamiento	5V
Voltaje de entrada	7-9 V
Pines E/S digital	14 (de las cuales 6 pueden ser usadas como salidas PWM)
Pines entrada analógica	8 (de las cuales 4 se extienden en pines)
DC Corriente continua por pin E/S	40 mA
Memoria Flash	16 KB (de las cuales 2 KB son usadas por el bootloader)
SRAM	1 KB
EEPROM	512 bytes
Velocidad de reloj	16 MHz

Los Shields de Arduino

Los Shields son placas que se colocan encima de la placa Arduino y que amplían una nueva función para que sea controlada desde Arduino, para controlar diferentes aparatos, adquirir datos, etc.

XBee Shield

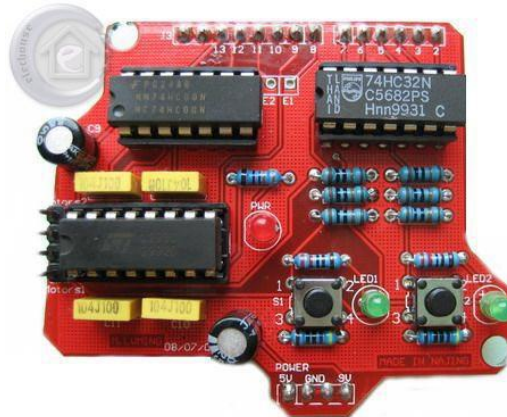


XBee shield

La XBee shield permite a una placa Arduino comunicarse de forma inalámbrica usando Zigbee. Está basada en el módulo XBee de MaxStream. El módulo puede comunicarse hasta 100ft (30 metros) en interior o 300ft (90 metros) al aire libre (en visión directa). Puede ser usado como reemplazo del puerto serie/USB o puedes ponerlo en modo de comandos y configurarlo para una variedad de opciones de redes broadcast o malladas.

La shield tiene pistas desde cada pin del XBee hasta un orificio de soldar. También provee conectores hembra para usar los pines digitales desde 2 hasta 7 y las entradas analógicas, las cuales están cubiertas por la shield (los pines digitales de 8 a 13 no están cubiertos por la placa, así que puedes usar los conectores de la placa directamente).

Motor Control v1.1



Motor shield

Este shield permite a Arduino controlar motores eléctricos de corriente continua, servos y motores paso a paso y leer encoders. Esta shield está todavía bajo testeó, pero puedes echarle un ojo al esquema, y usarlo en tus diseños.

Arduino Ethernet Shield



Ethernet shield

La Arduino Ethernet Shield permite a una placa Arduino conectarse a internet. Está basada en el chip Ethernet Wiznet W5100. El Wiznet W5100 provee de una pila de red IP capaz de TCP y UDP. Soporta hasta cuatro conexiones de sockets simultáneas. Usa la librería Ethernet para escribir programas que se conecten a internet usando la shield.

El actual diseño de la Ethernet shield no es compatible con el Arduino Mega.

La Ethernet shield dispone de unos conectores que permiten conectar a su vez otras placas encima y apilarlas sobre la placa Arduino.

Arduino usa los pines digitales 10, 11, 12, y 13 (SPI) para comunicarse con el W5100 en la Ethernet shield. Estos pines no pueden ser usados para e/s genéricas.

La shield provee un conector Ethernet estándar RJ45

El botón de reset en la shield resetea ambos, el W5100 y la placa Arduino.

La shield contiene un número de LEDs para información:

PWR: indica que la placa y la shield están alimentadas

LINK: indica la presencia de red y parpadea cuando la shield envía o recibe datos

FULLD: indica que la conexión de red es full duplex

100M: indica la presencia de una conexión de red de 100 Mb/s (de forma opuesta a una de 10Mb/s)

RX: parpadea cuando la shield recibe datos

TX: parpadea cuando la shield envía datos

COLL: parpadea cuando se detectan colisiones en la red

El jumper soldado marcado como "INT" puede ser conectado para permitir a la placa Arduino recibir notificaciones de eventos por interrupción desde el W5100, pero esto no está soportado por la librería Ethernet. El jumper conecta el pin INT del W5100 al pin digital 2 de Arduino. Y el slot SD en la shield no está soportado por el software Arduino.

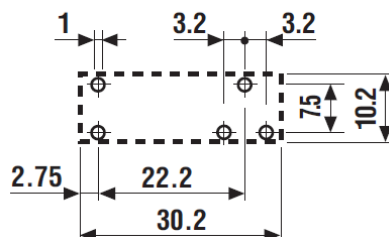
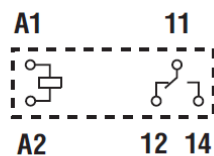
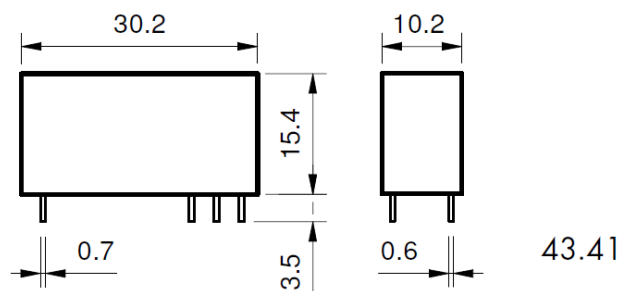
7.11. Anexo Relés 5V

Datos técnicos

Relé de protección RT II

Tipo	43.41.7.005.0000
Tensión nominal	5 V / DC
Tipo de contacto	1 contacto conmutado
Rango de voltaje DC	3,5 a 11,4 V
Vida mecánica	(AC / DC) 10×10^6 conmutaciones
R _{Coil}	110 Ω
Corriente nominal	10 A
Aire / fuga distancia	(Entre bobina y contactos) 10 mm
Vida eléctrica	(AC1) 100×10^3 Schaltspiele
Cumple la directiva RoHS	sí
Dimensiones	(L x W x H) 30,2 x 10,2 x 15,4 mm
Material de los contactos	AgNi
Especificación	DC-bobina
Corriente de conmutación	10 A
Tensión de conmutación	250 V / AC (AC1) / 30 V / DC

Dimensiones:



Anexo B: Programas de los ejercicios con Arduino

7.12. Mando del motor

```
int izquierda =5;

int derecha =6;

int inicio =7;

int sensor =8;

int verde =9;

int blanca =10;

int manuauto = 4;

void setup ()
{
  Serial.begin(9600);

  pinMode (blanca, OUTPUT);

  pinMode (verde, OUTPUT);

  pinMode (izquierda, OUTPUT);

  pinMode (derecha, OUTPUT);
}
```

```
void loop ()
{
    // Configuración del funcionamiento:
    int estadosentido = digitalRead(manuauto);
    int estadomarcha = digitalRead(inicio);
    if ( estadomarcha == HIGH){
        if ( estadosentido == LOW){
            avance();
        } else {
            retroceso();
        }
    } else {
        paro();
    }
}

// Funcionaes:

void avance () {

    digitalWrite(verde , HIGH);
    digitalWrite(derecha , HIGH);
    digitalWrite(blanca , LOW);
    digitalWrite(izquierda , LOW);
}
```



```
void retroceso () {  
  
    digitalWrite(verde , LOW);  
    digitalWrite(derecha , LOW);  
    digitalWrite(blanca , HIGH);  
    digitalWrite(izquierda , HIGH);  
}
```

```
void paro () {  
  
    digitalWrite(verde , LOW);  
    digitalWrite(derecha , LOW);  
    digitalWrite(blanca , LOW);  
    digitalWrite(izquierda , LOW);  
}
```

7.13. Uso del paro de emergencia

```
int izquierda =5;  
int derecha =6;  
int inicio =7;  
int sensor =8;  
int salida =0;  
int marcha =0;
```

```
int estadamarcha;

int previomarcha =0;

int pieza =0;

int estadosensor ;

int previosensor =0;

long time =0;

long espera =20; //tiempo que debe estar activado el sensor

void setup ()

{

  Serial.begin(9600);

  digitalWrite (sensor, HIGH); //resistencia pull-up

  pinMode (izquierda, OUTPUT);

  pinMode (derecha, OUTPUT);

}

void loop ()

{

  //Configuración del sensor para no tener rebotes:

  int optico;

  estadosensor=digitalRead(sensor);

  if (estadosensor!=previosensor){ //al cambiar de estado LOW => HIGH

    time=millis(); //cuenta el tiempo en que está HIGH

  }

  if ((millis()-time)>espera){ // si es superior a 20ms
```

```

    optico=estadosensor;        // Es decir pasa HIGH
    if (optico==HIGH){          //Si está activado hay pieza y sino no.
        pieza=HIGH;
    }else{
        pieza=LOW;
    }
}

previosensor = estadosensor;

//Configuración del pulsador como interruptor sin interrupciones:
int value;
estadomarcha=digitalRead(inicio);
if (estadomarcha!=previomarcha){ //al cambiar de estado LOW => HIGH
    value=estadomarcha;        // Es decir pasa HIGH
    if (value==HIGH && pieza==LOW){ //Si está activado hay pieza y sino no
        marcha=HIGH;
    }
}

previomarcha = estadomarcha;

// Configuración del funcionamiento:
if (marcha==HIGH){
    avance();
}
}

```

```
// Funciones:  
  
void avance () {  
    digitalWrite(derecha , HIGH);  
    digitalWrite(izquierda , LOW);  
}
```

7.14. Paro por el sensor óptico

```
int izquierda =5;  
int derecha =6;  
int inicio =7;  
int sensor =8;  
  
int salida =0;  
int marcha =0;  
int estadomarcha;  
int previomarcha =0;  
  
int pieza =0;  
int estadosensor ;  
int previosensor =0;  
long time =0;  
long espera =20; //tiempo que debe estar activado el sensor
```

```
int E0;

int E1;

int E2;

void setup()
{
    Serial.begin(9600);
    pinMode (izquierda, OUTPUT);
    pinMode (derecha, OUTPUT);
}

void loop()
{
    //Configuración del sensor para no tener rebotes:
    int optico;
    estadosensor=digitalRead(sensor);
    if (estadosensor!=previosensor){    //al cambiar de estado LOW => HIGH
        time=millis();                //cuenta el tiempo en que está HIGH
    }
    if ((millis()-time)>espera){    // si es superior a 20ms
        optico=estadosensor;        // Es decir pasa HIGH
        if (optico==HIGH){          //Si está activado hay pieza y sino no.
            pieza=HIGH;
            E2=1;
        }else{
```

```
    pieza=LOW;
}
}
previosensor = estadosensor;

//Configuración del pulsador como interruptor sin interrupciones:
int value;
estadomarcha=digitalRead(inicio);
if (estadomarcha!=previomarcha){ //al cambiar de estado LOW => HIGH
    value=estadomarcha; // Es decir pasa HIGH
    if (value==HIGH){ //Si está activado hay pieza y sino no.
        E2=0;
        marcha=HIGH;
    }
}
previomarcha = estadomarcha;

// Configuración del funcionamiento:
if (E2==0 && E1==0){
    E0=1;
}
if (E0==1 && marcha==HIGH && pieza==LOW){
    E1=1; //Si está E0=1 && se pulsa marcha && no hay pieza
    E0=0; //activa E1 y desactiva E0
}
```

```
if (E1==1){
    avance();
}
if (E1==1 && pieza==HIGH){
    E2=1;           //Si está E1=1 && hay pieza
    E1=0;           //activa E2 y desactiva E1
    paro();
}
if (E2==1){           // El temporizador ha llegado a 5000ms paro
    paro();
}
}

// Funciones:
void avance () {
    digitalWrite(derecha , HIGH);
    digitalWrite(izquierda , LOW);
}

void paro () {
    digitalWrite(derecha , LOW);
    digitalWrite(izquierda , LOW);
}
```

7.15. Temporizadores

```
int izquierda =5;

int derecha =6;

int inicio =7;

int sensor =8;

int verde =9;

int blanca =10;

int salida =0;

int marcha =0;

int estadomarcha;

int previomarcha =0;

int pieza =0;

int estadosensor ;

int previosensor =0;

long time =0;

long espera =20; //tiempo que debe estar activado el sensor

long temporizador =0;

long intervalo =5000;

int E0;

int E1;

int E2;

int E3;
```



```
void setup ()
{
  Serial.begin(9600);
  digitalWrite (sensor, HIGH); //resistencia pull-up
  pinMode (blanca, OUTPUT);
  pinMode (verde, OUTPUT);
  pinMode (izquierda, OUTPUT);
  pinMode (derecha, OUTPUT);
}

void loop ()
{
  //Configuración del sensor para no tener rebotes:
  int optico;
  estadosensor=digitalRead(sensor);
  if (estadosensor!=previosensor){ //al cambiar de estado LOW => HIGH
    time=millis(); //cuenta el tiempo en que está HIGH
  }
  if ((millis()-time)>espera){ // si es superior a 20ms
    optico=estadosensor; // Es decir pasa HIGH
    if (optico==HIGH){ //Si está activado hay pieza y sino no.
      pieza=HIGH;
    }else{
      pieza=LOW;
    } }
  previosensor = estadosensor;
```

```

//Configuración del pulsador como interruptor sin interrupciones:

int value;

estadomarcha=digitalRead(inicio);

if (estadomarcha!=previomarcha){ //al cambiar de estado LOW => HIGH

    value=estadomarcha; // Es decir pasa HIGH

    if (value==HIGH){ //Si está activado hay pieza y sino no.

        E3=0; //Reinicia el programa una vez que se ha parado

        marcha=HIGH;

    }

}

previomarcha = estadomarcha;

// Configuración del funcionamiento:

if (E2==0 && E1==0 && E3==0){

    E0=1;

}

if (E0==1 && marcha==HIGH && pieza==LOW){

    E1=1; //Si está E0=1 && se pulsa marcha && no hay pieza

    E0=0; //activa E1 y desactiva E0

}

if (E1==1){

    avance();

}

if (E1==1 && pieza==HIGH){

    E2=1; //Si está E1=1 && hay pieza

```

```

    E1=0;                //activa E2 y desactiva E1
    temporizador=millis();    //comienza el temporizador
    Serial.println(temporizador);
}

if (E2==1){            // Si está activada la E2 retroceso
    retroceso();
}

if (E2==1 && (millis()-temporizador)>intervalo){
    E3=1;                // Si está E2=1 y se cumple el temporizador
    E2=0;                // desactiva E2
    Serial.println(millis());
}

if (E3==1){            // El temporizador ha llegado a 5000ms paro
    paro();
}
}

// Funciones:
void avance () {
    digitalWrite(verde , HIGH);
    digitalWrite(derecha , HIGH);
    digitalWrite(blanca , LOW);
    digitalWrite(izquierda , LOW);
}

```

```
void retroceso () {  
    digitalWrite(verde , LOW);  
    digitalWrite(derecha , LOW);  
    digitalWrite(blanca , HIGH);  
    digitalWrite(izquierda , HIGH);  
}
```

```
void paro () {  
    digitalWrite(verde , LOW);  
    digitalWrite(derecha , LOW);  
    digitalWrite(blanca , LOW);  
    digitalWrite(izquierda , LOW);  
}
```

7.16. Uso del encoder magnético

```
int encoder =4;  
int izquierda =5;  
int derecha =6;  
int inicio =7;  
int sensor =8;  
int verde =9;  
int blanca =10;
```

```
int salida =0;

int marcha =0;

int estadomarcha;

int previomarcha =0;

int pieza =0;

int estadosensor;

int estadoencoder;

int previosensor =0;

int previoencoder =0;

long time =0;

long espera =20; //tiempo que debe estar activado el sensor

int vuelta =0;

int contador =0;

int E0;

int E1;

int E2;

int E3;

int E4;

void setup ()

{

  Serial.begin(9600);
```

```

digitalWrite (sensor, HIGH); //resistencia pull-up
digitalWrite (encoder, HIGH); //resistencia pull-up
pinMode (blanca, OUTPUT);
pinMode (verde, OUTPUT);
pinMode (izquierda, OUTPUT);
pinMode (derecha, OUTPUT);
}

void loop ()
{
//Configuración del sensor para no tener rebotes:
int optico;
estadosensor=digitalRead(sensor);
if (estadosensor!=previosensor){ //al cambiar de estado LOW => HIGH
    time=millis(); //cuenta el tiempo en que está HIGH
}
if ((millis()-time)>espera){ // si es superior a 20ms
    optico=estadosensor; // Es decir pasa HIGH
    if (optico==HIGH){ //Si está activado hay pieza y sino no.
        pieza=HIGH;
    }else{
        pieza=LOW;
    }
}
}
previosensor = estadosensor;

```

```
//Configuración del pulsador como interruptor sin interrupciones:

int value;

estadomarcha=digitalRead(inicio);

if (estadomarcha!=previomarcha){ //al cambiar de estado LOW => HIGH

    value=estadomarcha; // Es decir pasa HIGH

    if (value==HIGH){ //Si está activado hay pieza y sino no.

        E3=0; //Reinicia el programa una vez que se ha parado

        marcha=HIGH;

    }

}

previomarcha = estadomarcha;

// Configuración del funcionamiento:

if (E2==0 && E1==0 && E3==0 && E4==0){

    E0=1;

}

if (E0==1 && marcha==HIGH && pieza==LOW){

    E1=1; //Si está E0=1 && se pulsa marcha && no hay pieza

    E0=0; //activa E1 y desactiva E0

}

if (E1==1){

    avance();

}

if (E1==1 && pieza==HIGH){

    E2=1; //Si está E1=1 && hay pieza
```

```

    E1=0;           //activa E2 y desactiva E1
}

if (E2==1){       // Si está activada la E2 retroceso
    retroceso();

    //Configuración del contador
    if (digitalRead(encoder) == HIGH ) {
        if (digitalRead(encoder) == LOW ){
            contador++;
        } }

if(contador==5){
    E3=1;          // Si está E2=1 y se cumple el temporizador
    E2=0;          // desactiva E2
}}

if (E3==1){      // El temporizador ha llegado a 5000ms paro
    paro();
    contador=0;
} }

// Funciones:
void avance () {
    digitalWrite(verde , HIGH);
    digitalWrite(derecha , HIGH);
    digitalWrite(blanca , LOW);
    digitalWrite(izquierda , LOW);
}

```



```
void retroceso () {  
  
    digitalWrite(verde , LOW);  
    digitalWrite(derecha , LOW);  
    digitalWrite(blanca , HIGH);  
    digitalWrite(izquierda , HIGH);  
}
```

```
void paro () {  
  
    digitalWrite(verde , LOW);  
    digitalWrite(derecha , LOW);  
    digitalWrite(blanca , LOW);  
    digitalWrite(izquierda , LOW);  
}
```