

Sistema de fácil análisis y almacenamiento de medidas con Arduino a través de protocolo de comunicación Xbee

Hannier S. Dávila Delgado
Jerry Vargas Jaramillo

Sistema de fácil análisis y almacenamiento de medidas con Arduino a través de protocolo de comunicación Xbee.

Hannier S Dávila Delgado
Jerry Vargas Jaramillo

Trabajo de grado presentado como requisito
parcial para optar al título de Ingeniero
Electricista

Pereira, abril de 2018
UNIVERSIDAD TECNOLÓGICA DE PEREIRA
Programa de Ingeniería Eléctrica.



Sistema de fácil análisis y almacenamiento de medidas con Arduino a través de protocolo de comunicación Xbee.

© Hannier S. Dávila Delgado

© Jerry Vargas Jaramillo

Director: Ph.D (c). Luis Hernando Ríos González
Docente de la Universidad Tecnológica de Pereira

Pereira, abril de 2018

Programa de Ingeniería Eléctrica.

Universidad Tecnológica de Pereira

La Julita. Pereira(Colombia)

TEL: (+57) (6)3137122

www.utp.edu.co

Versión web disponible en: http://recursosbiblioteca.utp.edu.co/***/index.html

Dedicatoria

Dedico este trabajo de grado a mi familia, motivo fundamental para alcanzar cada una de las metas propuestas durante mi vida, por mostrarme siempre el camino correcto, por su abnegado esfuerzo y su compañía a pesar de la distancia.

-Hannier S. Dávila Delgado

Dedicada a mis padres, por su apoyo incondicional en esta etapa.

-Jerry Vargas Jaramillo

Agradecimientos

A mis padres, porque gracias a ellos se cumplió esta meta, por su apoyo y esfuerzo. A mi hermano, por su amistad y compañía durante todo este tiempo. A mis amigos, siempre incondicionales en los momentos más complicados. A los docentes de la Universidad Tecnológica de Pereira por su entrega y dedicación. Al ingeniero Luis Hernando Ríos, por su colaboración y apoyo en cada paso de este proyecto.

-Hannier S. Dávila Delgado

A mis padres, porque gracias a ellos se cumplió esta meta, por su apoyo y esfuerzo. A los docentes de la Universidad Tecnológica de Pereira por su entrega y dedicación.

-Jerry Vargas Jaramillo

Tabla de Contenido

1. Introducción	1
1.1. Justificación	2
1.2. Objetivos	2
1.2.1. General	2
1.2.2. Específicos	2
1.3. Estado del arte	3
1.4. Estructura del trabajo de grado	4
2. Marco Teórico	5
2.1. Redes inteligentes	6
2.2. Comparativa de tecnologías	7
2.2.1. Arduino	7
2.2.2. Beaglebone Black	9
2.2.3. Raspberry Pi	10
2.2.4. Eleccion de tarjeta	11
2.3. Características de sensores	12
2.3.1. Sensor de corriente STC-013	13
2.3.1.1. Sensor de corriente STC-013/100 (salida de corriente)	14
2.3.1.2. Sensor de corriente STC-013/030 (salida de voltaje)	15
2.3.2. Transformador de tensión	15
2.3.3. Sensor de humedad (Higrómetro FC-28)	17
2.4. Comparativa de lenguajes de programación	17
2.4.1. C	18
2.4.2. Python	19
2.4.3. C++	19
3. Aplicación y resultados	20
3.1. Planteamiento con sensor de tensión y sensor de corriente	21
3.2. Planteamiento con sensor de tensión, sensor de corriente y sensor de humedad	26
3.3. Transferencia de datos por medio de protocolo de comunicación Xbee	28
3.4. Base de datos e interfaz de usuario	30
4. Conclusiones, recomendaciones y trabajo futuro	32
Bibliografía	33

Capítulo 1

Introducción

Con el desarrollo tecnológico que ha venido impactando de manera significativa a la humanidad, en los últimos años se ha tomado conciencia de los graves daños que el mal manejo de los recursos puede ocasionar al planeta, gracias a esto cada vez son más las fuentes de energía limpia que se implementan alrededor del mundo. De estas situaciones parte el concepto de Smart grid, o red eléctrica inteligente, un tema ligado de manera estrecha con el presente tema de investigación.

Como se indica en [1] un concepto más estandarizado de lo que son las redes eléctricas inteligentes sería: una red que integra de manera inteligente las acciones de los usuarios que se encuentran conectados a ella, entre los que se encuentran generadores y consumidores, o los usuarios que hacen parte de las dos. Todo esto con el fin de lograr conseguir un suministro de energía eléctrica eficiente, eficaz y confiable, con altos estándares de calidad.

Hablando del marco colombiano, se puede decir que hasta el momento es muy poco el avance que se ha dado con respecto a este tema, debido en su mayoría a los costos que conlleva la implementación de esta tecnología, y sin ir más allá a la poca cultura ciudadana, política y social que se tiene, ya que, en parte, es gracias a estos factores que hay un atraso considerable con respecto a países vecinos.

Aun así, ya es hora de considerar planes para que el desarrollo de estas nuevas tecnologías traiga beneficios a los ciudadanos y también al medio ambiente, y más aún, considerando que el siglo XXI está llamado a ser el siglo de las ciudades. Según la Organización de las Naciones Unidas (ONU) más del 50% de la población mundial es residente en áreas urbanas y se estima que alrededor del año 2050 este porcentaje se eleve al 70%. [2]

Este proyecto pretende aportar al desarrollo de las nuevas ciudades y a la concientización del mal gasto de los recursos, se presenta como un sistema de monitoreo, control y análisis para diversos ámbitos naturales donde los factores ambientales son de gran influencia para diversas aplicaciones industriales que contribuirán al desarrollo de las Smart grids.

1.1. Justificación

Debido al constante cambio en la tecnología enfocada a disminuir el impacto que ha producido el ser humano al medio ambiente. Han surgido diversas alternativas para que la afectación a procesos como la agricultura, consumo de energía, etc. Sean mínimas para el usuario y de forma bilateral se logre una mejora sustancial en el uso de los recursos ambientales.

En los últimos años, en el ámbito colombiano se ha pasado por diversas crisis ambientales, principalmente desencadenadas en gran parte por el llamado fenómeno del niño, llevando a crisis en varios ámbitos que mueven los sectores sociales y económicos del país, tales como la agricultura, ganadería y la generación de energía eléctrica.

Teniendo en cuenta el planteamiento anterior, surge la necesidad de poder verificar las variables ambientales para lograr tener una perspectiva de como es el comportamiento de las mismas en el tiempo y en que pueden afectar a las diversas operaciones que dependen de estas. Y de esta forma obtener un plan de acción preventivo para que la afectación sea mínima. La solución propuesta intenta generar conciencia en la sociedad, mostrando como es la variación de estos factores y como pueden afectar a procesos que dependen de estos, de igual manera se quiere que las personas puedan darse cuenta de cómo los recursos naturales se ven afectados por la falta de conciencia de la sociedad.

1.2. Objetivos

1.2.1. General

Implementar un sistema de fácil análisis y almacenamiento de medidas con Arduino a través de protocolo de comunicación Xbee, con de sensor de corriente STC 013 y transformador de potencial.

1.2.2. Específicos

- Realizar la búsqueda del estado del arte haciendo énfasis en desarrollo de aplicaciones para smartgrids por medio de plataformas de código abierto.
- Desarrollar capacidades de programación en lenguaje de programación para Arduino.
- Realizar la programación de la tarjeta Arduino Uno y verificar su ejecución.
- Verificar mediante pruebas, los resultados obtenidos del sistema implementado.
- Plantear propuestas para aplicar el sistema implementado usando otro tipo de sensores.

1.3. Estado del arte

En los últimos años el desarrollo de aplicaciones para la llegada de las redes inteligentes, ha sido foco de atención para la comunidad tecnológica en general, por este motivo se ha encontrado gran avance en lo que se refiere a sistemas de medidas inteligentes, algunos de los más relevantes, que sirvieron como base sólida para esta investigación fueron los siguientes:

En enero de 2013, Marco Herrera, estudiante de ingeniería eléctrica de la Escuela Politécnica Nacional, realiza una investigación denominada: “Descripción de redes inteligentes (Smart Grids) y su aplicación en los sistemas de distribución eléctrica”. En el cual aporta contenido de introducción a lo que son las redes eléctricas inteligentes, conceptos, tecnología usada para su desarrollo y su evolución a nivel mundial. Además, realiza un análisis de la situación en ese entonces, de la red eléctrica de Ecuador y la necesidad de mejora, para que el atraso sea mínimo en el momento de implementar las redes inteligentes. Así mismo se enfoca en el análisis desde el punto de vista de los usuarios finales del servicio, buscando una forma en que estos logren administrar, controlar y monitorear su consumo de energía. [3]

Investigando en bases de datos de la IEEE se encontró, que, en el año 2014, se realizó una conferencia denominada Conference on Power System Technology, en la cual se presentó un artículo bastante interesante que va ligado a esta investigación, su nombre es “Smart Grid Meter Analytics for Revenue Protection” [4]. Presentado por Zhilin Wu, Tong Zhao, Lihan He y Xiangming Shem. En este artículo, se analizan las funciones de las Smart grids, indicando detalladamente cada etapa que las conforman, además de mostrar el análisis matemático de cómo deberían funcionar las redes de distribución de una forma ideal.

Otro artículo interesante con respecto a este tema que ha ido a evolucionando con el transcurso de los años, también disponible en la base de la IEEE es el de “Real-time Energy Control Approach for Smart Home Energy Management System” [5]. Que es dedicado a la investigación de un control de energía en tiempo real para un hogar, incluyendo variables como calentadores de agua eléctrica, secadoras y tecnologías que hacen parte de las redes inteligentes, como vehículos eléctricos, sistema de almacenamiento de energía o baterías, celdas fotovoltaicas, etc. En este caso se utilizaron formas de control difusas, para lograr un óptimo resultado en cuestión de carga y descarga de baterías, y mecanismos para permitir a los hogares interactuar con los servicios de respuesta a la demanda.

En el contexto nacional, el 30 de enero de 2014, por medio de la Universidad Nacional de Colombia, Natalio Castaño Jaramillo, Carlos Jaime Franco Cardona y Juan David Velásquez Henao, publicaron un artículo en la revista DYNA, llamado “Smart Meters Adoption: Recent Advances and Future Trends” o en español “Adopción de Medidores Inteligentes: Avances Recientes y Tendencias futuras” [6]. En el cual hacen énfasis en la necesidad de que los usuarios tengan control de su pico de demanda máximo y poder ayudar a reducirlos, para esto proponen la implementación de medidores inteligentes, que arrojen la información necesaria para que los usuarios puedan controlar su consumo.

1.4. Estructura del trabajo de grado

El contenido del trabajo de grado inicia con las descripciones previas de la información involucrada en el mismo, como es introducción, justificación, objetivos y estado del arte, continua con el marco teórico en el cual se explican algunos conceptos básicos de los sistemas de medición, tarjetas electrónicas con las que se pueden realizar diversas aplicaciones, sensores, su funcionalidad y características para luego estudiar diversos protocolos de comunicación y el porqué de la selección realizada, sus ventajas y desventajas con respecto a otros tipos. Posteriormente se mostrará el planteamiento realizado, desarrollo del código de programación su ejecución y la metodología para la realización de la aplicación para varios tipos de sensores. Finalmente se presentan los resultados obtenidos, recomendaciones y trabajos futuros con base al presente.

Capítulo 2

Marco teórico

En este capítulo se abordarán los aspectos teóricos de los ámbitos implicados en este proceso de investigación partiendo de un concepto más profundo de que son las redes inteligentes, sus sistemas de comunicación. Se hará una comparación de las tecnologías disponibles para la aplicación, se explicará el funcionamiento de los sensores utilizados para realizar la implementación. Se realizará un análisis de los lenguajes de programación de código abierto, tales como C, Python, C++. etc. debido a que en la actualidad son los de mayor utilización para aplicaciones como la que se muestra en este proyecto.

2.1. Redes inteligentes

Las redes inteligentes son redes que trata de integrar diferentes recursos que están interconectados de forma remota. Una definición bastante centrada de lo que son las redes inteligentes es la de la asociación de electrotecnia de Argentina que la describe como: “La conjunción de la red eléctrica tradicional con tecnologías modernas de la información y la comunicación. Integra datos provenientes de los dos distintos puntos de la cadena eléctrica, desde generador hasta el usuario final; y transformarlos en información y acciones que lleven a una mejora en su gestión. Sus objetivos es elevar la eficiencia, confiabilidad, sustentabilidad, calidad, para hacer frente a nuevos desafíos”. [7] [8]

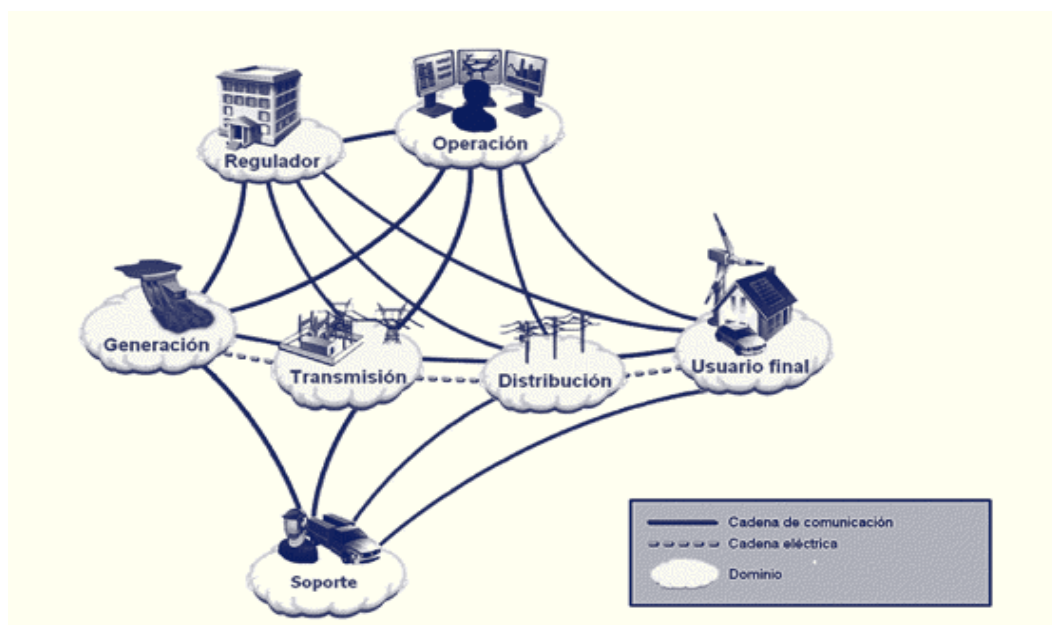


Figura 2.1. Esquema de una red inteligente [9]

Los sistemas de comunicación de las redes inteligentes permiten la interacción entre varios equipos tales como sensores, medidores inteligentes, controladores, etc.

Entre los sistemas de comunicación que se usan con frecuencia están:

- Enlaces PLC: Cuyas siglas minimizan su nombre, Power Line Communications, estos enlaces dan conectividad con redes locales a través de la instalación eléctrica. En otras palabras, PLC usa el cableado existente en las instalaciones para transmitir datos además de las variables eléctricas. [10]

- Wireless Fidelity: Más conocido como conexión Wi-Fi, esta tecnología permite obtener comunicación inalámbrica por medio de ondas, el plus que posee es que brinda interoperabilidad entre los equipos. Actualmente se encuentran dos tipos de comunicación Wi-Fi; el 802.11b que emite a 11mB/s y el 802.11g que es más rápida, debido a su velocidad de 54 mB/s. [11]
- Zig Bee: Es el nombre que se le da a un conjunto de protocolos de comunicación inalámbrica, su utilización se da con radiodifusión digital. Las velocidades de transmisión están en un rango entre 20kB/s y 250 kB/s y sus rangos de alcance van entre los 10m a 95m en línea vista. [12]
- Xbee: Son módulos con soluciones integradas, son un medio inalámbrico para la conexión y comunicación entre dispositivos. Utilizan un protocolo de red llamado IEEE 802.15.4 para realizar conexiones de redes punto a punto o punto a multipunto. Entre sus características está su gran desempeño en aplicaciones con alto tráfico de datos y su baja latencia. Es basado en el protocolo ZigBee. [13]

2.2. Comparativa de tecnologías

En la actualidad con el auge de las nuevas tecnologías se han venido desarrollando varias posibilidades capaces de lograr aplicaciones futuristas, como la implementación de sistemas basados en domótica, sistemas automatizados, en los cuales los sensores son parte fundamental, además de que los usos de estas nuevas tecnologías van encaminadas a una reducción del impacto al medio ambiente. Las tecnologías como Arduino, Raspberry Pi y BeagleBone Black son el foco de estudio para esta aplicación debido a su explosión comercial en los últimos años, por esta razón la comparativa se centrará en estas.

2.2.1. Arduino

Arduino, es una plataforma electrónica de código abierto, se basa en un hardware y en un software de fácil uso. Se caracterizan por leer entradas de cualquier tipo, como la luz, energía, temperatura etc. Las placas Arduino son relativamente económicas en comparación con otras plataformas con controladores.

El software consiste en dos elementos, el primero es el entorno de desarrollo denominado Arduino IDE que se ejecuta en sistemas operativos Windows, Macintosh OSX y Linux, la mayoría de microcontroladores se enfoca en desarrollarse a Windows, mientras esta plataforma posee más variedad de sistemas operativos. El segundo elemento es el cargador de arranque o llamado bootloader, que se ejecuta dentro del microcontrolador en el momento que este se enciende. A este tipo de entorno que se les denomina de código libre, se desarrolla en su mayoría a través de bibliotecas C++, de igual manera se puede agregar código AVR C directamente en los programas de Arduino. [14] [15]

El hardware consiste en una placa con un microcontrolador y puertos de entrada y salida, tanto análogas como digitales. Las placas se clasifican en dos tipos, de desarrollo o de expansión; para el

caso de las placas de desarrollo los más conocidos son: Arduino Uno, Arduino Galileo, Arduino Due, Arduino Mega, Arduino Mega, Arduino Mini, Arduino Nano, entre otros. [15]



Figura 2.2. Vista superior Arduino Uno. [16]

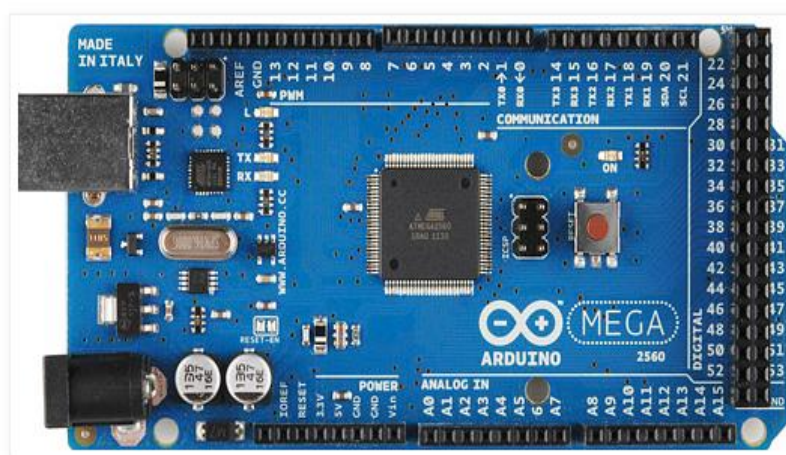


Figura 2.3. Vista superior Arduino Mega [17]

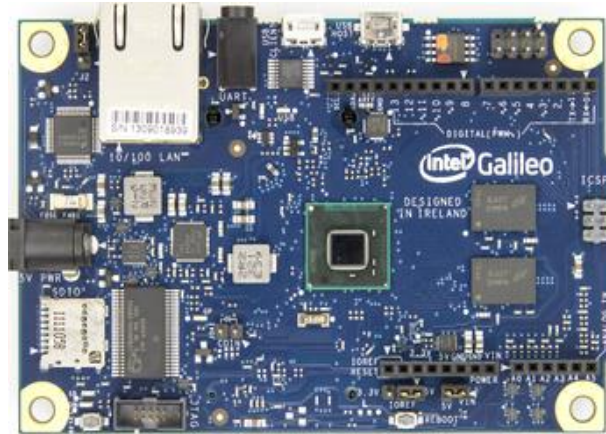


Figura 2.4. Vista superior Arduino Galileo [18]

Una variable interesante de la plataforma Arduino, es la capacidad de acoplarse a otros lenguajes de programación sin tener ninguna consecuencia en sus procesos, algunos de los más populares son C, C++, Python, etc.

2.2.2. BeagleBone Black

La Beaglebone Black es una tarjeta de desarrollo, con un procesador hecho por la conocida compañía Texas Instrument que posee una arquitectura ARM Cortex A8. Esta tarjeta presenta grandes capacidades de procesamiento, periféricos y GPIOs. Una de las desventajas es que solo tiene un puerto USB Host. Un aspecto importante del hardware es que es totalmente abierto, esto significa que todos los archivos de fabricación de la board (PCB) y el esquema de la tarjeta están disponibles para ser revisados por el público. [19]

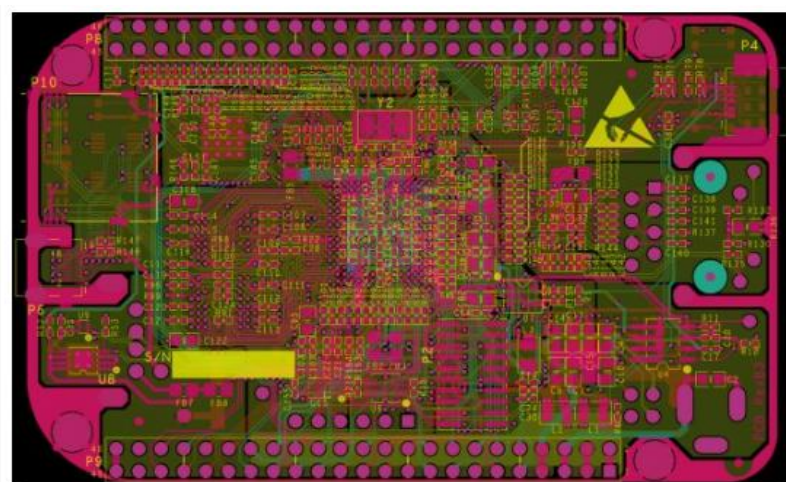


Figura 2.5. PCB de tarjeta Beaglebone Black [19]

La tarjeta viene pre cargada de fábrica con un sistema operativo, puede ser Linux o Debian, además es compatible con software comerciales como lo son Android, Ubuntu, Cloud, etc. Otra ventaja interesante de la tarjeta es que puede ser programada por varios lenguajes, tales como C, C++, Python, etc. Hay otra versión interesante de esta tarjeta de desarrollo que se llama Beaglebone Blue, con características similares a la que fue considerada para este proyecto. [19] [20]

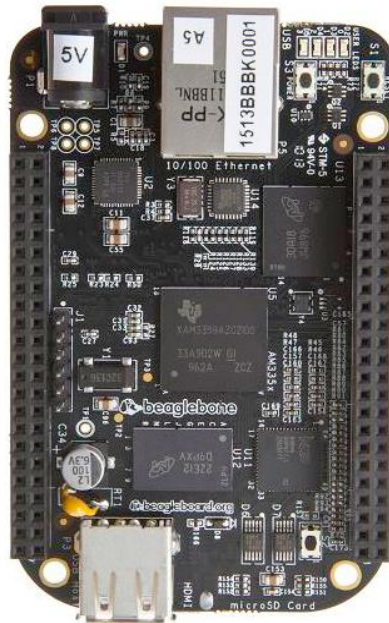


Figura 2.6. Vista superior de Beaglebone Black. [20]



Figura 2.7. Vista superior de Beaglebone Blue. [21]

2.2.3. Raspberry Pi

Es un micro ordenador o una placa de computadora, al igual que las otras tecnologías mencionadas líneas arriba, es de las tecnologías denominadas de código abierto, tanto en hardware como en software. Su sistema operativo es una versión modificada y adaptada de Debian, llamada Raspbian, aunque al ser de código abierto puede usar otros sistemas operativos sin alterar sus procesos, para esta clase de tarjetas se promueve mucho el lenguaje de programación Python, aunque también es compatible con otros lenguajes, como Ruby, TinyBasic, C, C++. [22]

Al ser un micro ordenador su tamaño es muy reducido, más o menos del tamaño de una tarjeta bancaria, a pesar de sus dimensiones posee un procesador ARM con potencia hasta 1GHz, cuenta con 512 mB de memoria RAM, GPU, Video Core IV, lo necesario para que la tarjeta no sea solo programable, sino que también tenga entre sus funciones ejecutar programas, navegar en internet, entre otras funciones.

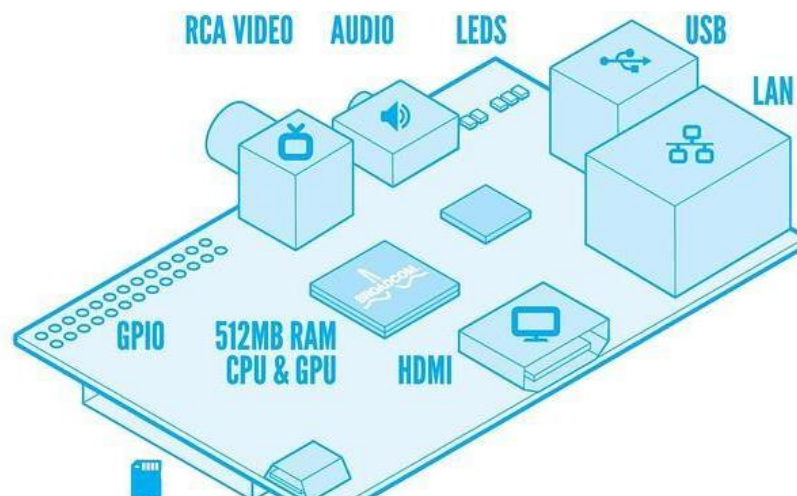


Figura 2.8. Esquema general de Raspberry Pi. [23]

Es importante resaltar que dentro de la fabricación de Raspberry Pi se hicieron dos modelos, denominados A y B, de los cuales nacen nuevas actualizaciones provocando que esta tarjeta se mantenga en constante innovación, las únicas diferencias entre los dos modelos se basan en la capacidad de memoria, debido a que el modelo B posee 512 mB mientras que el modelo A posee 256 mB, en el consumo energético, el modelo B con 3.5 W y el modelo A con 2.5 W, la conectividad de red, ya que el modelo A no lo posee y la última diferencia se puede identificar en los puertos USB, donde el modelo B posee 2 puertos y el modelo A solamente 1. [22]

Por ultimo con respecto a este micro ordenador, dado su economía ha sido parte de grandes desarrollos en los últimos años, pero, lo que en muchas ocasiones se desconoce son los accesorios que se necesitan para poder lograr un óptimo resultado, lo que conlleva a un gasto mayor al que se puede plantear inicialmente.

2.2.4 Elección de la tarjeta

Como se puede verificar en los literales, 2.2.1, 2.2.2 y 2.2.3 la variedad de características y beneficios que ofrecen las tarjetas analizadas son de gran aporte para la aplicación, pero una variable fundamental, que se debe analizar para escoger la más acertada es el factor económico, los beneficios que ofrecen las tarjetas se pueden ver en resumen en la tabla, la cual tiene como objetivo facilitar la tarea.

	Arduino	Beaglebone Black	Raspberry pi
Costo	Bajo	Medio-Alto	Medio-Alto
Beneficios	Capacidad de conectar sensores de distinta naturaleza. Microcontrolador capaz de embeber software y programable desde cualquier PC.	Conectar sensores de distinta naturaleza. 512mB de RAM DDR3. Sistema operativo seleccionable. Fast Ethernet 10/100 Mbps.	Similar a un computador, con potencia limitada. Interconectar sensores de distinta naturaleza
Aprendizaje	Fácil	Medio	Fácil
Facilidad de uso	Alta	Medio-Alto	Alto
Documentación	Mucha	Media-Baja	Mucha
Adaptación a la aplicación	Alta	Alta	Alta

Tabla 1. Características y beneficios de las tres tecnologías investigadas.

Con ayuda de la tabla 1, la tarjeta escogida fue la de Arduino, ya que su bajo costo, adaptabilidad para la aplicación, facilidad de manejo y cantidad de documentación hacían que la aplicación se desarrollara con mayor facilidad. Es de resaltar que cualquiera de las tres se adaptaba al proyecto, pero factores como los mencionados anteriormente además del económico hacían que la posibilidad el Arduino fuera más viable.

2.3. Características de los sensores

Un sensor se define como un dispositivo que está en la capacidad de detectar estímulos o señales externas y responder de una forma pre-determinada. Estos equipos son capaces de transformar medidas de magnitudes físicas o químicas en señales de carácter eléctrico. Para el caso de este estudio se realizó la implementación con dos tipos de sensores, uno de corriente y otro de voltaje, además se propuso una forma de acondicionar un sensor de humedad para funcionar en el sistema propuesto.

2.3.1. Sensor de corriente STC-013

Estos sensores trabajan como transformadores, la corriente que circula por el cable en el cual se va a medir la corriente, actúa como el devanado primario, este devanado se considera como una sola espira. Internamente, el sensor posee un devanado secundario, que dependiendo del modelo del sensor puede tener hasta 2000 espiras. Es importante saber que este tipo de sensores son de carácter no invasivo, esto significa que se puede medir la corriente que atraviesa el conductor sin necesidad de cortarlo o modificarlo. La medición se realiza por inducción electromagnética. [25] [24]

Dentro de la clasificación de sensores SCT-013, hay dos modelos, uno que proporciona la medida tomada como una salida de corriente y otro que entrega una señal de tensión. La precisión es de 1-2%, esto depende de que el núcleo ferromagnético se cierre de forma adecuada, ya que si queda mal cerrado puede provocar desviaciones de hasta el 10%. [25]

Con respecto a su funcionamiento, se puede decir que tiene similitudes con un transformador de tensión, pero buscan medir magnitudes distintas, por esta razón la construcción y el diseño es diferente. En la figura 2.9 se observa la construcción.

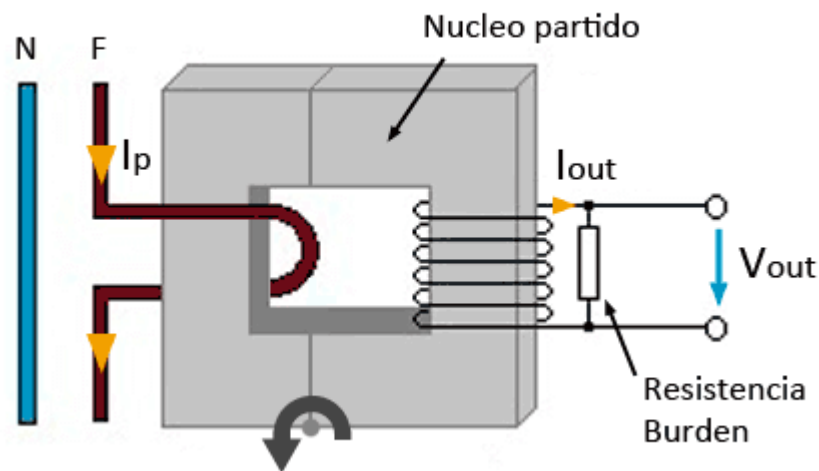


Figura 2.9. Esquema del sensor STC 013. [25]

Un transformador de corriente (CT) busca generar una corriente en el secundario que sea proporcional a la corriente que atraviesa en el lado primario. Por esta razón en el lado primario el número de espiras es reducido. El funcionamiento es muy sencillo, cuando la corriente circula por el conductor se genera un flujo magnético en el núcleo ferromagnético, que a su vez genera una corriente en el secundario.

La relación de transformación de corriente depende de la relación entre el número de espiras, como se muestra en la ecuación (1).

$$\frac{V_p}{V_s} = \frac{I_s}{I_p} = \frac{N_p}{N_s} \quad (1)$$

Donde V_p : Tensión en el lado primario, V_s : Tensión en el lado secundario, I_s : Corriente en el lado secundario, I_p : Corriente en el lado primario, N_p : Numero de espiras en el primario y N_s : Numero de espiras en el secundario.

Como se observa en la figura 2.9. El núcleo que maneja el sensor es partido, lo cual hace que el sensor adquiera una forma de pinza, por lo tanto, se hace mucho más manejable y practico a la hora de ubicarlo en algún conductor.

2.3.1.1 Sensor de corriente STC-013/100 (Salida de corriente)

Este sensor mide una corriente de hasta 100A, teniendo como señal de salida 50mA, justo para la entrada analógica de Arduino, por tanto, su relación es de 100A: 50mA. En la figura 2.10. se observa la forma del sensor, mientras que en la figura 2.11 se aprecia el esquema circuital. [24]



Figura 2.10. Sensor STC-0130/100. [24]

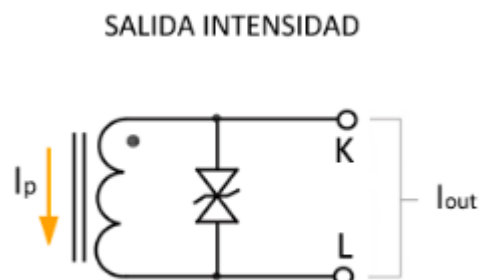


Figura 2.11. Esquema circuital del sensor STC-013/100

2.3.1.2. Sensor de corriente STC-013/030. (Salida de tensión)

Permite realizar medidas en un rango de 30A. Este modelo tiene una resistencia de carga interna, la cual hace que la salida sea una señal de tensión. La relación de este modelo es de 30A : 1V. Este modelo fue el escogido para realizar la implementación debido a que se ajusta al rango de la tarjeta Arduino, además de que su esquema circuital permite que no sea necesario un acondicionamiento. En la figura 2.12 se observa físicamente mientras que en la figura 2.13. aparece el diagrama circuital. [24]



Figura 2.12. Sensor STC-013/030. [24]

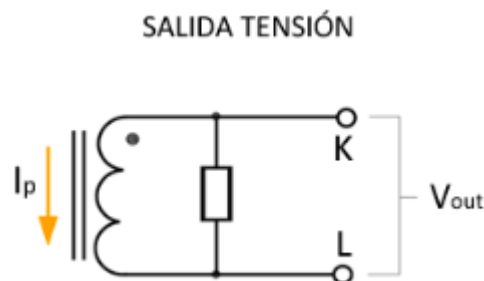


Figura 2.13. Esquema circuital sensor STC-013/030. [25]

2.3.2. Transformador de tensión

Es un dispositivo utilizado para aumentar o disminuir la tensión en un circuito eléctrico, manteniendo la potencia. Su principio de funcionamiento es la transferencia de energía eléctrica de uno o más circuitos, a uno o más circuitos por medio de la inducción electromagnética, de esta forma se eleva o se reduce las magnitudes de corriente o voltaje.

En un transformador de tensión que se usa como equipo de medida, por lo general su funcionamiento es de reductor de tensión. Tiene por lo general una potencia muy baja, dado que su objetivo principal

es suministrar una muestra de voltaje. En la figura 2.14 se muestra el esquema de transformador usado para la aplicación. [26]

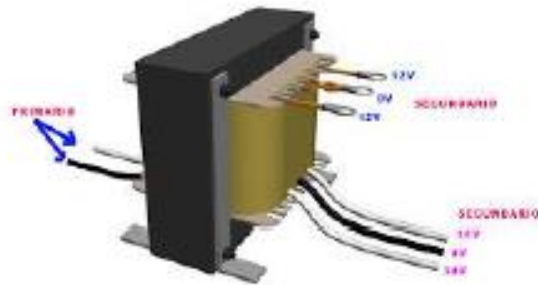


Figura 2.14. Esquema de transformador de tensión.

El circuito equivalente de un transformador de medida es muy similar a un transformador convencional con dos bobinas. Tal como se observa en la figura 2.15.

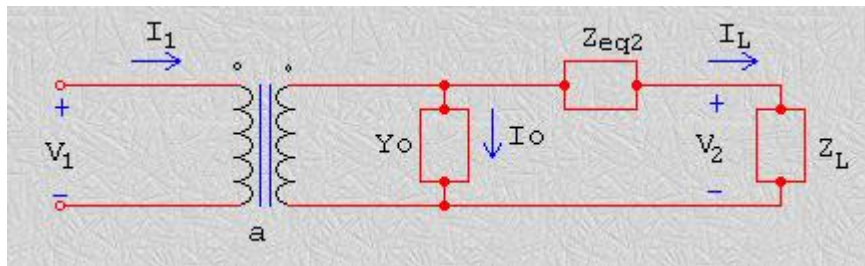


Figura 2.15. Circuito equivalente del transformador de tensión. [26]

Al referir el circuito al secundario del transformador, se obtiene que las equivalencias son las siguientes: Z_{eq2} : Impedancia equivalente, referida al secundario, Z_L : Impedancia del instrumento (voltímetro), V_2 : Tensión secundaria, refleja de la tensión primaria, $Y_o = 0$.

Con estas consideraciones la ecuación de la malla en el secundario del circuito es:

$$\frac{V_1}{V_2} = I_L \cdot Z_{eq2} + V_2 \cdot I_L = \frac{V_2}{Z_L} \quad (2)$$

Por lo tanto

$$\frac{V_1}{V_2} = \left(\frac{Z_{eq2}}{Z_L + 1} \right) \cdot a \quad (3)$$

Se observa que la relación de transformación V_1/V_2 difiere en a con el coeficiente:

$$\frac{Z_{eq2}}{Z_L + 1} \quad (4)$$

Con las ecuaciones (2), (3) y (4) se observa a que existe un error de magnitud y fase, debido a que el resultado es un número complejo.

2.3.3. Sensor de humedad (Higrómetro FC-28)

Esta parte es el cumplimiento del objetivo específico, en el cual se plantea la posibilidad de usar el sistema implementado con otro tipo de sensores, en este caso se medirá la humedad del suelo por medio de un higrómetro FC-28.

Este sensor mide la humedad del suelo, como se mencionó antes, se emplean en sistemas de riego, este sensor es muy sencillo, mide la humedad por la variación de su conductividad. El sensor se distribuye con una placa que permite obtener la medición como una señal analógica o una señal de salida digital, esta se activa cuando la humedad superar un valor, esta placa tiene un circuito comparador y un potenciómetro. En si el funcionamiento es muy básico, cuando el valor medido esta entre 0-500 se concluye que el suelo está húmedo, mientras si sus valores oscilan entre 500 - 1023 se concluye que la humedad es casi nula, se puede decir que el suelo esta árido. [27] [28]

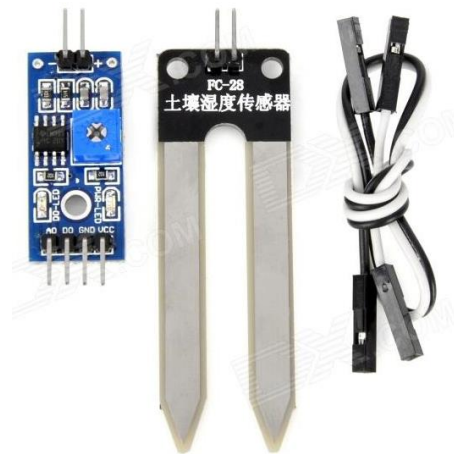


Figura 2.16. Composición del sensor de humedad. [29]

2.4. Comparativa de lenguajes de programación

Actualmente se posee una amplia gama de lenguajes de programación. Estos lenguajes presentan algunas prestaciones diferentes dependiendo de cuál se escoja, y su uso puede venir dado por la aplicación que se tenga planeado desarrollar. Un punto bastante importante a la hora de escoger cierto lenguaje es la dificultad que puede presentar para el desarrollador, lo cual puede facilitar o complicar la aplicación.

En los últimos años, la popularidad de algunos lenguajes ha crecido más que otros, como se muestra en la figura 2.17.

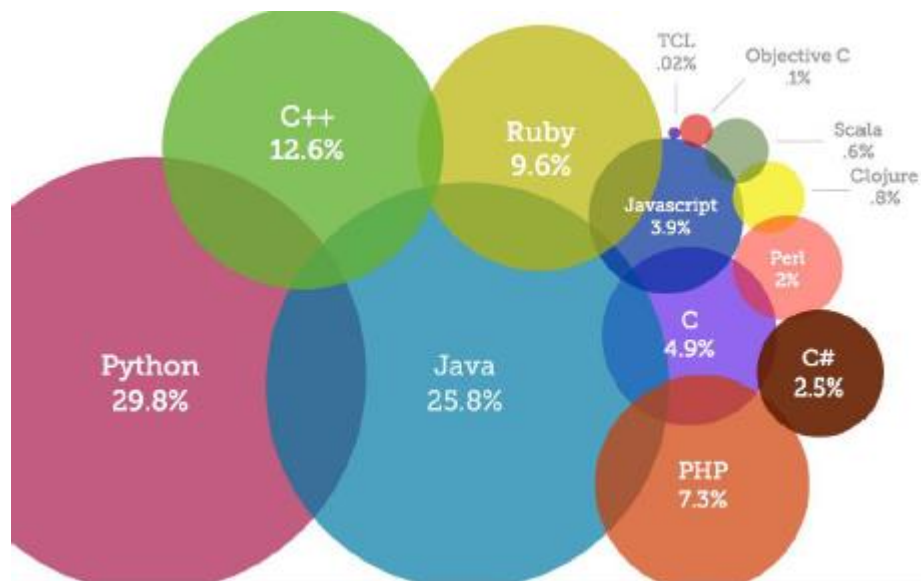


Figura 2.17. Popularidad de algunos lenguajes de programación en el año 2013. [30]

Como se mencionó líneas arriba, esta comparativa se centrará básicamente en tres lenguajes de programación, C, C++ y Python.

2.4.1. C

Es un lenguaje de programación desarrollado por Dennis Ritchie entre 1969 y 1972 como evolución del lenguaje anterior B, que a su vez fue basado en BCPL.

Este lenguaje es orientado a la implementación de sistemas operativos Unix. La ventaja de C es que aparte de la creación de sistemas operativos también es utilizado para crear aplicaciones, debido a la eficiencia de su código.

C es considerado un lenguaje de alto nivel debido a que es estructurado y posee sentencias y funciones que simplifican su compilación. Al mismo tiempo se tiene la posibilidad de programar a bajo nivel (memoria, tocando los registros, entre otros). Fue desarrollado principalmente para incentivar la programación de una máquina de forma independiente.

Las ventajas de este lenguaje es que es muy eficiente porque ofrece la posibilidad de utilizar sus características de bajo nivel para realizar aplicaciones de gran calidad, además de proporcionar facilidades para realizar programas para modular y utilizar bibliotecas existentes para el lenguaje.

Mientras que el mayor inconveniente de este lenguaje es con los datos dinámicos, por la poca velocidad de desarrollo, esto ocurre porque el compilador de este lenguaje se limita a la traducción de su código solamente. Por otra parte, el mantenimiento es un poco más costoso y complicado que con otros lenguajes de alto nivel. [31]

2.4.2. Python

Se trata de un lenguaje de programación multiparadigma, esto significa que tiene varios usos, en este caso este lenguaje soporta la programación orientada a objetos, imperativa y en una menor escala la programación funcional. Este lenguaje tiene un tipado dinámico y es multiplataforma. Se considera un lenguaje muy simple de aprender, pero a su vez muy poderoso, cuenta con estructuras de datos muy eficientes y de alto nivel. Su lenguaje para scripting, desarrollo rápido y capacidad de integrarse fácilmente con lenguajes como C y C++, hacen de este lenguaje una gran herramienta. [32]

Las ventajas de este lenguaje de programación son bastantes, algunas de ellas son: Que es un lenguaje muy simple, por lo que es muy sencillo iniciarse en este tipo de programación, debido a su naturaleza de open source puede funcionar en diversas plataformas como Macintosh, Solaris, OS/2, Linux, Windows, etc. Una característica importante es que el código no es necesario compilarlo, por lo cual el desarrollo es más rápido. [33]

Algo que le juega en contra a esta herramienta es que por su naturaleza de no compilación es un poco lento al ejecutarse, dado que los lenguajes de interpretación son mucho más lentos que los lenguajes de compilación.

2.4.3. C++

Es un lenguaje de programación diseñado a mediados de 1980 por Bjarne Stroustrup. En primera instancia la intención fue extender el lenguaje C, por esta razón C++ es un lenguaje híbrido, posteriormente se añadieron facilidades para una programación genérica, posteriormente se añadieron facilidades para realizar programación genérica, los cuales llevaron a una programación orientada a objetos, por esta razón se dice que este lenguaje es un lenguaje de programación multiparadigma.

En la actualidad C++ es un lenguaje muy versátil, potente y muy general. Su éxito y aplicación entre desarrolladores le ha llevado a ser una herramienta muy utilizada, llevando a ocupar el primer puesto en el desarrollo de aplicaciones. Este lenguaje mantiene las ventajas de C en cuanto a riqueza, flexibilidad y eficiencia.

Algunos de los problemas de este lenguaje son la falta de reflexión o de recolector de basura, además de que sus tiempos de compilación son lentos.

En el caso de la implementación que se plantea en este escrito, es bien sabido que el lenguaje de programación de arduino es basado en C++, por tal razón se decidió hacer énfasis en conocimientos sobre este lenguaje de programación para la realización de la implementación.

Capítulo 3

Aplicación y resultados

En este capítulo se mostrará el desarrollo de la propuesta planteada. El presente documento ha sido desarrollado con el propósito mostrar los resultados obtenidos después de realizar pruebas de una implementación, cuyo propósito es el de almacenar y analizar en una base datos medidas de variables tales como corriente, tensión eléctrica, humedad, etc. Para de esta forma, obtener una tendencia de cómo puede ser el comportamiento de las variables en un futuro y como pueden afectar los procesos industriales y el avance de la sociedad.

Se propuso una alternativa para obtener medidas de variables eléctricas, con el fin de que el usuario logre tener un monitoreo del consumo de energía, esto con algunas condiciones, como que la carga la cual sea objeto de monitoreo sea netamente o mayormente resistiva. Esta alternativa puede ser muy útil en acometidas domiciliarias donde aún no predominan los electrodomésticos que no inyecten interferencia a la red, un ejemplo de estos la tecnología LED.

Para hacer más amplia la gama de variables analizadas se hizo una modificación en el código para obtener el funcionamiento de un sensor de humedad, este sensor es muy útil para aplicaciones como cultivos, ya que hay plantaciones más delicadas que otras, donde la humedad puede afectar de manera significativa, de igual manera con ayuda del análisis grafico que ofrece la base de datos se puede detectar en que periodos del año la tierra es más húmeda o más árida, lo que hace de esta propuesta una herramienta muy útil para tener planes de contingencia para que la afectación sea mínima.

3.1. Planteamiento con sensor de tensión y sensor de corriente.

La estructura básica de un programa en Arduino es bastante simple y divide su ejecución en dos segmentos setup y Loop. Setup() constituye la preparación del programa y Loop() es la ejecución en ciclo. En la función Setup() se incluye la declaración de las variables y se trata de la primera función que se ejecuta en el programa. Esta función se ejecuta una única vez y es empleada para configurar el pinMode (por ejemplo: si un determinado pin digital es de entrada o salida) e inicializar la comunicación serie. La función Loop() incluye el código a ser ejecutado continuamente (leyendo las entradas de la placa, o salidas de los sensores ya anteriormente mencionados). A continuación, se presenta el código

```
1. #include <LiquidCrystal I2C.h>
2. #include<Wire.h>
3. #include<math.h>
4. LiquidCrystal_I2C lcd(0x3F,16,2);
5. const int analogInPin = A0;
6. const int analogInPin1 = A1;
7. const int sensorPin = A2;
8. int retardo = 2 ; // (tiempo (s.) entre visionados)
9. float lectura,lectural,ff,pKW,iA,vV,vS,S_Ratio,Y;
10. long x;
11. int i=0;
12.
13. void setup() {
14.     lcd.init();
15.     lcd.backlight();
16.     lcd.clear();
17.     lcd.setCursor(0,0);
18.     lcd.print("P=      I= ");
19.     lcd.setCursor(0,1);
20.     lcd.print("V=      );
21.     Serial.begin(9600);
22.     S_Ratio = 36.5; // Sensor/ratio (mV/mA ) : 36.5
23.     vV = 120; // valor de tension a computar
24.     ff = 4.15; // freq. factor / (50Hz -> 5 / 60Hz -> 4.15)
25. }
26.
27. float smoothreadx(float fx){ // fc (factor corrector)
28. int ni = 35; // n. de iteraciones => smooth
29. // (ni) => rango 10 a 50 mejor promedio [smoothing]
30. float retorno = 0.0;
31. for (int x = 0; x< ni; x++){
32. do { // espero paso por cero
33.     delayMicroseconds(100);
34.     } while (analogRead(1) != 0) ;
35.     delay (ff); // espera centro de ciclo
36.     delay (10); // estabilizacion CAD
37.     retorno = retorno +(analogRead(1)*fx);
38.     }
39. return retorno / ni;
40. }
41.
42. float smoothread(float fc){ // fc (factor corrector)
43. int ni = 35; // n. de iteraciones => smooth
44. // (ni) => rango 10 a 50 mejor promedio [smoothing]
45. float retorno = 0.0;
46. for (int x = 0; x< ni; x++){
47. do { // espero paso por cero
48.     delayMicroseconds(100);
```

```

49.         } while (analogRead(0) != 0) ;
50.         delay (ff);           // espera centro de ciclo
51.         delay (10);           // estabilizacion CAD
52.         retorno = retorno +(analogRead(0)*fc);
53.     }
54.     return retorno / ni;
55.
56. }
57.
58. void loop() {
59.
60.     lectura = smoothread (1) / 1.41;    // lectura (rms)
61.     iA = (lectura * S_Ratio)*4.9259/1000-1;    // Intensidad (A)
62.     if (iA<0.5){
63.         iA=0;
64.     }
65.     else {
66.         iA=iA;
67.     }
68.     lectural = smoothreadx (1) / 1.41;    // lectura (rms)
69.     vV = (lectural *
70.     S_Ratio)*122*0.933435348/(1000*6.061739944);    // Tension (V)
71.     pKW = (vV * iA);    // Potencia (kW)
72.
73.     Serial.println( (String) "DATA,DATE,TIME," + vV + "," + iA);
74.     lcd.setCursor (2,0);
75.     lcd.print (pKW);
76.
77.     lcd.setCursor (11,0);
78.     lcd.print (iA);
79.
80.     lcd.setCursor (2,1);
81.     lcd.print (vV);
82.
83.     delay(retardo * 300);
84. }

```

Aunque el desarrollo del programa en Arduino fue parte esencial, es igualmente de vital importancia el almacenamiento de los datos obtenidos que es el enfoque final realmente, y para esto se hizo uso de una herramienta llamada PLX-DAQ. Inicialmente desarrollada en un entorno VBA por un programador anónimo conocido como NetDevil, pulida por la comunidad del fórum Arduino y finalmente modificada, adaptada y mejorada por nosotros obtener los resultados deseados, esta herramienta es un programa diseñado para establecer de manera sencilla la comunicación entre Microsoft Excel en una computadora Windows y cualquier dispositivo que soporte protocolo de puerto serial (optimizada para su uso con la placa Arduino).

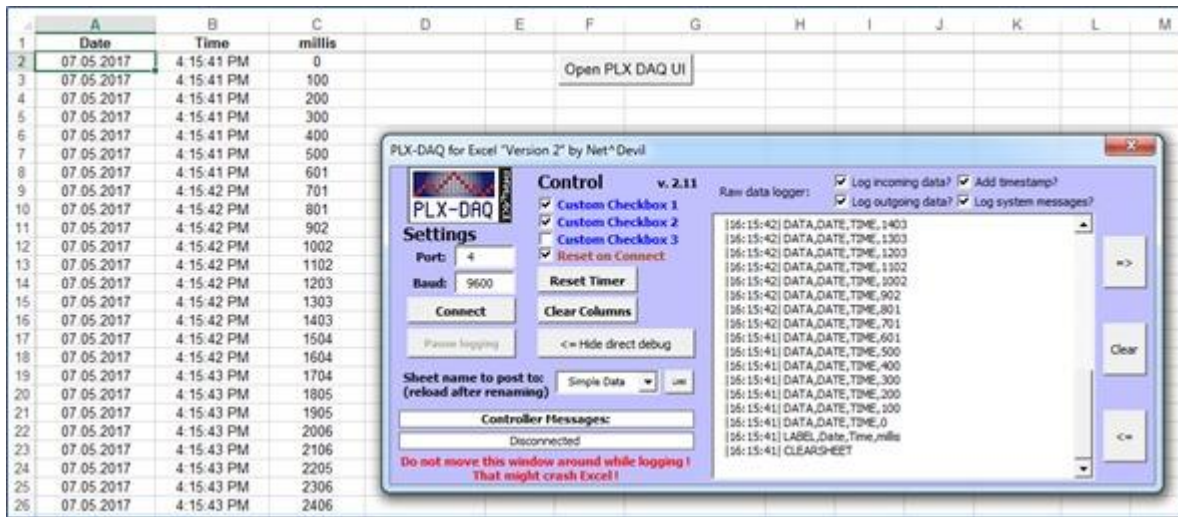


Figura 3.1. Base de datos PLX-DAQ en Excel

En la figura 3.1 se observa la base de datos usada para lograr un análisis sencillo de las variables. Esta base de datos fue diseñada para obtener solo una entrada y no mostrar análisis gráfico, por tal razón se vio la necesidad de ampliar el número de entradas y tener un módulo de visualización gráfica, además de mejorar su presentación para que esta fuera más interactiva, fácil e intuitiva con el usuario.

```

1.
2. Dim ChartNum As Integer
3.
4. Private Sub UserForm_Initialize()
5.     ChartNum = 1
6.     UpdateChart
7. End Sub
8.
9. Private Sub PreviousButton_Click()
10.     If ChartNum = 1 Then ChartNum = 3 Else ChartNum = ChartNum - 1
11.     UpdateChart
12. End Sub
13.
14. Private Sub NextButton_Click()
15.     If ChartNum = 3 Then ChartNum = 1 Else ChartNum = ChartNum + 1
16.     UpdateChart
17. End Sub
18.
19. Private Sub CloseButton_Click()
20.     Unload Me
21. End Sub
22.
23. Private Sub UpdateChart()
24.     Set CurrentChart = Sheets("Simple
25. Data").ChartObjects(ChartNum).Chart
26.     CurrentChart.Parent.Width = 700
27.     CurrentChart.Parent.Height = 500
28.     ' Save chart as GIF
29.     Fname = ThisWorkbook.Path & Application.PathSeparator &
30. "temp.gif"
31.     CurrentChart.Export FileName:=Fname, FilterName:="GIF"
32.     ' Show the chart
33.     Imagen1.Picture = LoadPicture(Fname)

```

```
34.     End Sub
35.
```

El código anterior es el encargado de actualizar, y permitir la visualización de las gráficas de los datos censados dentro de una userform. Las siguientes líneas son las encargadas de las acciones que deben ser ejecutadas al momento de abrir la aplicación además de preparar y establecer la configuración inicial por medio de la apertura de userform encargada de determinar los parámetros iniciales que deben ser cargados por el usuario para la comunicación de puerto serial con la placa arduino.

```
1. Option Explicit
2.
3. Dim shName As String
4. Dim Avail As Variant
5.
6. Private Sub Workbook_NewSheet (ByVal Sh As Object)
7.
8.     Dim i As Integer
9.
10.        frmStampDAQ.UpdateSheetList = True
11.
12.        frmStampDAQ.cmbBoxSheetName.Clear
13.
14.        For i = 1 To ActiveWorkbook.Worksheets.Count
15.            If (ActiveWorkbook.Worksheets(i).Name <> frmStampDAQ.get
SettingsSheetName) Then
16.                frmStampDAQ.cmbBoxSheetName.AddItem
ActiveWorkbook.Worksheets(i).Name
17.            End If
18.        Next i
19.
20.        frmStampDAQ.cmbBoxSheetName.Value = frmStampDAQ.getCurSheetN
ame
21.
22.        frmStampDAQ.UpdateSheetList = False
23.
24.    End Sub
25.
26.    Private Sub Workbook_Open()
27.        Application.Visible = False
28.        frmStampDAQ.Show
29.        Load UserForm1
30.    End Sub
31.
32.    Public Sub openUI ()
33.        frmStampDAQ.Show
34.    End Sub
35.
36.
37.
38.    Private Sub Workbook_SheetActivate (ByVal Sh As Object)
39.
40.        On Error Resume Next
41.
42.        Dim i As Integer
43.
44.        Avail = Sheets(shName).Range("A1")
45.        If Err.Number <> 0 Then
46.            frmStampDAQ.UpdateSheetList = True
47.
48.            frmStampDAQ.cmbBoxSheetName.Clear
```

```

49.
50.         For i = 1 To ActiveWorkbook.Worksheets.Count
51.             If (ActiveWorkbook.Worksheets(i).Name <> frmStampDAQ
52.                 .getSettingsSheetName) Then
53.                 frmStampDAQ.cmbBoxSheetName.AddItem
54.                 ActiveWorkbook.Worksheets(i).Name
55.             End If
56.         Next i
57.
58.         '## if deleted sheet was the one currently used then
59.         update sheet in form already
60.         If (frmStampDAQ.getCurSheetName = shName) Then
61.             frmStampDAQ.UpdateSheetList = False
62.             frmStampDAQ.cmbBoxSheetName.Value =
63.             IIf(frmStampDAQ.UpdateSheetList, frmStampDAQ.getCurSheetName,
64.                 ThisWorkbook.ActiveSheet.Name)
65.
66.         frmStampDAQ.UpdateSheetList = False
67.         End If
68.
69.         On Error GoTo 0
70.
71.     End Sub
72.
73. Private Sub Workbook_SheetDeactivate(ByVal Sh As Object)
74.     shName = Sh.Name
75. End Sub

```

Después de realizar las modificaciones necesarias para que la base de datos se adaptara a las necesidades del sistema la parte visual de esta quedo como se muestra en la figura 3.2. y 3.3.

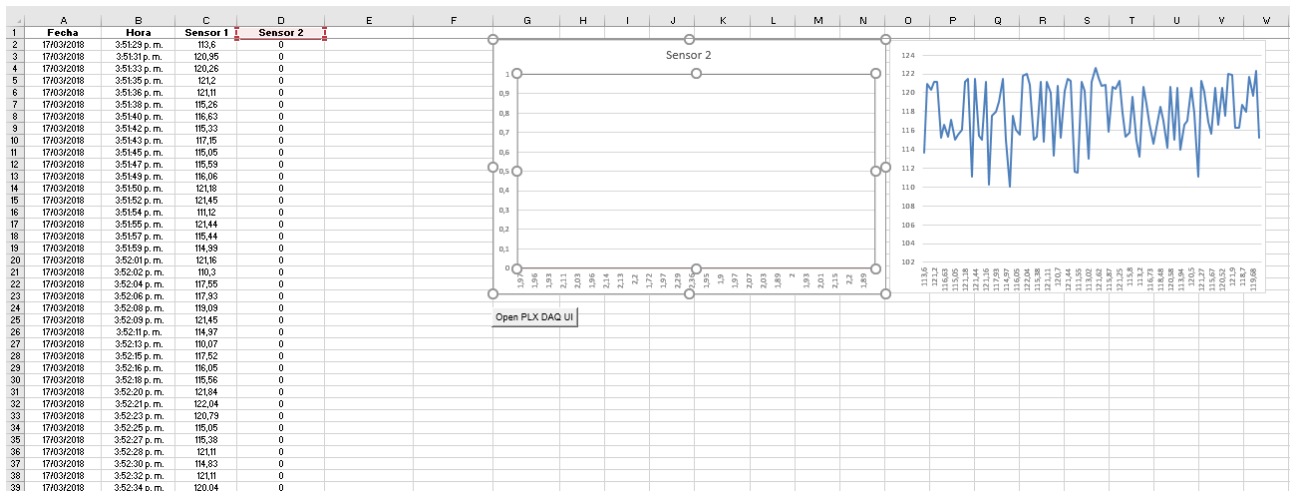


Figura 3.2. Base de datos con visualización de datos grafica para dos sensores.

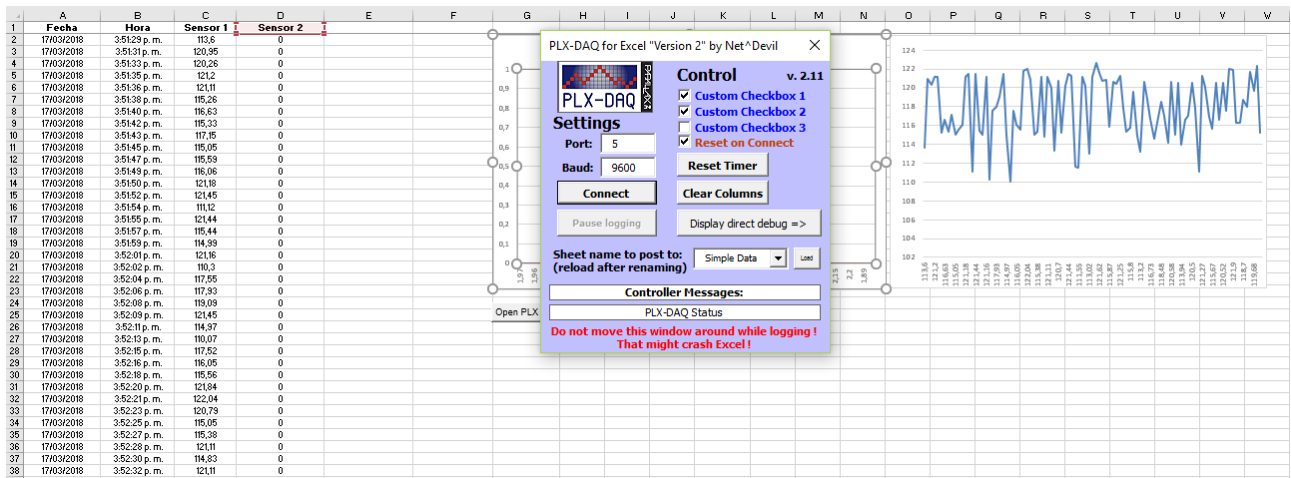


Figura 3.3. Base de datos con panel de control PLX-DAQ

Es de gran importancia mencionar que los gráficos que se muestran la figura 3.2 y la figura 3.3 son resultados de pruebas realizadas con cargas resistivas (bombillas) conectadas a una toma corriente. Donde el sensor 2, es el sensor de corriente y permanece en 0A durante un tiempo prudente para poder observar la oscilación de la tensión en la red.

3.2. Planteamiento con sensor de tensión, sensor de corriente y sensor de humedad.

Como se mencionó literales anteriores, uno de los objetivos de este sistema era su capacidad para poder acoplarse no solamente a variables eléctricas si no, también a variables ambientales, como la temperatura, la humedad o la presión, en este caso se añadió un sensor de humedad cuyas características están especificadas en el literal 2.3.3. La modificación realizada en el código e implementada para las pruebas se muestra a continuación:

```

1. #include <LiquidCrystal_I2C.h>
2. #include<Wire.h>
3. #include<math.h>
4. LiquidCrystal_I2C lcd(0x3F,16,2);
5. int aridez;
6. const int analogInPin = A0;
7. const int analogInPin1 = A1;
8. const int sensorPin = A2;
9. int retardo = 2 ; // (tiempo (s.) entre visionados)
10. float lectura,lectural,ff,pKW,iA,vV,vS,S_Ratio,Y;
11. long x;
12. int i=0;
13.
14. void setup() {
15.   lcd.init();
16.   lcd.backlight();
17.   lcd.clear();
18.   lcd.setCursor(0,0);
19.   lcd.print("P=      I= ");
20.   lcd.setCursor(0,1);
21.   lcd.print("V=      Ari=");
22.   Serial.begin(9600);
23.   S_Ratio = 36.5; // Sensor/ratio (mV/mA ) : 36.5
24.   vV = 120; // valor de tension a computar

```



```

25.     ff = 4.15; // freq. factor / (50Hz -> 5 / 60Hz -> 4.15)
26.     }
27.
28.     float smoothreadx(float fx){ // fc (factor corrector)
29.     int ni = 35; // n. de iteraciones => smooth
30.     // (ni) => rango 10 a 50 mejor promedio [smoothing]
31.     float retorno = 0.0;
32.     for (int x = 0; x< ni; x++){
33.     do { // espero paso por cero
34.         delayMicroseconds(100);
35.         } while (analogRead(1) != 0) ;
36.         delay (ff); // espera centro de ciclo
37.         delay (10); // estabilizacion CAD
38.         retorno = retorno +(analogRead(1)*fx);
39.     }
40.     return retorno / ni;
41.     }
42.
43.     float smoothread(float fc){ // fc (factor corrector)
44.     int ni = 35; // n. de iteraciones => smooth
45.     // (ni) => rango 10 a 50 mejor promedio [smoothing]
46.     float retorno = 0.0;
47.     for (int x = 0; x< ni; x++){
48.     do { // espero paso por cero
49.         delayMicroseconds(100);
50.         } while (analogRead(0) != 0) ;
51.         delay (ff); // espera centro de ciclo
52.         delay (10); // estabilizacion CAD
53.         retorno = retorno +(analogRead(0)*fc);
54.     }
55.     return retorno / ni;
56.
57.     }
58.
59.     void loop() {
60.
61.         lectura = smoothread (1) / 1.41; // lectura (rms)
62.         iA = (lectura * S_Ratio)*4.9259/1000-1; // Intensidad (A)
63.         if (iA<0.5){
64.             iA=0;
65.         }
66.         else {
67.             iA=iA;
68.         }
69.         lectural = smoothreadx (1) / 1.41; // lectura (rms)
70.         vV = (lectural * S_Ratio)*122*0.933435348/(1000*6.061739944);
// Tension (V)
71.
72.         pKW = (vV * iA); // Potencia (kW)
73.
74.         int humedad = analogRead(sensorPin);
75.         aridez= map(humedad,0,1023,0,100);
76.
77.
78.         Serial.println( (String) "DATA,DATE,TIME," + vV + "," + iA + "," +
aridez );
79.         lcd.setCursor(2,0);
80.         lcd.print(pKW);
81.
82.         lcd.setCursor(11,0);
83.         lcd.print(iA);

```

```

84.
85.     lcd.setCursor(2,1);
86.     lcd.print(vV);
87.
88.     lcd.setCursor(13,1);
89.     lcd.print(aridez);
90.     delay(retardo * 300);
91.     }

```

Como se observa las modificaciones son mínimas, pero de esta forma se logra garantizar el correcto funcionamiento de los sensores propuestos.

En las figuras 3.2 y 3.3 se aprecia que la base de datos en ese momento funciona y enseña las gráficas de solamente dos sensores, por lo tanto, para lograr una inclusión efectiva del sensor de humedad fue necesario modificar la base de datos para obtener un resultado óptimo de la aplicación.

3.3. Transferencia de datos por medio de protocolo de comunicación Xbee

Como se hizo referencia en [13] el protocolo de comunicación Xbee es un medio de transferencia de datos inalámbrica, que funciona con dos dispositivos, un emisor y un receptor de muy fácil manejo. Para llevar esta herramienta al proyecto se necesitó hacer una configuración por medio de un software llamado X-CTU, en este caso el dispositivo Xbee es de la serie 2.

Lo primero es descargar e instalar el programa X-CTU y conectar el Xbee que se configurará como receptor a la computadora por medio de USB, segundo, ejecutar el programa se abrirá la pantalla que se muestra en la figura 3.4.

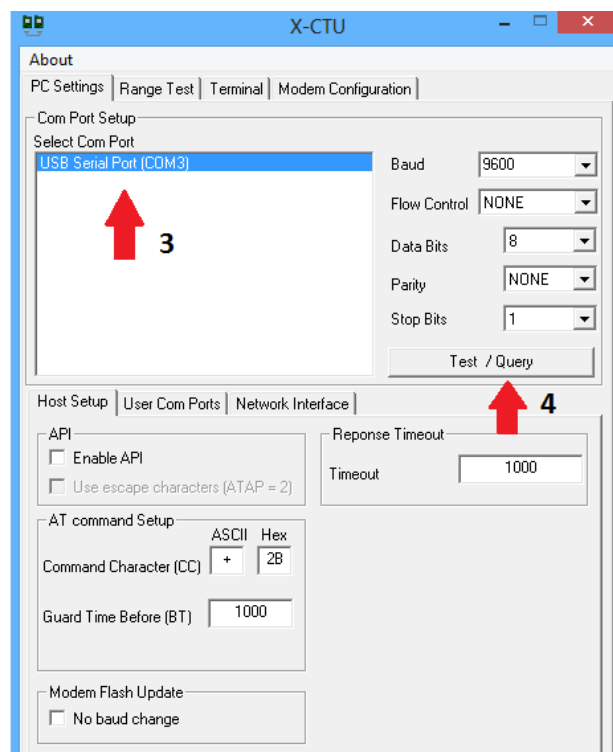


Figura 3.4. Configuración módulo Xbee.

En la figura 3.4 donde apunta la flecha 3 se muestra el puerto COM donde está conectado el dispositivo, el paso 4 es para realizar la prueba de que el módulo tiene un funcionamiento correcto. Para verificar esto se da click y esperar una respuesta de OK.

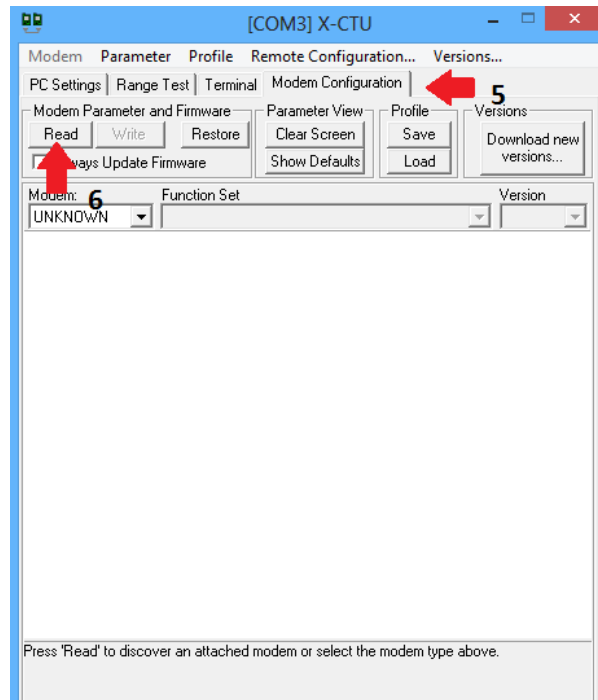


Figura 3.5. Paso 5 y paso 6.

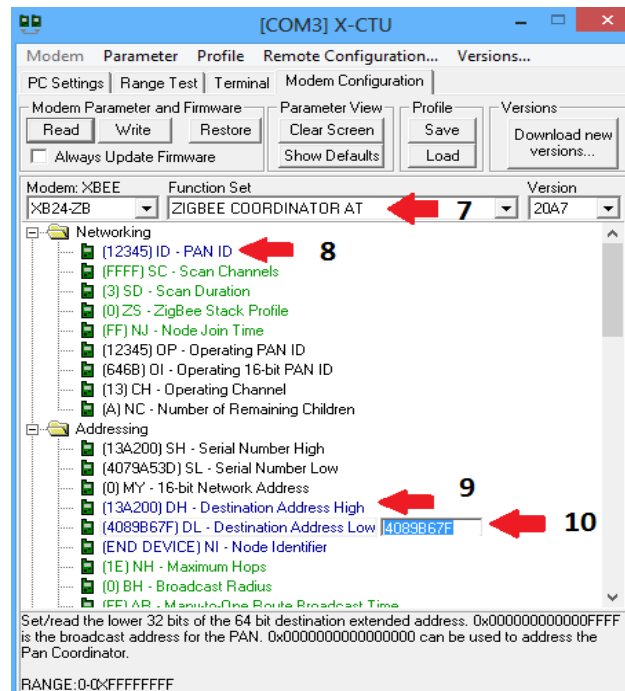


Figura 3.6. Pasos de configuración módulo Xbee como receptor

De esta forma se configura el módulo xbee como receptor, para configurarlo como emisor, basta con realizar unos pequeños cambios. Por ejemplo, en la figura 3.6 en la flecha número 7 en lugar de la opción ZigBee Coordinator AT se escoge la opción ZigBee end device AT. De esta forma se

configura el segundo módulo Xbee como emisor. La conexión del módulo Xbee emisor al arduino se observa en la figura 3.7.

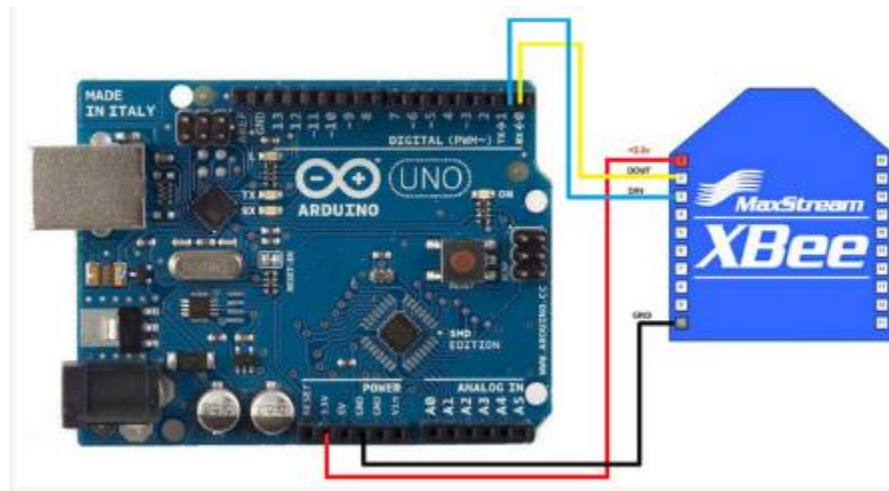


Figura 3.7. Conexión de Xbee emisor a arduino. [34]

3.4. Base de datos e interfaz de usuario

Por necesidad de obtener un buen funcionamiento de los tres sensores, fue necesario implementar ciertas modificaciones en la base de datos para que se visualice el comportamiento de las variables, de esta manera se hicieron algunos cambios en el código de Visual Basic para lograr conjuntamente una interfaz para que el usuario tenga mayor facilidad de visualización, análisis y configuración. El resultado se muestra en la figura 3.8.

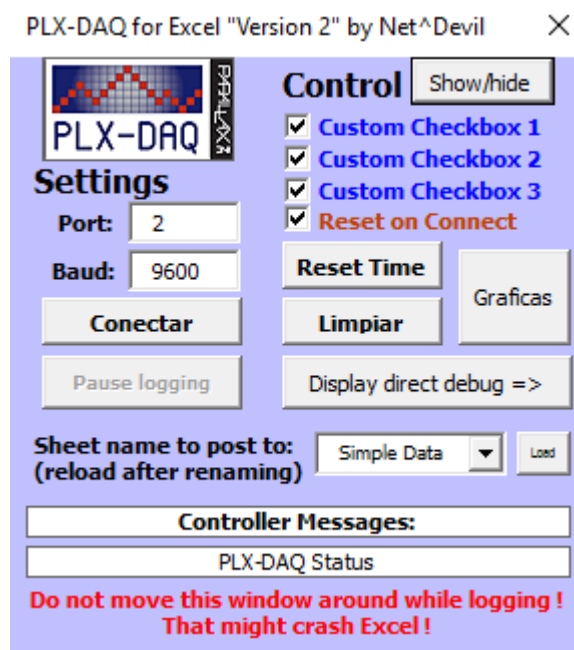


Figura 3.8. Panel de control de la interfaz de usuario y base de datos.

Diferenciando el panel mostrado en la figura 3.3. el manejo tanto de la base de datos como de las gráficas de análisis. Los botones agregados con el nombre de show/hide y graficas hacen muy sencillo la manipulación de los datos.

Una explicación sencilla del uso puede ser la siguiente:

Paso 1. Verificar en que puerto COM está conectado el receptor de datos, esto se puede comprobar entrando a la opción Equipo, hacer click derecho y escoger la opción administrar, inmediatamente se abrirá una ventana llamada administrador de dispositivos, luego en el listado buscar puerto COM_LPT, ahí se podrá verificar cual es el puerto COM en donde se encuentra el emisor. En la figura 3.8 muestra como ejemplo el puerto 2 (Port 2) y cambiar de ser necesario su puerto.

Paso 2. Al presionar el botón conectar de forma inmediata hará conexión con el arduino y se registrará en la base de datos lo que se esté midiendo.

Paso 3. Para verificar las mediciones en base de datos bastara con presionar el botón show/hide y este de forma rápida enviara al usuario a la base de datos de Excel mostrando los datos tomados.

Paso 4. Si solamente se desea observar el comportamiento de las gráficas el botón “Graficas” muestra las 3 graficas en tiempo real de esa forma se puede detallar el comportamiento de las variables.

Si se desea limpiar la lista de datos guardados y limpiar las gráficas se deben presionar los botones “limpiar” y “reset time” de esta forma se iniciará desde cero.

De esta forma el usuario tendrá un acceso fácil tanto a la base de datos como a la visualización de las variables de forma gráfica.

En conclusión, la base de datos final se muestra a continuación junto con resultados de las últimas pruebas realizadas el día 4 de abril del año en curso.

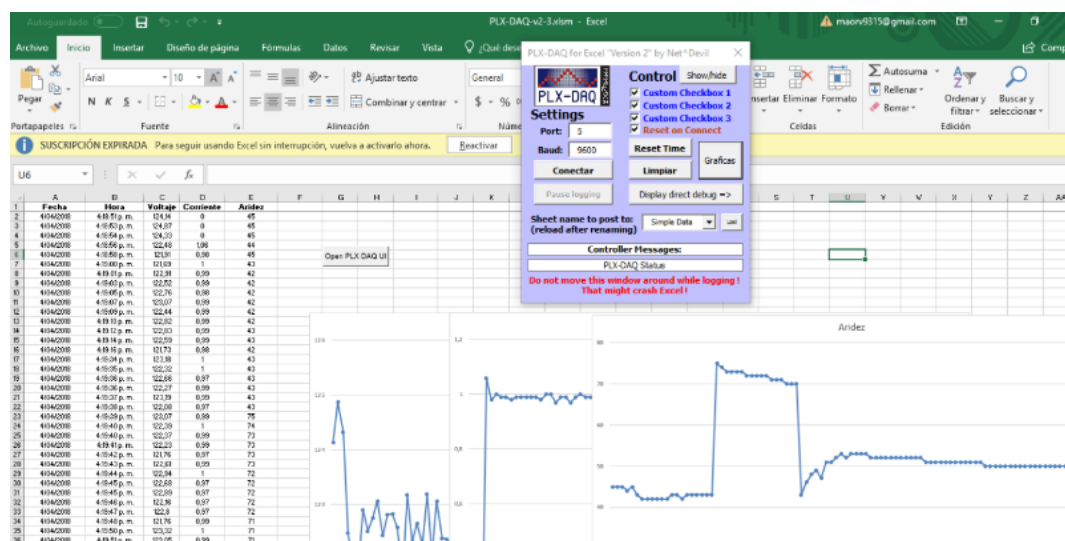


Figura 3.9. Se indica la base de datos y la interfaz completa. Datos tomados el 04/04/2018

La interfaz de la base de datos tuvo como objetivo ser más amigable y de fácil manejo, de tal forma que el usuario tenga la alternativa tanto de visualizar la toma de medidas en tiempo real o ver su

comportamiento gráficamente sin necesidad de ingresar a Excel, los resultados se muestran se presentan en las siguientes figuras.

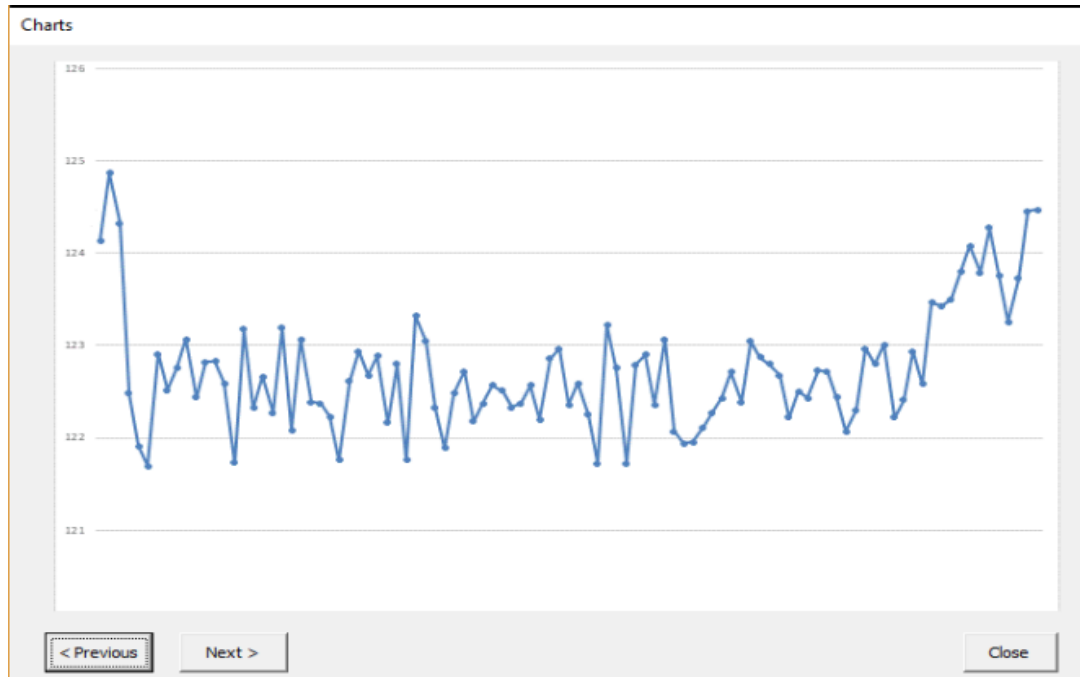


Figura 3.10. Se muestra el análisis grafico de las medidas de voltaje. Datos del 04/04/2018

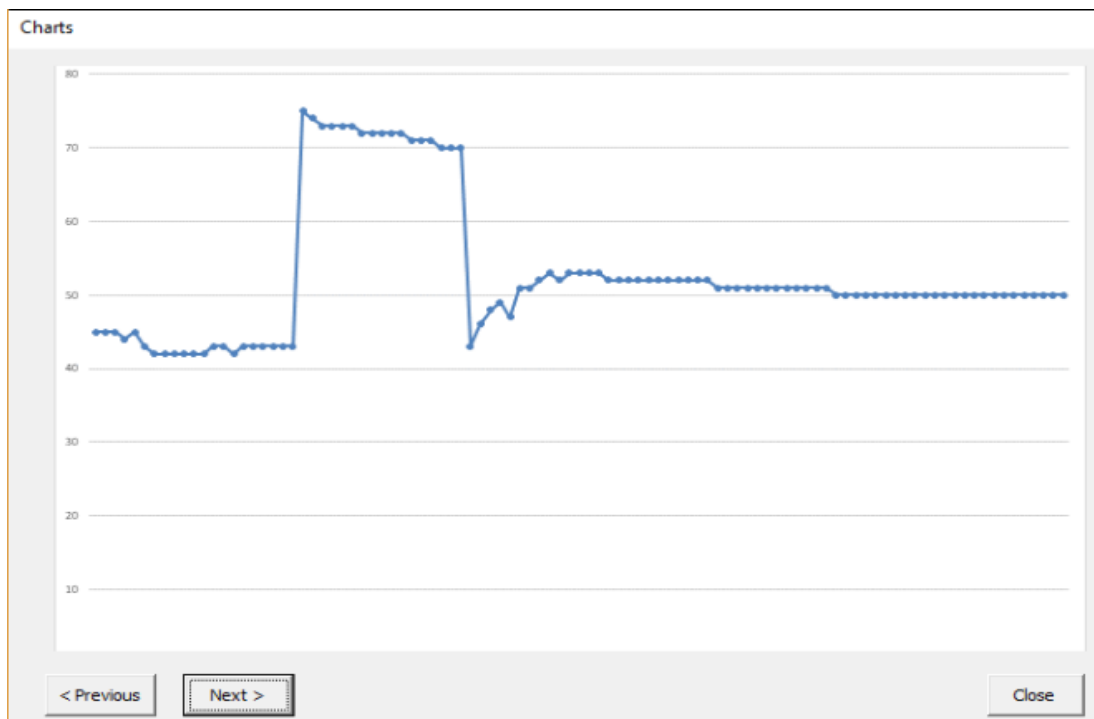


Figura 3.11. Se muestra el análisis grafico de las medidas de aridez, medidas en porcentaje. Datos del 04/04/2018.

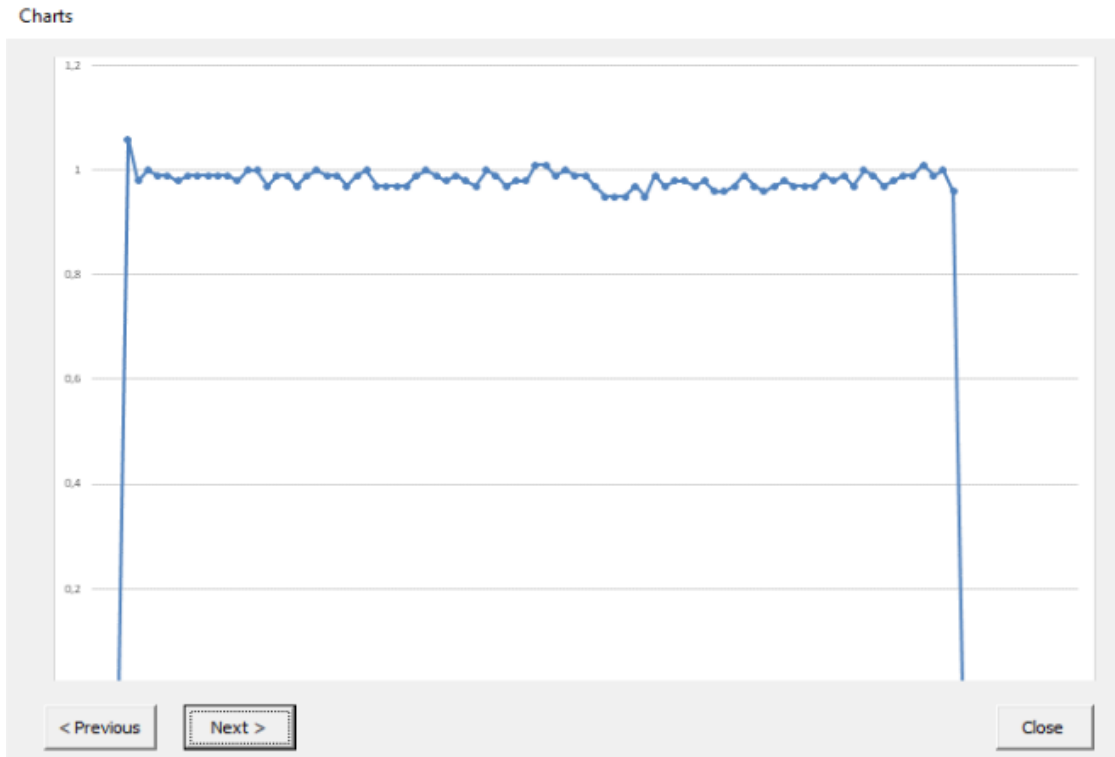


Figura 3.12. Se muestra el análisis gráfico de las medidas de corriente. Datos del 04/04/2018.

Una de las ventajas de esta interfaz es que el usuario podrá desplazar las gráficas por medio de los botones “previous” y “next” para revisar la anterior y la siguiente, respectivamente.

A manera de análisis de los resultados obtenidos en las gráficas se podría decir que:

En la figura 3.10. en la que se relaciona el voltaje se observa la oscilación proveniente de la red, que tiene su tope en aproximadamente 125V y se reduce a aproximadamente 122.8V, en varias de las pruebas realizadas se pudo comprobar que esta oscilación es constante e incluso llega a disminuir hasta 115V aproximadamente, en este caso se mantuvo por encima de la tensión nominal en una red que es de 120V.

En la figura 3.11. en la que se muestra los resultados de medición de aridez se nota la variación del porcentaje de aridez en el momento en el cual el higrómetro es humedecido por medio de una servilleta mojada, los resultados varían entre el 35% y el 32%. En el momento que se quita el elemento húmedo la aridez incrementa a un 75% más o menos, de esta manera se puede controlar la humedad en plantaciones de ser necesario.

Y por último en la figura 3.12. el análisis de la medición de la corriente se puede comprobar su iniciación en 0A, esto debido a que en el instante de iniciar la prueba no se había conectado ninguna carga, en el momento en el que se conecta un banco de bombillas inmediatamente la magnitud aumenta a 1.2A aproximadamente, de forma instantánea empieza a oscilar entre 1A y 0.95A. Se puede asociar este cambio a las oscilaciones de tensión que presenta la red.

En la siguiente imagen se observa la conexión física de los dispositivos usados para el montaje.

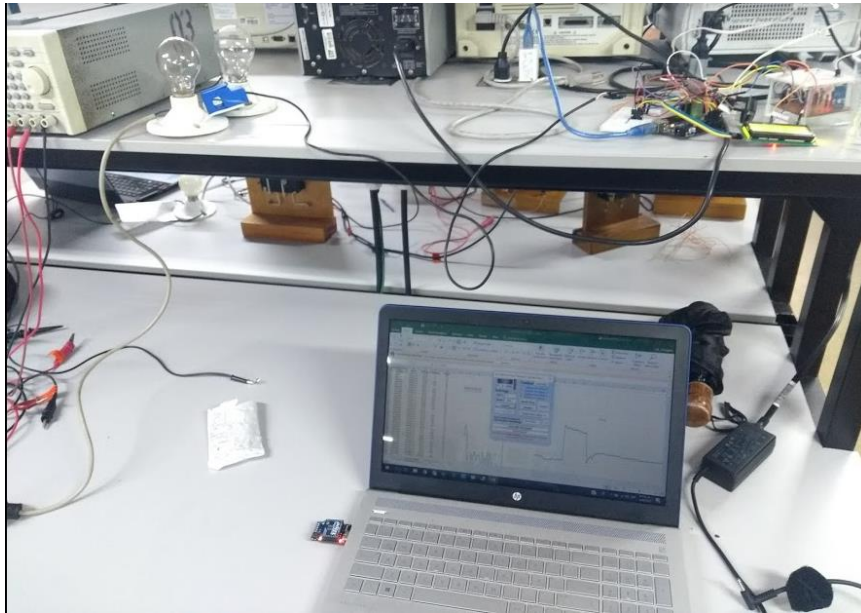


Imagen 3.1. Se muestra la conexión física de los componentes de la implementación.

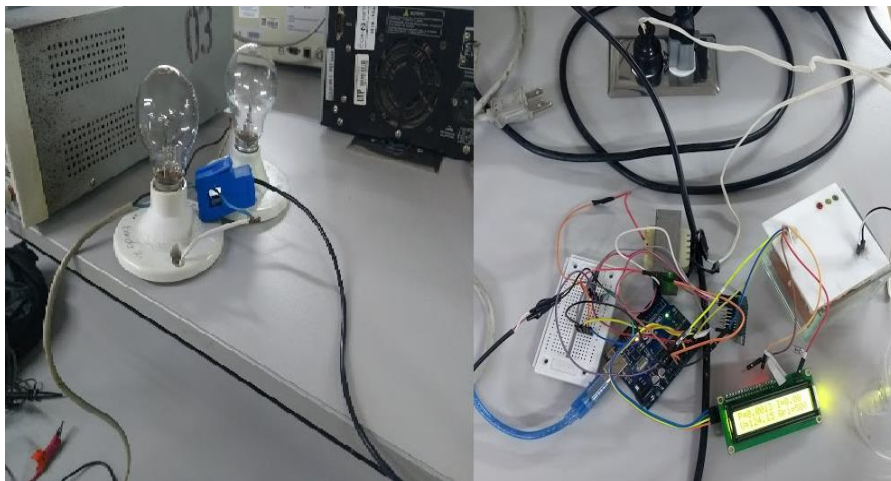


Imagen 3.2 Conexión de sensores a arduino.

Capítulo 4

Conclusiones, recomendaciones y trabajos futuros

- Las plataformas de código abierto como lo es Arduino provoca una revolución en la tecnología de la actualidad, debido a sus diversas aplicaciones y funcionalidades, haciendo accesibles dispositivos para mejorar la calidad de vida y disminuir la afectación al medio ambiente.
- El sistema implementado puede ser usado como un dispositivo de control y monitoreo de consumo de energía eléctrica en hogares, siempre y cuando en la red domiciliaria predominen las cargas resistivas o cargas con factor de potencia igual a 1.
- El protocolo de comunicación Xbee posee características que para este tipo de aplicaciones son de gran ventaja, debido a su capacidad de transportar alto tráfico de datos en poco tiempo, además de manejar una baja latencia lo que hace que la visualización de los datos sea prácticamente en tiempo real.
- La base de datos se puede considerar como una de las mejores formas de presentar datos valga la redundancia, para lograr un análisis fácil y detallado del comportamiento de las variables de estudio.
- Si alguien decidiera enfocarse en realizar una investigación sobre aplicativos de medición de energía eléctrica en tiempo real podría tener como base este proyecto, gracias a la gran precisión obtenida en las pruebas con los sensores de corriente y tensión. Una recomendación sería explorar en el mercado una tarjeta capaz de tener un tiempo de muestreo superior, debido a que, si el enfoque es lograr una medición precisa en cualquier red domiciliaria, se tiene que tener en cuenta la cantidad de armónicos que son inyectados por la tecnología LED.
- Al realizar pruebas con el transformador de tensión se pudo comprobar la constante oscilación de la red, esto en algunos casos puede ser perjudicial para los elementos conectados a esta, debido a los transitorios y sobretensiones causando un mal funcionamiento incluso dañándolos.

Bibliografía

- [1] Energía y Sociedad. Redes eléctricas inteligentes, 2010, [En línea]. Disponible en: <http://www.energiaysociedad.es/wp-content/uploads/pdf/smartgrids.pdf>
- [2] El país, “En el siglo XXI convertirse en una ciudad inteligente es la única opción”, 2017, [En línea]. Disponible en: https://elpais.com/tecnologia/2017/11/22/actualidad/1511370108_620353.html
- [3] M. V. Herrera. “Descripción de redes inteligentes (Smart Grids) y su aplicación en sistemas de distribución eléctrica,” Tesis de pregrado, Escuela Politecnica Nacional, Facultad de ingeniería eléctrica y electrónica, Quito, Ecuador, 2013. [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/5542>
- [4] Wu, Z., Zhao, T., He, L., & Shen, X. (2014). Smart grid meter analytics for revenue protection. In *POWERCON 2014 - 2014 International Conference on Power System Technology: Towards Green, Efficient and Smart Power System, Proceedings* (pp. 782–787). <https://doi.org/10.1109/POWERCON.2014.6993869>
- [5] Zhou, S., Wu, Z., Li, J., & Zhang, X. (2014). Real-time Energy Control Approach for Smart Home Energy Management System. *Electric Power Components and Systems*, 42(3–4), 315–326. <https://doi.org/10.1080/15325008.2013.862322>
- [6] CASTAÑO JARAMILLO, N., FRANCO CARDONA, C. J., & VELÁSQUEZ HENAO, J. D. (2014). SMART METERS ADOPTION: RECENT ADVANCES AND FUTURE TRENDS. *DYNA*, 81(183), 221. <https://doi.org/10.15446/dyna.v81n183.38148>
- [7] Energía y Sociedad, “Redes eléctricas inteligentes”, 2010, [En línea]. Disponible en: <http://www.energiaysociedad.es/>
- [8] Ministerio de energía y minería argentina, O. Medina. [En línea]. Disponible en: <http://www.energia.gov.ar/>
- [9] J.R.Gómez, P.R.Viego. Redes eléctricas inteligentes. Universidad de Cienfuegos, Cuba. [En línea]. Disponible en: <http://www.cubasolar.cu/biblioteca/Energia/Energia62/HTML/Articulo06.htm>
- [10] Xatakahome, Redes PLC: Que son y para qué sirven. 2013, [En línea]. Disponible en: <https://www.xatakahome.com/la-red-local/redes-plc-i-que-son-y-para-que-sirven>
- [11] Aula clic, Artículo 5: WiFi. La comunicación inalámbrica. [En línea]. Disponible en: <https://www.aulaclic.es/articulos/wifi.html>
- [12] EcuRed, ZigBee, [En línea]. Disponible en: <https://www.ecured.cu/ZigBee>
- [13] Xbee, ¿Qué es Xbee?, [En línea]. Disponible en: <http://xbee.cl/que-es-xbee/>

- [14] Arduino, What is Arduino?, [En línea]. Disponible en: <https://www.arduino.cc/en/Guide/Introduction>
- [15] Wikipedia, Arduino, 2018, [En línea]. Disponible en: https://es.wikipedia.org/wiki/Arduino#Lenguaje_de_programaci.C3.B3n_Arduino
- [16] BricoGeek, Arduino Uno, [En línea]. Disponible en: <http://tienda.bricogeek.com/arduino/305-arduino-uno-0805833349009.html>
- [17] Blogspot, Arduino en español, 2016, [En línea]. Disponible en: <http://manueldelgadocrespo.blogspot.com.co/p/arduino-mega-2560.html>
- [18] Arduino, Intel Galileo, [En línea]. Disponible en: <https://www.arduino.cc/en/ArduinoCertified/IntelGalileo>
- [19] Hetpro, Beaglebone Black: Introduccion, 2015, [En línea]. Disponible en: <https://hetpro-store.com/TUTORIALES/beaglebone-black-introduccion/>
- [20] Beagleboard.org, BeagleBone Black, 2017, [En línea]. Disponible en: <https://beagleboard.org/black>
- [21] Makezine, BeagleBone Blue, 2017, [En línea]. Disponible en: <https://makezine.com/product-review/beaglebone-blue/>
- [22] Wikipedia, Raspberry Pi, 2018, [En línea]. Disponible en: https://es.wikipedia.org/wiki/Raspberry_Pi
- [23] ABC, ¿Qué es Raspberry Pi y para que se usa?, 2013, [En línea]. Disponible en: <http://www.abc.es/tecnologia/informatica-hardware/20130716/abci-raspberry-como-201307151936.html>
- [24] Naylamp Mechatronics, Tutorial sensor de corriente AC no invasivo STC-013, [En línea]. Disponible en: http://www.naylampmechatronics.com/blog/51_tutorial-sensor-de-corriente-ac-no-invasivo-s.html
- [25] Luis Llamas, Sensor de corriente electrica no invasivo con arduino y STC 013, 2017, [En línea]. Disponible en: <https://www.luisllamas.es/arduino-sensor-corriente-sct-013/>
- [26] Patricio Concha, Transformadores de medida, [En línea]. Disponible en: http://patricioconcha.ubb.cl/transformadores/transformadores_de_medida.htm
- [27] Luis Llamas, Medir la humedad del suelo con arduino e higrómetro FC-28, 2016, [En línea]. Disponible en: <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>
- [28] Madness electronics, Tutorial sensor de humedad de suelo, 2016, [En línea]. Disponible en: <http://www.madnesselectronics.com/tutorial-sensor-de-humedad-de-suelo/>
- [29] Brunetica, Sensor humedad suelo FC-28, 2013, [En línea]. Disponible en: <https://brunetica.wordpress.com/2013/11/21/sensor-humedad-suelo-fc-28-c/>

[30] Data Virtualizer, Popularity vs Productivity vs Performance, 2013, [En línea]. Disponible en: <http://datavirtualizer.com/popularity-vs-productivity-vs-performance/>

[31] Wikipedia, C (lenguaje de programación), 2018, [En línea]. Disponible en: https://es.wikipedia.org/wiki/C_%28lenguaje_de_programaci%C3%B3n%29

[32] Wikipedia, Python, 2018, [En línea]. Disponible en: <https://es.wikipedia.org/wiki/Python>

[33] Cuatro ríos, Principales características del lenguaje Python, [En línea]. Disponible en: http://www.cuatrorios.org/index.php?option=com_content&view=article&id=161:principales-caracteristicas-del-lenguaje-python&catid=39:blogsfeeds

[34] Forum arduino. Señales análogas con xbee, 2015. [En línea]. Disponible en: <https://forum.arduino.cc/index.php?topic=313720.0>

