



“Responsabilidad con pensamiento positivo”

UNIVERSIDAD TECNOLÓGICA ISRAEL

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:
INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES**

TEMA:

**PROTOTIPO DE SISTEMA DE MONITOREO Y CONTROL DE CONSUMO DE
ENERGÍA ELÉCTRICA PARA UN DOMICILIO APLICANDO EL CONCEPTO DE
INTERNET DE LAS COSAS.**

AUTOR:

Sánchez Oñate Ricardo Mauricio

TUTOR:

PhD Millard Escalona

QUITO, ECUADOR

2020

DECLARACIÓN

Yo, Ricardo Mauricio Sánchez Oñate, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, perteneciente a la Universidad Tecnológica Israel, declaro que el contenido aquí descrito es de mi autoría y de mi absoluta responsabilidad legal.

Quito D.M., marzo 2019

Ricardo Mauricio Sánchez Oñate

C.I.: 1716905060

AGRADECIMIENTO

Mi agradecimiento primero que nada a Dios, a mi familia mis padres y todos los que han aportado con un granito de arena, que con su apoyo incondicional ha hecho posible alcanzar mis metas y llegar a estas instancias. Este logro es fruto de su sacrificio, es una forma de demostrarle mi gratitud.

Ricardo Mauricio Sánchez Oñate

DEDICATORIA

Este trabajo es dedicado a mi familia mis pequeños Kevin y Francisco mis dos tesoros, mi esposa Alicia el soporte de mi vida, mis padres que con sus palabras de aliento no me han dejado desfallecer desde el inicio de mi vida estudiantil hasta el día de hoy que se ha conseguido un logro más a nivel académico y personal, no me queda más que dedicarle el presente, que más que un reto ha sido un trabajo prácticamente interminable, como querer llegar a la cima de una montaña, muchas veces ya no se divisa la cumbre, solo queda mirar los pies y caminar con las palabras de cariño y aliento de las personas que nos rodean y aman.

Ricardo Sánchez

RESUMEN

El desarrollo del presente proyecto tiene la finalidad de brindar una herramienta para el monitoreo del consumo de energía a nivel domiciliario basado en el concepto de la internet de las cosas, utilizando las diferentes tecnologías y plataformas disponibles, para obtener información y publicarlo en la nube, pudiendo utilizarla de diferentes maneras y principalmente para el beneficio de la comunidad como también ahorrar este recurso no renovable tanpreciado, es así que el sistema de monitoreo permite realizar medidas del consumo de energía de la acometida como también a tres aparatos electrónicos mediante el uso de sensores. La comunicación de los sensores se lo realiza en una red inalámbrica que trabaja en la frecuencia 2.4GHz para esto se utiliza el *transceiver* NRF 24 L01+, para configuración del nodo central se utilizó un Raspberry pi Zero, encargada de recolectar la información acumulada en los sensores y cada hora publicarla en la nube (*ThingSpeak*). En una aplicación para sistema Android, se accede a los datos gráficamente, la aplicación también indicará, si se encuentra conectada una carga siempre y cuando esta consuma una potencia mayor a 5[W], como también desconectarla de la toma eléctrica. Por último, el sistema de monitoreo y control de consumo de energía también presenta datos reales de consumo, que luego de analizar la información se ubica el aparato de mayor consumo con relación al total, como también determinar si algún equipo, electrodoméstico está averiado al presentar un comportamiento diferente en su consumo de energía.

Palabras clave; monitoreo, IoT, comunicación, inalámbrica, sensores, NRF 24L01+, Arduino, Raspberry, nube, potencia, energía.

ABSTRACT

The development of this project aims to provide a tool for monitoring energy consumption at the household level based on the concept of the Internet of Things, using the different technologies and platforms available, to obtain information and publish it in the cloud, Being able to use it in different ways and mainly for the benefit of the community as well as saving this precious non-renewable resource, so the monitoring system allows measurements of the power consumption of the connection as well as three electronic devices through the use of sensors. The communication of the sensors is carried out in a wireless network that works on the 2.4GHz frequency, for this the NRF 24 L01 + transceiver is used, for the configuration of the central node a Raspberry pi Zero was used, in charge of collecting the information accumulated in the sensors and every hour publish it in the cloud (ThingSpeak). In an application for Android system, the data is accessed graphically, the application will also indicate, if a load is connected if it consumes a power greater than 5 [W], as well as disconnect it from the electrical outlet. Finally, the monitoring and control system for energy consumption also presents real consumption data, which after analyzing the information, the device with the highest consumption in relation to the total is located, as well as determining if any equipment or appliance is damaged by presenting a different behavior in their energy consumption.

Key words; monitoring, IoT, communication, wireless, sensors, Nrf24l01 +, Arduino, Raspberry, cloud, power, energy.

ÍNDICE

INTRODUCCIÓN.....	1
1. ANTECEDENTES DE LA SITUACIÓN OBJETO DE ESTUDIO	1
2. PROBLEMA DE INVESTIGACIÓN: PRESENTACIÓN Y JUSTIFICACIÓN 3	
3. OBJETIVOS	3
a. Objetivo general:.....	3
b. Objetivos específicos:	4
4. Descripción de los capítulos.....	4
CAPÍTULO 1	0
ENERGÍA ELÉCTRICA.....	0
1.1. Energía eléctrica como producto	0
1.1.1. Cálculo de la potencia.	1
1.2. Origen de la electricidad	3
1.3. Sistemas para la lectura de energía eléctrica.....	5
1.3.1. Contador electromagnético.	6
1.3.2. Contador electrónico	7
1.3.3. Pliego tarifario residencial.	8
1.4. Electrónica del proyecto.	8
1.4.1. Fuentes de alimentación	9
1.4.2. Relé- Relevador.....	10
1.4.3. Arduino	11
1.4.4. Entradas analógicas	13
1.4.5. Precisión de la medición	13
1.4.6. Precisión relativa	13
1.4.7. Referencia de tensión analógica (AREF).....	14
1.4.8. Convertidor Análogo / Digital	14
1.5. Sensores	15
1.5.1. Sensor de corriente SCT 013.....	15
1.5.2. Transformador de voltaje AC - ZMPT101B.....	17
1.5.3. Raspberry Pi	18
1.6. Redes de comunicación de datos	21
1.6.1. Internet.	21
1.6.2. Internet de las cosas IoT.....	21
1.6.3. Plataformas IoT	22

1.6.4.	Red de comunicación de datos.....	23
1.7.	Tipo de redes inalámbricas	28
1.7.1.	Modo infraestructura	28
1.7.2.	Modo Ad hoc.....	28
1.7.3.	Redes Wifi.....	29
1.7.4.	Tecnología Zigbee.....	30
1.8.	Tipos de dispositivos zigbee	31
	CAPÍTULO 2	32
	MARCO METODOLÓGICO	32
2.	Metodología documental	32
2.1.	Método experimental e inductivo	32
	CAPÍTULO 3	34
3.	LA PROPUESTA	34
3.1.	Módulos inalámbricos para la toma de medidas de voltaje y corriente.....	35
3.1.1.	Técnica de medición	37
3.1.2.	Valor RMS o eficaz.....	37
3.1.3.	Medición de voltaje.....	38
3.1.4.	Medidor de corriente.....	38
3.1.5.	Procesador de datos.....	39
3.1.6.	Arduino nano.....	40
3.1.7.	Arduino IDE.....	40
3.2.	Implementación de una red de sensores, en topología estrella.	41
3.2.1.	Características de los nodos remotos	41
3.2.2.	Características del nodo central	42
3.2.3.	Comunicación Inalámbrica.	43
3.2.4.	Módulo Transceiver NRF 24L01	43
3.2.5.	Protocolo de comunicación SPI	45
3.2.6.	Algunas otras características	46
3.2.7.	Canales de recepción y paquetes de datos.....	46
3.3.	Plataforma IoT y su comunicación	47
3.3.1.	Visualización de datos de consumo de energía.....	49
3.3.2.	Herramienta de desarrollo de aplicación para Android.....	49
3.3.3.	Diseño de la aplicación observar datos	52
3.4.	Control de Consumo del Sistema IoT basado en la Lectura de Factor de Potencia	54

3.4.1.	Planteamiento del Esquema de Control	54
3.5.	Automatización del sistema	57
3.5.1.	Cron.....	57
3.5.2.	Crontab.....	57
3.6.	Diagrama de la maqueta para pruebas	59
	CAPÍTULO 4	62
4.1	Diseño e implementación de la tarjeta electrónica del módulo de energía principal.....	62
4.2.	Diseño e implementación de la tarjeta electrónica de los módulos de energía remotos.	64
4.2.1.	Implementación de software del prototipo de adquisición de datos.	71
4.2.2.	Implementación de Software del Prototipo para Acometida:	73
4.3.	Calibración.....	74
4.3.1.	Calibración del sensor de corriente.	75
4.3.1.	Calibración del sensor de voltaje	75
4.3.2.	Cálculo de potencia y energía	79
4.4.	Configuración de la red de nodos.	81
4.5.	Comunicación Inalámbrica.	83
4.5.1.	Instalar y configurar de las librerías requeridas para el control de NRF 24L01+ en Raspberry pi.....	84
4.5.2.	Instalar y configurar las librerías requeridas para el control de GPIO.....	85
4.5.3.	Ejecución del programa.....	85
4.5.4.	Instalación y configuración de librerías NRF 24L01+ en Arduino.....	87
4.5.5.	Ejecución del Programa.	87
4.5.6.	Test de comunicación bidireccional.....	88
4.6.	Recepción y envío de datos desde Arduino a Raspberry y viceversa.....	89
4.7.	Configuración del servidor de la nube.....	90
4.8.	Configuración de la comunicación entre el nodo central y el servidor.....	92
4.9.	Elaboración de la aplicación	93
4.9.1.	Pantalla de Inicio de sesión	93
4.9.2.	Pantalla para visualización de dispositivos conectados.	94
4.9.3.	Pantalla para visualización de datos de consumo de energía.....	97
4.10.	Configuración del Crontab.....	98
4.11.	Análisis de los resultados medidos	100
4.11.1.	Medidor de energía 1	100
4.11.2.	Medidor de energía 2.....	102

4.11.3.	Medidor de energía 3.....	104
4.11.4.	Medidor de energía 4 Acometida.....	106
4.12.	Diagrama de bloques total del sistema.....	109
CONCLUSIONES.....		110
RECOMENDACIONES		111
REFERENCIAS BIBLIOGRÁFICAS		112
ANEXOS.....		115
ANEXO A		116
MANUAL DE USUARIO		116
ANEXO B		125
MANUAL DEL TÉCNICO		125
ANEXO C		139
ANEXO D		142
ANEXO E.....		143

ÍNDICE DE FIGURAS

Figura. 0.1. Potencia aparente	6
Figura. 0.2. Triangulo de potencias	7
Figura. 0.3. Esquema generación y transporte energía	9
Figura. 0.4. Diagrama contador electromagnético	11
Figura. 0.5. Contador electrónico.	12
Figura. 0.6. Fuente lineal	14
Figura. 0.7. Fuente conmutada	15
Figura 0.8 Esquema de un relé	15
Figura. 0.9. Tipos placas de Arduino.....	16
Figura. 0.10. Convertidor analógico	19
Figura. 0.11. Los sensores emulan los sentidos	20
Figura. 0.12. Sensor de corriente no invasivo	21
Figura. 0.13. Sensor de corriente no invasivo esquema eléctrico.....	21
Figura 0.14 Transformado de voltaje	22
Figura. 0.15. Raspberry Pi 3 Modelo B.....	24
Figura. 0.16. Raspberry Pi Zero W	25
Figura. 0.17. ThingSpeak	27
Figura. 0.18. Topología en anillo	29
Figura 0.19 Topología bus	30
Figura 0.201 Topología tipo árbol.....	30
Figura 0.21 Topología tipo malla	30
Figura. 0.32. Cable de red categoría 5	31
Figura. 0.43. Espectro electromagnético bandas comerciales	32
Figura. 0.54. Modo infraestructura	33
Figura. 0.25. Modo Ad-hoc	34
Figura. 0.26. Dispositivos Zigbee	36
Figura. 0.6. Arquitectura del prototipo.	39
Figura. 0.7. Diseño del módulo transmisor, receptor	40
Figura. 0.8. Diseño del circuito de control relé	41
Figura. 0.9. <i>Valor eficaz de una señal sinodal</i>	42
Figura. 0.10. Diagrama sensor ZMPT101B	43
Figura. 0.11. Esquema de medición de voltaje	44

Figura. 0.12. Arduino Nano	45
Figura. 0.13. Módulo transceiver NRF 24L01	48
Figura 0.914 Estructura principal del protocolo PSI	50
Figura. 0.150. Recepción de información por varios canales.	51
Figura. 0.11. Representación del paquete de datos.	52
Figura. 0.12. Diagrama de dispositivos conectados a ThingSpeak	53
Figura. 0.13. Logo de página web para el desarrollo de APP	54
Figura. 0.14. Interfaz de APP Inventor para elegir los componentes	55
Figura. 0.165. Interfaz de APP Inventor diagrama de bloques	56
Figura. 0.176. Interfaz de APP Inventor simulador.....	56
Figura. 0.187. Diseño pantalla 1 ingreso del usuario	57
Figura. 0.198. Diseño pantalla 2 estado de dispositivo de monitoreo	58
Figura. 0.19.20 Diseño Screen 3 estado de presentación de datos	58
Figura. 3.20. Pliego tarifario factor de potencia	59
Figura. 3.21. triangulo de potencias	60
Figura. 0.22. Comando para iniciar el Crontab	63
Figura. 0.23. Editor de texto Crontab	63
Figura. 3.34. Diagrama de la maqueta	64
Figura. 0.25. Esquema de distribución de módulos de monitoreo	65
Figura. 0.26. Esquema de distribución de módulos de monitoreo	66
Figura. 0.21. Diagrama en tercera dimisión tarjeta integradora módulos secundarios.	67
Figura. 0.22. Diagrama de la electrónica de la placa PCB modulo principal	68
Figura 0.23 Diseño de ruteo de pistas tarjeta integradora	69
Figura. 0.24. Diagrama en tercera dimensión de la tarjeta integradora, módulos secundarios.	70
Figura. 0.25. Diagrama de la electrónica de la placa PCB	70
Figura. 0.26. Diseño de ruteo de pistas tarjeta integradora	71
Figura. 0.27. Impresión en baquelita mediante calor.	72
Figura. 0.28. Proceso químico retiro de cobre	72
Figura. 0.29. Baquelita lista para la soldadura de elementos	73
Figura. 0.30. Baquelita lista para la soldadura de elementos	73
Figura. 0.31. Diseño de ruteo de pistas tarjeta integradora	73
Figura. 0.32. Diseño de ruteo de pistas tarjeta integradora	73
Figura. 0.33. Implementación del módulo principal.	74

Figura. 0.34. Módulo principal con visualizador de datos.	74
Figura. 0.35. Módulo remoto	74
Figura. 0.36. Diagrama de bloques módulos remotos	75
Figura. 0.37. Librería “EmonLib” en Arduino.	77
Figura. 0.38. Diagrama de bloques medidor de energía acometida	77
Figura. 0.39. Diagrama para la conexión de los sensores de corriente y voltaje	78
Figura. 0.40. Código de programación, calibración de sensores	80
Figura. 0.41. Medida de la corriente en miliamperios.	81
Figura. 0.42. Medida de voltaje.	81
Figura. 0.43. Parámetros obtenidos con el sensor de medida, monitor serial Arduino IDE.	81
Figura. 0.44. Código de programación, cálculo de potencia real.	83
Figura. 0.45. Código de programación, cálculo de energía.	84
Figura. 0.46. Presentación de datos Arduino IDE	84
Figura. 0.47. Valores medidos de potencia almacenado en el servidor IoT.	85
Figura. 0.48. Conexión del módulo NRF 24L01+a la Raspberry pi.	86
Figura. 0.49. Disposición de pines módulos NRF 24L01+	86
Figura. 0.50. Instalación de librería de comunicación.....	88
Figura. 0.51. Instalación librerías GPIO	89
Figura. 0.52. Librería en Python para NRF 24	90
Figura. 0.53. Gestor de librerías Arduino	91
Figura. 0.54. Ejecución de la librería NRF 24L01	92
Figura. 0.55. Pruebas de conexión módulos inalámbricos	93
Figura. 0.56. Algoritmo de programación envío y recepción de datos	93
Figura. 0.57. Algoritmo de envío de datos al servidor	94
Figura. 0.58. Ingreso a la cuenta base de datos	95
Figura. 0.59. Campos De Almacenamiento	95
Figura. 0.60. Configuración de librerías para acceder a los datos en la nube	96
Figura. 0.61. Pantalla de inicio	97
Figura. 0.62. Diagrama de bloques de la programación pantalla principal	98
Figura. 0.63. Estado de switch de cada dispositivo	99
Figura. 0.64. Lectura de estados en los dispositivos de medida	99
Figura. 0.65. Estado de relés	100
Figura. 0.66. Tercera pantalla visualización de datos	101

Figura. 0.67. Diagrama de bloques para la visualización de datos	102
Figura. 0.68. Configuración Crontab	103
Figura. 0.69. Maqueta con los sensores de adquisición de datos	103
Figura. 0.50. Comportamiento del consumo de energía módulo 1	106
Figura. 0.51. Comportamiento del consumo de energía módulo 2	108
Figura. 0.52. Comportamiento del consumo de energía módulo 3	110
Figura. 4.53. Consumo de energía módulo 4 acometida	112

ÍNDICE DE TABLAS

Tabla. 1.1. Proyectos energéticos del Ecuador.....	10
Tabla. 1.2. Cargos tarifarios Quito	13
Tabla. 1.3. Cuadro comparativo Arduino	17
Tabla. 1.4. Valores de tensión de referencia Arduino	19
Tabla. 1.5. Valores de tensión de referencia del circuito de acoplamiento	23
Tabla. 1.6. Propiedades del transformador	23
Tabla. 1.7. Características Raspberry pi Zero W	25
Tabla. 1.8. Parámetros del protocolo Zigbee	36
Tabla. 3.1. Características técnicas transceiver.....	49
Tabla. 3.2. Librerías de red NRF 24L01.....	49
Tabla. 4.1. Contantes de calibración	80
Tabla. 4.2. Valores de energía teóricos durante 30 minutos	85
Tabla. 4.3. Disposición de conexión de tarjeta inalámbrica con las diferentes placas empleadas como nodos	87
Tabla. 0.4. Tabla de direccionamiento	90
Tabla. 0.5. Combinaciones de estados entregados a la nube.	98
Tabla. 0.6. Valores entregados desde la aplicación para control de relés	100
Tabla. 0.7. Valor promedio del costo kWh	104
Tabla. 0.8. Consumo de energía módulo 1 – Refrigeradora	105
Tabla. 0.9. Consumo de energía módulo 2 – Televisión	107
Tabla. 0.10. Consumo de energía módulo 3 Lámpara	109
Tabla. 0.11. Consumo de energía módulo total de la línea.	111

INTRODUCCIÓN

1. ANTECEDENTES DE LA SITUACIÓN OBJETO DE ESTUDIO

Como es de conocimiento público, la energía es un recurso imprescindible para el desarrollo de sus actividades y el bienestar de la sociedad, pero a su vez su generación afecta directamente al calentamiento global, en muchas ocasiones se debe intervenir fuentes naturales para su generación, poniendo en riesgo la forma de vida de las personas. Bajo esta se han planteado dos formas de control de energía la primera el sentido común y la segunda el uso de la tecnología.

En este trabajo se plantea una alternativa para el monitorear el consumo de energía eléctrica en domicilios y que el uso de esta se lo realice de una de una manera más eficiente

Algunos proyectos similares al tema propuesto se exponen a continuación han sido utilizados como base para el desarrollo del presente proyecto.

- “Diseño e implementación de un medidor de energía electrónico para vivienda con orientación de la prevención de consumo y ahorro energético” Este proyecto permite el acceso a la información de la lectura del consumo energético ya sea a través de un mensaje de texto a través de un módulo GSM, como también acceder a la información del servidor mediante una interfaz web, se utiliza para la adquisición de datos una placa Arduino y para para el almacenamiento de datos una Raspberry pi, en la cual se administra una base de datos (Samaniego Idrovo & Valesaca Orellana, 2016).
- “Sistema electrónico de monitoreo y control para la distribución de energía eléctrica en los hogares” este proyecto utiliza un computador, o un dispositivo móvil conectado en la misma red, en la etapa de adquisición de datos, utiliza un sensor de corriente no invasivo y para la el voltaje un transformador reductor dicha información es procesada en una placa de adquisidor de datos, información

adquirida es visualizada en una pantalla LCD (Pallo Noroña & Salazar Vásquez, 2017).

- En la universidad Israel se realizó la “Implementación de un equipo de contrastación para contadores de energía eléctrica monofásica, para el laboratorio de medidores de la empresa eléctrica Quito” el cual funciona al recibir información de una tarjeta con un opto acoplador para la adquisición de datos y una Raspberry Pi para el procesamiento de información recolectada, el sistema funciona con cargas fantasmas colocadas en los terminales de monitoreo, simulando un aparato eléctrico o electrónico en funcionamiento, la información obtenida es comparada con las lecturas obtenidas del medidor de la empresa eléctrica, con el propósito de verificar la eficiencia con la que trabaja el medidor (Chacón Cajo, 2018).

En el mercado existen dispositivos que monitorean el consumo de energía para domicilios y tienen un costo elevado, su función es monitorear el tendido eléctrico que se origina en el tablero de distribución y en la mayoría de los casos, con un costo adicional, presentan módulos para monitorear o controlar las tomas de dos equipos, se accede a la información a través de un portal web

El presente proyecto plantea un sistema que permitirá obtener información del monitoreo de consumo de energía obtenido en tiempo real, el cual podrá ser visualizado desde una PC, o un dispositivo móvil, al analizar los datos adquiridos, también se podrá identificar si existe un consumo excesivo de un equipo (dañado), además permitirá detectar a un aparato que no debería estar consumiendo energía, el análisis de datos también permitirá determinar las horas en que los equipos consumen mayor cantidad de energía, determinar las causas del consumo y podría diseñar un plan para optimizar el consumo energético.

2. PROBLEMA DE INVESTIGACIÓN: PRESENTACIÓN Y JUSTIFICACIÓN

Con el paso del tiempo y los adelantos tecnológicos, el ser humano siempre buscara la forma de facilitar su estilo de vida, llevándolo al desarrollo de nuevos aparatos eléctricos y electrónicos, aumentando el consumo de energía en los ambientes en que se desenvuelve diariamente, así como también en el hogar.

También cabe mencionar que en la vida cotidiana el usuario espera mes a mes la planilla de consumo de energía y se puede encontrar con un valor elevado por consumo de energía eléctrica, es en ese momento que se ve obligado a buscar las posibles causas que originan el problema.

Por tal motivo el sistema de monitoreo de energía será una herramienta útil para el usuario el cual podrá observar el comportamiento de consumo de energía de manera gráfica en tiempo real, así como también acceder al historial de consumo, desde cualquier dispositivo con acceso a internet y poder realizar correcciones oportunas en el consumo de energía. Si se identificaran anomalías de comportamiento de los aparatos, esto lo hará comparando lecturas de consumo en tiempos determinados, permitiendo ubicar las causas de consumo de energía innecesario.

3. OBJETIVOS

a. Objetivo general:

Elaborar un prototipo de sistema de monitoreo y control de consumo de energía eléctrica de un sistema monofásica para un domicilio, aplicando el concepto de Internet de las cosas.

b. Objetivos específicos:

- Analizar la información adquirida por el sistema, en diferentes ambientes, en los cuales deberá trabajar el dispositivo
- Definir la etapa de comunicación y visualización de los aparatos para monitorear y controlar el consumo de energía.
- Diseñar el prototipo electrónico del sistema de monitoreo y control desde una aplicación para teléfono celular con sistema Android.
- Construir la aplicación en Android para acceder y visualizar gráficamente el historial de consumo.
- Implementar el diseño electrónico.
- Construir la maqueta que simule tres ambientes; sala, comedor y dormitorio, donde se implementará el prototipo de monitoreo y control de consumo de energía
- Presentar el historial de consumo de energía utilizando gráficas.
- Realizar pruebas de funcionamiento para la validez del prototipo.

4. Descripción de los capítulos

El presente proyecto se enfoca en la implementación de un sistema prototipo para monitoreo de consumo de energía para hogares (bifilar), basado en el concepto de internet de las cosas. A continuación, se describen los siguientes capítulos:

- El Capítulo I. Fundamento teórico se ha recopilado información a manera general de los diferentes conceptos y funcionamiento de los medidores de energía eléctrica.
- En el Capítulo II marco metodológico, se describe la metodología empleada para ejecución del presente proyecto.
- El Capítulo III. Propuesta, este capítulo es uno de los más extensos ya que se detallan los diferentes circuitos electrónicos y las diferentes conexiones que se han empleado para el funcionamiento hardware. También se expone la lógica de control o software.
- En el Capítulo IV trata sobre la construcción del proyecto, en este capítulo es donde se realizan las diferentes pruebas de funcionamiento y procesamiento de datos del medidor inteligente.
- Para terminar, se han realizado las conclusiones y recomendaciones del proyecto.

CAPÍTULO 1

1. FUNDAMENTACIÓN TEÓRICA

ENERGÍA ELÉCTRICA

La energía eléctrica hoy en día se ha convertido en una de los principales fuentes de energía no renovable, se la utilizada en todo ámbito en que el ser humano desarrolla la vida y sus diferentes actividades ya sea en el trajinar diario de la oficina, en labores domésticas, como también en actividades de ocio El uso de la energía eléctrica es ilimitado, por ejemplo en el hogar se utiliza, en cocinas eléctricas, alimentadores electrodomésticos, ventiladores, televisores, en la industria en sistemas de alumbrado, motores eléctricos, climatización, etc., su uso tan frecuente se debe a que es fácil de controlar, es versátil, de pronto resultados a más de ser una energía limpia en el sitio de consumo. (Joaquin, s.f.)

1.1. Energía eléctrica como producto

La electricidad se ha convertido en una fuente de energía no renovable, se consume en el momento que es generada y para su transportación requiere un sistema de líneas de transmisión específicas que debe cumplir con las diferentes leyes de la física como las Kirchhoff, a más de lo mencionado anteriormente la energía eléctrica no se puede almacenar.

La mayoría de los países tienen la necesidad de crear gigantescas, costosas y sofisticadas estructuras para generar y comercializar la energía. La unidad de medida básica es el Joule y el segundo (s) es la unidad básica de la unidad de tiempo es decir:

$$Potencia = \frac{Energia(E)}{Tiempo (t)}$$

De tal manera que la energía es igual a:

$$Energia = Potencia(P) \times Tiempo(t) \text{ EC 2}$$

Es decir, el consumo eléctrico se lo obtiene del producto de la potencia en watts, por el tiempo en horas (Enríquez, 2009).

Unidad de medida de la energía: **Wh**

1.1.1. Cálculo de la potencia.

El cálculo de potencia en circuitos alimentados por señales alternas, debido a la naturaleza de los componentes de las cargas origina que la corriente adelante 90° al voltaje en un circuito de carga capacitiva X_C , o la corriente retrase al voltaje 90° en un circuito con carga inductivo X_L . Para un circuito resistivo puro se considera que el ángulo entre la corriente y el voltaje es 0° . En un circuito resistivo, inductivo y capacitivo el cálculo de la potencia es más complejo debido a que existen diferencias de fase de la corriente y voltaje.

- **Potencia aparente “S”** Al realizar el cálculo de potencias en un circuito de la Figura 1.1, se debe multiplicar el voltaje por el valor de la corriente $p= VI$ considerando que la carga es resistiva, si la carga es resistiva y reactiva al realizar la multiplicación, este valor no representara ni la potencia real tampoco la reactiva, en este caso VI , representa la potencia aparente que se representa por la letra S, donde V e I son los valores RMS del voltaje y corriente respectivamente (González, 2008).

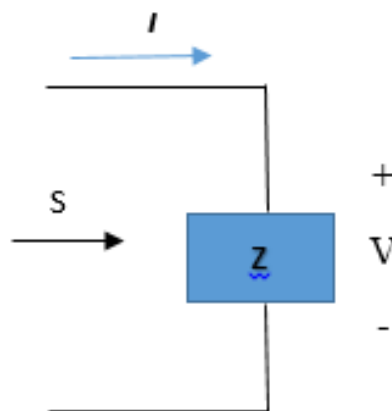


Figura. 0.20. Potencia aparente

Fuente: El autor

La potencia aparente se calcula de la fórmula:

$$S = V I \text{ [VA]}$$

Donde:

S = potencia aparente cuya unidad de medida es volta- Amper

V= voltaje en voltios [V]

I = corriente en amperios [A]

En la Figura. 1.2. En el triángulo de potencia se observa los diferentes elementos que intervienen para obtener la potencia aparente, para el ejemplo se utiliza una carga inductiva, parte imaginaria positiva Q y una parte real P.

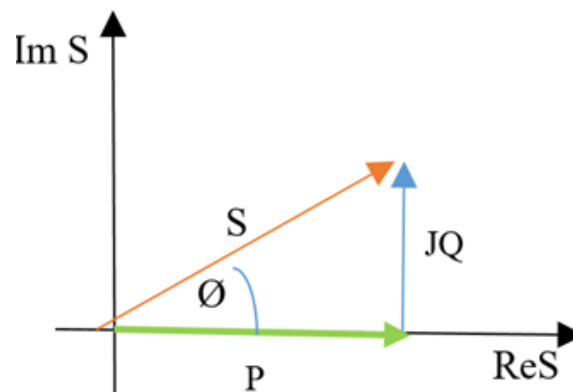


Figura. 0.21. Triangulo de potencias

Fuente: El autor

- **Potencia reactiva** La potencia reactiva no genera trabajo útil, es la energía consumida por una bobina para generar un campo magnético para generar el movimiento de un motor o aquella energía almacenado en capacitor, el vector capacitivo tiene polaridad negativa y él vector inductivo es positivo, por lo que se les puede sumar matemáticamente. (Ecuared, 2013)

Fórmula de cálculo:

$$Q = VI \sin \varnothing$$

Q = potencia reactiva unidad de medida [VAR]

V= voltaje en voltios [V]

I = corriente en amperios [A]

- **Potencia activa** La potencia activa es aquella energía eléctrica cuyo trabajo útil es utilizado para generar otras formas de energía como calefactores, lumínica, térmica, química.

- Las cargas de esta potencia se las considera netamente resistivas es decir la corriente se encuentra en fase con el voltaje. (ECUARED, 2013)

Fórmula de cálculo:

$$P = VI \cos \emptyset$$

P = potencia activa cuya unidad de medida vatio [W]

V= voltaje en voltios [V]

I = corriente en amperios [A]

\emptyset = Angulo de fase = 0

1.2. Origen de la electricidad

La electricidad se origina en un fenómeno donde existe una diferencia de potencial, que se origina al inducir una fuerza ajena al sistema para arrancar un electrón de un átomo, que genera un flujo de electrones en el interior de un material conductor capaz de crear energía lumínica, calor o fenómenos electromagnéticos. En todo el mundo existen varias centrales energéticas que generan energía utilizando los diferentes recursos naturales como:

- **Centrales Térmicas.** - La energía eléctrica se origina de la combustión de combustibles sólidos (carbón natural), combustibles líquidos y gaseosos, es decir de la energía de combustión térmica.
 - El vapor de agua generado en las calderas permite el movimiento de las turbinas sobre su propio eje, así como también las turbinas pueden ser impulsadas por gases obtenidos de la quema de gas natural, o aceite de petróleo destilado
 - **Centrales Hidrotérmica.** - Esta central utiliza la energía térmica almacenada en grandes extensiones de agua como el océano.
 - **Centrales Solares o Heliotérmicas.** - utiliza la energía proveniente del sol en forma de radiación electromagnética, para esto se utiliza paneles solares, la recolección de este tipo de energía depende de las condiciones meteorológicas, en un día despejado se puede recolectar 1kW/m².
 - **Centrales Eólicas.**- estas centrales utilizan el viento o corrientes de aire para mover sus generadores de energía, la velocidad mínima requerida es de 6 m/s.
- Centrales Nucleares.**- la energía eléctrica que es producida en una central nuclear, tiene origen en la generación de energía nuclear que utiliza como combustible principal el

uranio, en el reactor se produce la fisión, que libera gran cantidad de energía llegando a temperatura suficientemente altas para generar vapor agua capaz de mover las turbinas generadoras de energía, el uranio es el único de los tres elementos nucleares que se pueden obtener de la naturaleza.

- **Centrales hidroeléctricas.** Son infraestructuras que utilizan la fuerza del agua la cual es represada, entubada y controlada para mover grandes turbinas de la hidroeléctrica, la generación de energía eléctrica en las centrales hidroeléctricas es a bajo costo, pero su implementación requiere una fuerte inversión.

De las diferentes centrales eléctricas se debe transportar la energía hasta el punto de consumo, para ello es necesario elevar el voltaje en cámaras elevadoras en un rango de 66kV a 400kV, la energía distribuida ingresa a cámaras de transformadoras reductoras de voltaje donde se maneja voltajes en el rango 30 a 20kV o también llamado tensiones medias, estas cámaras reductoras se las puede localizar próximos a sectores urbanos a los que están destinados la energía. (Sánchez, 2008). En la Figura. 1.3. se observa el esquema de generación y transporte de energía

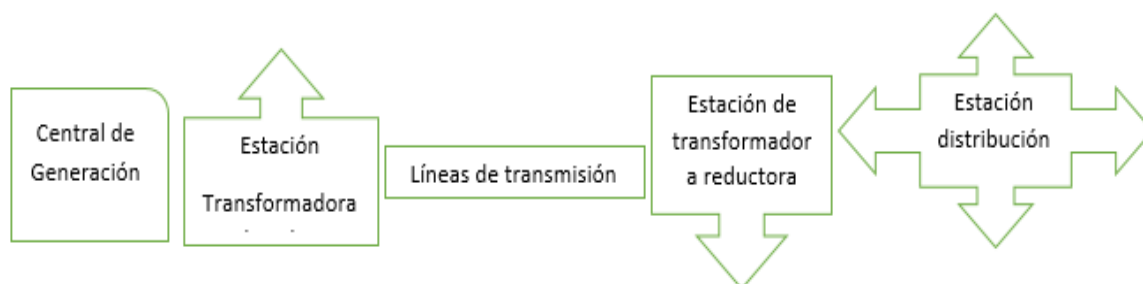


Figura. 0.22. Esquema generación y transporte energía

Fuente: El autor

En la última década se han realizado varios proyectos energéticos en el país, los cuales buscan aprovechar el caudal de agua de los diferentes ríos y algunos otros recursos naturales como el viento y utilizarlos en la generación de energía eléctrica.

El objetivo de estos numerosos proyectos es obtener independencia energética a mediano y largo plazo, reduciendo la emisión de CO₂, reducir la importación de energía y generando una fuente de empleos. En la Tabla. 1.1. se realiza un listado de los diferentes

proyectos energéticos realizados en el país donde se puede apreciar su localización geográfica y su capacidad de generación en megavatios [MW].

Tabla. 1.1. Proyectos energéticos del Ecuador

PROYECTO	GENERACIÓN DE ENERGÍA	UBICACIÓN
Hidroeléctrica Coca codo Sinclair	1500 MW	Sucumbíos y Napo
Hidroeléctrica Manduriacu	65 MW	Pichincha e Imbabura
Hidroeléctrico Quijo	50MW	Pichincha
Hidroeléctrica Masar Dudas	20.82MW	Cañar
Hidroeléctrico Minas San Francisco	270 MW	Azuay - El Oro
Hidroeléctrica sopladora	487 MW	Azuay y Morona Santiago
Hidroeléctrica Toachi Pilatón	254.40	Pichincha, Santo domingo de los Tsáchilas y Cotopaxi
Hidroeléctrica Delsintanisagua	180MW	Zamora Chinchipe
Eólico Villonaco	16.5 MW	Loja

Fuente: (Enríquez, 2009)

1.3. Sistemas para la lectura de energía eléctrica.

Un sistema de medición de energía es aquel que permite realizar medidas del consumo de energía de un determinado sistema o servicio eléctrico utilizado en la industria o a un nivel domiciliario, que permiten calcular el consumo de energía en kW/h ya sea de un servicio eléctrico o de un sistema completo. En Ecuador la lectura del consumo eléctrico de los domicilios es realizada mediante contadores de energía electromecánicos, los cuales almacenan valores de energía consumida durante un periodo de tiempo. La toma de las

lecturas de estos dispositivos es realizada por personal de la empresa eléctrica una vez al mes para poder calcular el consumo mensual de energía de los diferentes domicilios, con estos valores y realizando algunas operaciones matemáticas obtienen el valor a pagar de la planilla.

Para los diferentes tipos de suministros existen contadores: Monofásicos, Bifásicos y Trifásicos

1.3.1. Contador electromagnético.

El medidores tradicional o electromagnético, cuyo funcionamiento se basa en el principio descubierto por Galileo Ferraris en el siglo XIX, en la que se emplea dos bobinas, una de corriente contrapuesta a una de voltaje, que hacen girar un disco rotatorio, este dispositivo ha sido utilizado por varios años gracias a sus alta precisión de medición de consumo de energía para bajas frecuencias, pero presenta grandes dificultades para realizar lecturas en diferentes componentes en alta frecuencia.

En la Figura. 1.4. se observa el esquema de un contador electromagnético y sus diferentes componentes básicos.

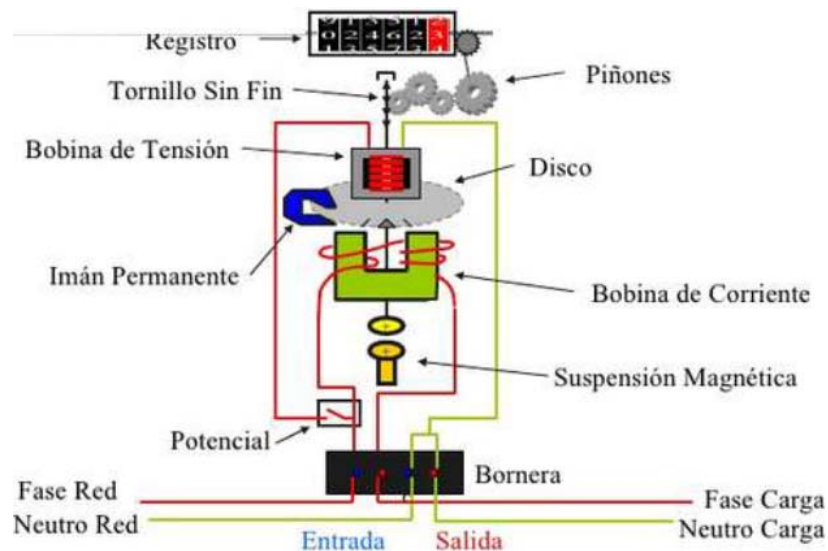


Figura. 0.23. Diagrama contador electromagnético

Fuente: (Vega, 2016)

1.3.2. Contador electrónico

El contador electrónico de la Figura. 1.5. es un dispositivo totalmente programable en el cual el usuario puede definir mediante software, los tipos de lecturas que debe realizar el dispositivo de estado sólido, se puede obtener lecturas del consumo de energía ya sea reactiva, activa o aparente, también la demanda máxima de potencia, valores de voltaje y corriente, factor de potencia, factores importantes de la red que permiten medir la calidad de energía. (Vega, 2016)

El contador electrónico dispone de un procesador y una capacidad de memoria suficiente para almacenar los diferentes datos adquiridos por los sensores, así como también almacenar los resultados obtenidos de los diferentes procesos, la presentación de la información se lo puede realizar en una PC, o en un display propio del aparato.

Según la función que cumple este tipo de contadores se los clasifica en:

- **Contadores de Demanda.** - durante un periodo de 24 horas realizan la medida y el almacenamiento del consumo de energía eléctrica en un solo periodo y con una tarifa única (Vega, 2016)
- **Contadores Multitarifa.** - se realiza la lectura del consumo de energía ya sea por tramos del sistema o el sistema completo, además de realizar las diferentes mediciones del consumo de energía en diferentes periodos dentro de un tiempo de 24 horas (Vega, 2016)



Figura. 0.24. Contador electrónico.

Fuente: (Vega, 2016)

1.3.3. Pliego tarifario residencial.

La categoría residencial está destinada al uso doméstico no comercial, donde el consumo de energía por abonado relativamente baja al comprarlos con los demás sectores ya sea el comercial o el industrial. ARCONEL es la encargada de establecer los parámetros de cobro de valores de consumo de energía, donde el sector residencial dispone del valor más bajo por consumo de energía, este sector comprende valores menores a 600 voltios (ARCONEL, 2019). En la Tabla. 1.2., se observa la categorización según el consumo de energía en kWh.

Tabla. 1.2. Cargos tarifarios Quito

Rango de consumo kWh	Estrato consumo	Energía USD /kWh	Comercialización (USD/Consumidor)
1-50	E	0.091	
51-100	E	0.093	
101-150	D	0.095	
151-200	C	0.097	1.414
201-250	C	0.099	
251-300	B	0.101	
301-350	B	0.103	
351-500	A	0.105	
Categoría		Residencial temporal	
0.128		1.414	

Fuente: (ARCONEL, 2019)

1.4. Electrónica del proyecto.

Para el desarrollo el presente proyecto, incorporará varias tecnologías como Arduino, Raspberry PI, también elementos pasivos y activos entre estos se mencionan: sensores para adquisición de datos, dispositivos de comunicación inalámbrica NRF 24, display LCD 16x2, dispositivos móviles Android “celular” para visualización de datos, los diferentes dispositivos electrónicos y algunas de sus características técnicas se detallan en los siguientes párrafos. Estos permitirán adquirir, procesar, almacenar y mostrar los diferentes datos obtenidos en un sistema de consumo de energía.

1.4.1. Fuentes de alimentación

La fuente de alimentación es un dispositivo electrónico que convierte una señal de tensión alterna en una señal de tensión continua. En la Figura. 1.6. se observa un ejemplo de una fuente no conmutada, normalmente este tipo de dispositivos utiliza elementos como rectificadores y filtros capacitivos para estabilizar la señal de la red eléctrica de alterna a una señal eléctrica continua, siendo una desventaja su no linealidad que provoca un alto número de armónicos a la red eléctrica. (Alava, Campos, & Falcones).

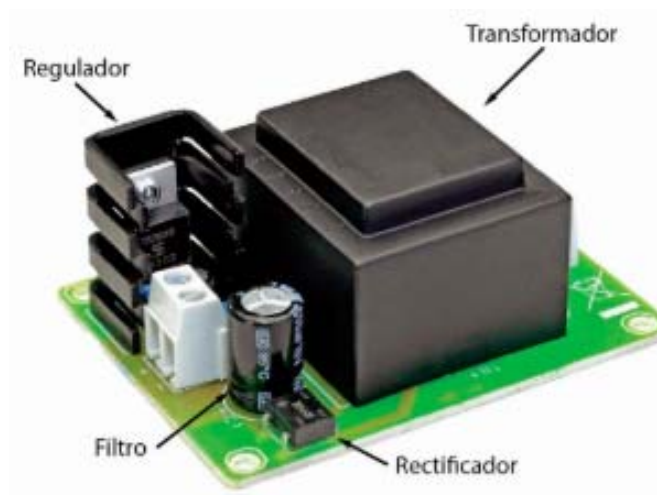


Figura. 0.25. Fuente lineal

Fuente: (Blogger, 2015)

En la Figura. 1.7. se observa una fuente conmutada, este tipo de fuentes según avanza la tecnología cada vez tiene una mayor la demanda de fuentes AC / DC y requiere un mayor rendimiento. Debido a que los equipos actuales que son utilizados por el usuario requieren voltajes más bajos y son más susceptibles a daños por variaciones de tensión, para esto se desarrollaron fuentes conmutadas que a comparación con las fuentes lineales son mucho más pequeñas y reducen el factor de potencia. (Alava, Campos, & Falcones)



Figura. 0.26. Fuente conmutada

Fuente: (Blogger, 2015)

1.4.2. Relé- Relevador

En la Figura. 1.8. se muestra el esquema de un relé y sus diferentes componentes principales como:

- La bobina,
- Contactos de conexión que son normalmente abiertos (NO) y normalmente cerrado (NC).
- La armadura,
- Terminales de energización de la bobina.

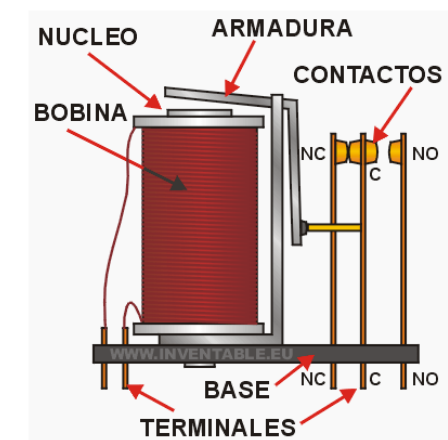


Figura 0.27 Esquema de un relé

Fuente: (Inventarle Electrónica Simple Y Clara, 2013)

El relevador básicamente es un switch que se activa o desactiva al inducir un voltaje en los terminales del electroimán, el cual atrae o repela un pequeño brazo llamado armadura, permite que exista conexión o no entre dos o más terminales.

1.4.3. Arduino

Arduino es una plataforma electrónica que integra desde un microcontrolador de la marca Atmega, también brinda circuitería extra como reguladores de voltaje y un puerto USB, para su configuración y realizar pruebas de control electrónico y de comunicación con el chip, desde cualquier pc de una manera sencilla.

En la Figura. 1.9. se aprecia físicamente los diferentes modelos de Arduino que existen el mercado donde se aprecia en especial la diferencia de tamaño entre uno y otro modelo.



Figura. 0.28. Tipos placas de Arduino

Fuente: (PE, 2017)

La tarjeta Arduino se alimenta de 5V, de voltaje continua, además la tarjeta Arduino pone a disposición varios pines el numero dependerá del modelo de tarjeta, se los puede utilizar como entrada o salida de señales digitales de cualquier dispositivo que sea capaz de enviar o recibir señales digitales entre valores de 0 a 5 voltios. (Aprendiendo Arduino, 2016). La placa Arduino también pone a disposición varias entradas y salidas analógicas, las cuales permiten la lectura de señales obtenidas por sensores en forma de variaciones de voltaje.

De los tipos de placas de Arduino existentes, se busca el que brinda el número de entradas analógicas y digitales necesarias para realizar el procesamiento de la señal adquirida por los sensores de corriente y voltaje, como también permita integrar el dispositivo para comunicación inalámbrica. (PE, 2017).

En la Tabla. 1.3. que se aprecia a continuación, se realiza una comparación entre los diferentes tipos de Arduino, donde se escogerá uno que se ajuste a los requerimientos necesarios para desarrollar el presente proyecto. Al comparar varias tarjetas Arduino se observa bastante similitud en sus características técnicas, con algunas diferencias como la capacidad del procesador, el voltaje a la salida de sus terminales y su tamaño.

Tabla. 1.3. Cuadro comparativo Arduino

Arduino	Procesador	Frecuencia	RAM	Pin Digital Entrada /Salida	Tensión /Corriente En Los Pines	Resolución
Pro Mini	AVR Atmega 168 `O 328 8 BITS	16 Mhz	2 kIB	14/14	3.3 V-5V 40mA	10 bits 1024valores
Nano	AVR Atmega 168 `O 328 8 BITS	16 Mhz	2 kIB	14/14	5V 40mA	10 bits 1024 valores
Uno	AVR Atmega 328 8 BITS	16 Mhz	2 kIB	14/14	5V 40mA	10 bits 1024 valores
Mega/Mega 2560	AVR Atmega 2560 8 BITS	16 Mhz	8 kIB	54/54	5V 40mA	10 bits 1024 valores
Micro	AVR Atmega 32UE 8 BITS	16 Mhz	2.5 kIB	20/20	5V 40mA	10 bits 1024 valores

Fuente: (PE, 2017)

1.4.4. Entradas analógicas

Las señales analógicas pueden tomar cualquier magnitud física que varía con el tiempo, espacio dentro de un intervalo V_{CC} y $-V_{CC}$.

Vale señalar que ingresan en las entradas analógicas deben estar dentro del rango de $0 - 5V$, las entradas analógicas son escasas y su procesamiento más lento en comparación con las entradas de señales digitales.

Una entrada digital, trabaja únicamente con dos valores un alto $5 V$ y un bajo $0 V$, a lo contrario de una entrada análoga, proporciona un valor codificado con un numero de N bits (10 bits)

1.4.5. Precisión de la medición

Arduino nano tiene embebido un convertidor análogo digital (ADC), el cual se encarga de convertir una señal analógica en una señal digital codificada con N número de bits. Arduino nano, proporciona 1024 niveles digitales, presume una precisión de ± 2.44 mV.

1.4.6. Precisión relativa

La precisión de medida de un Arduino para una señal que está dentro del rango de $0 - 5V$ utilizando el convertidor análogo digital de 10 bits con una precisión aproximada a $4.88mV$ con relación al voltaje de $5V$, teniendo una precisión relativa de 0.1% .

En la práctica las señales analógicas presentan variaciones en todo el rango de la señal de referencia, a manera de ejemplo se considera una señal entre 0 y $1 V$, En este caso, se mantiene la relación relativa $4.88mV$, es decir la precisión caería a un 0.5% .

1.4.7. Referencia de tensión analógica (AREF)

Para realizar una corrección de la referencia relativa, Arduino da la facilidad de cambiar el voltaje de referencia por el conversor analógico con la función *AnalogRef*. En la Tabla. 1.4. se puede observar los diferentes voltajes de referencia para la placa de Arduino. Al trabajar con los voltajes de referencia se debe no superar el límite para no malograr al Arduino.

Tabla. 1.4. Valores de tensión de referencia Arduino

Referencia analógica	Valores de voltaje V_{CC}
Por defecto	5V O 3.3 V
Internal	1.1 V (Atmega 168 Y 328)
Extenal	p.IN Vref (0- V_{CC})
Internal 1 V1, Internal 2V 56	1.1 V y 2.56 (solo Arduino Mega)

Fuente: (Arduino, n.d.)

1.4.8. Convertidor Análogo / Digital

En la Figura. 1.10. se presenta el diagrama de bloques de un convertidor análogo digital que donde se puede observar el proceso que sufre la señal analógica al atravesar los diferentes bloques hasta obtener a la señal digital.

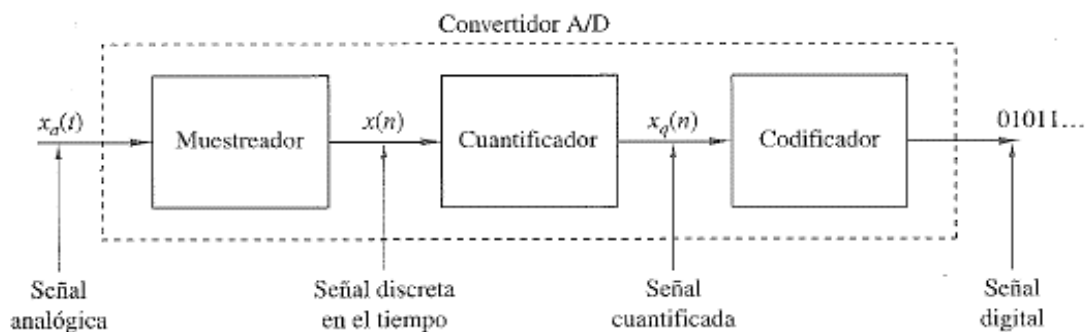


Figura. 0.29. Convertidor analógico

Fuente: (Proakis, 2007)

El convertidor análogo digital integrado en el controlador Atmega de Arduino nano pone a disposición 8 canales con una resolución de 10 bits, que entregaran valores entre 0 y 1023. La función de las entradas analógicas tiene como función principal leer datos obtenidos por los diferentes sensores conectados a la placa, estos pines también se les puede programar para que cumplan funciones de entrada y salida de propósito general. Similares a los pines 0 al 13. (Arduino, n.d.)

1.5. Sensores

En las actividades diarias en la vida de los seres humanos, todo el tiempo censando a través de los sentidos los cuales están recibiendo información que es procesada por nuestro cerebro, en la Figura. 1.11. se observa una ilustración donde a manera de ejemplo se aprecia como los sensores emulan los sentidos de las personas. La tecnología trata de desarrollar sensores basados en el sentido de los seres humanos con los cuales pretende censar los diferentes eventos que suceden en un ambiente, diseñando un sinnúmero de dispositivos que recoja información y la envíe al procesador para obtener información requerida para un proceso (Especcator, 2012).



Figura. 0.30. Los sensores emulan los sentidos

Fuente: (Especcator, 2012)

1.5.1. Sensor de corriente SCT 013

En la siguiente Figura. 1.12. se aprecia la imagen del sensor de corriente no invasivo es un transformador por cuyo núcleo atraviesa el cable conductor de un circuito, puede ser

este la línea o el neutro, pero no los dos juntos, son muy prácticos y se los puede utilizar incluso para realizar mediciones del consumo de energía de un edificio si fuese necesario, es un sensor no invasivo así que se lo puede conectar si intervenir físicamente al cable. (ANEXO E)



Figura. 0.31. Sensor de corriente no invasivo

Fuente: (OpenEnergyMonitor)

Se compone básicamente de un bobinado primario, un núcleo y un devanado secundario, el cable de la acometida al edificio o la línea del circuito eléctrico el bobinado primario, el devanado secundario utiliza un cable bajo calibre y un número de espiras alto. (*OpenEnergyMonitor*). En la Figura 1.13 se observa los componentes del sensor de corriente no invasivo de la familia SCT.

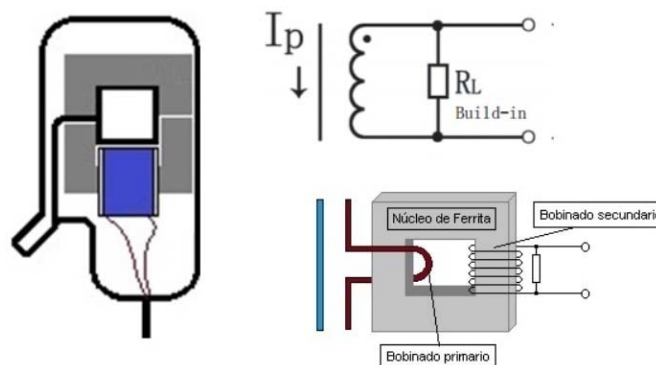


Figura. 0.32. Sensor de corriente no invasivo esquema eléctrico

Fuente: (Mechatronics, 2016)

El devanado secundario se encuentra aislado galvánicamente del núcleo, para evitar un contacto físico entre el devanado y el núcleo.

El sensor para utilizar en este proyecto tiene incorporado una resistencia de carga es decir entrega un voltaje a la salida de los terminales, los sensores de la familia SCT-013-000, contiene embebido un sistema de protección con la intención de reducir el riesgo de un shock eléctrico.

1.5.2. Transformador de voltaje AC - ZMPT101B

El transformador que se muestra en la Figura. 1.14. permite tomar medidas directamente de la toma de la red eléctrica doméstica y entrega a su salida una señal de voltaje que está dentro de los parámetros de funcionamiento en las entradas analógicas de Arduino que trabaja con valores positivos en el rango de 0- 5V (Mechatronics, 2016).

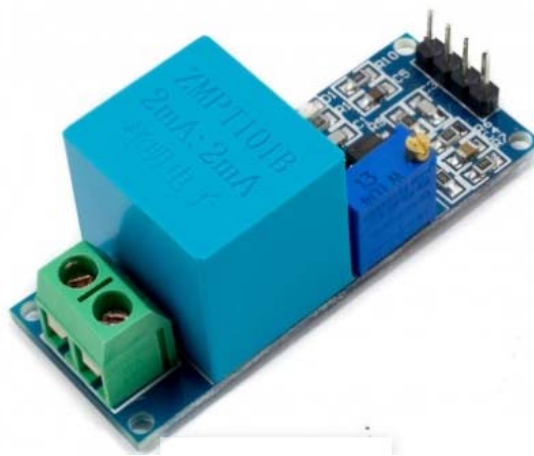


Figura 0.33 Transformado de voltaje

Fuente: (Mechatronics, 2016)

En el devanado primario del transformador se conecta la señal alterna soportada por el transformador que es de 250 V, a la salida del devanado secundario es necesario referenciar el voltaje además de utilizar un amplificador operacional, el voltaje de desplazamiento depende del voltaje de alimentación V_{CC} , el módulo también permite ajustar el voltaje pico con la ayuda de un potenciómetro incorporado en la placa. (Mechatronics, 2016)

Con la señal entregada a la salida de este transformador, cualquier dispositivo podrá procesar la señal y obtener valor de voltaje. En la Tabla. 1.5. se dan a conocer algunas características técnicas del circuito de acoplamiento del transformador de voltaje AC - ZMPT101B. (ANEXO D), así como las propiedades del transformador en la Tabla. 1.6.

Tabla. 1.5. Valores de tensión de referencia del circuito de acoplamiento

Voltaje de funcionamiento:	3.3 V- 5VDC
Voltaje de entrada:	250 VCA máximo
Voltaje del secundario: (señal sinusoidal)	5 VAC

Fuente: (Mechatronics, 2016)

Tabla. 1.6. Propiedades del transformador

Corriente entrada y salida:	2mA
Aislamiento eléctrico entrada	300V
Ratio in – out	1000:1000
Diferencia de fase	< 30 ⁰ (a 50 ohm)
Rango lineal	0-3mA (a 50 ohm)
Linealidad	1%
Precisión	0.2%

Fuente: (Mechatronics, 2016)

1.5.3. Raspberry Pi

Es un miniordenador diseñado para la implementación de pequeños proyectos e incentivar el aprendizaje de sistemas de computación en los centros de estudio, en la Figura. 1.15. muestra un modelo de Raspberry Pi.

Se han desarrollado varios sistemas operativos para la placa Raspberry, siendo el principal basado en Linux con algunas modificaciones del sistema original, *Debian* o también llamado *Rasbian* para que este sea funcional en una placa con menores requisitos de hardware, a diferencia de los computadores tradicionales, Raspberry Pi utiliza un procesador con arquitectura ARM de tipo RISC (*Reduced Instructions Set Computer*), esta arquitectura utiliza instrucciones realmente simples, con la finalidad de reducir el consumo de energía (Raspberrypiparatorpes, 2013)



Figura. 0.34. Raspberry Pi 3 Modelo B

Fuente: El autor

En la actualidad se han desarrollado otros sistemas operativos enfocados en el uso de la Raspberry Pi incluido la versión de Windows 10 IoT. Core. El uso de un sistema operativo u otro difiere en la mayoría de los casos por el uso de licencias, o las características propias de cada sistema operativo por ejemplo, Windows 10 IoT. Core, es un sistema cerrado en comparación a *Rasbian*, un sistema operativo abierto y se ajusta con mayor facilidad a los requerimientos del desarrollador, también el entusiasta que desee desarrollar sus proyectos en Raspberry cuenta con ayuda de la comunidad que brinda soporte para el uso del sistema operativo, en la actualidad se desconoce la existencia de una ayuda en línea para Windows 10 IoT Core. (Raspberrypiparatorpes, 2013),

- En la Figura 1.16 se observa una Raspberry Pi Zero W, que en comparación es mucho más pequeña que la Raspberry pi B, del mismo modo se dispone de un número menor de entradas ejemplo la conexión a internet se la realiza inalámbricamente Wifi.



Figura. 0.35. Raspberry Pi Zero W

Fuente: (Raspberrypiparatorpes, 2013)

En el mercado existe varios tipos de placas Raspberry, para la elaboración del presente proyecto se escogido la Raspberry pi Zero W, que es una variable de la Raspberry Pi convencional, de bajo costo y de tamaño bastante reducido que cumple con las necesidades. (ANEXO C) y especificaciones como muestra la Tabla. 1.7.

Tabla. 1.7. Características Raspberry pi Zero W

CPU	Broadcom BCM 2835 un solo núcleo 1Ghz
RAM	512 MB
Ranura de almacenamiento	Micro SD
Video output	Mini HDMI y salida de video compuesto sin conector montado
Wifi	Utiliza el microchip Cypress CYW43438, soporta 802.11/n
Bluetooth	Bluetooth 2.0/4.1
Entradas y salidas de propósito general (GPIO)	40 pines
Voltaje de alimentación	5 V, 2.5 A mediante micro USB
Dimensiones	65mm x 30mm x 5mm

Fuente: (Raspberrypiparatorpes, 2013).

1.6. Redes de comunicación de datos

1.6.1. Internet.

La internet es la infraestructura tecnológica más grande que el hombre ha podido crear, esta se encarga de intercomunicar a millones de personas alrededor del mundo, en un principio la red de la internet utilizaba únicamente computadores de escritorios, según la tecnología avanza, es mayor el número de dispositivos electrónicos como computadora, televisores, tablets, celulares, etc., que requieren acceso a internet, en el idioma de los computadores a todos estos dispositivos se los denomina host o terminales. (Garnet, 2014)

En el 2015 el número de terminales conectadas a internet era mayor a 5000 millones, se estima que para el 2020 este número ascenderá a 25000 millones. En el 2015 el número de beneficiarios conectados a internet a nivel mundial era superior a 3.200 millones, que correspondía al aproximadamente el 42% de la población mundial (ITU, 2015).

1.6.2. Internet de las cosas IoT.

En la actualidad el uso del internet se ha convertido en una necesidad imperante, un claro ejemplo es el uso del celular. Se puede encontrar en el mercado tecnológico un sinnúmero de equipos electrónicos que permiten conectividad a internet desde una alfombra hasta un vehículo, a manera de ejemplo un *Smart Watch*, toma lecturas del comportamiento de una persona y da alarmas de distancia recorrida, tiempo de ocio, horas de sueño en el transcurso del día, todo esto queda registrado y puede ser observado desde nuestro celular con la ayuda de una aplicación. (Garnet, 2014)

La evolución de la tecnología y la necesidad de crear artefactos inteligentes, han abierto una puerta que puede ser aprovechada por hacker, quienes pueden hacer uso del llamado internet de las cosas y acceder a la información de los servidores de datos, utilizar juguetes conectados a internet para hablar con niños, por estas razones de inseguridad y de confidencialidad, muchos de las personas que aún no utilizan esta tecnología han querido conocerla, lo que ha detenido su aplicación total.

1.6.3. Plataformas IoT

En la actualidad se dispone de un sinnúmero de plataformas que pueden ser utilizadas en diferentes proyectos enfocados en el internet de las cosas, entre ellas se puede nombrar: *Arduino Cloud*, *Cayenne*, *Thingierio*, *aREST*, *ThingSpeak*, *Artika Cloud*, etc. Todas estas plataformas están disponibles para que el usuario haga uso de sus prestaciones tecnológicas, el uso de una u otra plataforma está marcado a su facilidad de uso o a las prestaciones que entregan como el número de datos que permiten almacenar, la compatibilidad con las diferentes tarjetas electrónicas como: *Arduino*, *Raspberry Pi*, *Módulo wifi ESP8266*, *Lora Wan*, etc , el tiempo de actualización de datos, las librerías para acceder a estas, son algunas pautas que permitirán elegir la más idónea para el desarrollo del presente proyecto. A continuación en la Figura. 1.17, se muestra el entorno de la plataforma *ThingSpeak*.

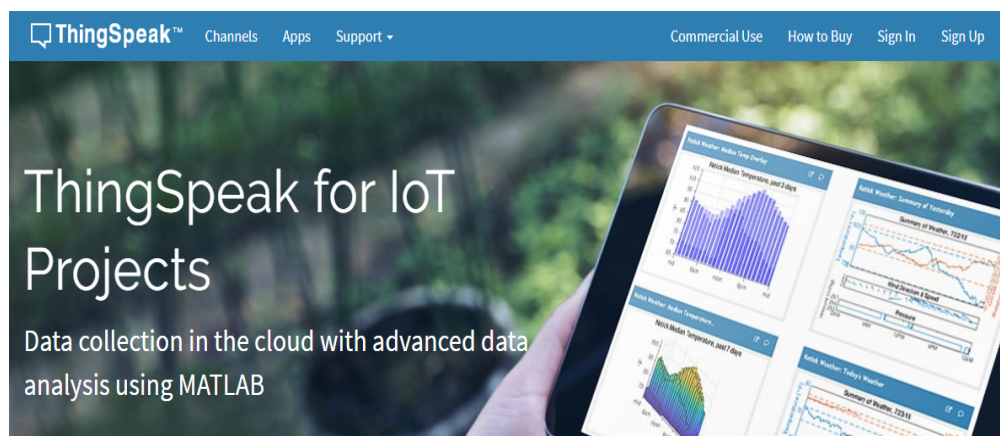


Figura. 0.36. ThingSpeak

Fuente: (ThingSpeak, 1994-2020)

ThingSpeak, en una plataforma de código libre y abierto, pone a disposición varios canales, 8 en su versión free y almacena 8200 datos por canal, así como también permite acceso a Mathlab, para realizar varias operaciones matemáticas, permitiendo una mejor visión de los datos adquiridos. Esta plataforma es amigable, con Raspberry PI o Arduino, la transmisión de datos recolectados por sensores o que se encuentran en base de datos, se lo realiza utilizando el protocolo HTTP ya sea para enviar al internet o una red de área local. (ThingSpeak, 1994-2020)

Esta plataforma admite realizar un monitoreo de los objetos o sensores ya que consigue la integración sencilla de terceros como twitter, en la cual se puede visualizar el cambio de lecturas de los sensores a través de un cambio de estado en la cuenta de la red social. Al utilizar la versión gratuita y al ser de código libre, la hace bastante utilizada en la implementación de cosas a la web.

1.6.4. Red de comunicación de datos.

Es un sistema conformado por *hardware* y *software*, que se comunican entre sí utilizando medios físicos, inalámbricos o cualquier medio de comunicación que les permita la transmisión de datos mediante el envío ya sea de pulsos eléctricos y ondas electromagnéticas para brindar un servicio o recurso como el internet. (Prieto Blázquez)

- **El hardware** se considera al cableado estructurado, estaciones de trabajo, equipos nodos de red.
- **El software** se considera a los diferentes sistemas operativos que se utilizan para administrar la red (Network operating system).

En las redes de comunicación se puede enumerar cuatro ítems importantes que intervienen en su concepto de funcionamiento:

- **Protocolo de Comunicación.** - Es el conjunto de reglas, normas que dan todas las facilidades para la comunicación para que se pueda realizar el intercambio de información en un mismo lenguaje entre el emisor y receptor. Existen un sin número de protocolos de red pero el más utilizado es el TCP/IP, que es un protocolo utilizado por el internet.
- **Topología.** - se describe la estructura de conexión de los diferentes nodos conectados entre sí de una red, las topologías conocidas son: anillo, árbol, estrella, bus punto a punto.
 - **Topología en anillo.** En esta tipología, los terminales de la red se encuentran conectados uno a continuación de otro (serie), es decir el ultimo terminal se conecta con el primero, al existir un camino único de comunicación se pierde

la comunicación si uno de los elementos deja de funcionar, En la Figura. 1.18 se observa la topología en anillo.

Este tipo de tecnología utiliza un token o testigos, que es el encargado de entregar o recolectar información de un terminal a otro con el objetivo de evitar pérdidas de la información. (Kurose & Ross, 2017)

Vale señalar que en un inicio la comunicación se realizaba en un solo sentido el terminal tiene una tarjeta de transmisión y una de recepción de la información. En la actualidad se está empleando esta tecnología con comunicación bidireccional.

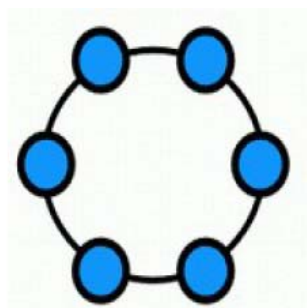


Figura. 0.37. Topología en anillo

Fuente: (Molina)

- **Topología en Estrella.** - Esta topología presenta un nodo central donde se conectan todos los terminales, es decir al encontrarse un terminal defectuoso no se altera el funcionamiento de la red de datos, toda la comunicación pasa a través del nodo central, la dificultad de esta topología es el envío de los paquetes de información, el router se encarga de esta enviar estos a su destino.
- **Topología en bus.** Esta topología como se aprecia en la Figura. 1.19. utiliza un solo bus de datos en los cuales se conectan los terminales o host mediante unidades interfaz y derivadores, la ruptura del bus o canal de comunicación hace que los terminales queden desconectados y al no existir ningún otro canal de comunicación, la red queda deshabilitada. (Kurose & Ross, 2017)

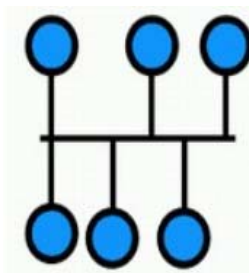


Figura 0.38 Topología bus

Fuente: (Molina)

- **Topología en árbol.** Esta topología se podría resumir como la suma de varias redes tipo estrella donde los nodos se conectan entre ellos, la falla de un nodo no impide la intercomunicación entre equipos, siempre y cuando no estén conectados a este nodo, en la Figura. 1.20. se aprecia este tipo de topología.

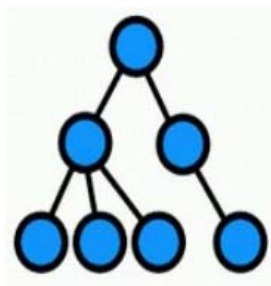


Figura 0.2070 Topología tipo árbol

Fuente: (Molina)

- **Topología en malla.** Es aquella en la cual los nodos se encuentran conectados de uno en uno o con varios de ellos como se muestra en la Figura. 1.21 la importancia de no depender de una ruta única evita que se pierda información, dispone de varios caminos para llegar a su destino, cada servidor dispone de su propio recurso de conexión.

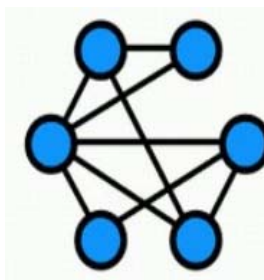


Figura 0.711 Topología tipo malla

Fuente: (Molina)

- **Seguridad.** - Son los diferentes parámetros que se encargan de responder la confidencialidad, autenticidad, e integridad de datos
- **El medio de Transmisión.** - son las vías o caminos que utiliza la tecnología para enviar la información, a través de pulsos eléctricos o electromagnéticos.

Se diferencia en los medios de transmisión:

- **Medio físico** como su nombre lo indica es la transmisión de información, se lo realiza a través de medios físicos como hilos de cobre que en redes de computadoras el cable multipar es el más utilizado en la Figura. 1.22. se muestra un *patch cord* categoría 5, según avanza la tecnología se requiere de medio físicos que brinden una mayor velocidad como por ejemplo la fibra óptica.

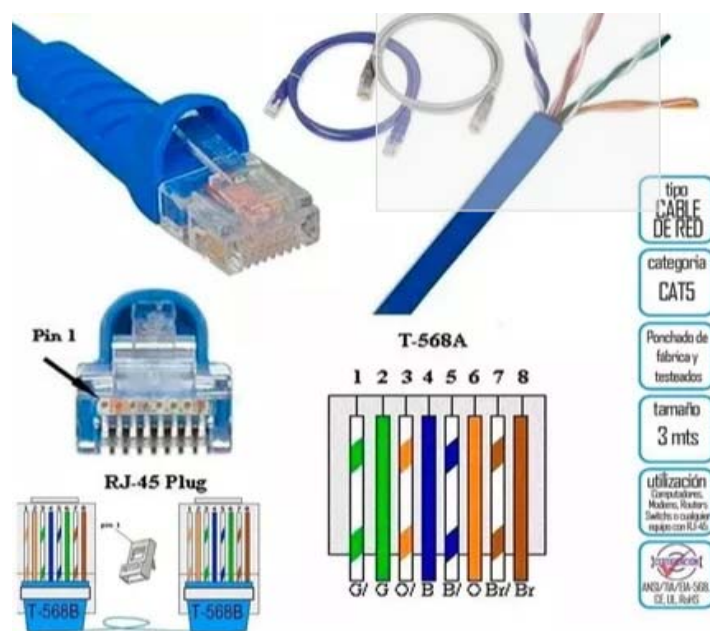


Figura. 0.722. Cable de red categoría 5

Fuente: (Molina)

- **Medio inalámbrico** utiliza como medio de transmisión el aire o espacio libre, o técnicamente hablando el espectro electromagnético comprendiendo el rango de frecuencias desde 3KHz a 300GHz. (Enciclopedia-Britanica), en la Figura. 1.23. se observa el rango de frecuencias del espectro electromagnético utilizado en telecomunicaciones.

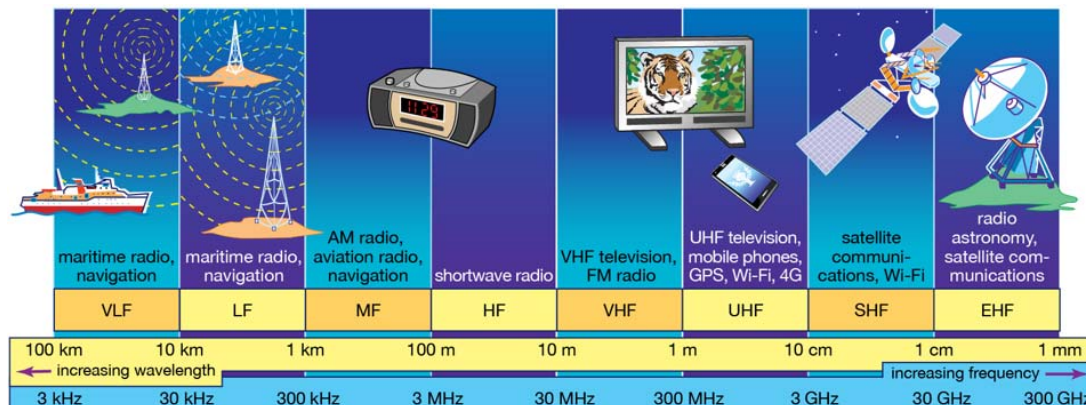


Figura. 0.733. Espectro electromagnético bandas comerciales

Fuente: (Enciclopedia-Britanica)

El rango de frecuencias más utilizadas en comunicaciones inalámbricas es:

- **Frecuencia Infrarrojos IR (300 GHz y 400THz).** Son utilizadas para comunicaciones punto a punto, teniendo como limitante que no atraviesan objetos, se lo utiliza comúnmente en control remoto de televisores, también se los utilizaba no comúnmente para comunicación de equipos electrónicos que prácticamente se encontraban uno al lado de otro
- **Microondas (1GHz a 300GHz)** el rango de frecuencias es utilizado especialmente en telecomunicaciones satelitales, terrestres punto a punto, son una alternativa del cable coaxial y de la fibra óptica. Son utilizadas también en comunicaciones inalámbricas más habituales como las *WLAN*, *Bluetooth*, *UMTS*. Las microondas son ondas mayormente direccionales.
- **Radio frecuencia RF (300 Hz 300 GHz).** Este espacio electromagnético es utilizado especialmente en transmisión de FM, AM y transmisión de televisión digital terrestre, pueden atravesar obstáculos con relativa facilidad y son omnidireccionales.

- **Demás frecuencias** del espectro radioeléctrico eléctrico como: rayos X, rayos Gamma de mayor frecuencia, en teoría brindarían mayores prestaciones, no se los utiliza debido a que pueden ser peligroso para los seres vivos, además de su dificultad de producir y su modulación.

1.7. Tipo de redes inalámbricas

1.7.1. Modo infraestructura

Este tipo de red inalámbrica tiene un punto de acceso WLAN como parte central de la red. La comunicación entre los dispositivos se realiza a través del enrutador, el cual permite la comunicación con una red cableada, de esta forma los dispositivos podrán comunicarse dentro de la red inalámbrica como también tendrían acceso a la red cableada. (Brother, 2014), en la Figura. 1.24. se aprecia un ejemplo de conexión utilizando el modo infraestructura.

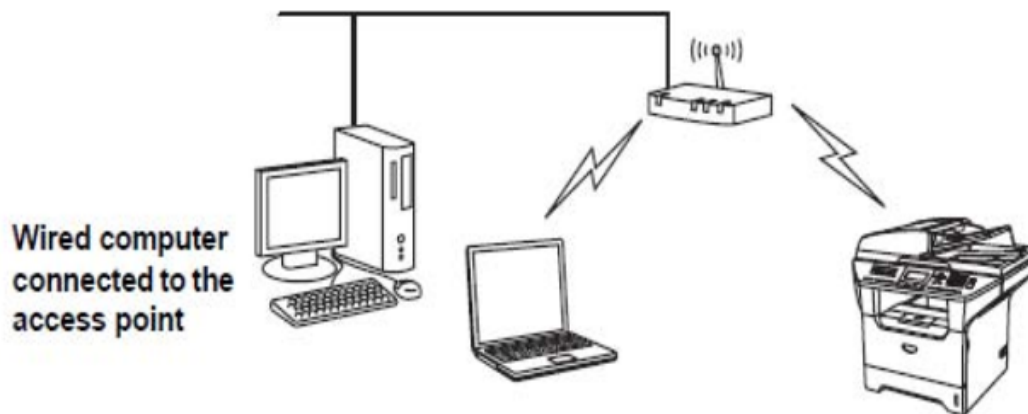


Figura. 0.744. Modo infraestructura

Fuente: (Brother, 2014)

1.7.2. Modo Ad hoc

Es un modo de red inalámbrica que se muestra en la Figura. 1.25. y que a diferencia del modo infraestructura, no dispone de un enrutador o punto central, cada dispositivo inalámbrico se comunica directamente con el otro. (Brother, brother at your side, 2014).

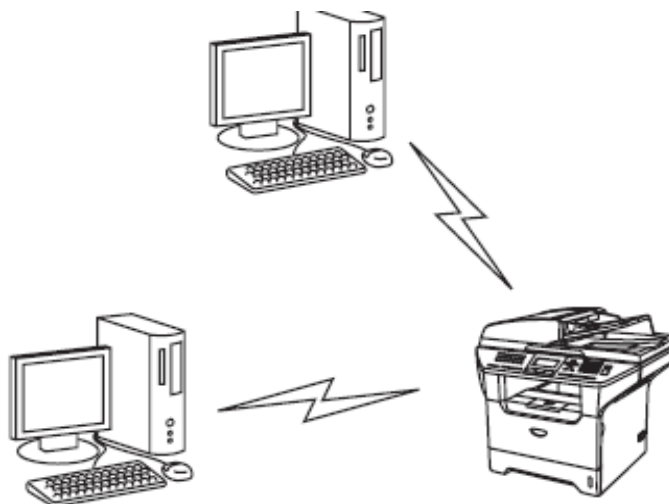


Figura. 0.755. Modo Ad-hoc

Fuente: (Brother, 2014)

1.7.3. Redes Wifi

A nivel mundial es cada vez más fácil conectarse a internet inalámbricamente, según la norma 802.11, (dedicada a redes inalámbricas banda ancha). El acceso a internet se ha convertido en una necesidad básica para el desarrollo de la vida moderna es así como instituciones estatales: municipios, universidades, bibliotecas han creado infraestructura para acceder libremente a redes inalámbricas con la finalidad de reducir la desigualdad digital e infraestructura.

En muchos países se han planteado crear lugares de acceso a internet, debido a que la tecnología Wifi no requiere licencia de uso del espacio electromagnético, pero a su vez su gran limitante es el alcance para cubrir una zona similar a la que cubre una antena celular 4G, la cual paga millones de dólares por el uso del espectro electromagnético esto se ha convertido en un obstáculo financiero para la implementación de estas redes. (Molina)

El estándar 802.11, según la IEEE pone en conocimiento una serie de especificaciones de las cuales se mencionan las siguientes (IEEE.ORG, 2011).

- **IEEE 802.11a** este protocolo es dirigido a la transmisión de paquetes, aunque no brinda una calidad de servicio óptima, la velocidad de transmisión es de 54Mbps, en la frecuencia de 2.4 GHz
- **IEEE 802.11b** inicialmente conocida como WI-FI, la velocidad de transmisión es de hasta 11Mbps, en la banda de frecuencias 2.4GHz.
- **IEEE 802.11g**. Evolución de la IEEE 802.11b, la velocidad máxima de transmisión es de 54Mbps, frecuencia de trabajo 2.4GHz.
- **IEEE 802.11i** se desarrolló con la finalidad de mejorar las seguridades en protocolos de autenticación y codificación, este estándar abarca los protocolos 802.1x, TKIP y AES a través del protocolo WPA2.
- **IEEE 802.11n**, trabaja en las frecuencias 2.4 GHz y 5 GHz, con velocidad máxima de 600 Mbps, este tipo de protocolo es compatible con los diferentes dispositivos según la norma IEEE 802.11, brinda la facilidad de trabajar en la frecuencia de 5 GHz, lo que la hace más fiable debido a que esta banda no está saturada.

1.7.4. Tecnología Zigbee

El desarrollo del protocolo 802.15.4, tiene como función principal la optimización de los recursos de una red, es decir su uso se enfoca, en redes de sensores o redes inalámbricas WSN, debido al tiempo en que se realiza el censado de información se los realiza en tiempos establecidos, no requieren una red tan robusta como una red de computadores.

Zigbee es considerado el protocolo más prometedor en lo que se refiere a redes de sensor inalámbrico así se observan sus características en Tabla. 1.8. el protocolo define la capa física y de control de acceso a las redes inalámbricas. La topología de configuración Zigbee es: árbol, malla o estrella. (Non-IP Smart Object Technologies, 2010).

Tabla. 1.8. Parámetros del protocolo Zigbee

Frecuencia de trabajo	2.4GHz
Alcance (Transmisor 1mW)	100m línea de vista 30m en interiores.
Velocidad	Hasta 256 Kbps
Numero de dispositivos por red	65535

Fuente: (Aprendiendo Arduino, 2016)

1.8. Tipos de dispositivos zigbee

Se presenta la confirmación de una red Zigbee de la Figura. 1.26.

- Coordinador Zigbee responsable de que la red encienda y funcione correctamente en su conjunto.
- Router Zigbee, es responsable de realizar el enrutamiento para el envío de los paquetes
- Por último, el dispositivo final (dispositivo de monitoreo),

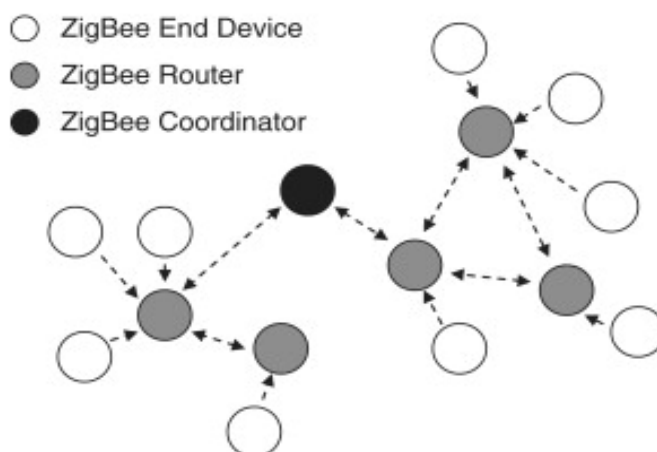


Figura. 0.26. Dispositivos Zigbee

Fuente: (Non-IP Smart Object Technologies, 2010).

CAPÍTULO 2

MARCO METODOLÓGICO

2. Metodología documental

La metodología utilizada en el proyecto será de tipo documental puesto que se aplicarán principios descubiertos en el funcionamiento de dispositivos electrónicos y el proceso de datos. Esta permitirá escoger los elementos idóneos para acondicionar los diferentes circuitos electrónicos y de control, partiendo de las características técnicas que entrega el fabricante en sus hojas técnicas, para un correcto uso de los componentes electrónicos como sensores para la adquisición de datos, procesadores de información, antenas *transceiver* para la comunicación bidireccional entre otros dispositivos. También partir de experiencias anteriores del manejo y configuración de estos dispositivos y comparar sus experiencias

Los componentes deben cumplir ciertos requisitos para su correcta calibración para asegurar un funcionamiento óptimo, así como respetar valores de consumo de energía cuando están en uso, para ello se debe obtener la suficiente información provista por el fabricante para utilizarlos efectivamente.

2.1. Método experimental e inductivo

Para la parte de diseño se utilizará el método experimental e inductivo, debido a que se deberá acoplar componentes electrónicos y componentes de comunicación, experimentar en la práctica su funcionamiento.

El método experimental consiente en relacionar el hardware con el software del proyecto, puesto que realizará el monitoreo y se obtendrá mediciones a través de sensores

de los fenómenos eléctricos, dichos valores medidos podrán ser comparados con valores adquiridos por instrumentos de medida comerciales.

CAPÍTULO 3

3. LA PROPUESTA

El proyecto se compone de las siguientes etapas, como se identifica en la Figura. 3.1.

- Módulos inalámbricos para la toma de medidas de voltaje y corriente.
- Implementación de una red de sensores, en topología estrella.
- Recopilación de datos en ThingSpeak.
- Visualización de datos de consumo de energía
- Control consumo
- Automatización del sistema

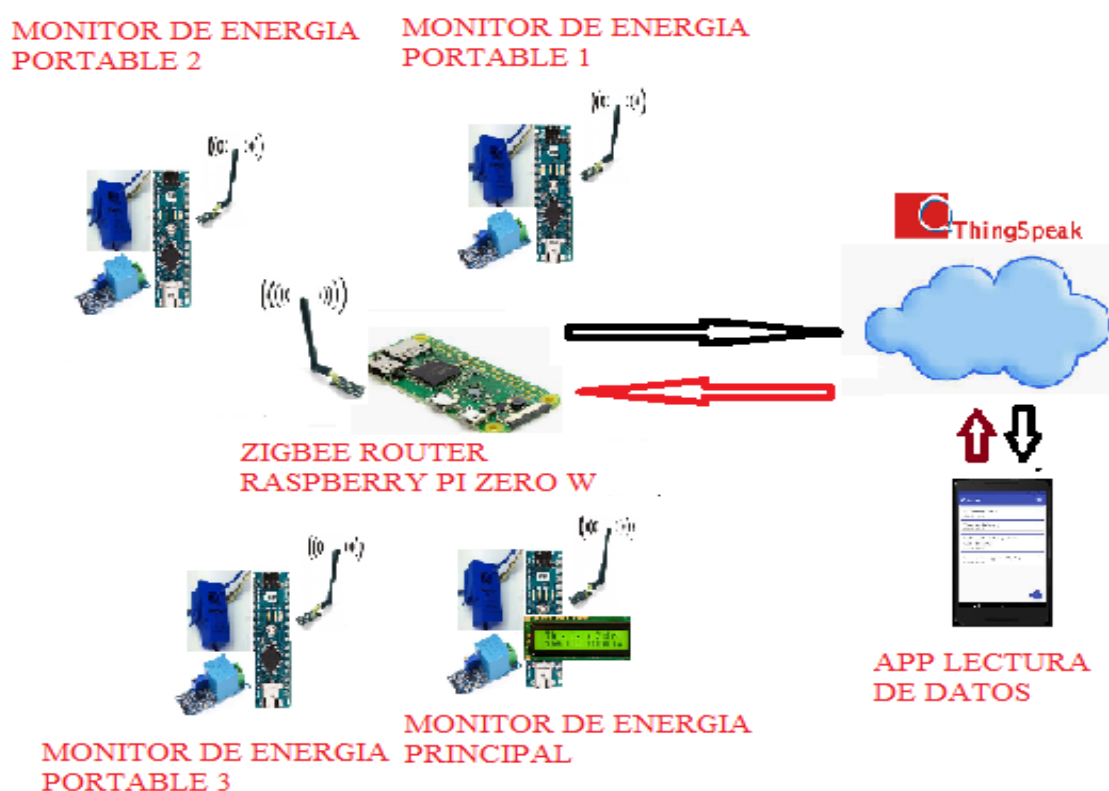


Figura. 0.76. Arquitectura del prototipo.

Fuente: el autor

3.1. Módulos inalámbricos para la toma de medidas de voltaje y corriente.

El sistema de monitoreo consta de tres terminales inalámbricos portables y uno principal. Los terminales portables están conformados en su interior por el Arduino nano, al cual se conectarán dos sensores en su entradas del convertidor análogo digital ADC, el primero es un sensor de corriente no invasivo SCT 013 y un segundo sensor de voltaje ZMPT101B, los dos sensores toman muestras de los valores de corriente y voltaje que es consumido por el aparato eléctrico, conectado en el módulo de monitoreo, dichos valores son almacenados en una variable flotante y son utilizados para calcular la potencia consumida, se realiza un muestreo de la señal de entrada en un periodo de tiempo de 10 minutos tiempo en que recibe la orden del nodo central, para recolectar la información

El terminal principal, a diferencia de los portables presenta valores de voltaje, corriente y potencia consumida por todo el sistema eléctrico del domicilio en un LCD de 16x2. En la Figura. 3.2. se observa el diagrama de bloques del módulo de adquisición de datos transmisor

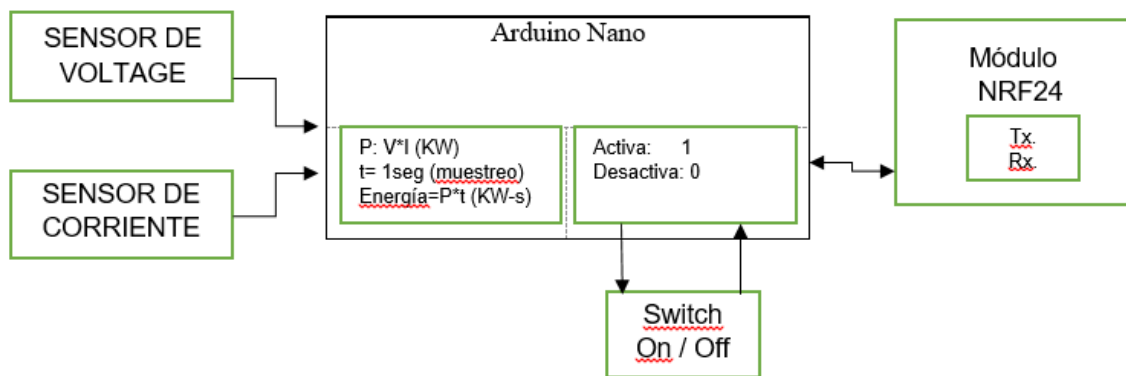


Figura. 0.77. Diseño del módulo transmisor, receptor

Fuente: El autor

En cuanto al acondicionamiento de señal de Relé, además de permitir un monitoreo del consumo de energía, en cada módulo se ha asignado una salida digital de la plataforma Arduino para controlar un relé. En la Figura. 3.3. se muestra la configuración implementada a la salida digital de la placa Arduino, que al aplicar un voltaje en la base del transistor configurado en la zona de corte y saturación permite o no activar el relé. El diodo garantiza

la descarga de corriente almacenada en la bobina durante los períodos de transición entre encendido y apagado.

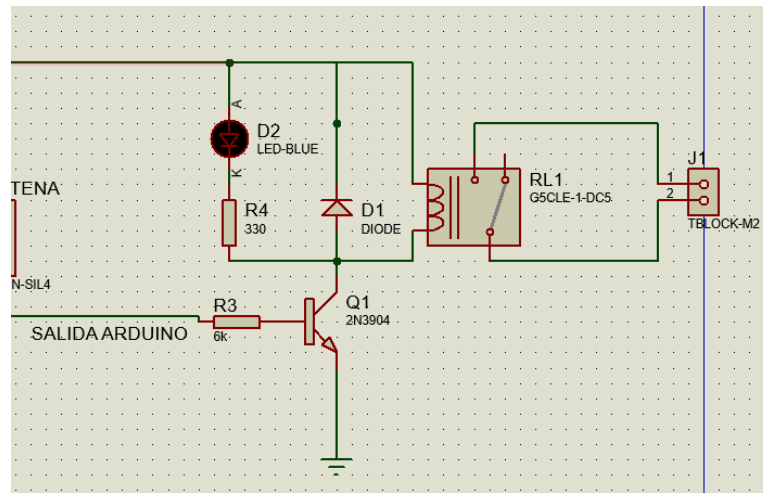


Figura. 0.78. Diseño del circuito de control relé

Fuente: El autor

Respectos a los cálculos, es necesario considerar que el transistor PNP trabaje en corte y saturación:

- | | | |
|--------|-------------------------|--------------------------|
| DATOS: | $V_{CE}=0\text{ V}$ | $\beta=100$ (Data sheet) |
| | $V_{CC}=V_C=5\text{ V}$ | $I_C=\beta I_B$ |
| | $R_{Relé}=70\Omega$ | |

Malla 1

$$V_{CC}=V_C+V_{CE}$$

$$V_C=R_C \cdot I_C$$

$$I_C = \frac{V_C}{R_C} = \frac{5}{70\Omega} = 71\text{mA}$$

Malla 2

$$V_{SA}=V_{RB}+V_{BE}$$

$$I_B=714\mu\text{A}$$

$$R_B = \frac{V_{SA}-V_{BE}}{I_B} = \frac{4.3}{714\mu\text{A}} = 6020[\Omega]$$

3.1.1. Técnica de medición

Para calcular la energía total consumida por una vivienda, es necesario realizar la medición tanto de la corriente como el voltaje que ingresa a través del cable de acometida. De la misma manera para monitorear el consumo de varias cargas puntuales dentro del domicilio, se deben tomar el valor de voltaje y corriente que circulan en el conductor que tiene origen en el tablero de distribución y termina en él toma corriente de un ambiente del domicilio.

3.1.2. Valor RMS o eficaz.

El valor RMS, es aquel que se obtiene después de realizar la raíz cuadrada del valor medio al cuadrado. El valor RMS equivalente de una señal sinusoidal es aproximadamente $1/\sqrt{2} = 0.707$ del valor máximo de su amplitud también denominada valor eficaz, calculado a partir de la expresión:

$$v_{RMS} = \sqrt{\frac{1}{T} \int_0^T V^2 * dt}$$

A continuación en la Figura 3.4 se observará una señal sinusoidal en la cual se indican los valores RMS y máximo de la señal.

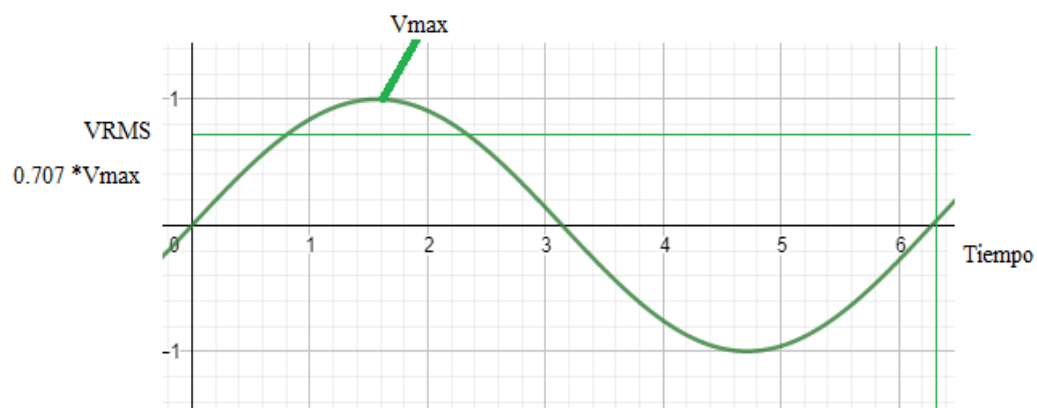


Figura. 0.79. Valor eficaz de una señal sinodal

Fuente: el autor

3.1.3. Medición de voltaje

Para realizar la medida de los diferentes suministros de energía en puntos determinados o principales de una red eléctrica se utiliza sensores, los cuales son parte de un sistema de medición de voltaje, toman muestras de la señal sinusoidal y el algoritmo en la programación de Arduino, calcula el valor de voltaje RMS, ecuación 3. En la Figura. 3.5. se muestra el diagrama del circuito de acoplamiento que permite referenciar la señal que ingresa al convertidor análogo digital, que ingresa en la entrada analógica del microprocesador.

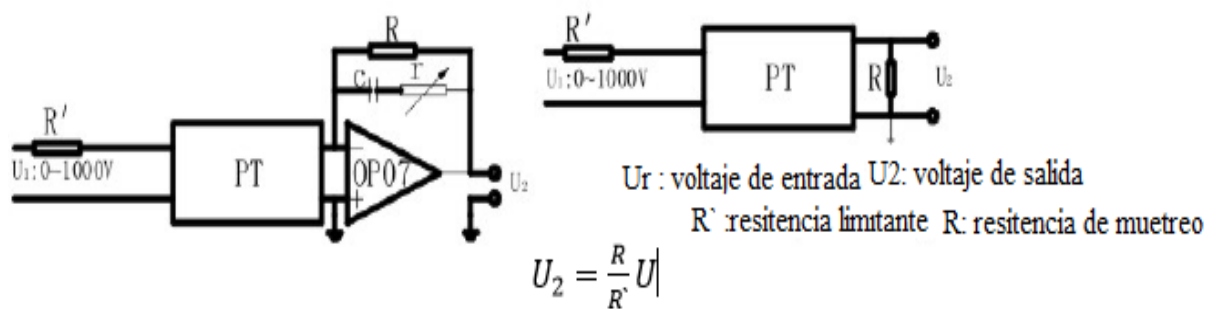


Figura. 0.80. Diagrama sensor ZMPT101B

Fuente: (Nanjing Zeming Electronic Co., n.d.)

Se hace referencia a la señal de entrada al microprocesador con un valor de 2.5 V, teniendo en consideración que la placa Arduino no trabaja con valores negativos es decir, el circuito de acoplamiento proporciona a las entradas de Arduino los valores entre 0-5 VAC.

3.1.4. Medidor de corriente.

Los sensores o transductores de corriente obtienen el valor de corriente que circula en un conductor, entregando a la salida un voltaje o una corriente proporcional a la corriente que circula por el conductor, debido a su facilidad de uso los sensores de efecto hall, son muy utilizados ya que no son invasivos, tipo pinza aunque también existen este tipo de invasivo es decir se debe intervenir el conductor en el cual se quiere adquirir los datos.

En la Figura. 3.6. se puede observar el circuito de acoplamiento del sensor, donde se detallan el punto de referencia a 2.5 V DC, con la finalidad de únicamente suministrar valores positivos de voltaje, entre 0 y 5 V DC.

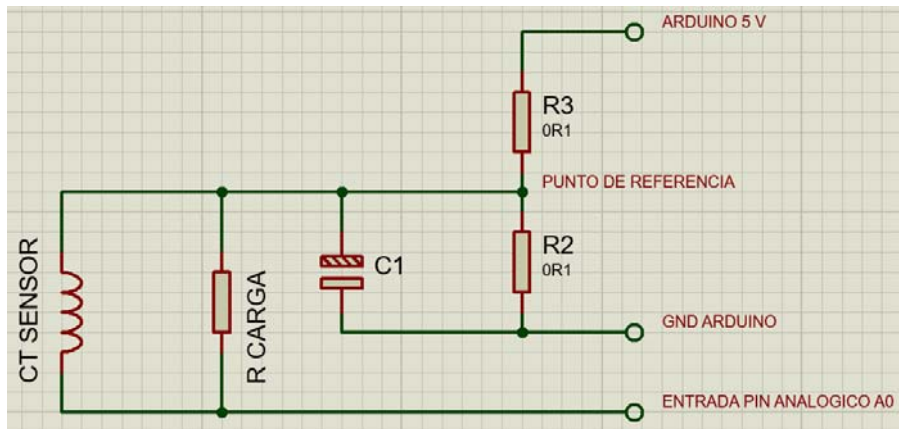


Figura. 0.81. Esquema de medición de voltaje

Fuente: (OpenEnergyMonitor)

DATOS:

$$V_0 = 5 \text{ V}$$

$$V_{REF} = 2.5 \text{ V}$$

$$V_{REF} = \frac{R_2}{R_1 + R_2} * V_0$$

$$R_1 = R_2$$

3.1.5. Procesador de datos

Para procesar los datos recolectados por los sensores de corriente y voltaje y calcular la energía consumida en la vivienda o en ambientes específicos, se utilizará una placa electrónica de desarrollo Arduino. En la actualidad Arduino se ha constituido en una de las placas para el desarrollo de proyectos electrónicos más utilizadas en la implementación de proyectos, su software y hardware están basados en tecnología libre, así como también el usuario tiene a disposición un sinnúmero de librerías que facilitan su uso, el lenguaje de programación que utiliza para su programación es C++.

3.1.6. Arduino nano.

Es una placa bastante compacta como se aprecia en la Figura. 3.7. con dimensiones de 1.83 cm x 4.25 cm, difiere del Arduino UNO el tipo de conector USB tipo B por un *micro Jack* de alimentación tipo B y sus pines son tipo *header*. (ANEXO C).

Arduino nano pone a disposición 14 pines que pueden ser utilizados como salida o entrada digital, de estos, 6 pueden utilizarse como PWM, además se dispone de 6 entradas analógicas y un cristal de 16Mhz y lo más importante posee un procesador de tecnología Atmega328P.

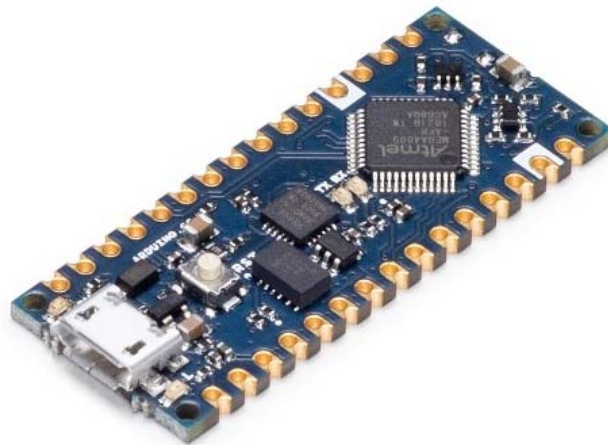


Figura. 0.82. Arduino Nano

Fuente: (Arduino, n.d.)

3.1.7. Arduino IDE

Para el desarrollo de los algoritmos para Arduino se dispone de una plataforma de *software* libre desarrollado por Arduino del mismo nombre, es de tipo IDE (entorno de desarrollo integrado), donde se puede realizar la compilación del código y enviarlo al chip mediante cable USB, otra herramienta que brinda este software es el monitor serial, la cual permite realizar pruebas rápidas de funcionamiento del código. El lenguaje de programación es una versión simplificada de C++, lo que permite importar y utilizar varias de sus librerías.

3.2. Implementación de una red de sensores, en topología estrella.

La topología de red utilizada para el desarrollo del presente proyecto será tipo estrella configurada en una red ad-hoc, este tipo de red se utiliza para realizar configuraciones de red de sensores: monitoreo general, salud, vigilancia, etc. Se utiliza redes de sensores para optimizar recursos o cuando no se dispone de una red inalámbrica modo infraestructura a su vez, utiliza una Raspberry Pi Zero W, configurada como nodo central y los sensores se los configurará como terminales.

Para la comunicación inalámbrica de los sensores, se utilizará el módulo NRF 24, que trabaja en la frecuencia 2.4GHz, el fabricante pone a disposición la librerías de código abierto tmrh20, que construye los parámetros TCP/IP sobre este módulo inalámbrico para su configuración, debido a su bajo costo en comparación con un módulo Zigbee ya que dispone de características similares de funcionamiento, hacen que este módulo sea bastante atractivo para el uso en redes de sensores. (Himadrinath, Shashwata , Shinjan , Soham , & Urmi , 2017)

El nodo central, se conecta a los terminales tanto los portables como el principal que cumplen la función de monitorear el consumo de energía del domicilio. Raspberry Pi es la encargada de solicitar la actualización de información adquirida por los módulos de medición cada diez minutos, dicha información es enviada para ser almacenada en la Raspberry pi que genera un documento de texto por cada dispositivo, la información recibida de cada módulo de monitoreo será almacenada durante 60 minutos, luego del cual Raspberry pi enviará esta información para su almacenamiento en la nube. (ThingSpeak)

3.2.1. Características de los nodos remotos

Con la intención de crear una red con modularidad, se ha decidido que los nodos remotos no se conozcan, con la intención que cada módulo sea independiente, es decir si existe la necesidad de reemplazar o modificar algún nodo, este proceso no afecta el funcionamiento de la red. También se consigue con esto que los nodos remotos únicamente se centren en la comunicación exclusiva con el nodo central y si ha futuro existiría la necesidad de crear nuevos módulos la implementación será más sencilla. Los nodos remotos

contaran para el procesamiento y control con Arduino nano ya que tiene las características necesarias para cumplir con las necesidades de la red planteada

3.2.2. Características del nodo central

Dado el trabajo que va a realizar el nodo central, este debe contar con la capacidad de almacenamiento de información, pues en este nodo es donde convergen las diferentes variables recogidas en los nodos remotos, el acceso y gestión de la información debe ser de manera pronta y fácil.

En función de las características requeridas, se ha escogido Raspberry pi puesto que puede brindar prestaciones similares a un PC de escritorio. Es un dispositivo de costo relativamente bajo.

Las características que debe cumplir el nodo central son:

- Conexión a una internet
- Interfaz de propósito general de entrada y salida.
- Un sistema operativo amigable, que permita la gestión de almacenamiento y configuración de las características anteriores.
- Un bajo consumo de energía.

Raspberry pi utiliza para almacenamiento un tarjeta micro SD, donde se instala el sistema operativo creado por los mismos desarrolladores de Raspberry.

Una vez que la Raspberry pi se encuentre operativa se debe configurar algunos parámetros como: asignar una IP estática, se habilita el protocolo SSH para poder acceder a las líneas de comando remotamente, la Raspberry también permitan la conexión a internet, sin olvidar la configuración de los puertos de entrada y salida de propósito general para el control de periféricos.

3.2.3. Comunicación Inalámbrica.

Una vez elegida la plataforma en la cual se trabajara, al mismo tiempo se generó la necesidad de contar con un medio de comunicación inalámbrica que permita la comunicación entre los diferentes nodos, luego de comparar algunas alternativas en el mercado, se concluyó que es una buena opción trabajar con el módulo NRF 24L01+, debido a su bajo consumo de energía, su bajo costo, tiene la suficiente potencia para cumplir con los requisitos del proyecto, además tiene a disposición de una gran cantidad de información documentada, como la facilidad de trabajar con las plataformas de control como: Arduino y Raspberry y todo esto a un costo relativamente bajo.

3.2.4. Módulo Transceiver NRF 24L01

Es un dispositivo electrónico desarrollado por la compañía *Nordic Semiconductor*, cuyo chip integra un *transceiver* completo que trabaja en la frecuencia 2.4GHz, además está provisto de un preamplificador LNA (*low noise amplifier*), también tiene un monitor de banda base (*Enhanced ShockBurst™*), diseñado para operar en redes inalámbricas de bajo consumo de energía.

El módulo NRF 24L01, utiliza el estándar de comunicación SPI (*Serial Peripheral Interface*) y 8 pines para la su interfaz de conexión en la Figura. 3.8. se aprecia el *transceiver* que es de tamaño reducido y tiene un gran alcance.



Figura. 0.83. Módulo transceiver NRF 24L01

Fuente: (LOZURITECH, 2016)

En la siguiente Tabla. 3.1. se presenta varias de las características técnicas del módulo para la comunicación inalámbrica NRF 24LO1, (ANEXO C).

Tabla. 3.1. Características técnicas transceiver

Voltaje de funcionamiento:	1.9 V – 3.3 V
Voltaje puerto IO	0~3.3v / 5v
Nivel de salida	7 dB
Sensibilidad RX	≤ -90dB
Alcance	15 – 30 metros, espacios cerrado Hasta 100 metros línea de vista
Velocidad de transmisión de datos	250Kbps- 1Mbps hasta 2 Mbps
Consumo de energía	En el orden de los 20 μ A

Fuente: (LOZURITECH, 2016)

TMRH20 ha desarrollado varias librerías que pueden ser utilizadas para configurar el módulo NRF 24 para trabajar con Arduino como también es compatible para trabajar con Raspberry Pi. En la Tabla. 3.2. se observan las librerías TMRH20 y su equivalente en TCP/IP (Himadrinath, Shashwata , Shinjan , Soham , & Urmi , 2017).

Tabla. 3.2. Librerías de red NRF 24L01

Librería TMRH20 RF 24	TCP/IP Equivalente
RF 24 MESH	Capa de aplicación (DHCP)
RF 24 ETHERNET	Capa de transporte
RF 24 NETWORK	Capa de internet/red
RF24	Capa de enlace

Fuente: (Himadrinath, Shashwata , Shinjan , Soham , & Urmi , 2017)

El chip NRF implementa una variación del protocolo ANT (hasta la capa de transporte). El enrutamiento de paquetes en la red multi hop es manejado por la biblioteca TMRH20, driver que implementa un algoritmo basado en el protocolo para MANET llamado "Administrador y protocolo de enrutamiento seguro basado en fidelidad" (AFSR), que garantiza el enrutamiento seguro a través de la red, eligiendo un nodo administrador en función de la voluntad y la fidelidad, después de lo cual un nodo solo se comunica con esa seguridad. (Himadrinath, Shashwata , Shinjan , Soham , & Urmi , 2017).

3.2.5. Protocolo de comunicación SPI

La comunicación PSI, se la puede entender como el envío de información desde un transmisor y la recepción de esta por parte de un receptor, o sencillamente intercambio de información entre dos dispositivos.

El protocolo de comunicación inalámbrica que utiliza NRF 24 es el SPI, protocolo de comunicación serial presentado en la Figura. 3.9. Hoy en día es uno de los protocolos más populares para la comunicación entre microcontroladores debido a su simplicidad y velocidad de transmisión. El protocolo SPI trabaja a modo full dúplex, permite la comunicación bidireccional entre el nodo central con las terminales remotas por canales diferentes, al ser un protocolo síncrono se dispone de una canal adicional para el sincronismo.

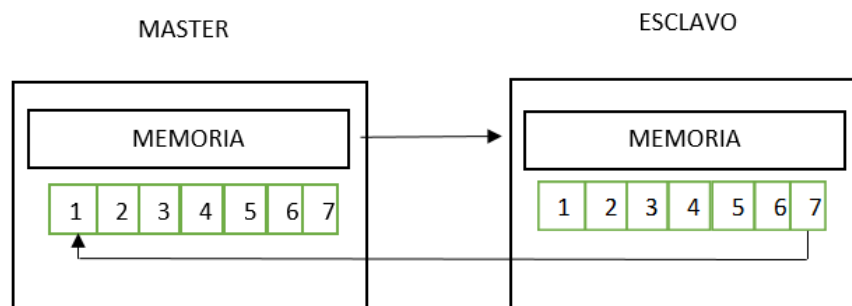


Figura 0.984 Estructura principal del protocolo PSI

Fuente: (Sanz Fernández, 2014)

3.2.6. Algunas otras características

- Dispone de un sistema de acuse de recibido, o de auto reconocimiento automático sin la intervención del microcontrolador este sistema se lo conoce como *Enhanced Shockurst*. Al enviar el transmisor un paquete de información automáticamente el receptor enviara un mensaje de haber recibido un paquete de datos. (Sanz Fernández, 2014)
- La verificación y validación de la información se lo realiza mediante una dirección de 3- 5 bytes así como también dispone de un código verificación de redundancia cíclica (CRC) de 1 o 2 bytes.
- Es un sistema de seis canales para la recepción de datos.
- Tasa de transmisión al aire libre de 1 a 2 Mbps
- Dispone de 125 canales de transmisión de RF
- El chip requiere una alimentación de entre 1.9 y 3.5 VDC

3.2.7. Canales de recepción y paquetes da datos

El módulo de comunicación inalámbrica NRF 24L01+ dispone de un canal de transmisión TX y 6 canales de recepción RX es decir, puede recibir información de seis dispositivos diferentes los cuales se encuentran debidamente identificados desde su origen por la dirección del paquete, que permite la comunicación de un dispositivo a la vez, en la Figura. 3.10. se ilustra una posible configuración entre un nodo central y seis transmisores, en una misma frecuencia de trabajo.

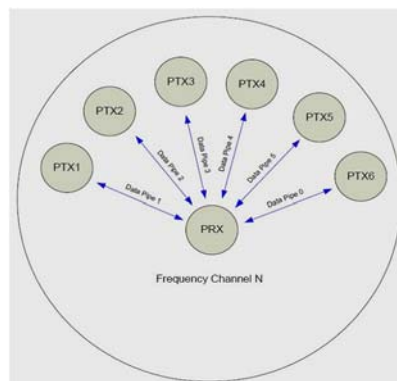


Figura. 0.850. Recepción de información por varios canales.

Fuente: (Sanz Fernández, 2014)

El envío y recepción de información por varios canales es factible gracias a su estructura de paquete que incluye de 3 a 5 bytes (por defecto 5) para la verificación y validación de la información, según Figura. 3.11.

Preamble 1 byte	Address de 3 – 5 bytes	9 bits	Payload 1-32 bytes	CRC 1-2 bytes
--------------------	---------------------------	--------	-----------------------	------------------

Figura. 0.11. Representación del paquete de datos.

Fuente: (Sanz Fernández, 2014)

Protocolo Enhanced StockBurst

Este protocolo basa su funcionamiento en el reconocimiento automático para la validación, verificación y reenvío de paquetes de datos de hasta 32 bits. Durante el proceso de recepción de datos, se establece un protocolo de seguridad para evitar fallos, únicamente se procesa el resto del paquete de información cuando se ha validado la señal de validación correctamente, entonces se procede a la verificación de CRC, solo si el este código de validación es correcto, se traslada el paquete información al buffer RX FIFO, para su transmisión o almacenamiento.

3.3. Plataforma IoT y su comunicación

ThingSpeak es una plataforma creada para desarrollo de proyectos IoT, esta plataforma de software libre almacena toda la información en la nube, a la cual el usuario puede acceder de una manera fácil y rápida, pudiendo utilizarla como más le convenga. Los datos de la plataforma se encuentran protegidos por una contraseña API, la cual es controlada por el usuario de la cuenta, al iniciar sesión el beneficiario puede acceder a su información e incluso descargarla si es necesario, los formatos en los que se accede a los datos acumulados en la nube mediante el desarrollo de aplicaciones son: CSV o JSON, a través API REST la cual permite crear, actualizar gráficos y canales. La transferencia de estado representativo de tipo (REST), es un estilo arquitectónico diseñado como un modelo de solicitud-respuesta que se comunica a través de HTTP, para utilizar el API REST es obligatorio la palabra clave adecuada. (TINKSPEAK, 1994-2020).

En la Figura. 3.12, se presenta la estructura de configuración de una red de sensores, los cuales publican la información en un servidor de internet como ThingSpeak

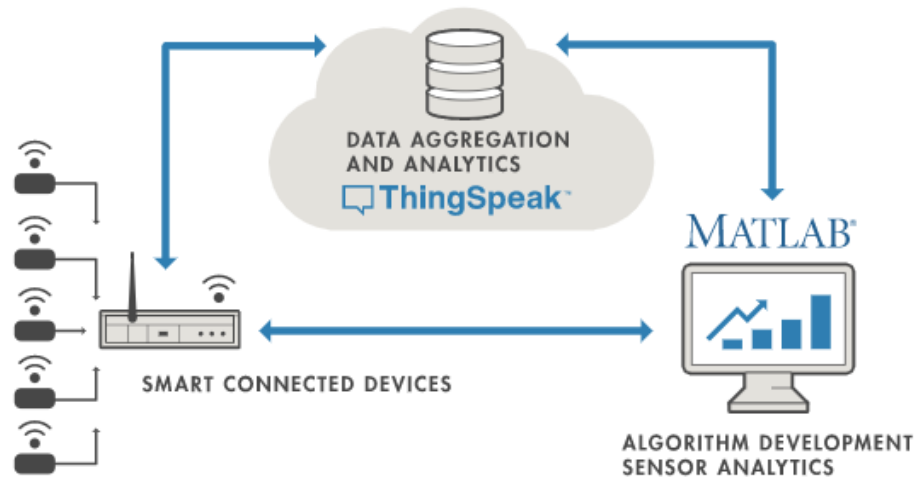


Figura. 0.12. Diagrama de dispositivos conectados a ThingSpeak

Fuente:(TINKSPEAK, 1994-2020)

ThingSpeak, es la plataforma de almacenamiento de 8200 mensajes al día y alrededor de 3.000.000 de mensajes por año en su versión gratuita será utilizada en este prototipo. Para el cálculo de envíos por día se utiliza la ecuación.

$$\begin{aligned} \frac{\text{Envios}}{\text{Dia}} &= 4 \text{ envios (1 por sensor cada sensor) } * 24 \text{ horas} \\ &= 96 \text{ envios por dia toto el sistema} \end{aligned}$$

Al realizar los cálculos, donde la información que será enviada cada hora al día en un total de 96 datos a los diferentes canales se dispone de 8104 datos que se podrán utilizar para la actualización de estados de los sensores. Para el cálculo del total de envíos por mes se empleará la ecuación:

$$\frac{\text{Envios}}{\text{mes}} = 96 \text{ envios total del sistema} * 30 \text{ dia} = 2880 \text{ envios al mes.}$$

Luego de analizar los cálculos obtenidos, el uso de la plataforma ThingSpeak en su versión gratuita brinda la suficiente capacidad para el almacenamiento de datos que se generaran en el proyecto.

3.3.1. Visualización de datos de consumo de energía

Para acceder y mostrar los datos almacenados en la nube (ThingSpeak), se empleará una herramienta web de uso gratuito y fácil acceso a sus recursos donde se utilizará una aplicación para teléfono celular con plataforma Android que accederá a la información de la cuenta y presentará gráficamente los valores de consumo de energía de cada sensor de medida.

3.3.2. Herramienta de desarrollo de aplicación para Android



Figura. 0.13. Logo de página web para el desarrollo de APP

Fuente: (MIT APP INVENTOR, 2012-2020)

El diseño de la aplicación del celular se realizará en la plataforma web de desarrollo visual e intuitivo en línea que cuenta con más de seis millones de usuarios registrados con una cuenta de Google o Gmail, esta herramienta que basa su funcionamiento en bloques permite que el usuario elabore aplicaciones para tabletas o celulares en poco tiempo. MIT App inventor pretende que el usuario convencional no solo utilice la tecnología, sino que desarrolle la suya. en la siguiente Figura. 3.13. se muestra el logo de página MIT aAPP inventor.

La programación basada en bloques busca el desarrollo intelectual creativo de los jóvenes, niños y aficionados, que marque diferencia, con esto lograr un impacto social. Para acceder a la plataforma MITI APP inventor basta escribir la dirección Url: <http://appinventor.mit.edu/>, donde basta dar un clic en crear aplicaciones y tener acceso a las herramientas para él diseño de la aplicación requerida.

Entorno gráfico de desarrollo: En la Figura. 3.14. se presenta la interfaz de diseño, la cual se divide en cuatro partes.

- **Paleta**, ubicada a lado izquierdo de la pantalla y facilita todos los componentes necesarios para la creación de la aplicación.
- **Visor**, este espacio muestra una pantalla virtual de un celular, la que permite visualizar el diseño de la aplicación así como también se irá agregando los componentes que formaran parte de la aplicación.
- **Componentes**, presenta un listado de todos los componentes utilizados para el diseño de la aplicación.
- **Propiedades** permite seleccionar los componentes, acceder y modificar las propiedades de estos (texto, agregar imágenes, cambiar colores, etc.)

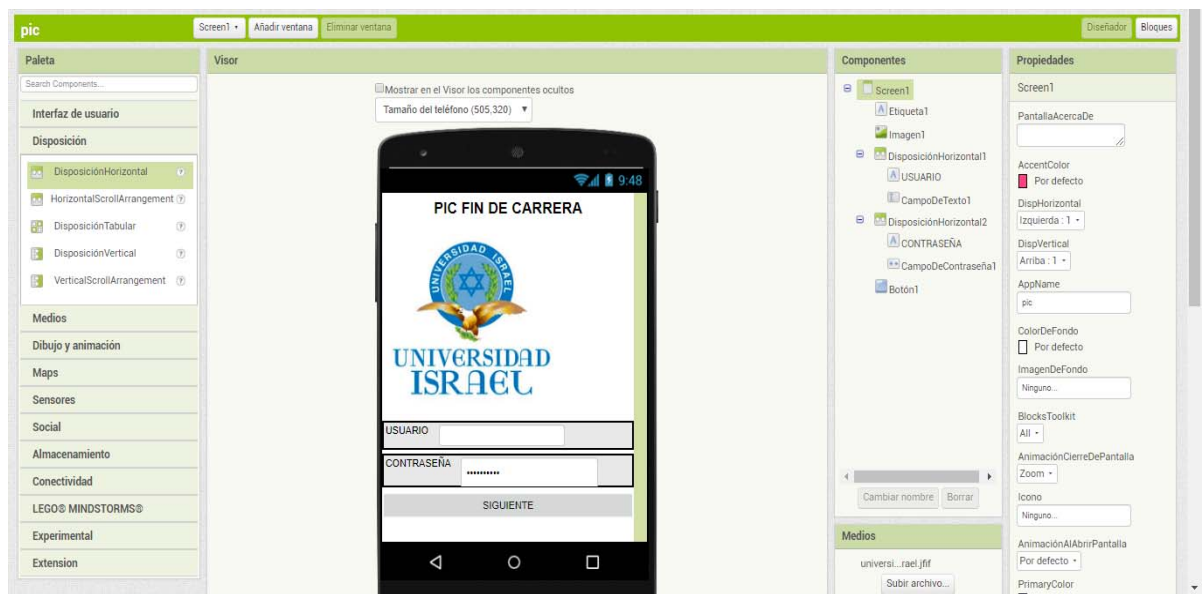


Figura. 0.14. Interfaz de APP Inventor para elegir los componentes

Fuente: autor

Una vez concluido el diseño de la aplicación, se prosigue al diseño de bloques.

Diseño de bloques en la Figura. 3.15. se presenta el entorno para programar en bloques, es donde se realiza la programación que deberán cumplir los diferentes elementos.

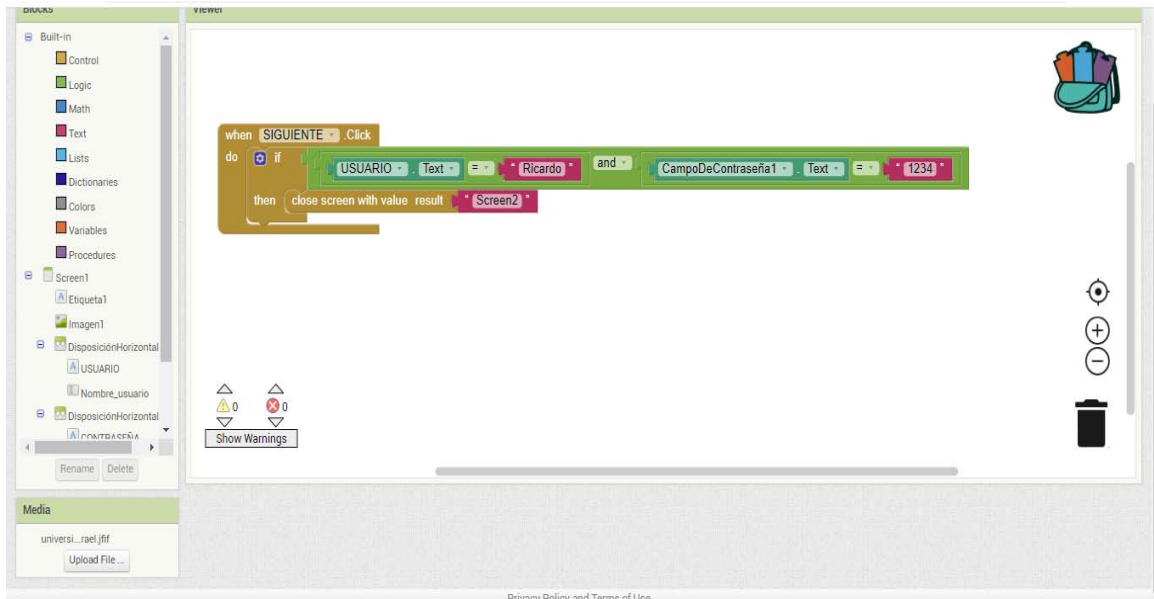


Figura. 0.865. Interfaz de APP Inventor diagrama de bloques

Fuente: autor

Emulado, en la Figura. 3.16 se presenta un emulador del sistema operativo Android, donde se realiza pruebas que permiten probar las configuraciones según se va desarrollando el programa.

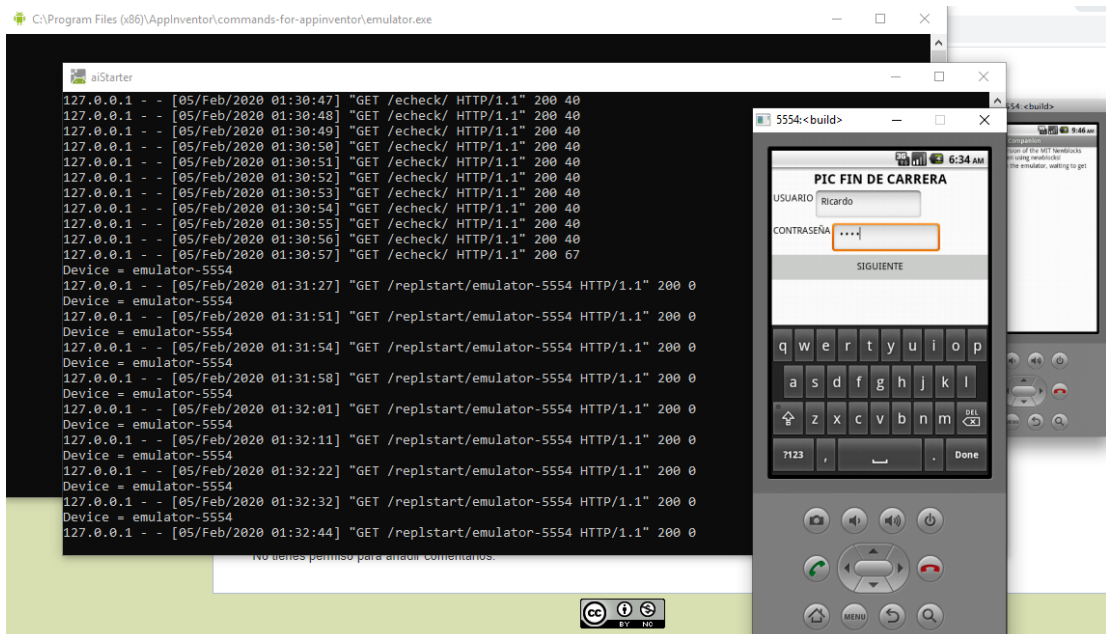


Figura. 0.876. Interfaz de APP Inventor simulador

Fuente: El autor

3.3.3. Diseño de la aplicación observar datos

Para acceder a la información en la base de datos se desarrollará una aplicación que funcionará en la plataforma Android donde esta tendrá:

- **Entorno 1.**-Figura. 3.17 se muestra la primera pantalla para usuario, donde deberá ingresar un usuario y una clave para acceder a los datos almacenados en la nube.



Figura. 0.887. Diseño pantalla 1 ingreso del usuario

Fuente: El autor

- **Entorno 2.**- En la segunda pantalla como se muestra en la Figura. 3.18, el usuario observará los tres módulos de monitoreo, como también mediante alarma grafica sabrá que módulo está activo y con carga. Desde este entorno el usuario tendrá acceso a activar o desactivar cualquier módulo si tiene la necesidad de hacerlo.

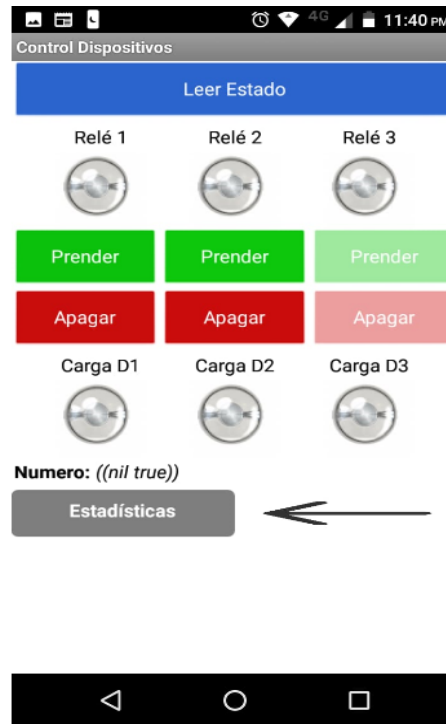


Figura. 0.898. Diseño pantalla 2 estado de dispositivo de monitoreo

Fuente: El autor

- **Entorno 3.-** En el tercer entorno como se puede apreciar en la Figura. 3.19, el usuario podrá escoger el módulo que desea monitorear de manera diaria, semanal y mensual, los datos se mostraran gráficamente en barras.



Figura. 0.19.90 Diseño Screen 3 estado de presentación de datos

Fuente: autor

3.4. Control de Consumo del Sistema IoT basado en la Lectura de Factor de Potencia

3.4.1. Planteamiento del Esquema de Control

Se propone un modelo de control de consumo basado en el monitoreo del Factor de Potencia. A través de la lectura de dicha variable, es posible verificar que el conjunto de cargas conectadas al sistema no genere un valor de Factor de Potencia demasiado bajo, demostrando una poca capacidad de dichas cargas para absorber la potencia activa.

Un bajo Factor de Potencia supone efectos negativos sobre la línea de distribución hacia las viviendas, porque se requieren corrientes mayores para entregar un determinado valor de potencia que demanda la carga.

Por esta razón se incluye en el pliego tarifario un recargo o penalización aplicado a una determinada residencia que no cumpla con un mínimo Factor de Potencia, tal como se muestra a continuación.

$$FP_r = \begin{cases} P_{BFP} = 0 & \text{si } FP_r \geq 0.92 \\ P_{BFP} = B_{FP} \times FSPEE_i & \text{si } FP_r < 0.92 \rightarrow B_{FP} = \frac{0.92}{FP_r} - 1 \end{cases}$$

Donde:

FP_r = Factor de potencia registrado

P_{BFP} = Penalización por bajo factor de potencia

B_{FP} = Factor de penalización

$FSPEE_i$ = Factura por servicio público de energía eléctrica inicial

Figura. 3.20. Pliego tarifario factor de potencia

Fuente: (ARCONEL, 2019)

Como se observa en la Figura. 3.20. la penalización por bajo factor de potencia aplica para consumidores cuyo Factor de Potencia es menor a 0.92. A partir de ello, se propone implementar el esquema de control de consumo precisamente en dicho valor límite, con el fin de garantizar que el sistema sirva como una guía para que el usuario evite ser penalizado.

De esta manera se propone el modelo de control de consumo como un sistema de alerta capaz de notificar al usuario que, el conjunto de cargas que ha sido conectado al sistema se encuentra generando un Factor de Potencia que conducirá a una penalización en facturación por parte de la Empresa Eléctrica. Este esquema de control de consumo además recomendará, a través de la aplicación móvil, el valor del capacitor que debería ser conectado en paralelo para corregir el Factor de Potencia hacia un valor de 0.92.

Considerando el triángulo de potencia de la Figura. 3.21. que se muestra a continuación, es posible definir el procedimiento a seguir para el cálculo del banco de capacitores.

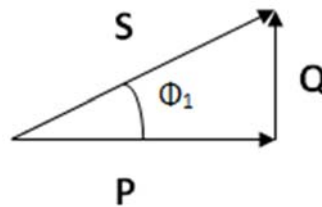


Figura. 3.21. triángulo de potencias

Fuente: (González, 2008).

El Factor de Potencia que entrega el sistema se define como fp_1 , mientras que el Factor de Potencia deseado se define como fp_2 , cuya relación con el triángulo de potencia es:

$$Fp_1 = \cos \phi_1$$

$$Fp_2 = \cos \phi_2$$

Por otro lado, las potencias reactivas pueden calcularse de la manera siguiente:

$$\operatorname{Tg}(\phi_1) = \frac{Q_1}{P} \Rightarrow Q_1 = \operatorname{Tg}(\phi_1) \cdot P$$

$$\operatorname{Tg}(\phi_2) = \frac{Q_2}{P} \Rightarrow Q_2 = \operatorname{Tg}(\phi_2) \cdot P$$

De esta forma, se observa que el valor de Potencia Reactiva que deberá compensarse viene dado por la diferencia entre la Potencia Reactiva existente, menos la P. R. que se busca tener.

$$dif = Q_1 - Q_2$$

Finalmente, se puede determinar el valor de Capacitancia necesaria a través de la ecuación:

$$C = \frac{P}{V^2 \cdot \omega}$$

Donde:

$$\omega = 2 \cdot \pi \cdot f$$

$$P = dif$$

3.4.2. Implementación a nivel del Módulo de Acometida

Se desarrolla para el módulo de la acometida una función que permita calcular el valor del capacitor deseado teniendo en cuenta el factor de potencia al cual se desea llegar y el factor de potencia que se está obteniendo en el tiempo en el cual se realiza la medición esto es, cada hora, lo cual permitirá realizar el cálculo de la potencia reactiva que se debe manejar para llegar a dicho factor de potencia y así obtener el valor de la reactancia requerido. Al final se procede a calcular el valor del capacitor necesario para compensar el factor de potencia y esté dentro de los márgenes que estipula la ley.

$$Q_c = P_{act} \tan(\phi_a) - P_{act} \tan(\phi_d)$$

$$Q_c = V^2 / X_c = V^2 / (1/\omega C)$$

$$C = Q_c / (V^2 * \omega)$$

3.4.3. Implementación a nivel de Raspberry y Servidor

Se desarrolla un programa en Python el cual pide el dato del capacitor calculado del medidor de la acometida mediante comunicación de radio frecuencia. Este script se ejecutará

cada hora en el minuto 57 mediante el uso del Crontab y se lo guardará en un archivo de texto llamado “capacitor.txt”.

En el script “nube.py” encargado de subir los datos de energía a la nube también se utilizará para enviar los datos del capacitor que se necesita para asegurar un factor de potencia mayor a 0.92. Para ello se usa el Field 7 del canal IoT ya utilizado de la página “ThingSpeak”.

3.4.4. Implementación a nivel de la Aplicación Móvil

Se desarrolla una función que lea el último dato del Field 7 en formato .xml del canal IIoT, el cual será ejecutado cada 20 minutos. Si el dato recibido es diferente de cero se notifica el mensaje del capacitor que necesita el usuario poner en su acometida y así asegurar un fp. mayor a 0.92 y evitar sanciones.

3.5. Automatización del sistema

Con la finalidad de que el sistema de monitoreo sea autónomo, es decir que los diferentes módulos no necesiten la manipulación del administrador, sino únicamente se enchufen los diferentes módulos de monitoreo a la toma eléctrica y empiece a trabajar todas sus funciones, se utilizara **Cron**, que se inicia una vez que arranca el sistema operativo.

3.5.1. Cron

Cron proviene del griego cronos = tiempo, es un programa administrador de procesos en segundo plano, que permite ejecutar actividades que han sido programadas a realizarse en una hora y fecha determinada, estos procesos se registran en el fichero Crontab

3.5.2. Crontab

Crontab es un programador de tareas, que utiliza al demonio cron para ejecutar dichas actividades. Crontab es un archivo de texto donde se almacenan una lista de comandos que

se ejecutaran en tiempos especificados por el usuario. Crontab se encarga de confirmar la hora, fecha y permisos de ejecutar de un programa esto se realizará en un segundo plano por lo que es imperceptible para el usuario, para acceder a Crontab, en la se muestra Figura. 3.22. se observa los códigos para empezar a configurar el comando Crontab.

```

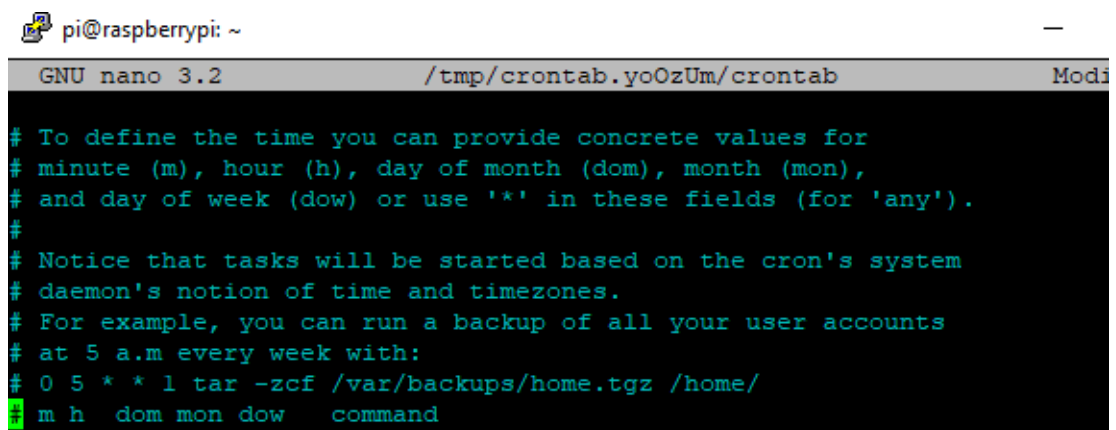
a pi@raspberrypi:~ $ export EDITOR="nano"
pi@raspberrypi:~ $ crontab -l
no crontab for pi
pi@raspberrypi:~ $ crontab -e
no crontab for pi - using an empty one

```

Figura. 0.22. Comando para iniciar el Crontab

Fuente: El autor

En la Figura. 3.23. se observar el fichero del Crontab donde se edita los periodos de tiempo que se realizaran actividades específicas.



```

pi@raspberrypi: ~
GNU nano 3.2 /tmp/crontab.yoOzUm/crontab
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
# m h dom mon dow  command

```

Figura. 0.23. Editor de texto Crontab

Fuente: El autor

Donde:

- m:** minuto en el que se va a ejecutar el script, valor de 0-59
- h:** horas exactas, formato 24 horas valores, 0-23
- dom:** se especifica el día de mes en que se ejecutara el script
- mon:** mes en el que se ejecutara el script.
- dow:** día de la semana 0- 6 donde cero es domingo.
- comando:** ruta completa del script o comando a ejecutar

3.6. Diagrama de la maqueta para pruebas

En la Figura. 3.24. se visualiza el diagrama de un domicilio con varios ambientes, de los cuales se han seleccionado tres para realizar las pruebas de funcionamiento, así como también se ha emulado un tablero de distribución para instalar el módulo principal.

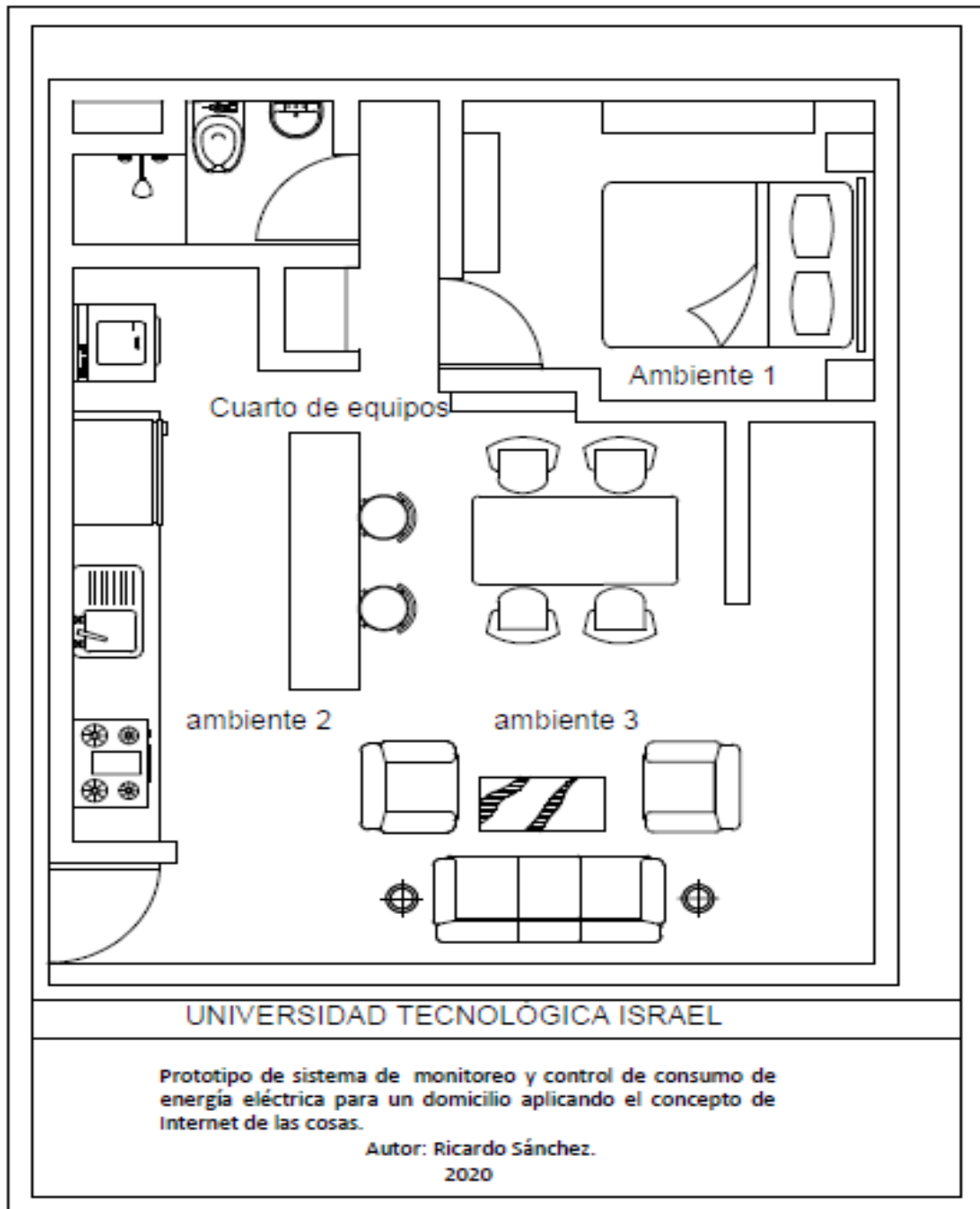


Figura. 0.914. Diagrama de la maqueta

Fuente: El autor

En la Figura. 3.25. se aprecia el esquema de distribución de la red de sensores en cada ambiente.

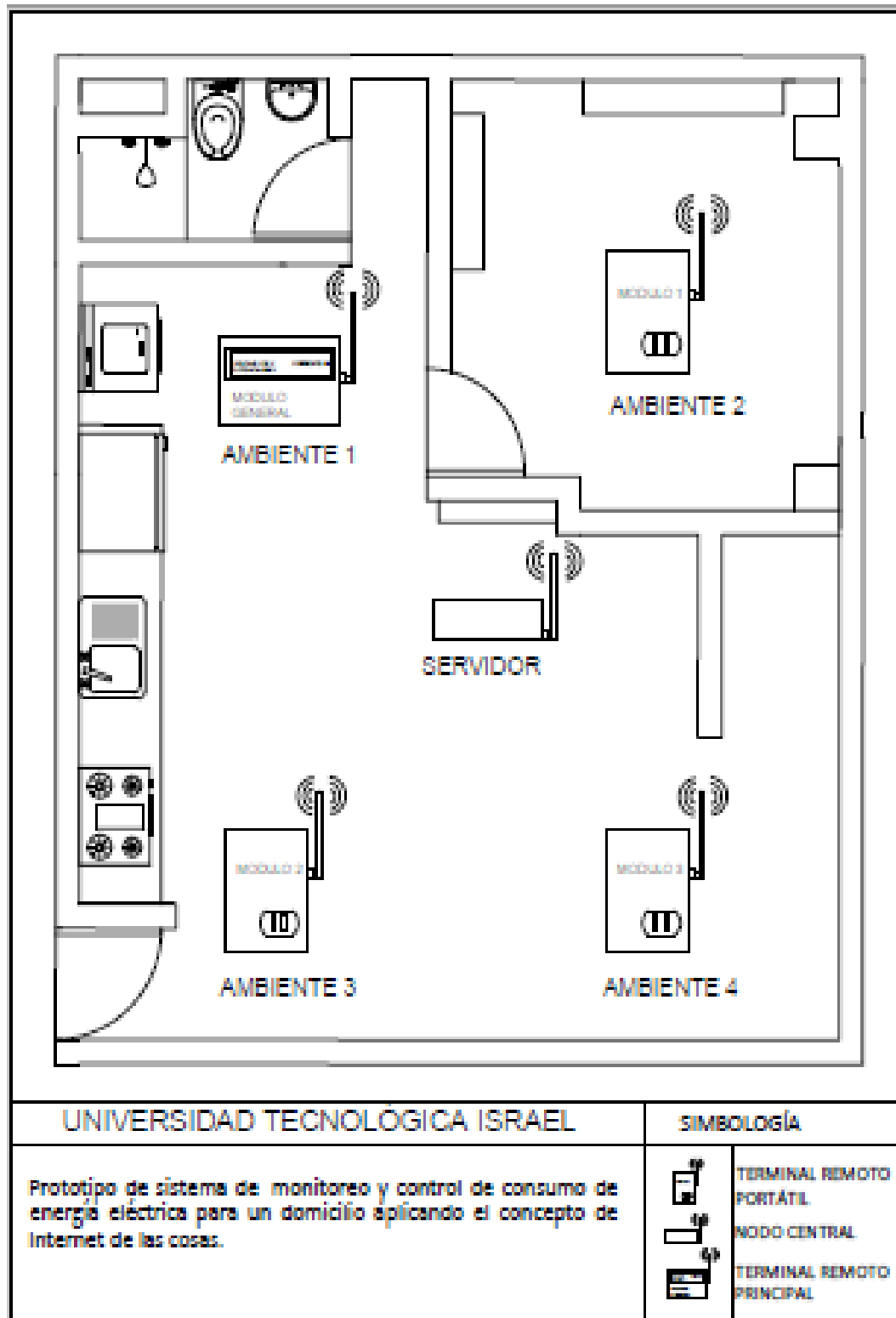


Figura. 0.25. Esquema de distribución de módulos de monitoreo

Fuente: El autor

En la Figura. 3.26. se detalla la presentación de la maqueta la cual tendrá una longitud de 60 cm y ancho 50 cm,

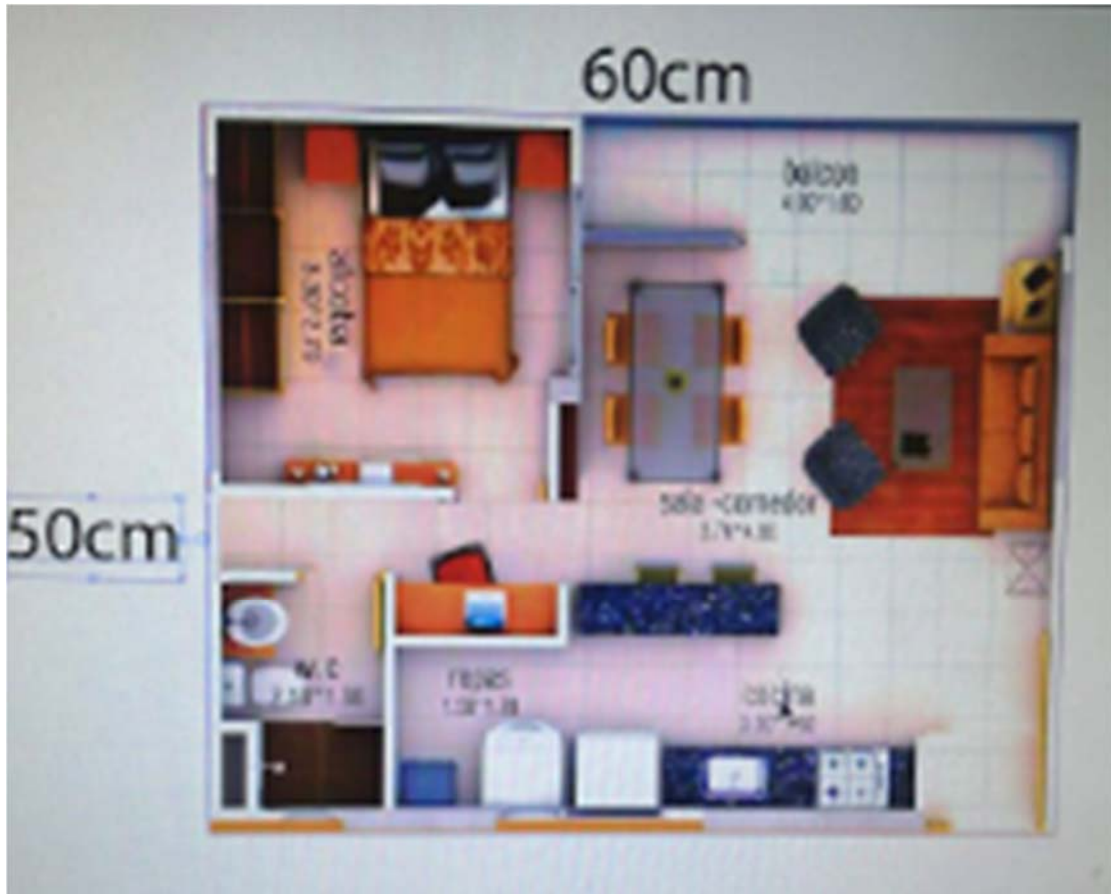


Figura. 0.26. Esquema de distribución de módulos de monitoreo

Fuente: El autor

CAPÍTULO 4

4. IMPLEMENTACIÓN

Es este capítulo se explica el diseño e implementación del prototipo de monitoreo de consumo de energía para domicilio basado en el internet de las cosas. A continuación se detalla la secuencia de pasos y procedimientos para el desarrollo del prototipo.

4.1 Diseño e implementación de la tarjeta electrónica del módulo de energía principal.

En la Figura. 4.1. se presenta en tercera dimensión el diseño de la tarjeta electrónica de monitoreo de energía para la acometida eléctrica.

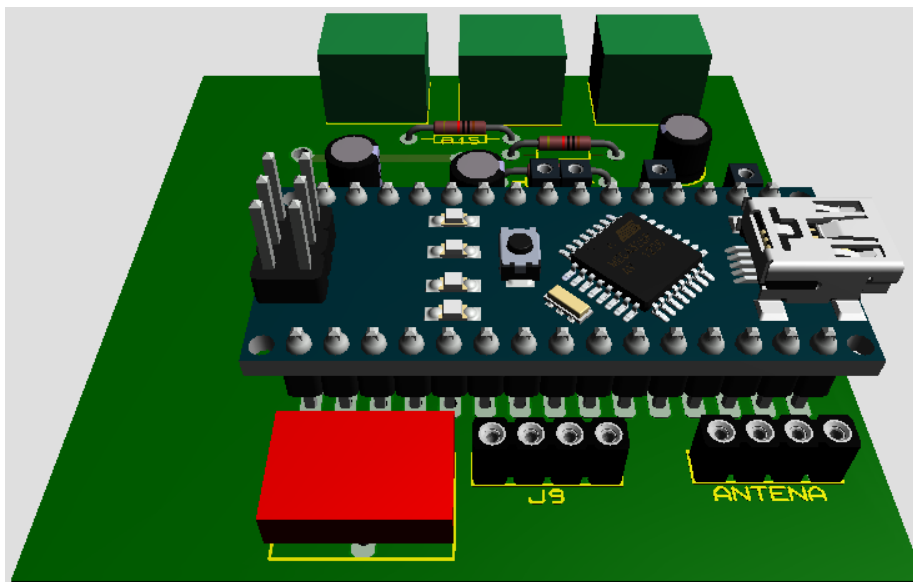


Figura. 0.92. Diagrama en tercera dimisión tarjeta integradora módulos secundarios.

Fuente: El autor

En la Figura. 4.2. se presenta el esquema electrónico realizado en el software de diseño electrónico Proteus.

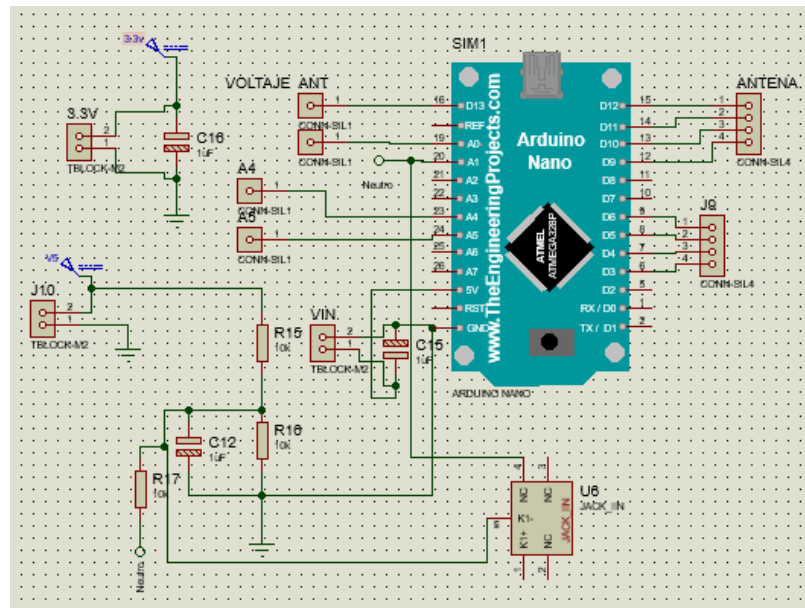


Figura. 0.93. Diagrama de la electrónica de la placa PCB modulo principal

Fuente: El autor

En este módulo la información de consumo se reporta a través de una LCD, además de ser enviada hacia la nube, representa la totalidad de corriente consumida por los tres módulos, así como el voltaje que se conecta en paralelo con cada punto de la residencia. El presente módulo no cuenta con la capacidad de habilitar o deshabilitar el flujo de corriente hacia la carga, puesto que su función radica únicamente en el monitoreo de la corriente que ingresa hacia toda la instalación. De esta manera, la etapa de “Actuar y Definir una Respuesta” no incluye la configuración de ninguna salida digital, puesto que no existe un relé instalado.

Elementos para utilizar en el sensor de monitoreo

- pinza SCT013-30A
- medidor Voltaje AC ZMPT101B
- arduino NANO
- transceiver NRF 24L01
- resistencias 10 K
- resistencia 1.5 K

- capacitor 47Uf
- módulo Alimentación Voltaje (Fuente Conmutada) 5V / 650 mA
- 1 módulo LCD I2C (Para módulo de Acometida)
- 4 toma corriente 15 A
- 1 metro Cable eléctrico 14 AWG
- cables de conexión

La Figura. 4.3. se presenta el diseño del circuito impreso listo para ser transferido a una baquelita

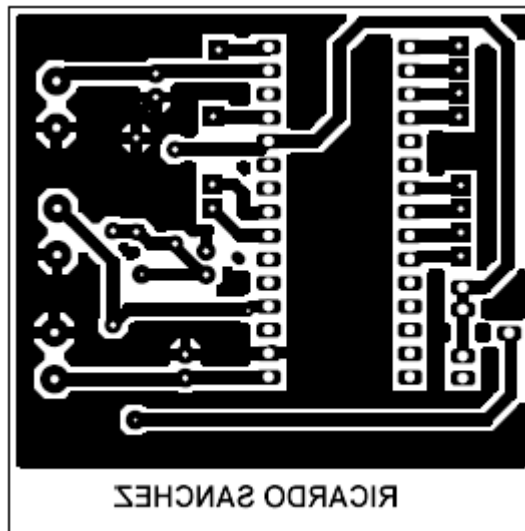


Figura 0.94 Diseño de ruteo de pistas tarjeta integradora

Fuente: el autor

4.2. Diseño e implementación de la tarjeta electrónica de los módulos de energía remotos.

Una vez determinado los materiales a utilizar en el presente proyecto, se procedió a diseñar la tarjeta integradora PCB para los sensores remotos, la Figura. 4.4. se observa el diseño en tercera dimensión de la tarjeta integradora y en la Figura. 4.5. encuentra el diagrama electrónico elaborado en el *software* de diseño electrónico Proteus.

Esta tarjeta integradora permitirá el trabajo en conjunto de los elementos, además incluirá la electrónica adicional necesaria para cada uno de los módulos de monitoreo portátiles.

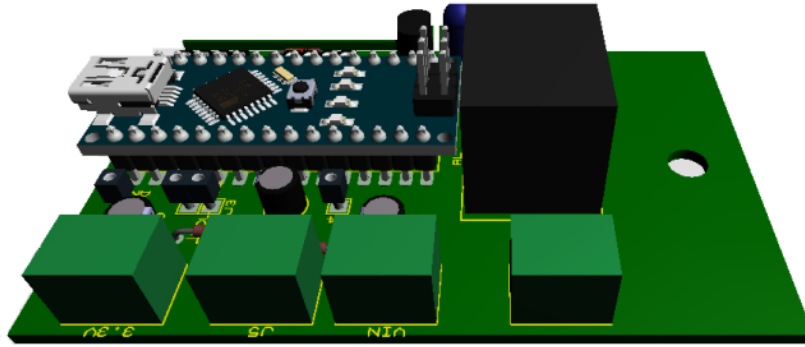


Figura. 0.95. Diagrama en tercera dimensión de la tarjeta integradora, módulos secundarios.

Fuente: El autor

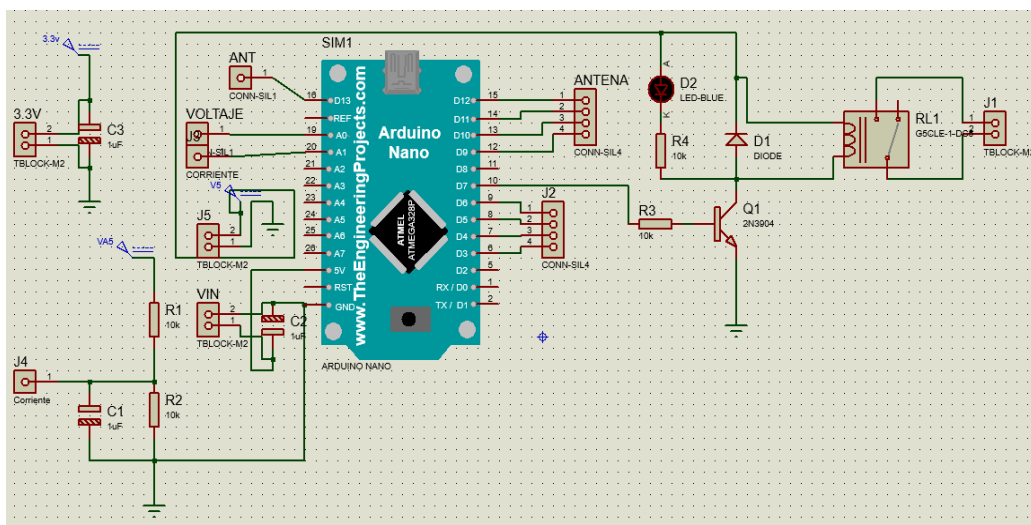


Figura. 0.96. Diagrama de la electrónica de la placa PCB

Fuente: El autor

Elementos para utilizar en el sensor de monitoreo

- 4 pinza SCT013-30A
- 4 medidor Voltaje AC ZMPT101B
- 4 arduino NANO
- 4 transceiver NRF 24L01
- 4 resistencias 10 K
- 4 resistencia 1.5 K
- 4 capacitor 47Uf
- 4 transistor 2N2222A
- 4 diodo Uso Común 1N4007
- 3 relé 5V / 15 A @ 110 VAC

- 4 módulo Alimentación Voltaje (Fuente Conmutada) 5V / 650 mA
- 1 módulo LCD I2C (Para módulo de Acometida)
- 4 toma corriente 15 A
- 1 metro de Cable eléctrico 14 AWG
- Cables de conexión

En la Figura. 4.6. se encuentra el diseño del circuito impreso para los sensores remotos realizado en Proteus, esta será trasferida a la baquelita utilizando una de las técnicas de impresión transfer, para luego mediante proceso químico retirar el exceso de cobre.

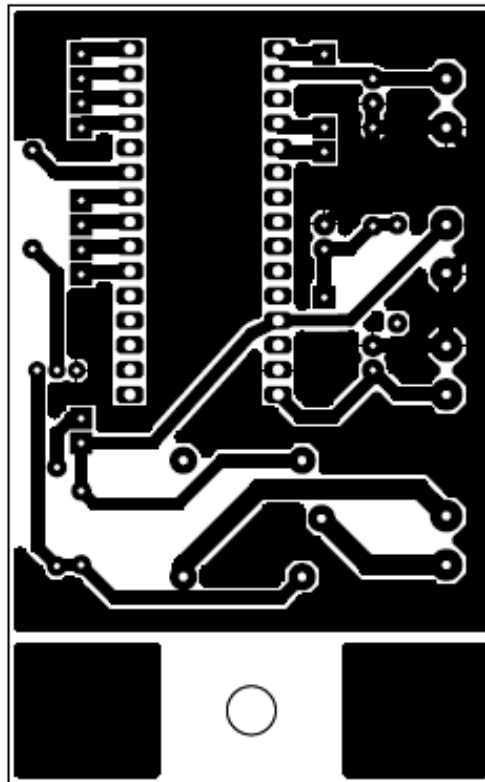


Figura. 0.97. Diseño de ruteo de pistas tarjeta integradora

Fuente: el autor

A continuación en la Figura. 4.7. se realiza el proceso de elaboración del circuito impreso.

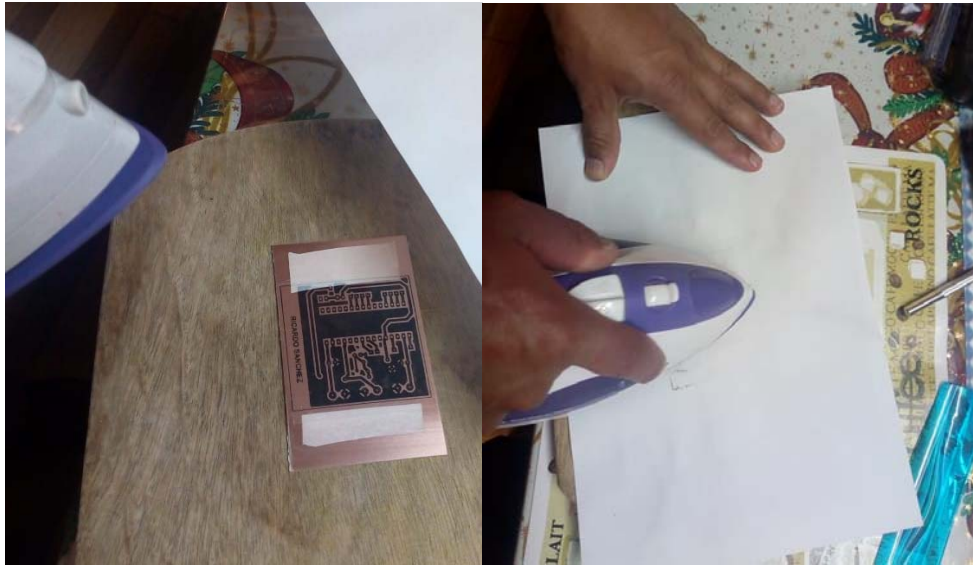


Figura. 0.98. Impresión en baquelita mediante calor

Fuente: El autor

En la Figura. 4.8. se observa el proceso de limpieza de cobre



Figura. 0.99. Proceso químico retiro de cobre

Fuente: El autor

Una vez que ha concluido el proceso químico, se realiza la limpieza de la superficie con la ayuda de una lija de agua, con la finalidad de retirar todos los excesos de ácido, tinta

del marcador permanente, para proceder a la perforación y a soldar los elementos soldados a la baquelita. Así se evidencia en la Figura. 4.9. y Figura. 4.10. respectivamente.

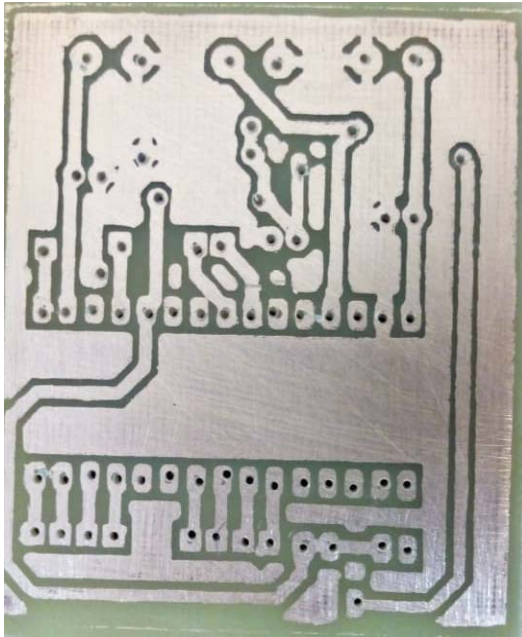


Figura. 0.100. Baquelita lista para la soldadura de elementos

Fuente: El autor

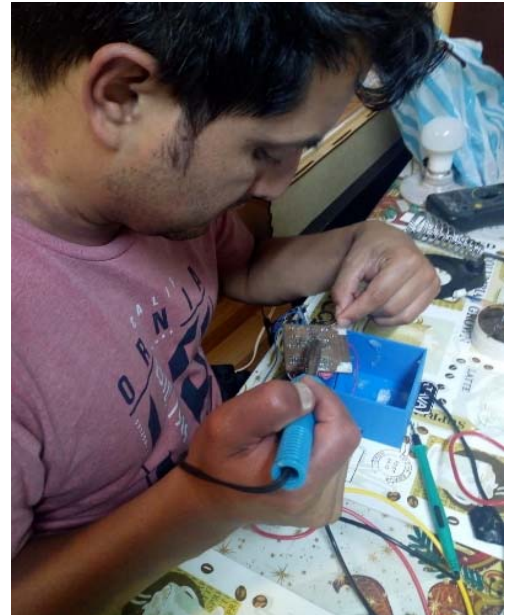


Figura. 0.101. Baquelita lista para la soldadura de elementos

Fuente: El autor

Este proceso de preparación de las tarjetas electrónicas es utilizado para todos los demás dispositivos, en las Figuras 4.11 y 4.12 se encuentra la imagen posterior y frontal de la tarjeta integradora y los elementos que la conforman los sensores de monitoreo.

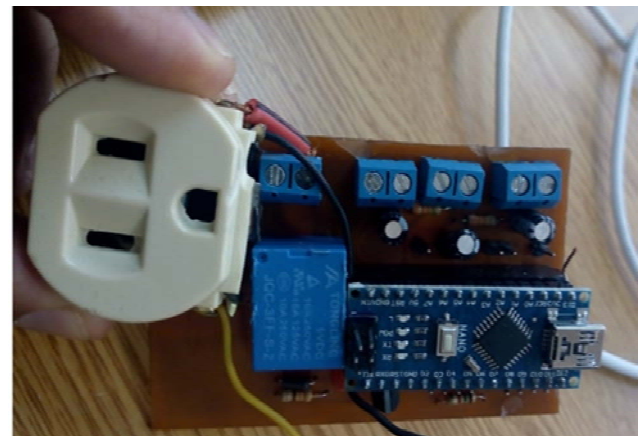
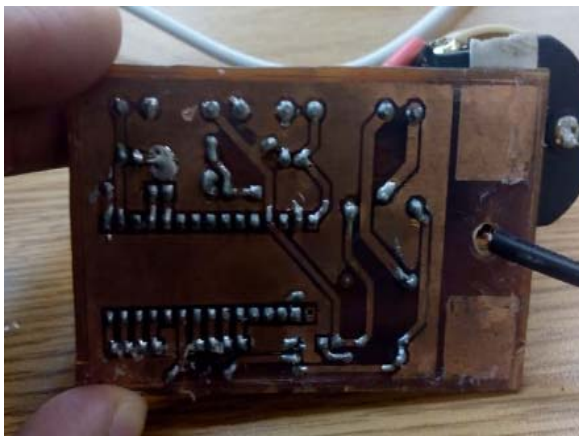


Figura. 0.102. Diseño de ruteo de pistas tarjeta integradora

Fuente: el autor

Figura. 0.103. Diseño de ruteo de pistas tarjeta integradora

Fuente: el autor

Montaje y pruebas de funcionamiento de los diferentes módulos electrónicos, de comunicación entre la tarjeta electrónica y el nodo central mostrado en la Figura. 4.13.

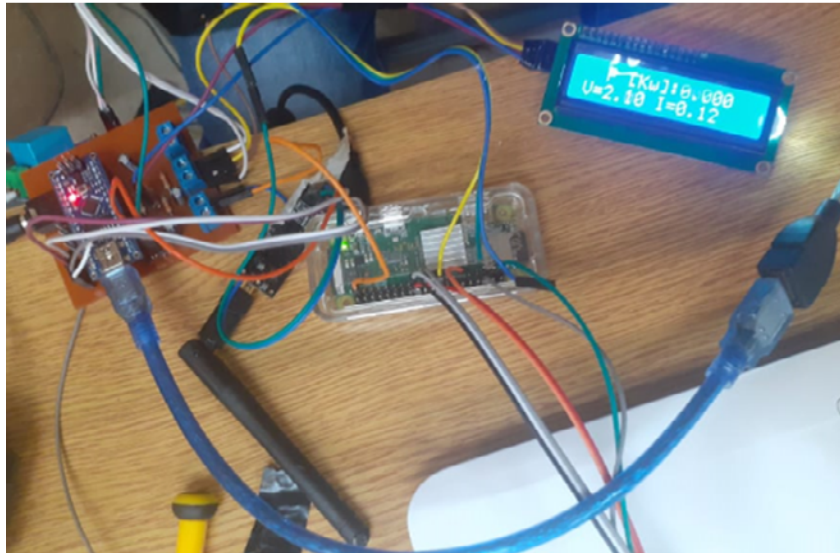


Figura. 0.104. Implementación del módulo principal

Fuente: El autor

A continuación se observan los módulos finalizados: Figura. 4.14. con el módulo principal y en la Figura. 4.15. se encuentra el sensor remoto.



Figura. 0.105. Módulo principal con visualizador de datos

Fuente: El autor

Figura. 0.106. Módulo remoto

Fuente: El autor

4.2.1. Implementación de software del prototipo de adquisición de datos.

En la Figura. 4.16. se observa el diagrama de bloques que explica el funcionamiento del algoritmo de programación en los módulos de energía remotos. (ANEXO F)

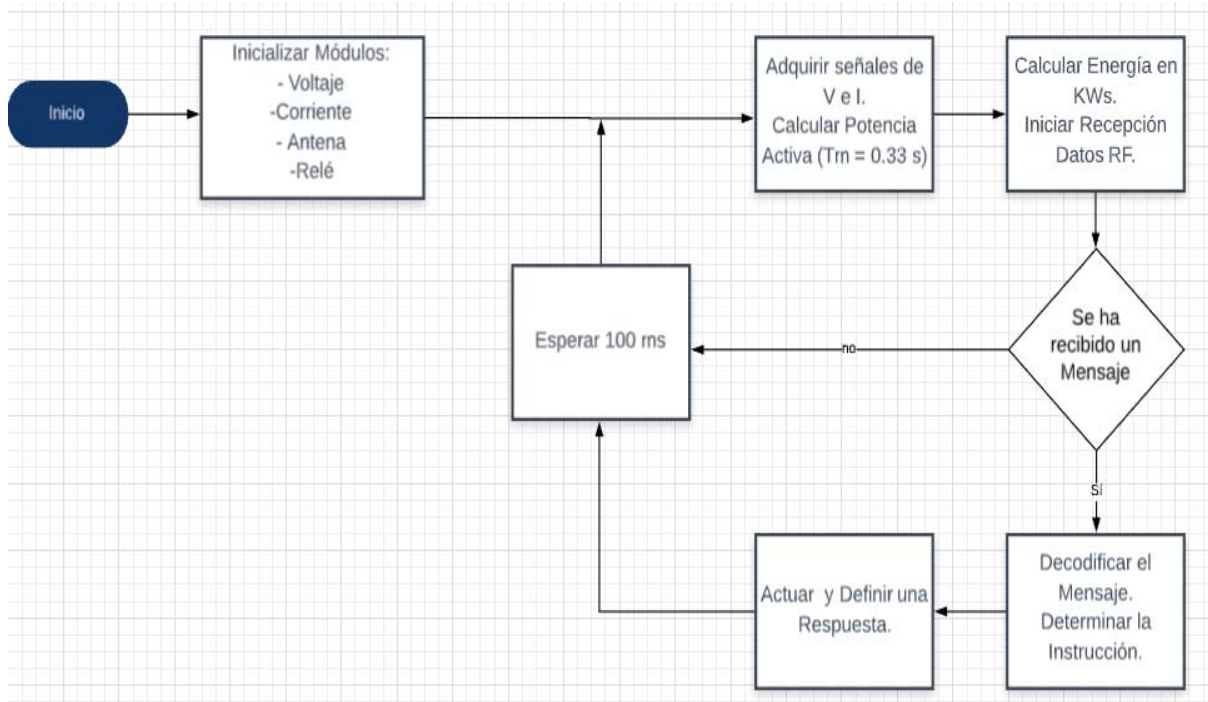


Figura. 0.107. Diagrama de bloques módulos remotos

Fuente: El autor

Se describen brevemente las etapas:

Inicializar Módulos: Única etapa fuera del lazo de funcionamiento normal del Arduino, donde se procede a configurar cada una de las entradas (digitales y analógicas) como salidas digitales necesarias para realizar todas las acciones de medición, control y comunicación.

Calcular Potencia: La primera etapa dentro del lazo de repetición consiste en el proceso de adquisición de las señales analógicas con un tiempo de muestreo de 0.33 [s], equivalente a 3.03 Hz. Una vez decodificadas las señales de Voltaje y Corriente, se procede a realizar el cálculo de Potencia Activa, dato fundamental para el funcionamiento del sistema.

Calcular Energía. - Se describen las etapas:

- **Iniciar Recepción:** Durante esta etapa se procede a realizar el cálculo discreto de Energía mediante una variable “acumuladora” de los valores de potencia obtenidos durante el tiempo. Para evitar la saturación de esta variable, los datos son enviados durante un tiempo de 10 min. Para ello, el módulo central envía la instrucción correspondiente a cada dispositivo esclavo, de modo que el dicho dispositivo envía la variable de energía calculada y procede a encerrar dicho valor.
- **Se ha recibido un mensaje:** En este punto se determina si existe una instrucción procedente del nodo central (Raspberry), a partir de lo cual se toman las decisiones correspondientes. En caso de no existir un comando, el sistema pasa a la etapa final, caso contrario se actúa conforme la instrucción.
- **Decodificar el mensaje:** Esta etapa involucra todo el proceso necesario para determinar qué tipo de instrucción se encuentra contenida en el mensaje que ha llegado. Así se verifica exactamente qué es lo que se pide al dispositivo. Entre las instrucciones pueden encontrarse: Reportar Energía, Prender Carga, Apagar Carga.
- **Actuar y definir respuesta:** En esta etapa concretamente se procede a actuar según la instrucción dada, configurando la salida digital de tal manera que se permita o se impida conducción a través del relé.
Esperar 100 ms: Pequeño delay aplicado para garantizar el funcionamiento del lazo de programación en función de posibles interrupciones externas debidas al elemento de comunicación.

En la Figura. 4.17. muestra las librerías requeridas para la elaboración del algoritmo también se definen las diferentes variables almacenar los valores de voltaje, corriente, potencia, tiempo, energía y consumo de energía.

```

iot_nrf24 Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda

iot_nrf24
//SendReceive.ino

#include<SPI.h>
#include<RF24.h>
#include "printf.h"
#include "EmonLib.h"           // Include Emon Library
EnergyMonitor emon1;         // Create an instance
float KWs=0;
int c_rele=0;
// CE, CSN pins
RF24 radio(9, 10);
    
```

Figura. 0.108. Librería “EmonLib” en Arduino

Fuente: El autor

A continuación, se presenta el código desarrollado para la tarjeta Arduino NANO contenida en cada elemento de medición sin *display*.

4.2.2. Implementación de Software del Prototipo para Acometida:

El algoritmo de funcionamiento según la Figura. 4.18. puede explicarse a través del siguiente diagrama de bloques, es en esencia similar al anterior, tomando en cuenta adicionalmente que este módulo reporta el consumo visualmente a través de un LCD:

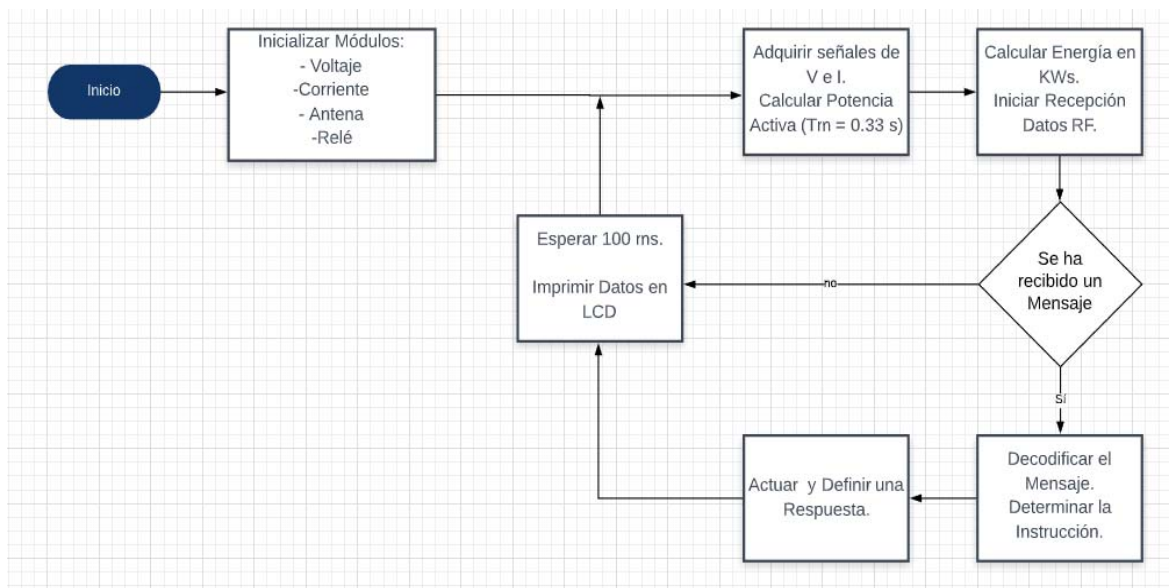


Figura. 0.109. Diagrama de bloques medidor de energía acometida.

Fuente: El autor

Este módulo se diferencia, entre otras cosas, en el hecho de que la información de consumo se reporta a través de una LCD, además de ser enviada hacia la nube. La información obtenida a través de este módulo representa la totalidad de corriente consumida por los tres módulos, así como el voltaje que se conecta en paralelo con cada punto de la residencia.

El presente módulo no cuenta con la capacidad de habilitar / deshabilitar el flujo de corriente hacia la carga, puesto que su función radica únicamente en el monitoreo de la corriente que ingresa hacia toda la instalación. De esta manera, la etapa de “Actuar y Definir una Respuesta” no incluye la configuración de ninguna salida digital, puesto que no existe un relé instalado.

4.3. Calibración

Una vez definido las diferentes librerías y variables, se procederá a calibrar los sensores de voltaje y corriente, esto se realiza siguiendo especificaciones técnicas provistas por el fabricante, como también realizando calibraciones necesarias.

Antes de iniciar el proceso de calibración se debe considerar ciertos factores de seguridad para evitar electrocuciones, manipular la red eléctrica donde el voltaje de línea tiene una magnitud aproximadamente 120 VAC podría ser peligroso, también se debe colocar una resistencia de carga para reducir el valor de corriente entregado a la salida de los terminales de los sensores, con el propósito de reducir controlar la corriente de salida y obtener un voltaje a la salida de 1 VAC referenciados a 2,5 Vcd, que entregará a la entradas analógica del procesador que únicamente utiliza valores positivos de hasta cinco voltios. En la Figura. 4.19. se presenta el diagrama de conexiones para el sensor de corriente y voltaje.

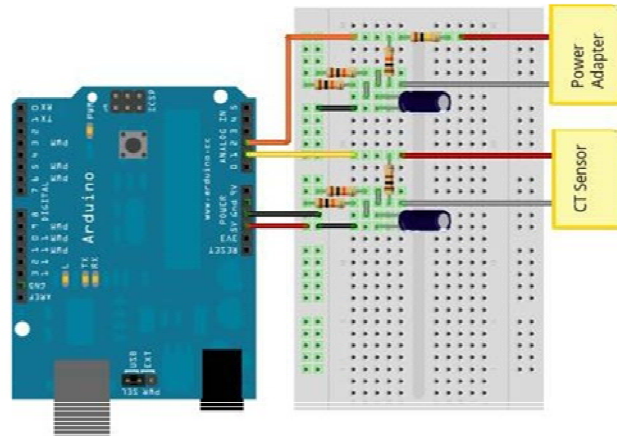


Figura. 0.110. Diagrama para la conexión de los sensores de corriente y voltaje

Fuente: (OpenEnergyMonitor)

4.3.1. Calibración del sensor de corriente.

El sensor de corriente que se utilizara en el proyecto es de la familia es el SCT 013, donde el cálculo del factor de calibración se realiza tomando en cuenta que a la salida del sensor el voltaje a manipular es aproximadamente a 1 VAC.

El procedimiento es el siguiente: se divide la corriente nominal del primario para el voltaje del secundario (voltaje se lo calcula de la corriente que atraviesa el bobinado secundario por la resistencia de carga). (*OpenEnergyMonitor*). Para el sensor SCT 30A/1V, no se requiere calcular la resistencia de carga puesto que ya tiene incluida de fábrica y entrega 1 VAC, lo mencionado se representada en la ecuación para el cálculo del valor de la constante de corriente

$$\text{Constante de la corriente} = \frac{30 [A]}{1[V]} \cong 30 \left[\frac{N * m}{C} \right]$$

4.3.1. Calibración del sensor de voltaje

El sensor que se utilizará para la lectura del voltaje que ingresa al sistema es el transformador de voltaje AC - ZMPT101B. De la misma manera para calcular el valor de voltaje de la entrada a la señal analógica A0, de Arduino se debe considerar el valor a la

entrada de esta, es decir se divide la señal del voltaje de la red eléctrica para el voltaje de la señal que debe ingresar a la entrada analógica A0. (OpenEnergyMonitor)

$$\text{Constante de voltaje} = \frac{120 [V]}{1[V]} \cong 120$$

En la Figura. 4.20. se observa el valor de las constantes de calibración para el sensor de voltaje como para el sensor de corriente, los valores pueden variar del calculado debido a la naturaleza de los materiales, para validar los valores medidos se realizó las diferentes pruebas de medición, que en la práctica se puede observar que el valor de la constante para el voltaje es de 154 y la constante de calibración para el sensor de corriente es de 27.

```
void setup(void) {
  pinMode(7, OUTPUT);
  emon1.voltage(A0, 164.5, 0); // V
  emon1.current(A1, 29);      // C
  printf_begin();
  while(!Serial);
  Serial.begin(9600);
}
```

Figura. 0.111. Código de programación, calibración de sensores

Fuente: El autor

En la Tabla. 4.1. se describen las constantes de calibración para cada uno de los sensores de monitoreo.

Tabla. 4.1. Contantes de calibración

CONTANTE DE CALIBRACION		
Dispositivo	Voltaje	Corriente
Modulo Principal	159	29
Módulo 1	164	28
Módulo 2	151	28
Módulo 3	151	29

Fuente: El autor

Prueba de funcionamiento, una vez finalizada la calibración de los sensores de adquisición de datos se realiza medidas de los valores que estos adquieren y se los compara con los valores teóricos o nominales de las cargas.

Módulo 1

Carga 1: lámpara incandescente de 40W:

$$P = V \cdot I$$

$$I [A] = \frac{40 [W]}{120 [V]} = 0.33 [A]$$

En la Figura. 4.21. y en la Figura. 4.22. se evidencia la medición de corriente y voltaje respectivamente además de los parámetros obtenidos para el módulo 1 en la Figura. 4.23.



Figura. 0.112. Medida de la corriente en miliamperios

Fuente: El autor



Figura. 0.113. Medida de voltaje

Fuente: El autor

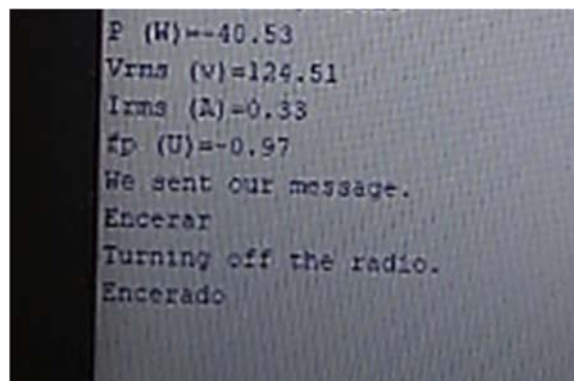


Figura. 0.114. Parámetros obtenidos con el sensor de medida, monitor serial Arduino IDE.

Fuente: el autor

El cálculo de error se lo realiza con la ecuación que se aprecia a continuación:

$$E \% = \left(\frac{\text{Calculado} - \text{Medido}}{\text{Calculado}} \right)$$

I Calculada [A]	I medida [A]
0.33	0.33

Error de lectura de la corriente:

$$E \% = \left(\frac{0,333 - 0,336}{0,333} \right) = 0,9\%$$

Potencia

Potencia Nominal [W]	Potencia Medida [W]
40	40.5

Error de la potencia calculada:

$$E \% = \left(\frac{40 - 40,5}{40} \right) = 1,25\%$$

Módulo 2:

I Calculada [A]	I medida [A]
0.33	0.33

$$E \% = \left(\frac{0,333 - 0,334}{0,333} \right) = 0,3\%$$

Potencia Nominal [W]	Potencia Medida [W]
40	39.89

$$E \% = \left(\frac{40 - 39,89}{40} \right) = 0,27\%$$

Módulo 3:

I Calculada [A]	I medida [A]
0.333	0.34

$$E \% = \left(\frac{0.333-0.34}{0.333} \right) = 0.3\%$$

Potencia Nominal [W]	Potencia Medida [W]
40	41.75

$$E \% = \left(\frac{40-41.75}{41.75} \right) = 4.37\%$$

4.3.2. Cálculo de potencia y energía

En la Figura. 4.24. se observa los parámetros calculados por el algoritmo de programación el cual mide y calcula todos los parámetros que intervienen en el triángulo de potencia presentado en el capítulo 1, estos valores se almacenan en las diferentes variables del programa y se los utiliza en los cálculos de consumo de energía.

```
float starttime=millis()/1000;
emon1.calcVI(20,2000); // Calculate all. No.of half wavelengths (crossings), time-out
// emon1.serialprint(); // Print out all variables (realpower, apparent power, Vrms, Irms, power factor)
float realPower = emon1.realPower; //extract Real Power into variable
```

Figura. 0.115. Código de programación, cálculo de potencia real.

Fuente: El autor

De todos los parámetros calculados, se considera únicamente el valor de la potencia real, que multiplicada por la variable de tiempo para obtener la energía consumida hasta ese instante, en la Figura. 4.25. se encuentra una la parte del lenguaje de programación donde se realiza el cálculo de la energía, también se observa el algoritmo para calcular el tiempo y se almacena en kW, vale señalar que se ha decidido manejar el valor del consumo de energía, con la referencia de tiempo en segundos ya que el valor en kWh es muy pequeño por lo cual

se convierte a kW para que no sea tan despreciable, este proceso se realiza hasta el momento en que reciba un mensaje de envío y recepción desde y hacia el servidor para su almacenamiento, una vez enviado y confirmada la recepción de dicha variable, el nodo central envía un nuevo mensaje para reiniciar la cuenta de la energía y reiniciar el proceso.

```
void loop(void) {
  emon1.calcVI(20,2000);          // Calculate all. No.of half wavelength
  //emon1.serialprint();          // Print out all variables (realpower
  float realPower      = emon1.realPower;
  float supplyVoltage  = emon1.Vrms; //extract Vrms into Variable
  float Irms           = emon1.Irms; //extract Real Power into variable
  float powerFactor    = emon1.powerFactor;
  float endtime=millis()/1000;
  if (endtime-starttime >=1){
  KWs= realPower/1000+KWs;
  starttime=millis()/1000;
  }
}
```

Figura. 0.116. Código de programación, cálculo de energía.

Fuente: El autor

En la Figura. 4.26. se observa los datos presentados en el Arduino IDE, con los parámetros de voltaje, corriente, factor de potencia y el consumo de una carga en kW

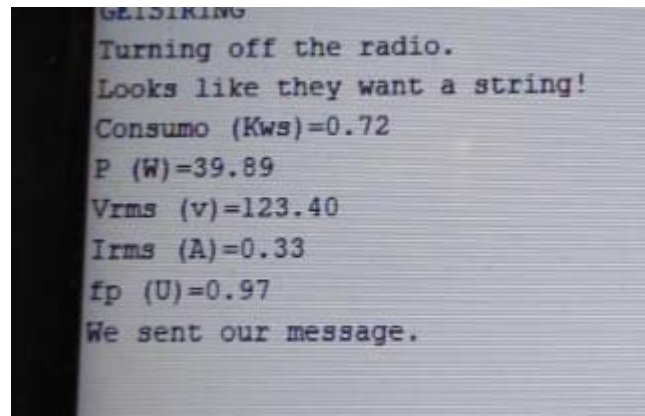


Figura. 0.117. Presentación de datos Arduino IDE

Fuente: El autor

En la Tabla. 4.2. se observa los valores de energía calculados durante un periodo de tiempo de 30 min, con un consumo de energía de 72kWs. En la Figura. 4.27. se observa el valor almacenado en el servidor IoT.

Tabla. 4.2. Valores de energía teóricos durante 30 minutos

	Hora de inicio	Carga	Unidad	Tiempo En segundos	Consumo teórico Ws	kWs	Valor en la aplicación	Hora final
Medidor 1	13:00	40	W	1800	72000	72		13:30

Fuente: El autor

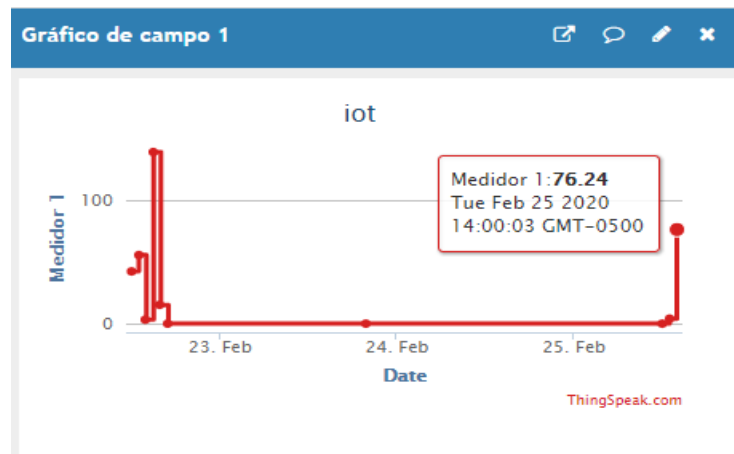


Figura. 0.118. Valores medidos de potencia almacenado en el servidor IoT

Fuente: El autor

4.4. Configuración de la red de nodos.

En la configuración de la red de nodos se describen los pasos:

Paso 1. Se instala el sistema operativo y configuraciones necesarias en la Raspberry Pi Zero W, para luego proceder a la conexión del NRF 24L01+ de comunicación inalámbrica a la Raspberry Pi como se observa en la Figura. 4.28.

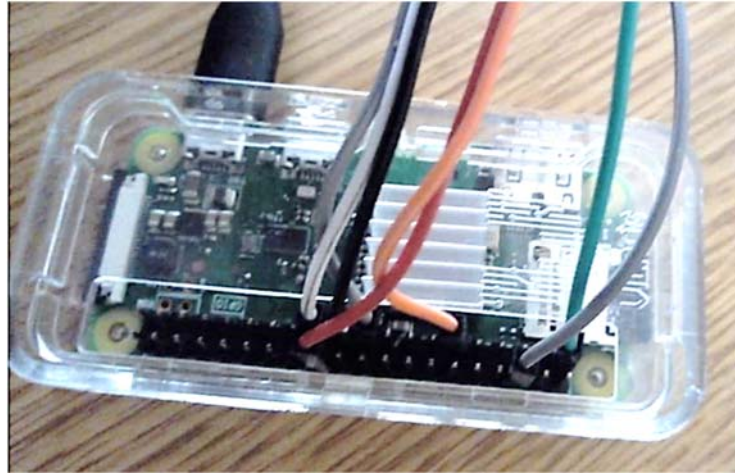


Figura. 0.119. Conexión del módulo NRF 24L01+a la Raspberry pi

Fuente: el autor

Paso 2. De la misma forma que se procedió con el servidor Raspberry, se procede a conectar el módulo NRF 24L01+ a la placa integradora con Arduino de cada uno de los módulos de monitoreo de energía, como muestra la Figura. 4.29. Este procedimiento se realizó en base al material provisto por el desarrollador de la librería TMRh20.

Disposición de los pines

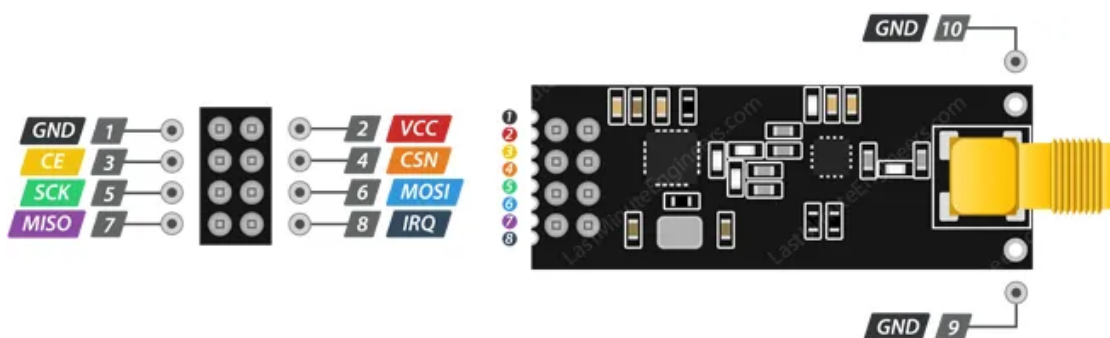


Figura. 0.120. Disposición de pines módulos NRF 24L01+

Fuente: (TEDIC, 2015)

En la Tabla. 4.3. se encuentra el resumen la distribución de pines y conexiones del dispositivo inalámbrico con las placas que se utilizaran como nodos de la red de sensores.

Tabla. 4.3. Disposición de conexión de tarjeta inalámbrica con las diferentes placas empleadas como nodos

Numero de pines	Disposición de pines		Disposición de pines Raspberry Pi
	NRF 24101+	Arduino nano	
1	GND	GND	6
2	VCC	3.3V	1
3	CE	D9	11
4	CSN	D10	24
5	SCK	D13	23
6	MOSI	D11	19
7	MISO	D12	21
8	IRQ	-	

Fuente: (TEDIC, 2015)

Al resolver el problema de hardware, se proseguirá con la etapa de configuración para la comunicación entre los nodos remotos y el central

4.5. Comunicación Inalámbrica.

Primero se debe establecer la comunicación inalámbrica bidireccional entre el nodo central (servidor) y Arduino nano utilizado como sensor y ejecutor de comandos, para ello se utilizará el lenguaje de programación Python para Raspberry y para la programación en Arduino se utilizará el lenguaje de programación propio de Arduino.

- El procedimiento que se debe cumplir en la Raspberry pi es el siguiente:
 - Instalar y configurar de las librerías requeridas para el control de NRF 24L01+
 - Instalar y configurar de las librerías requeridas para el control de GPIO.
 - Ejecución del Programa.

- El procedimiento que se debe cumplir en la Arduino es el siguiente:
 - Instalación y configuración de librerías NRF 24L01+
 - Ejecución del Programa.
- Test de comunicación bidireccional.

4.5.1. Instalar y configurar de las librerías requeridas para el control de NRF 24L01+ en Raspberry pi

Para utilizar los recursos existentes y darles una mayor funcionalidad a los nodos de la red inalámbrica, se elaboró un programa que permite llamar las funciones de la librería TMRH20, con los cuales se realiza la comunicación entre nodos, lo primero que se debe realizar es la instalación de las librerías para comunicación y la activación de las entradas GPIO. En la Figura. 4.30. se observa una parte del procedimiento empleado, en la consola de texto y los diferentes comandos empleados para la instalación de la librería NRF 24

```
this is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $ mkdir servidorRasPi
pi@raspberrypi:~ $ cd servidorRasPi
pi@raspberrypi:~/servidorRasPi $ wget https://raw.githubusercontent.com/jpbarrac
a/pynrf24/master/nrf24.py
```

Figura. 0.121. Instalación de librería de comunicación

Fuente: El autor

Para el direccionamiento los módulos inalámbricos NRF 24, utilizan 40 bits es decir, por cada dirección se requiere 5 bytes, también se dispone de un amplio rango de direcciones a utilizar.

4.5.2. Instalar y configurar las librerías requeridas para el control de GPIO.

```
pi@raspberrypi:~/Desktop $ sudo apt-get install python-rpi.gpio python3-rpi.gpio
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
python-rpi.gpio ya está en su versión más reciente (0.7.0~buster-1).
python3-rpi.gpio ya está en su versión más reciente (0.7.0~buster-1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
pi@raspberrypi:~/Desktop $
pi@raspberrypi:~/Desktop $ python
Python 2.7.16 (default, Oct 10 2019, 22:02:15)
[GCC 8.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import RPI.GPIO
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named RPI.GPIO
>>> import RPI.GPIO as GPIO
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named RPI.GPIO
>>> import RPi.GPIO as GPIO
>>>
[3]+ Detenido          python
pi@raspberrypi:~/Desktop $
```

Figura. 0.122. Instalación librerías GPIO

Fuente: el autor

Estas librerías según la Figura. 4.31. permiten designar que función cumplirán los diferentes pines de propósito general de la placa Raspberry, utilizadas en este caso para el control, envío y recepción de paquetes mediante el uso de la librería de comunicación inalámbrica, la asignación de los pines se los realizará según las recomendaciones del fabricante.

4.5.3. Ejecución del programa

En la Figura. 4.32. se observa la librería en Python, como también se observan algunos otros parámetros como las direcciones, de los diferentes módulos a los que se enviará y recibirá datos, también se ha configurado el canal de comunicación que en este caso es el 76.

```

pi@raspberrypi: ~/Desktop/servidorRasPi/NRF24L01
GNU nano 3.2 SendReceiveArduino.py
import RPi.GPIO as GPIO
from lib_nrf24 import NRF24
import time
import spidev

# CONFIGURACION DE PINES COMO BCM
GPIO.setmode(GPIO.BCM)
# DIRECCIONES DE LOS DISPOSITIVOS NRF24
pipes = [[0xE8, 0xE8, 0xF0, 0xF0, 0xE2], [0xF0, 0xF0, 0xF0, 0xF0, 0xE3], [0xE8, 0xE8, 0xF0, 0xF0, 0xE3], [0xE8, 0xE8, 0xF0, 0xF0, 0xE3]]
# CONFIGURACION SPI
radio = NRF24(GPIO, spidev.SpiDev())
# CONFIGURACION DEL MODULO RPI
radio.begin(0, 17)
radio.setPayloadSize(32)
radio.setChannel(0x76)
radio.setDataRate(NRF24.BR_1MBPS)
radio.setPALevel(NRF24.PA_MAX)
radio.setAutoAck(True)
radio.enableDynamicPayloads()
radio.enableAckPayload()
radio.openReadingPipe(1, pipes[1])
# IMPRIMIR DETALLES DEL MODULO NRF24 RPI
radio.printDetails()

# PRIMER MODULO
contador=0
while contador < 5:
    radio.openWritingPipe(pipes[0])
    print("Primer modulo envio:")
    print(contador)
    # PEDIR DATO DEL MODULO 1
    message = list("GETSTRING")
    while len(message) < 32:
        message.append(0)
    start = time.time()
    radio.write(message)
    print("Sent the message: {}".format(message))
    # RECIBIR DATO
    radio.startListening()
    
```

Figura. 0.123. Librería en Python para NRF 24

Fuente: El autor

En la Tabla. 4.4. se visualiza en las direcciones utilizadas en el direccionamiento entre el servidor y los terminales.

Tabla. 0.4. Tabla de direccionamiento

NODO	DIRECCIÓN
Maestro	0xf0f0f0f0e3
Dispositivo 1	0xe8e8f0f0e2
Dispositivo 2	0xe8e8f0f0e3
Dispositivo 3	0xe8e8f0f0e4
Dispositivo 4	0xe8e8f0f0e5

Fuente: El autor

El módulo inalámbrico tiene a su disposición 125 canales, es decir se podría implementar 125 nodos centrales trabajando al mismo tiempo con total independencia entre

ellos, cada canal puede utilizar hasta seis direcciones, es decir el nodo central se puede comunicar hasta con seis nodos terminales al mismo tiempo.

4.5.4. Instalación y configuración de librerías NRF 24L01+ en Arduino.

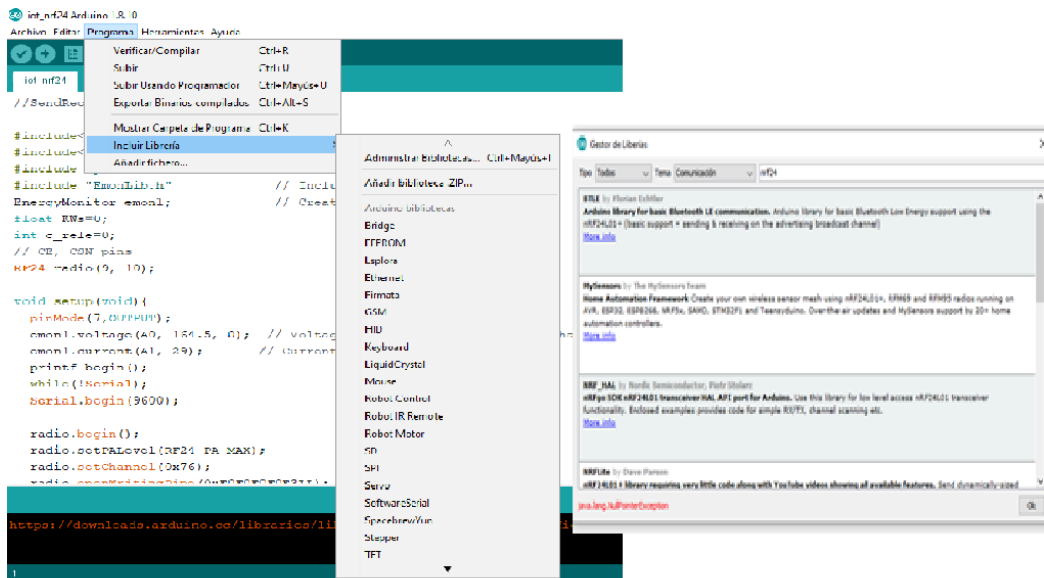


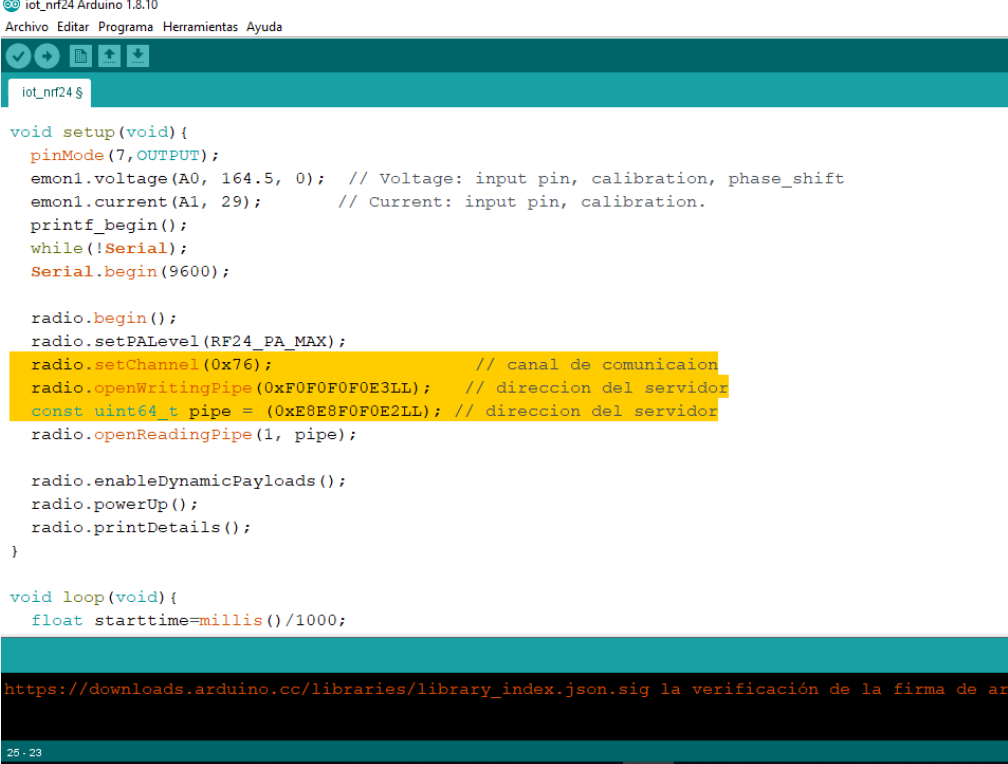
Figura. 0.124. Gestor de librerías Arduino

Fuente: El autor

Para la actualización de las librerías según la Figura. 4.33. se utilizó la herramienta propia del software de programación proporcionado por su fabricante donde se accede a las actualizaciones de las librerías e instalar nuevas, como la que se requiere para el control de la tarjeta de comunicación NRF 24L01+.

4.5.5. Ejecución del Programa.

Una vez que las librerías requeridas se han actualizado correctamente se procede a ingresar los parámetros que permitirán la comunicación entre el servidor y los terminales remotos, en la Figura. 4.34. se observa la configuración de los parámetros de direccionamiento entre el servidor y la terminal como también se registra el canal de comunicación.



```
iot_nrf24 Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda

iot_nrf24 $

void setup(void) {
  pinMode(7, OUTPUT);
  emon1.voltage(A0, 164.5, 0); // Voltage: input pin, calibration, phase_shift
  emon1.current(A1, 29);      // Current: input pin, calibration.
  printf_begin();
  while(!Serial);
  Serial.begin(9600);

  radio.begin();
  radio.setPALevel(RF24_PA_MAX);
  radio.setChannel(0x76); // canal de comunicacion
  radio.openWritingPipe(0xF0F0F0F0E3LL); // direccion del servidor
  const uint64_t pipe = (0xE8E8F0F0E2LL); // direccion del servidor
  radio.openReadingPipe(1, pipe);

  radio.enableDynamicPayloads();
  radio.powerUp();
  radio.printDetails();
}

void loop(void) {
  float starttime=millis()/1000;

  https://downloads.arduino.cc/libraries/library_index.json.sig la verificación de la firma de ar
25 - 23
```

Figura. 0.125. Ejecución de la librería NRF 24L01

Fuente: El autor.

4.5.6. Test de comunicación bidireccional.

El test de comunicación entre el servidor y el terminal de monitoreo se realiza mediante el envío de un mensaje codificado de verificación desde el servidor, a la terminal, dicho mensaje es recibido y decodificado, si el mensaje es falso, el terminal envía un mensaje de regreso al servidor con un mensaje de error, solicitando nuevamente el servidor reenviar el mensaje, este proceso se repite hasta que el receptor recibe y lee correctamente el mensaje, la terminal envía de regreso un mensaje recibido correctamente, este proceso se realiza de la misma manera para todos los terminales. En la Figura. 4.35. se observa la prueba de comunicación entre el servidor y el sensor.

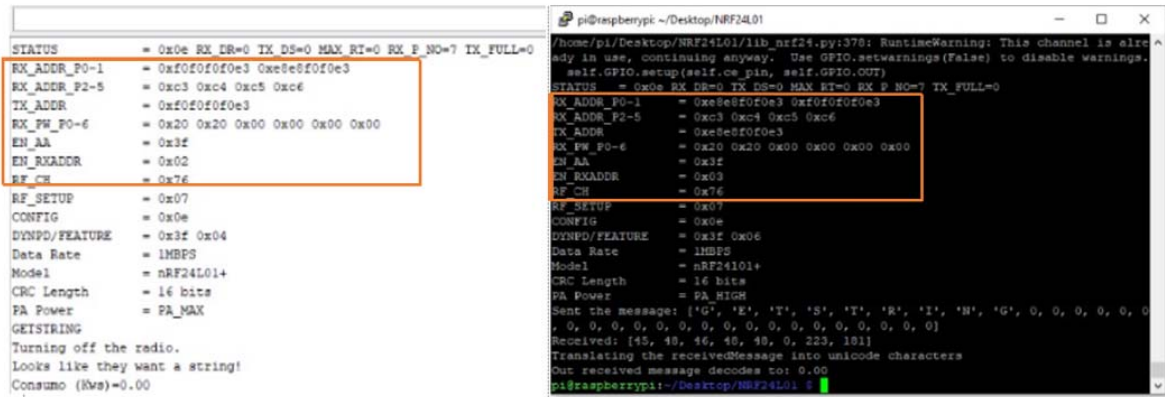


Figura. 0.126. Pruebas de conexión módulos inalámbricos

Fuente: El autor

4.6. Recepción y envío de datos desde Arduino a Raspberry y viceversa.

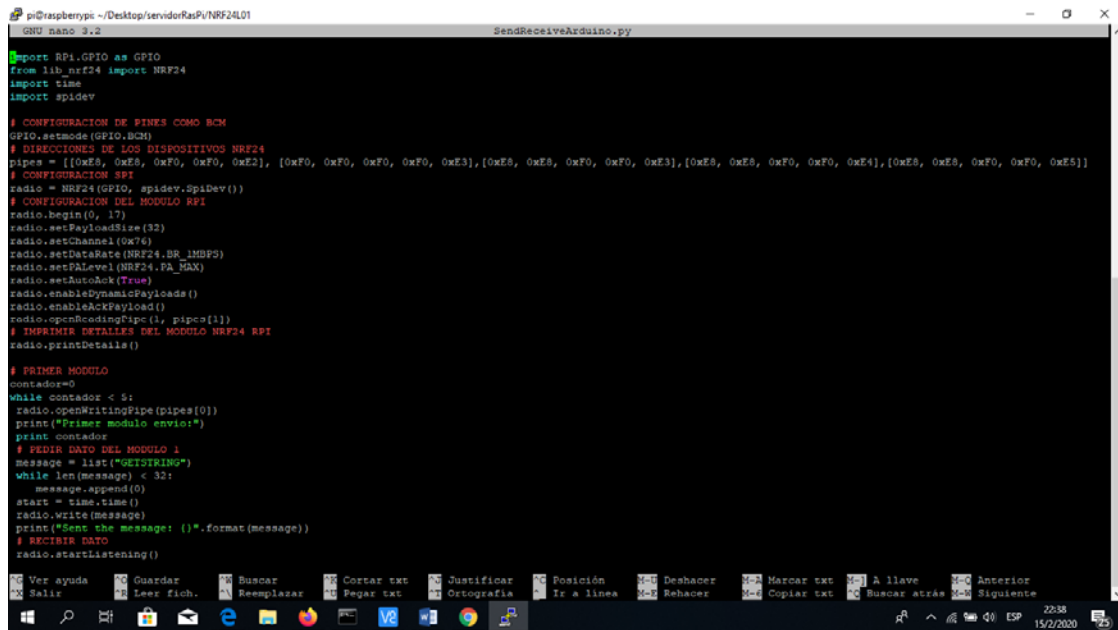


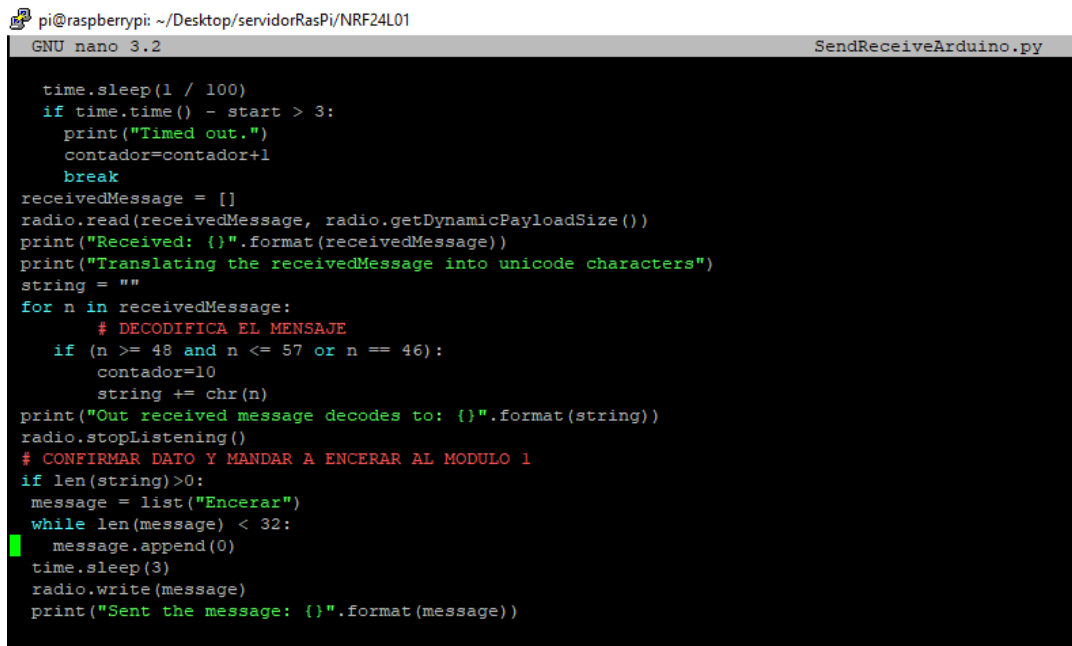
Figura. 0.127. Algoritmo de programación envío y recepción de datos

Fuente: El autor.

Cada medidor de energía toma los valores de la red a través de sensores y son almacenados en variables definidos en el programa, mediante la librería de comunicación NRF 24, se codifican y envían los datos al servidor, donde se decodifica la información, como muestra la Figura. 4.36. para luego, ser almacenados en un documento tipo texto, se

crea un documento para cada módulo de medida de energía. Este procedimiento se lo realiza cada 10 minutos, el servidor envía un mensaje “*getstring*” a los terminales, estos validan el mensaje recibido y envían la información.

En la Figura. 4.37, se observa parte del lenguaje de programación, donde la información es enviada y decodificada y validada en el servidor, este envía un mensaje para poner en cero el valor de energía medida hasta ese momento.



```
pi@raspberrypi: ~/Desktop/servidorRasPi/NRF24L01
GNU nano 3.2 SendReceiveArduino.py

time.sleep(1 / 100)
if time.time() - start > 3:
    print("Timed out.")
    contador=contador+1
    break
receivedMessage = []
radio.read(receivedMessage, radio.getDynamicPayloadSize())
print("Received: {}".format(receivedMessage))
print("Translating the receivedMessage into unicode characters")
string = ""
for n in receivedMessage:
    # DECODIFICA EL MENSAJE
    if (n >= 48 and n <= 57 or n == 46):
        contador=10
        string += chr(n)
print("Out received message decodes to: {}".format(string))
radio.stopListening()
# CONFIRMAR DATO Y MANDAR A ENCCERAR AL MODULO 1
if len(string)>0:
    message = list("Encerar")
    while len(message) < 32:
        message.append(0)
    time.sleep(3)
    radio.write(message)
    print("Sent the message: {}".format(message))
```

Figura. 0.128. Algoritmo de envío de datos al servidor

Fuente: El autor.

4.7. Configuración del servidor de la nube

Una vez definida la plataforma IoT se crea una cuenta de ThingSpeak, donde se almacenarán los datos del sistema de monitoreo de energía, se visualiza en la Figura. 4.38.

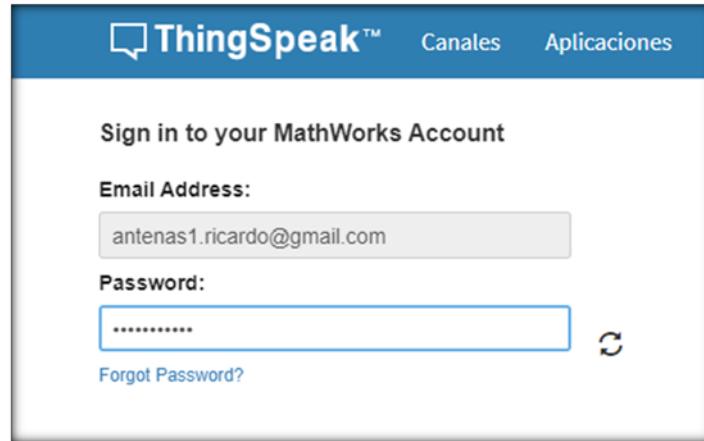


Figura. 0.129. Ingreso a la cuenta base de datos

Fuente: El autor

En la Figura. 4.39. se observa los campos que se deben ingresar para la configuración del canal, aquí se especifica donde se almacenarán los datos de los medidores de energía.



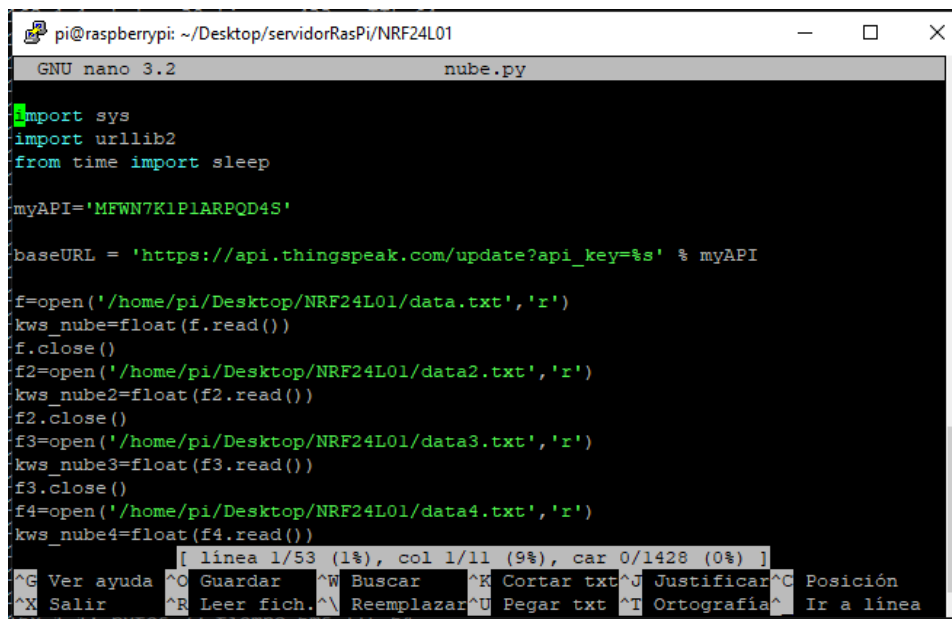
Figura. 0.130. Campos De Almacenamiento

Fuente: El autor

4.8. Configuración de la comunicación entre el nodo central y el servidor

Para el proyecto se ha creado una cuenta para uso de los recursos de la plataforma libre ThingSpeak. Para acceder a la información del servidor IoT, se dispone de una API para escritura de datos: **MFWN7K1P1ARPQD4S** y una para lectura de datos: **OZ4AP9MLBCS3Y7KQ**

Con el fin de acceder a la información de la nube se utiliza la librería “urllib2”, según la Figura. 4.40 que permite acceder y recuperar datos de internet utiliza el protocolo HTTP en este caso, se accede a la cuenta a la cuenta de ThingSpeak para escribir o leer información en cada canal.



```
pi@raspberrypi: ~/Desktop/servidorRasPi/NRF24L01
GNU nano 3.2 nube.py
import sys
import urllib2
from time import sleep

myAPI='MFWN7K1P1ARPQD4S'

baseURL = 'https://api.thingspeak.com/update?api_key=%s' % myAPI

f=open('/home/pi/Desktop/NRF24L01/data.txt','r')
kws_nube=float(f.read())
f.close()
f2=open('/home/pi/Desktop/NRF24L01/data2.txt','r')
kws_nube2=float(f2.read())
f2.close()
f3=open('/home/pi/Desktop/NRF24L01/data3.txt','r')
kws_nube3=float(f3.read())
f3.close()
f4=open('/home/pi/Desktop/NRF24L01/data4.txt','r')
kws_nube4=float(f4.read())

[ línea 1/53 (1%), col 1/11 (9%), car 0/1428 (0%) ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

Figura. 0.131. Configuración de librerías para acceder a los datos en la nube

Fuente: El autor

El algoritmo lee los documentos de texto que se han creado y guardado, para cada uno de los medidores de energía, antes de que los datos sean enviados, se valida el valor numérico de cada variable y se envía la información a la nube para ser almacenados en los diferentes campos del canal. Luego de este procedimiento se da la orden de poner en cero los documentos de texto, donde se almacenará la nueva información actualizada cada hora.

4.9. Elaboración de la aplicación

La aplicación desarrollada en la plataforma web MTI APP inventor, estará conformada por tres pantallas que son:

4.9.1. Pantalla de Inicio de sesión

En la pantalla de inicio Figura. 4.41. se visualiza el logo de la universidad Israel y en la parte inferior, de la pantalla, se visualiza dos campos que el usuario debe llenar, estos son el usuario y un clave de cuatro caracteres para las respectivas pruebas se ha colocado el usuario Ricardo y la clave es 1234



Figura. 0.132. Pantalla de inicio

Fuente: El autor

En el diagrama de bloques de la Figura. 4.42. se observa la lógica de programación y los campos que se han llenado para validar los campos especificados anteriormente.

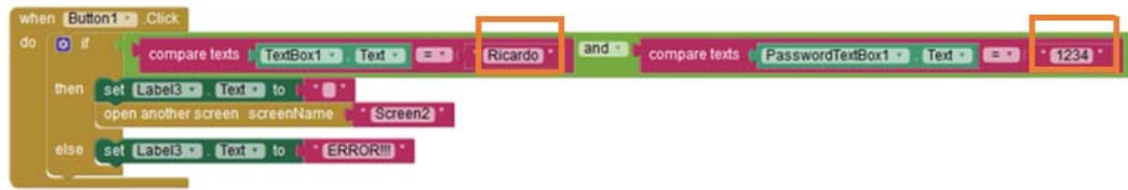


Figura. 0.133. Diagrama de bloques de la programación pantalla principal

Fuente: El autor

4.9.2. Pantalla para visualización de dispositivos conectados.

Esta pantalla se dispone de tres botones: el primer botón llamado “Leer Estado”, envía un mensaje al servidor en la nube y lo almacena, este campo es leído por el servidor de la red y ejecuta un algoritmo, el cual una vez finalizado envía un dato hacia la nube con la información del estado en que se encuentra el relay, (on – off), también informa si se encuentra conectada una carga mayor a 5W. En la Tabla. 4.5. se puede apreciar las diferentes comparaciones y combinaciones que entregara el algoritmo a la nube.

Tabla. 0.5. Combinaciones de estados entregados a la nube.

Estado 1	Estado 2	ESTADO
A	A	Sin carga y relé apagado
A	B	Sin carga relé encendido
B	B	Con carga y relé encendido

Fuente: El autor

Una vez recibida la respuesta solicitada, mediante una señal luminosa la cual es el estado del dispositivo es decir, si se encuentra activado el switch encenderá una alarma luminosa de color verde, caso contrario permanecerá apagada como muestra la Figura. 4.43.

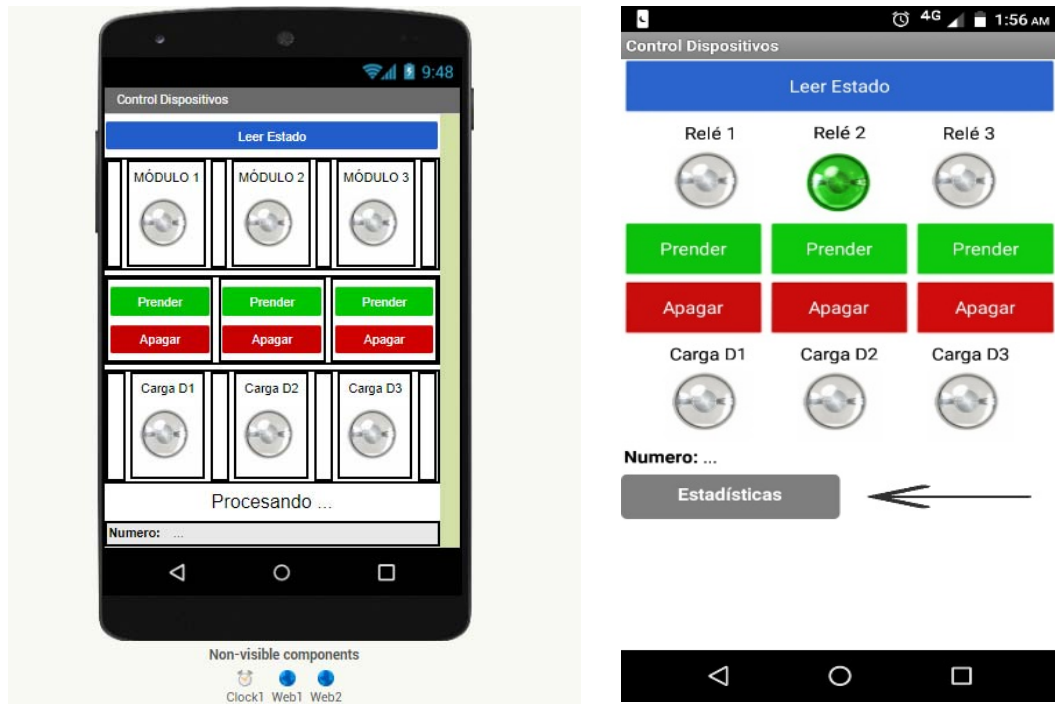


Figura. 0.134. Estado de switch de cada dispositivo

Fuente: El autor

De la misma manera, si existe carga conectada en el interfaz se encenderá el icono de carga como se observa en la Figura. 4.44.



Figura. 0.135. Lectura de estados en los dispositivos de medida

Fuente: El autor

Los botones de encender y apagar, al presionar uno de estos botones se enviará un valor a la nube que permitirá activar o desactivar un relé en cualquiera de los tres dispositivos de monitoreo, el dato se almacena en un campo llamado relé, dependiendo del valor que reciba, se activará o desactivará un switch del dispositivo seleccionado, como se especifica en la Figura. 4.45.

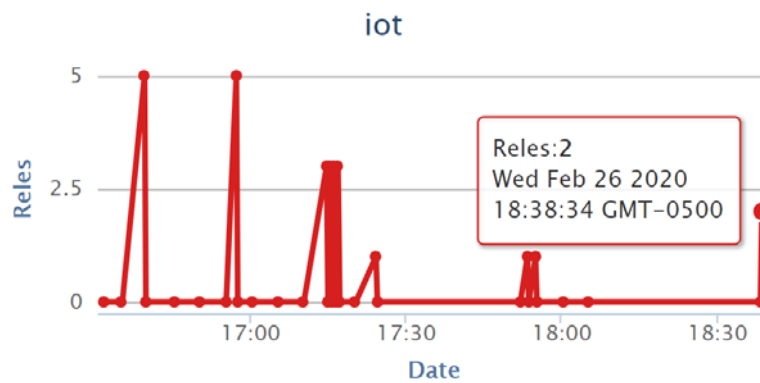


Figura. 0.136. Estado de relés

Fuente: el autor

En la Tabla. 4.6 se encuentra los valores que se envían desde la aplicación celular hacia la nube, se almacenan en el campo relés, los cuales son leídos por el servidor para realizar las diferentes acciones en los dispositivos.

Tabla. 0.6. Valores entregados desde la aplicación para control de relés

Dispositivo	Valor para activar	Valor para desactivar
1	1	2
2	3	4
3	5	6

Fuente: el autor

4.9.3. Pantalla para visualización de datos de consumo de energía.

Para acceder a la información de consumo de energía, se ha diseñado una tercera pantalla Figura. 4.46. donde se aprecian las lecturas realizadas. Para la visualización de los datos se ha configurado *web viewer*, la cual permite visualizarlos de manera gráfica luego de acceder a la dirección Url y extraer la información requerida.



Figura. 0.137. Tercera pantalla visualización de datos

Fuente: el autor

También se muestra parte del programa utilizado para la visualización de estos datos, Figura. 4.47.

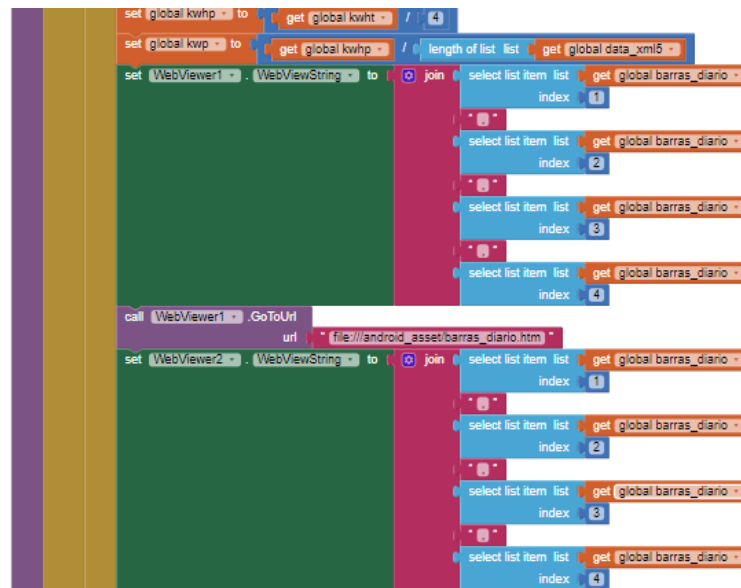


Figura. 0.138. Diagrama de bloques para la visualización de datos

Fuente: El autor

4.10. Configuración del Crontab

Configuración de Crontab, con la finalidad de que el sistema se inicie automáticamente al encender el servidor se han configurado dos actividades: el primero se ejecutará el programa llamado “SendReceiveArduino.py”, que es un algoritmo realizado en Python, el cual pide verificación de datos entre los diferentes módulos cada diez minutos, además solicita el envío de los valores almacenados en la memoria de Arduino de cada módulo, la información es leída y almacenada en un archivo tipo texto.

El segundo programa que se ejecutara es nube.py, el cual se encarga de publicar en la nube, información almacenada en el archivo de texto que se crea en el programa mencionado anteriormente, esta información es almacenada en el campo del servidor de la nube. Este proceso se ejecutará cada hora.

Como se observa en la Figura. 4.48 la configuración de Crontab que permite ejecutar los programas en tiempos establecidos.

```
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
@reboot python /home/pi/Desktop/NRF24L01/receivenube.py 1> /dev/null 2> /home/pi/Desktop/NRF24L01/error.log
#* * * * * sh ./home/pi/Desktop/NRF24L01/prueba.sh
*/10 * * * * python /home/pi/Desktop/NRF24L01/SendReceiveArduino.py > /home/pi/Desktop/log.txt
0 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23 * * * python /home/pi/Desktop/NRF24L01/nube.py > /home/pi/Desktop/d.txt
```

Figura. 0.139. Configuración Crontab

Fuente: El autor

Una vez implementado en su totalidad el prototipo y de realizar las diferentes pruebas de funcionamiento en la Figura. 4.49. se observa la red de sensores compuesta por el servidor central y los cuatros sensores de consumo de energía.

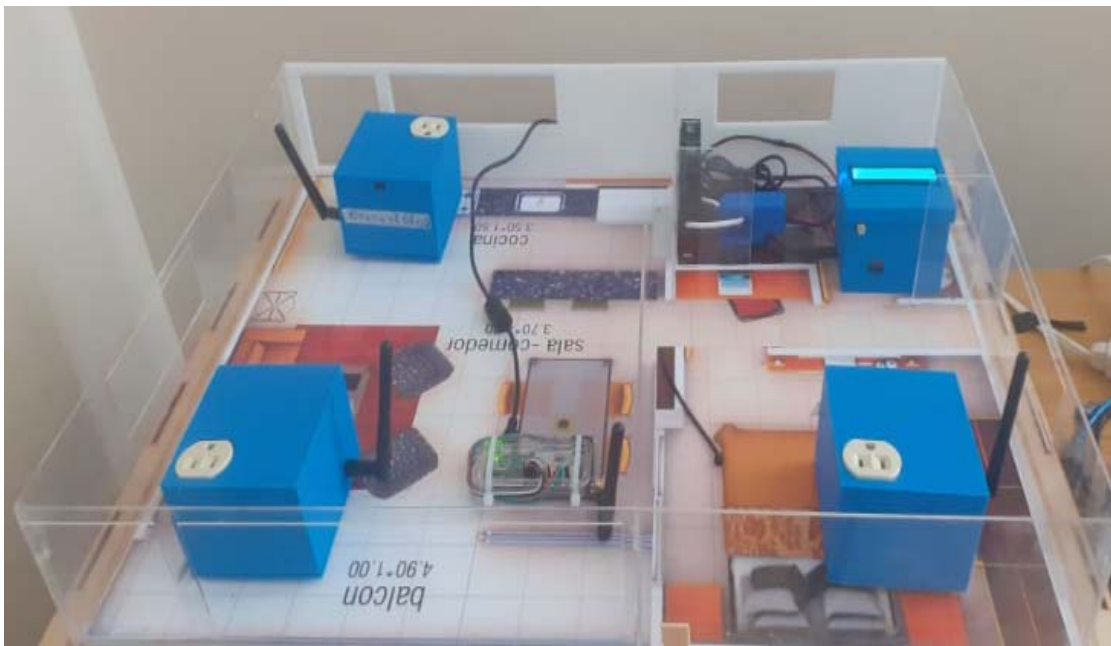


Figura. 0.140. Maqueta con los sensores de adquisición de datos

Fuente: El autor

4.11. Análisis de los resultados medidos

Para calcular un aproximado del consumo medido de energía en un domicilio, se ha tomado como referencia la Tabla. 1.2 del capítulo 1, emitida por ARCONEL que regula el valor kWh. En la Tabla. 4.7 se muestra el valor medio utilizado para realizar un aproximado del cálculo de consumo en dólares.

Tabla. 0.7. Valor promedio del costo kWh

Rango de consumo kWh	Estrato consumo	Energía USD /kWh	Comercialización (USD/Consumidor)
ene-50	E	0,091	1.414
51-100	E	0,093	
101-150	D	0,095	
151-200	C	0,097	
201-250	C	0,099	
251-300	B	0,101	
301-350	B	0,103	
351-500	A	0,105	
Valor promedio		0,098	

Fuente: el autor

Para conocer el comportamiento de los aparatos eléctricos en el hogar, se ha seleccionado los de mayor uso en el domicilio del autor, efectuando la toma de muestras durante un periodo de 4 días.

En los siguientes numerales se señalan los medidores de energía en el periodo establecido.

4.11.1. Medidor de energía 1

Se ha seleccionado un refrigerador ya que este prácticamente permanece conectado a la toma eléctrica todo el tiempo, en la Tabla. 0.8, se observa los datos descargados de la plataforma

de almacenamiento, con la finalidad de conocer el comportamiento de trabajo del aparato en función de las variaciones del consumo de energía.

Tabla. 0.8. Consumo de energía módulo 1 – Refrigeradora

Módulo 1 REFRIGERADORA [kWs]					
fecha de creación	Domingo	Lunes	Martes	Miércoles	Jueves
1/3/2020 0:00	63,33	330,63	0,03	175,72	401,61
1/3/2020 1:00	0	217,21	0	0	302,83
1/3/2020 2:00	17,36	0	0	0	248,56
1/3/2020 3:00	0	0	133,08	0	0
1/3/2020 4:00	0,02	0	289,67	0	0
1/3/2020 5:00	0,03	0	272,84	0	0
1/3/2020 6:00	0,02	9,67	144,12	0,09	53,67
1/3/2020 7:00	105,35	295,46	0,01	293,01	292,88
1/3/2020 9:00	0,33	283,94	0	301,6	271,84
1/3/2020 10:00	0,14	0,05	0	291,08	
1/3/2020 11:00	0,04	0	0	59,5	
1/3/2020 12:00	0	0	93,83	0,4	
1/3/2020 13:00	0	0	390,08	0	
1/3/2020 14:00	0,01	0,11	0	0	
1/3/2020 15:00	287,42	0	0	0	
1/3/2020 16:00	253,96	211,61	768,09	136,47	
1/3/2020 17:00	0	378,63	0	291,73	
1/3/2020 18:00	229,38	266,16	0,06	278,37	
1/3/2020 19:00	72,04	198,75	0	79,48	
1/3/2020 19:15	0	0	0,25	0,01	
1/3/2020 19:44	0,03	105,98	0,04	0,04	
1/3/2020 20:00	130,46	270,52	0	0,04	
1/3/2020 22:00	363,21	0,05	140,24	0	
1/3/2020 23:00	1523,13	0	290,03	151,24	
CONSUMO TOTAL	3046,26	2568,77	2522,37	2058,78	1571,39
Consumo total en kWh			3,26		
Consumo en dólares			0,29		

Fuente: El autor

Al analizar los valores de consumo de la refrigeradora se observa un mayor consumo el domingo, esto se debe a que el usuario utiliza con más frecuencia el equipo, es decir las puertas de la refrigeradora se han abierto un mayor tiempo, a diferencia de los días entre semana debido a que el usuario pasa un menor tiempo en el domicilio.

En la Figura. 0.141. se observa el comportamiento normal del electrodoméstico donde mantiene un comportamiento bastante continuo, es decir si el aparato no funciona correctamente se observaría consumos de energía por tiempos más prolongados o con picos de consumo de energía mayor.

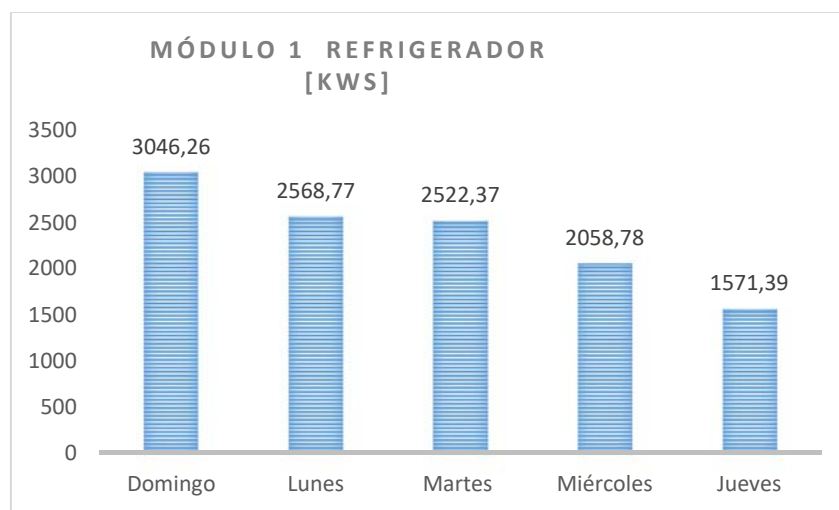


Figura. 0.50. Comportamiento del consumo de energía módulo 1

Fuente: El autor

4.11.2. Medidor de energía 2

El segundo aparato que se ha seleccionado para el análisis de datos es un televisor ya que este prácticamente suele estar más tiempo encendido, en la Tabla. 0.9. se observa los datos descargados de la plataforma de almacenamiento, donde permite conocer el comportamiento de trabajo del aparato, las variaciones de consumo de energía durante el tiempo que ha sido monitoreado durante un período de cuatro días

Gracias al sistema de monitoreo se ha observado que el aparato muchas veces se lo enciende sin que nadie lo utilice y genera un consumo de energía innecesario, en este periodo de

tiempo se trató de controlar los tiempos muertos y apagar el aparato, logrando una reducción en el consumo.

Tabla. 0.9. Consumo de energía módulo 2 – Televisión

Módulo 2 - TELEVISIÓN					
[kWs]					
Fecha de creación	Domingo	Lunes	Martes	Miércoles	Jueves
1/3/2020 0:00	14,33	159,39	5,88	4,08	4,44
1/3/2020 1:00	2,95	152,17	5,22	5,95	0,13
1/3/2020 2:00	6,71	3,62	5,45	0	0
1/3/2020 3:00	9,62	0	5,55	0,96	0
1/3/2020 4:00	0,11	0	5,62	0	0
1/3/2020 5:00	0,17	0	5,59	0	0
1/3/2020 6:00	0,14	0	5,33	0	0
1/3/2020 7:00	136,02	0	4,68	0	0
1/3/2020 8:00	187,69	0	3,95	0	0
1/3/2020 9:00	160,18	82,81	113,41	0	61,91
1/3/2020 10:00	70,42	160,94	85,62	0	173,5
1/3/2020 11:00	70,59	99,93	113,51	0	149,89
1/3/2020 12:00	66,82	155,97	151,17	0,02	0
1/3/2020 13:00	90,39	300,94	0	0	208,74
1/3/2020 14:00	158,08	66,81	0	125,96	0,72
1/3/2020 15:00	192,8	3,44	244,7	151,09	0,84
1/3/2020 16:00	249,34	3,61	176,23	77,62	0,67
1/3/2020 17:00	235,71	3,03	144,79	52,77	0,82
1/3/2020 18:00	88,79	4,18	0	99,28	0,76
1/3/2020 19:00	99,33	29,86	251,53	114,97	147,68
1/3/2020 20:00	35,9	11,3	209,72	183,21	183,19
1/3/2020 21:00	131,21	0	187,11	183,88	157,48
1/3/2020 22:00	189,08	449,33	0	0	147,92
1/3/2020 23:00	189,08	209,02	5,03	0	147,46
CONSUMO TOTAL	2385,46	1896,35	1730,09	999,79	1386,15
Consumo total en kWh			2,33		
Consumo aproximado en dólares			0,23		

Fuente: El autor

En la Figura. 0.51. se aprecia el comportamiento del uso del televisor, observando que al controlar los tiempos muertos y luego proceder con el apagado del equipo, se reduce el consumo de energía.

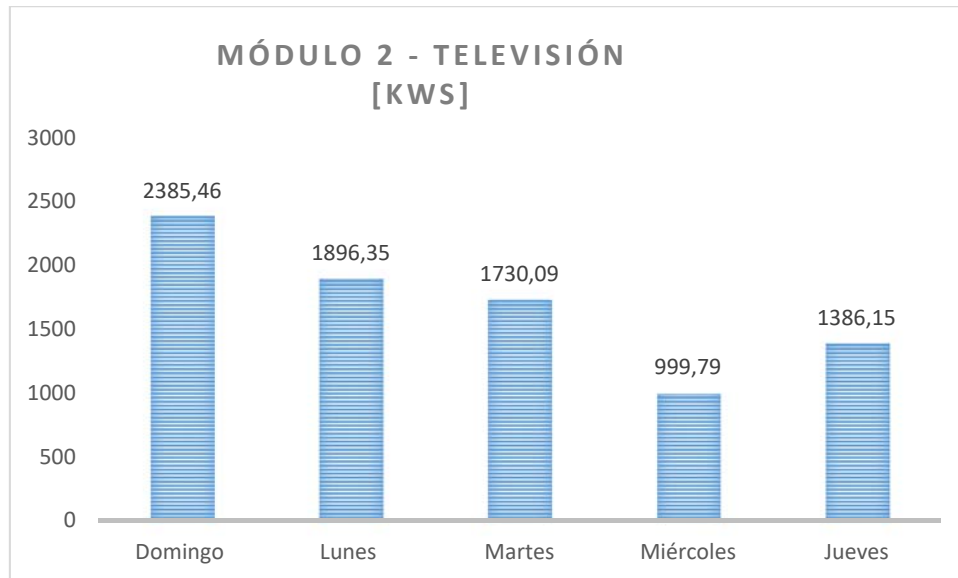


Figura. 0.51. Comportamiento del consumo de energía módulo 2

Fuente: El autor

4.11.3. Medidor de energía 3.

Para las pruebas de funcionamiento en el módulo tres, se ha utilizado una lámpara como carga la cual se la desconectaba remotamente desde la aplicación celular con la finalidad de reducir el consumo de energía y como en los casos anteriores se realiza un análisis del consumo de energía en la Tabla. 4.10., se muestra la información recolectada en un periodo de tiempo 4 días. Y en la Figura. 4.52 se observa la optimización del consumo de energía del mismo módulo.

Tabla. 0.10. Consumo de energía módulo 3 Lámpara

Módulo 3 LÁMPARA DE SALA [kWs]				
FECHA DE CREACION	LUNES	MARTES	MIERCOLES	JUEVES
1/3/2020 0:00	5,51	0	0	17,83
1/3/2020 1:00	5,12	0	0	7,2
1/3/2020 2:00	5,43	0	0	0
1/3/2020 3:00	5,42	0	12,68	0
1/3/2020 4:00	5,38	0	0	0
1/3/2020 5:00	5,33	0	0	0
1/3/2020 6:00	5,31	0	0	0
1/3/2020 7:00	5,39	0	0	0
1/3/2020 8:00	5,2	0	0	0
1/3/2020 9:00	5,75	10,49	0	0
1/3/2020 10:00	5,06	23,33	0	1,71
1/3/2020 11:00	4,65	15,67	7,35	0
1/3/2020 12:00	4,26	7,7	1,28	0
1/3/2020 13:00	10,05	0	0	0
1/3/2020 14:00	1,64	0	11,33	7,59
1/3/2020 15:00	5,2	13,86	5,55	13
1/3/2020 16:00	5,19	7,34	4,72	27,16
1/3/2020 17:00	6,62	17,47	5,82	0
1/3/2020 18:00	6,3	0	0	2,81
1/3/2020 19:00	1,81	19,95	1,19	24,35
1/3/2020 19:15	6	4,19	0	19,85
1/3/2020 19:44	30,75	19,13	3,12	14,25
1/3/2020 20:00	30,94	0	0	2,36
1/3/2020 22:00	22,77	1,24	12,68	6,5
1/3/2020 23:00	195,08	140,37	65,72	1,23
Consumo Total	390,16	280,74	131,44	145,84
Consumo total en kWh			0,26	
Consumo aproximado en dólares			0,026	

Fuente: El autor

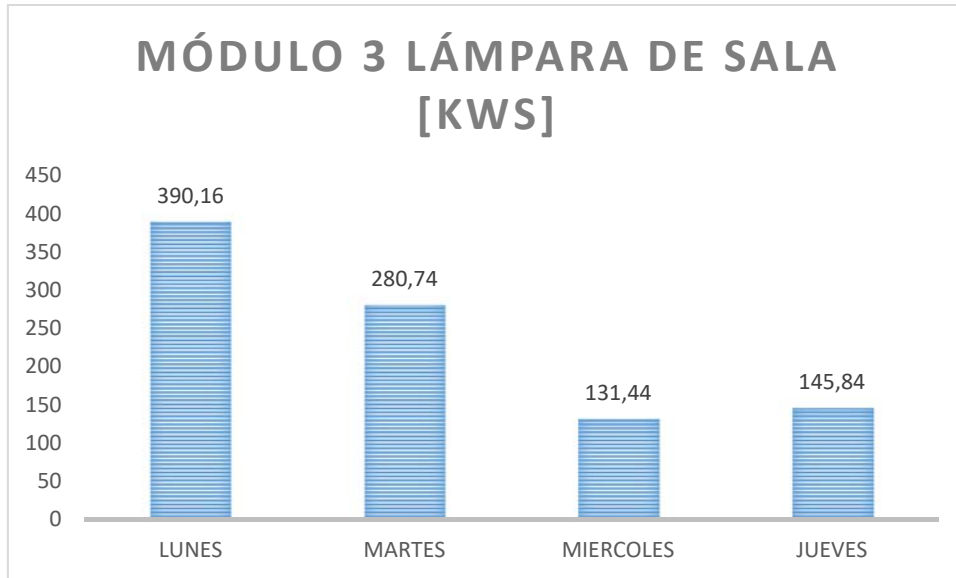


Figura. 0.52. Comportamiento del consumo de energía módulo 3

Fuente: El autor

4.11.4. Medidor de energía 4 Acometida.

El aparato se encarga de monitorear el consumo total de los equipos conectados a la red eléctrica dentro del domicilio, a manera de control se estima cuál de los aparatos conectados es el que tiene mayor consumo de energía luego de comparar las lecturas adquiridas en los módulos de cargas individuales, esto se observa en la Tabla. 4.11.

Tabla. 0.11. Consumo de energía módulo total de la línea.

Módulo 4 ACOMETIDA				
FECHA DE CREACION	LUNES	MARTES	MIERCOLES	JUEVES
1/3/2020 0:00	0	89,74	211,4	430,59
1/3/2020 1:00	430,69	33,87	163,77	249,36
1/3/2020 2:00	0	33,79	0	201,63
1/3/2020 3:00	89,74	76,65	164,62	193,95
1/3/2020 4:00	95,4	119,05	152,58	159,56
1/3/2020 5:00	63,9	107,65	25,48	29,41
1/3/2020 6:00	90,06	97,29	65,65	104,02
1/3/2020 7:00	29,3	69,32	309,11	212,04
1/3/2020 8:00	140,65	32,65	126,48	156,48
1/3/2020 9:00	209,3	253,38	117,06	308,57
1/3/2020 10:00	469,25	266,1	91,22	278,13
1/3/2020 11:00	279,9	272,33	86,09	261,93
1/3/2020 12:00	165,98	528,69	74,76	0
1/3/2020 13:00	64,07	0	0	961,4
1/3/2020 14:00	212,26	0	225,67	88,64
1/3/2020 15:00	175,31	882,84	309,32	132,66
1/3/2020 16:00	159,15	271,17	279,3	262,41
1/3/2020 17:00	266,69	248,51	223,28	251,28
1/3/2020 18:00	89,39	0	188,03	174,45
1/3/2020 19:00	80,08	330,78	338,97	355,72
1/3/2020 19:44	812,88	379,01	449,82	441,73
1/3/2020 20:00	439,99	257,42	361,74	516,04
1/3/2020 22:00	398,55	285,33	0	332,32
1/3/2020 23:00	341,1	284,64	579,34	329,57
Consumo Total	5103,64	4920,21	4543,69	6431,89
Consumo total en kWh			5,83	
Valor aproximado en dólares			0,57	

Fuente: El autor

En la Figura. 4.53. se aprecian los datos adquiridos durante el tiempo de monitoreo.

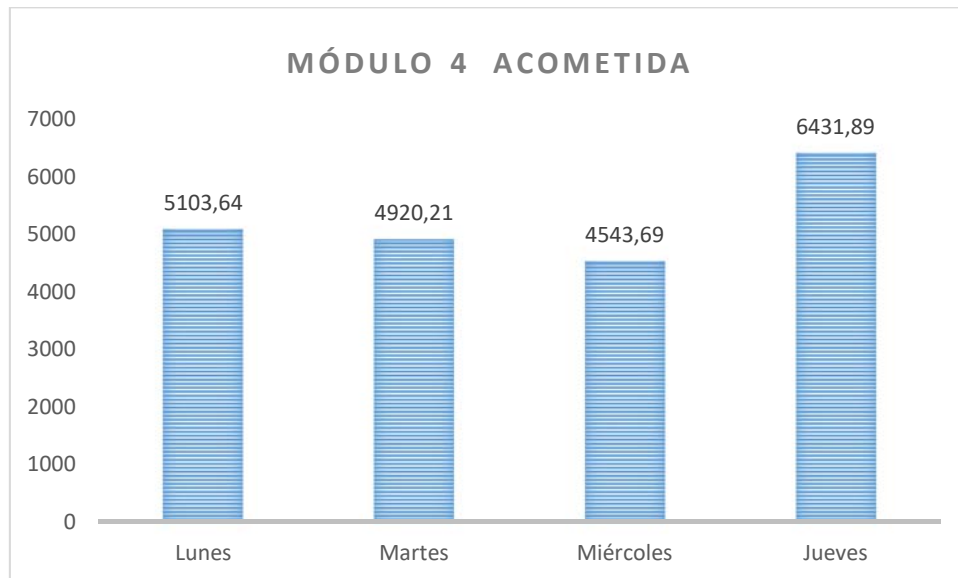
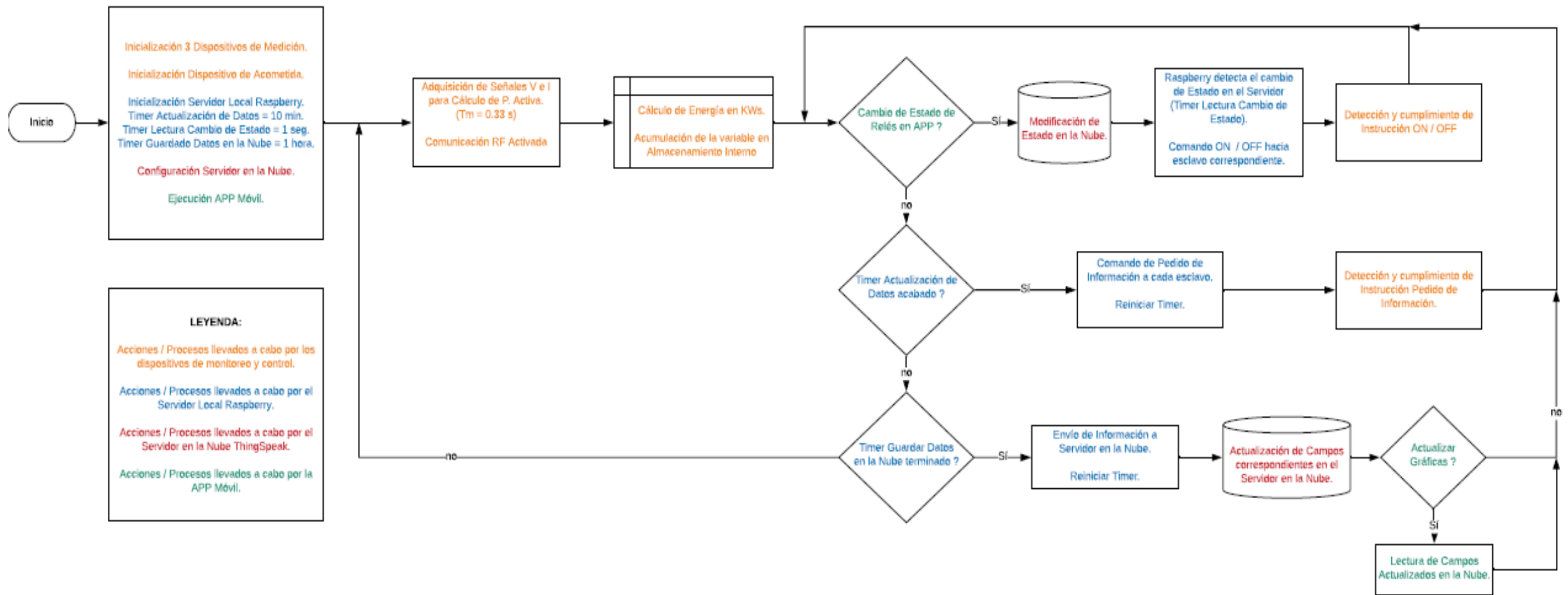


Figura. 4.53. Consumo de energía módulo 4 acometida

Fuente: El autor

Cabe señalar que el módulo central presenta un incremento del consumo, en el cuarto día de monitoreo, debido a un incremento en el uso de cargas que no ha sido considerado para obtener en la lectura de medidas y observar su comportamiento, tomando en cuenta que los aparatos escogidos para ser monitoreados constituyen aproximadamente el 48% de la carga utilizada es decir, en el sistema eléctrico existen varias cargas adicionales como impresoras, computadores, laptops etc, que no se consideran para el presente cálculo de consumo de energía.

4.12. Diagrama de bloques total del sistema



CONCLUSIONES

- La información se adquiere con ayuda de los sensores de consumo de energía, analizada en base a cuadros estadísticos guardados en la nube, durante un periodo de cuatro días, para las cargas de mayor consumo en la casa del autor.
- Para la etapa de comunicación entre los sensores y el servidor, se eligió el módulo NRF 24L01+ por su bajo costo y facilidad de configuración además, de proveer la suficiente documentación por parte del fabricante para su uso. En cuanto al control de consumo de energía se implementó una aplicación para sistema Android, la cual verifica que el factor de potencia este dentro del rango permitido $fp. = 0,92$.
- Mediante una aplicación diseñada en la plataforma MIT App inventor, se controla una etapa de electrónica que activa o desactiva un relé, suministrando o no, energía a cada carga.
- Mediante el uso de la plataforma MIT App inventor se creó una aplicación para sistema Androide en la cual el usuario visualiza el historial de consumo que se ha almacenado en la nube.
- Una vez definida las etapas de comunicación y control de los sensores de monitoreo, se procede a la construcción de la maqueta que simula tres ambientes: sala comedor y dormitorio, la cual se utilizó para exhibición del sistema.
- El historial se presentó con la ayuda de graficas estadísticas en las cuales se observa el comportamiento de consumo de los aparatos durante el periodo de tiempo de cuatro días, además se pude acceder a la información estadística de consumo utilizando la aplicación Android.
- Para la validación del prototipo se realizó pruebas de funcionamiento en compañía del tutor, que consistió en controlar las cargas ubicadas en el domicilio del autor a través de la aplicación Android, desde la universidad Israel mediante video conferencia.

RECOMENDACIONES

- Dado a la existencia de cargas en el hogar que funcionan con 220 V, se plantea el diseño de un sistema de monitoreo bifásico.
- El análisis en periodos de tiempo mayores brindarán una información más fiable y así diseñar un plan de para ahorro de energía.
- Es necesario evitar el uso de puentes o cables para la conexión del módulo NRF 24L01 para evitar la diafonía. En cuanto al factor de potencia este debe ser lo más próximo a la unidad, caso contrario podría llevar a penalización de consumo por un factor de potencia menor al permitido.
- Se recomienda el uso de la aplicación MIT App inventor ya que brinda un sin número de herramientas para el diseño de una aplicación para sistema Android.
- Es importante el uso de las plataformas libres como ThingSpeak, para el desarrollo de prototipos basados en el concepto de internet ya que brindan las suficientes herramientas para la ejecución de prototipos.
- Es recomendable habilitar un punto de acceso a internet eficaz, para realizar pruebas satisfactorias del funcionamiento del proyecto basadas en el internet de las cosas.

REFERENCIAS BIBLIOGRÁFICAS

- Alava, A., Campos, H., & Falcones, S. (s.f.). Diseño de una Fuente de Alimentación de Dos Etapas: AC-DC con Corrección de Factor de Potencia y DC-DC con un Convertidor Director (Forward Converter) e Implementación de su Etapa AC-DC. Diseño de una Fuente de Alimentación de Dos Etapas: AC-DC con Corrección de Factor de Potencia y DC-DC con un Convertidor Director (Forward Converter) e Implementación de su Etapa AC-DC. Escuela Superior Politécnica del Litoral (ESPOL), Guayaquil.
- Aprendiendo Arduino. (16 de 11 de 2016). Obtenido de <https://aprendiendoarduino.wordpress.com/>
- ARCOTEL. (2019). Obtenido de <https://www.regulacionelectrica.gob.ec/proyecto-hidroelectrico-mazar-dudas/>
- Arduino. (s.f.). Arduino. Obtenido de <https://www.arduino.cc/>
- Blogger. (14 de 04 de 2015). Obtenido de DIFERENCIAS DE LAS FUENTES LINEALES CON RESPECTO ALAS FUENTES CONMUTADAS: <http://fuentecommutada.blogspot.com/2015/04/diferencias-de-las-fuentes-lineales-con.html>
- Brother. (24 de marzo de 2014). brother at your side. Obtenido de brother at your side: <https://global.brother/en>
- Brother. (24 de marzo de 2014). Brother at your side. Obtenido de Brother at your side: <https://global.brother/en>
- Chacón Cajo, S. F. (09 de febrero de 2018). IMPLEMENTACIÓN DE UN EQUIPO DE CONTRASTACIÓN PARA CONTADORES DE ENERGÍA ELÉCTRICA MONOFÁSICO, PARA EL LABORATORIO DE MEDIDORES DE LA EMPRESA ELÉCTRICA QUITO. Quito, Pichincha, Ecuador.
- Ecuared. (23 de JUNIO de 2013). Potencia eléctrica. Obtenido de Ecuared: ecured.cu/Potencia_eléctrica
- ECUARED. (23 de JUNIO de 2013). Potencia eléctrica. Obtenido de ECUARED: ecured.cu/Potencia_eléctrica
- ECUATRAN. (5 de febrero de 2018). Obtenido de ECUATRAN: <https://www.ecuatran.com/blog/conoce-cuales-son-las-9-centrales-hidroelectricas-que-existen-en-ecuador/>
- Enciclopedia-Britanica. (s.f.). Enciclopedia Britanica. Obtenido de <https://www.britannica.com/search?query=Electromagnetic+spectrum+telecommunication>
- Enríquez, G. (2009). Tecnologías de generación de energía eléctrica. Limusa.
- Especator, E. (19 de 12 de 2012). INFORMATICA DEL FUTURO IMITARA LOS SENTIDOS.
- ESPECTADOR, E. (19 de 12 de 2012). INFORMATICA DEL FUTURO IMITARA LOS SENTIDOS. Colombia.

- Gartner. (2014). Gartner. Obtenido de <https://www.gartner.com/en/information-technology/insights/internet-of-things>
- González, S. R. (2008). Análisis de circuitos eléctricos. México: Cengage Learning.
- Himadri Nath, S., Shashwata, M., Shinjan, M., Soham, B., & Urmi, S. (05 de 07 de 2017). Comparative Performance Analysis between nRF24L01+ and XBEE ZB Module Based Wireless Ad hoc Networks. Comparative Performance Analysis between nRF24L01+ and XBEE ZB Module Based Wireless Ad hoc Networks. Calcuta, India.
- <http://www.ieee802.org/>. (27 de 01 de 2010). IEEE 802.15 WPAN. Obtenido de <http://www.ieee802.org/>
- IEEE.ORG. (2011). Obtenido de <http://grouper.ieee.org/groups/802/11/>
- IEEE802.15. (27 de 01 de 2010). IEEE 802.15 WPAN. Obtenido de <http://www.ieee802.org/>
- Inventable Electrónica Simple Y Clara. (2013). Obtenido de <https://www.inventable.eu/>
- ITU. (2015). ITU. Obtenido de Measuring the Information Society Report: <http://www.itu.int/en/ITU-D/>
- Joaquín, R. M. (s.f.). Energía eléctrica. Obtenido de http://newton.cnice.mec.es/materiales_didacticos/energia/electrica.htm
- Kurose, J. F., & Ross, K. W. (2017). Redes de computadores un enfoque descendente. Madrid: PEARSON EDUCACIÓN, S. A.
- LOZURITECH. (2016). Obtenido de <https://www.lozurytech.com/>
- Mechatronics, N. (2016). Tutorial sensor de corriente AC no invasivo SCT-013. Obtenido de https://naylampmechatronics.com/blog/51_tutorial-sensor-de-corriente-ac-no-invasivos.html
- MECHATRONICS, N. (2016). Tutorial sensor de corriente AC no invasivo SCT-013. Obtenido de https://naylampmechatronics.com/blog/51_tutorial-sensor-de-corriente-ac-no-invasivos.html
- MIT APP INVENTOR. (2012-2020). Obtenido de MIT APP INVENTOR: <https://appinventor.mit.edu/>
- Molina, C. E. (s.f.). Red TAUROS. Obtenido de <https://redtauros.com/site/fundamentos-de-redes/>
- Nanjing Zeming Electronic Co., L. (s.f.). ZMPT107 potential transformer. Obtenido de <http://www.zeming-e.com>
- Non-IP Smart Object Technologies. (s.f.). Obtenido de <https://www.sciencedirect.com/>
- Non-IP Smart Object Technologies. (2010). Obtenido de <https://www.sciencedirect.com/>
- OpenEnergyMonitor, L. (s.f.). YHDC SCT-013-000 Transformador de corriente.
- Pallo Noroña, J. P., & Salazar Vásquez, A. S. (febrero de 2017). Sistema electrónico de monitoreo y control para la distribución de energía eléctrica en los hogares. Ambato, Tungurahua, Ecuador.

- PE, I. (2017). como hacer.ue. Obtenido de Análisis comparativo de las placas Arduino (oficiales y compatibles)
- Prieto Blázquez, J. (s.f.). Introducción a los sistemas digitales de comunicacion. Catalunya.
- Proakis, J. (2007). Tratamiento digital de señales. Madrid: PEARSON EDUCACIÓN.
- Raspberrypiparatorpes. (2013). Obtenido de <https://rasberrypiparatorpes.net/>
- RASPBERRYPIPARATORPES. (2013). Obtenido de <https://rasberrypiparatorpes.net/>
- Renovable, M. d. (s.f.). Obtenido de <https://www.rekursosyenergia.gob.ec/proyecto-hidroelectrico-minas-san-francisco/#>
- Samaniego Idrovo, D. I., & Valesaca Orellana, D. F. (Agosto de 2016). Diseño e implementacion de un medidor de energia electronico. Cuenca, Azuay, Ecuador.
- sanchez, E. C. (2008). Instalaciones electricas en edificios de viviendas (Segunda ed., Vol. se).
- Sánchez, E. C. (2008). Instalaciones electricas en edificios de viviendas (Segunda ed., Vol. se).
- Sanz Fernández, P. (junio de 2014). Comunicación por RF entre microcontroladores PIC18 mediante el módulo NRF24L01. Obtenido de <https://upcommons.upc.edu/handle/2099.1/23530>
- TEDIC. (17 de 02 de 2015). hacklabasu. Obtenido de Cómo conectar una RaspberryPi y Arduino por RF: <https://hacklabasu.tedic.org/>
- ThingSpeak. (1994-2020). Retrieved from <https://thingspeak.com/>
- TINKSPEAK. (1994-2020). Retrieved from <https://thingspeak.com/>
- Vega, E. (Febrero de 2016). Obtenido de <http://cursosvega.blogspot.com/p/contador-de-energia.html>

ANEXOS

ANEXO A



“Responsabilidad con pensamiento positivo”

UNIVERSIDAD TECNOLÓGICA ISRAEL

Prototipo de sistema de monitoreo y control de consumo de energía eléctrica para un domicilio aplicando el concepto de Internet de las cosas.

MANUAL DE USUARIO

AUTOR:

Sánchez Oñate Ricardo Mauricio

QUITO, ECUADOR

2020

Tabla de contenido

Notas para el usuario:	118
Advertencias.	118
Requisitos para la instalación del sistema.	118
Componentes del sistema de monitoreo	119
Configuración del nodo central.	120
Puesta en funcionamiento de sensores de monitoreo	121
Aplicación para celular sistema operativo Android.	121
Pantalla 1	121
Pantalla 2	123

Notas para el usuario:

- Antes de utilizar cualquier dispositivo electrónico sírvase leer las diferentes instrucciones para el correcto uso de los equipos y evitar daños prematuros del sistema de monitoreo de energía para domicilio basado en el principio del internet de las cosas
- El sistema de monitoreo únicamente trabajara correctamente en un sistema monofásico o bifilar.

Advertencias.

- Tener respeto a la electricidad, utilizar el equipo adecuado para la manipulación del sistema eléctrico monofásico doméstico, ya que se puede perder la vida.
- Evite la manipular con las manos húmedas los equipos electrónicos
- Este sistema trabaja a 120V con una corriente nominal de funcionamiento de 15 Amperios

Requisitos para la instalación del sistema.

- Disponer de una fuente confiable de internet.
- Una PC de escritorio.
- Conocimientos de configuración de un dispositivo a una red
- Conocimientos básicos de electricidad.
- Un teléfono celular.
- Monitor entrada HDMI
- Cable mini HDMI a HDMI.
- Teclado y ratón dos en uno.
- Guantes.
- Herramientas varias.
- Multímetro.
- Sistemas de monitoreo de energía.

Componentes del sistema de monitoreo

Los partes del sistema de monitoreo se presentan desde la Figura. A.1. hasta la Figura. A.4.



Figura. A.1. Número del módulo monitoreo de energías portables de energía

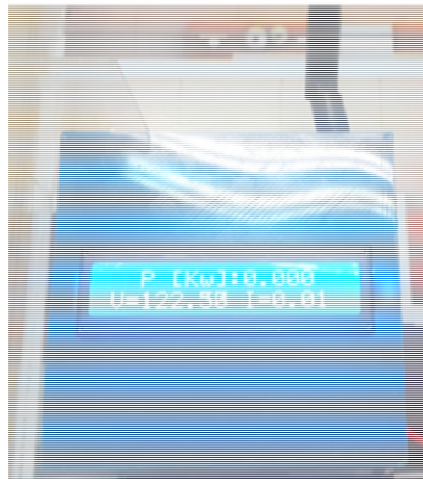


Figura. A.2. Número módulo de energía general 1

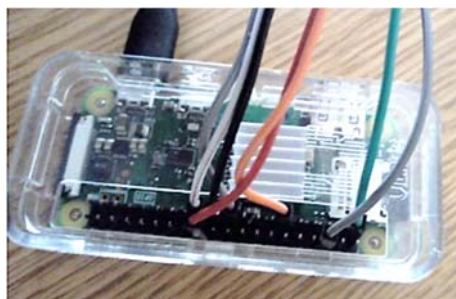


Figura. A.3. Nodo central (Raspberry) 1



Figura. A.4. Fuente energía nodo central (Raspberry) 1

Configuración del nodo central.

- Conectar los diferentes periféricos al servidor o nodo central (Raspberry)
- Encender el nodo central, Figura. A.5
- Configurar la red inalámbrica, mediante entorno gráfico.
- Verificar el acceso a internet desde el servidor o nodo central,
- El servidor debe ser colocado en la parte central de la red de sensores.
- Seleccionar los diferentes a aparatos eléctricos a monitorear, apagarlos.
- Conectar a la toma eléctrica el sensor de monitoreo
- Conectar el aparato al sensor de monitoreo ilustración 1.



Figura. A.5. Aparato eléctrico conectado al módulo de medida

Fuente: El autor

Puesta en funcionamiento de sensores de monitoreo

- Conectar el enchufe del aparato eléctrico al tomacorriente que es parte del sensor de monitoreo, en la
- Encender los sensores de monitoreo, espere unos minutos hasta que se inicializasen los parámetros, para empezar su funcionamiento.
- Instalar la aplicación en nuestro dispositivo Android.
- Ingresar el usuario (**Ricardo**) y clave de acceso (**1234**) como se muestra en la Figura. A.6.



Figura. A.6. Pantalla de inicio.

Fuente: El autor

Aplicación para celular sistema operativo Android.

En la aplicación una vez que se accedió se dispone de dos pantallas donde:

Pantalla 1

Antes de empezar a describir las funciones que cumplen los botones en la aplicación, se debe conocer que es permitido enviar un dato cada 15 segundos al servidor debido a que se ha configurado la cuenta en una versión gratuita. Es decir para enviar un nuevo dato hay que esperar un tiempo mayor a 15 segundos desde el ultimo envío.

En la Figura. A.7. se presenta de modo general el entorno principal para el control de los dispositivos de monitoreo.

Botones de Leer Estado

Al presionar este botón, se envía al servidor IoT, una solicitud para conocer si el sensor está activo o pasivo grafico Relé, como también conocer si tiene o no una carga mayor a 5 [W] conectada.



Figura. A.7. Pantalla control de módulos

Fuente: El autor

Botón prender: al presionar el botón de color verde, se acciona el switch del sensor de monitoreo

Botón apagar: al presionar este botón de color rojo, se desactiva el switch del sensor de monitoreo.

Nota: Una vez terminada la acción, esperar a que la aplicación se ponga lista para funcionar, presionar el botón “**Leer Estado**”, para actualizar la información en el dispositivo.

Botón estadísticas: Al presionar este botón color gris ubicado en la parte inferior derecha, se accede a la pantalla de estadísticas

Pantalla 2

A continuación en la Figura. A.8. se observa el entorno gráfico de la pantalla para la visualización de datos, en la cual se dispone de varios botones que permitirán acceder a la información obtenida desde la nube.



Figura. A.8. Pantalla control de módulos

Fuente: El autor

Botón para selección de módulos “M1” Al presionar este botón se despliega el listado de los módulos de monitoreo de energía, donde se seleccionará el módulo al cual se quiera acceder a la información.

Botón para selección de periodo de tiempo en la ilustración 4 se identifica el botón llamado “Mensual” el mismo, permite seleccionar el periodo de tiempo en el cual se desea verificar el consumo.

Botón OK: se presiona este botón para acceder a la gráfica de consumo de datos.

En la Figura. A.9. se observa las gráficas de la energía consumida por los diferentes aparatos eléctricos los datos obtenidos de la nube, vale señalar que los datos se presentan en las barras [kWs]. En la parte inferior de la gráfica se calcula el valor expresado de la energía

en kWh, como también un valor aproximado en dólares, se ha tomado como referencia 0.901 centavos según Tabla. 1.1. Capítulo 1.



Figura. A.9. pantalla de visualización de datos

Fuente: El autor

Para abandonar las pantallas se presiona la gráfica de la flecha tanto para regresar a la pantalla principal como para regresar a la pantalla inicio

Nota: para garantizar una buena comunicación del servidor y la nube, es necesario disponer de un buen proveedor de internet, ya que al no existir una buena comunicación existe fallos en el envío y recepción de datos.

Apagado del equipo.

1. Apagar el aparato conectado en módulo de monitoreo.
2. Apagar el switch del módulo “botón Apagar”
3. Apagado del nodo central, a través del terminal comando: sudo shutdown now
4. Desenergizar de la toma eléctrica la fuente de alimentación.

ANEXO B



“Responsabilidad con pensamiento positivo”

UNIVERSIDAD TECNOLÓGICA ISRAEL

Prototipo de sistema de monitoreo y control de consumo de energía eléctrica para un domicilio aplicando el concepto de Internet de las cosas.

MANUAL DEL TÉCNICO

AUTOR:

Sánchez Oñate Ricardo Mauricio

QUITO

ECUADOR

2020

125

MANUAL TÉCNICO

El manual técnico brinda algunas características de funcionamiento del sistema de monitoreo de energía para domicilio basado en el funcionamiento del internet de las cosas.

Funcionamiento:

El funcionamiento del equipo se lo podría describir en las siguientes etapas:

- Adquisición de datos del consumo de energía, para esto se utiliza sensores de voltaje ZMPT101B y para la corriente el sensor no invasivo SCT-013-30 los cuales generan una señal de voltaje que ingresa al convertidor análogo digital del microcontrolador, el cual se encarga de procesar la información.
- Procesamiento de datos adquirido con los sensores se lo realiza a través de Arduino nano, ya que presta las suficientes características técnicas para el trabajo requerido.
- La comunicación inalámbrica se utiliza el transceptor NRF 24L01+, trabaja a 2,4 GHz
- Para la configuración del nodo central se utiliza una minicomputadora Raspberry Pi Zero W
- El almacenamiento se lo realizara en la plataforma ThinkSpeak, la cual brinda excelentes prestaciones en la implementación de prototipos en los que se menciona, almacenar 8200 datos diarios y aproximadamente 3 millones de datos al año.
- El desarrollo de la aplicación se lo realiza en MIT APP Inventor el cual es un poderoso lenguaje de programación en bloques, y entregará un archivo en formato APK listo para la instalación en un dispositivo con sistema operativo Android.

Elementos utilizados para la implementación de Hardware del Prototipo de Adquisición de Datos

Fuente Conmutada 7V / 650 mA

La fuente conmutada de la Figura. B.1. es un dispositivo electrónico que es encargada de entregar a su salida una señal de voltaje continuo a su salida, mediante un proceso de rectificación de una señal de naturaleza alterna con la ayuda de transistores en conmutación.



Figura. B.1. Fuente de poder

Sensor de Corriente SCT 013-30A

SCT-013 es un sensor de la Figura. B.2. trabaja como transformador, donde el cable eléctrico se lo considera el devanado primario 1 espira, el devanado secundario se encuentra en el interior del sensor el cual dependiendo de sus características técnicas tiene aproximadamente 2000 espiras. El número de espiras representa la relación entre la corriente que atraviesa un conductor y la corriente que entrega el sensor a la salida del devanado secundario. Algunos modelos de estos sensores pueden tener a la salida de su devanado una resistencia con esto ya no entregan una corriente sino un voltaje.

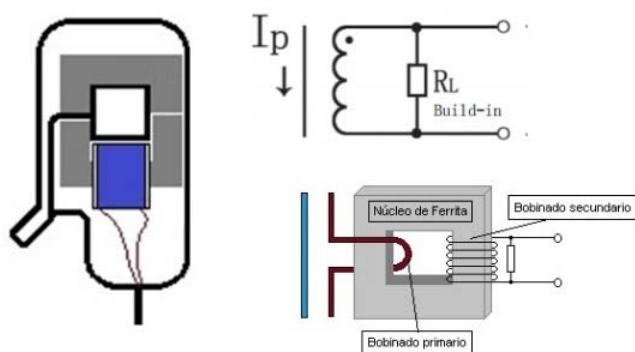


Figura. B.2 Sensor de corriente

Módulo de medición de Voltaje AC ZMPT 101B

El módulo transformador de voltaje alterno ZMPT101B de la Figura. B.3. permite medir voltaje alterno. El módulo está integrado por un transformador que cumple la función de aislamiento galvánico para mayor seguridad en el uso. El lado primario del transformador se conecta al voltaje alterno que se desea medir. En el lado secundario del transformador se encuentra un divisor de tensión y un circuito con amplificador operacional (OPAMP LM358) para adicionar un desplazamiento (offset) a la salida análoga.

Soporta voltajes de entrada de hasta 250VAC y entrega una onda senoidal de amplitud regulable por un potenciómetro en placa. La onda senoidal de salida está desplazada positivamente para que la onda no tenga voltajes negativos y así poder leer la onda completamente con el ADC.

- Voltaje de alimentación: 3.3V - 5VDC
- Voltaje alterno de entrada: 250VAC máx.
- Voltaje alterno de salida: Onda senoidal 5VAC máx.
- Señal de salida: analógica senoidal
- Dimensiones: 5 cm x 2 cm x 2.4 cm
- Propiedades del transformador:
- Corriente nominal de entrada y salida: 2mA

Ratio entrada-salida 1000:1000

- Diferencia de fase: $<30^\circ$ (a 50ohm)
- Rango lineal: 0-3mA (a 50ohm)
- Linealidad: 1%
- Precisión: 0.2%
- Aislamiento eléctrico entrada: hasta 3000V

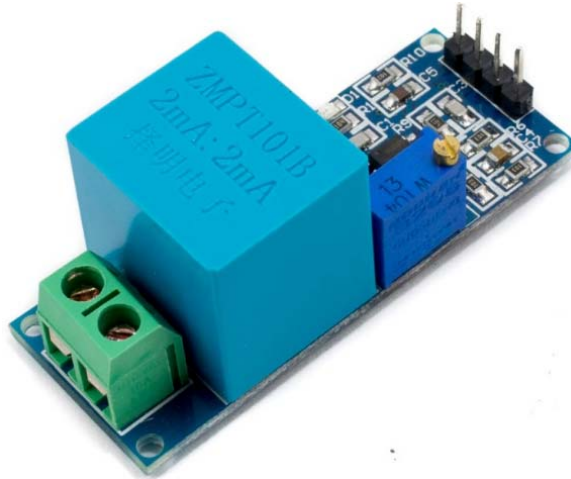


Figura. B.3. Sensor de voltaje

Arduino NANO

Arduino Nano de la ilustración 4 es una placa de desarrollo de tamaño compacto, completamente compatible con *protoboards*, basada en el microcontrolador Atmega328P. Tiene 14 pines de entrada/salida digital (de los cuales 6 pueden ser usando con PWM), 6 entradas analógicas, un cristal de 16Mhz, conexión Mini-USB, terminales para conexión ICSP y un botón de reseteo.

Posee las mismas capacidades que un Arduino UNO, tanto en potencia del microcontrolador como en conectividad, solo se ve recortado en su conector USB, conector Jack de alimentación y los pines cambia un formato de pines *header*. Esta placa posee las siguientes especificaciones:

Microcontrolador: ATmega328.
Voltaje de operación: 5V.
Voltaje de alimentación (Recomendado): 7-12V.
I/O Digitales: 14 (6 son PWM)
Entradas Analógicas: 8
Memoria Flash: 32KB.
EEPROM: 1KB.
Frecuencia de trabajo: 16MHz.
Dimensiones: 0.73" x 1.70"

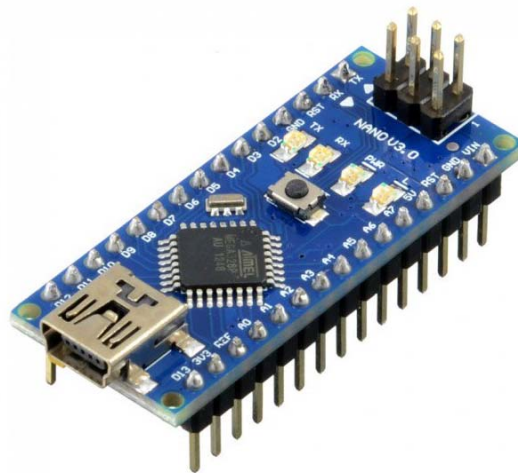


Figura. B.4.Arduino Nano

Módulo Transceiver NRF 24L01

El *Transceiver* NRF 24L01 de la Figura. B.5. opera en la banda de 2.4GHz (Industrial, Científica y Médica) y tiene características como su consumo ultra bajo (ULP). El chip integra un completo *transceiver* de 2.4Ghz, Sintetizador RF, y lógica como el mejorado *ShockBurst*™, que es un acelerador de protocolo por hardware para comunicación por SPI con microcontrolador.

- Fuente de alimentación: 1.9 ~ 3.6V
- voltaje de funcionamiento del puerto IO: 0 ~ 3.3v / 5v (Tolerante a 5V)
- Tasa de transmisión: + 7 dB
- Recepción de sensibilidad: \leq -90dB
- El alcance de transmisión: 250m en zona abierta

- Dimensiones: 15x29mm
- Funcionamiento en la banda ISM de 2,4 GHz, por lo que no necesita licencia y es libre en todo el mundo
- 3 velocidades de Datos seleccionables: 250Kbps, 1Mbps y 2Mbps
- Acelerador por hardware del protocolo SPI (ShockBurst™)
- Consumo energético ultra bajo, capaz de durar años utilizando una batería



Figura. B.5. NRF 24

Relé (Junto con Acondicionamiento)

Un relevador de la Figura. B.6. también conocido como relé de la ilustración 6 o relay, es un interruptor cuyo control corre por cuenta de un circuito eléctrico, a través de una bobina y un electroimán incide sobre diversos contactos para la apertura o el cierre de otros circuitos, que funcionan de manera independiente. Lo que hace la bobina es crear un campo magnético que lleva los contactos a establecer una conexión. El electroimán, por su parte, permite el cierre de los contactos. De esta forma, el relevador actúa como un interruptor que puede fomentar el paso de la corriente eléctrica o su interrupción. Cuenta con las siguientes características:

- Modelo SRD-5vdc-SL-C
- Contacto SPDT
- 5v
- 10A
- 5 pines
- 125VAC / 15A
- Dimensión de 19 mm x 15 mm x 15 mm



Figura. B.6. Relé

LCD I2C

El controlador de LCD I2C es un dispositivo que permite controlar una pantalla a través del bus I2C, usando únicamente dos cables. La pantalla tiene una retroiluminación de LED y puede mostrar dos filas con hasta 16 caracteres en cada fila. Puede ver los rectángulos para cada carácter en la pantalla y los píxeles que componen cada carácter. La pantalla es está diseñada para mostrar texto. El Módulo adaptador LCD a I2C está basado en el controlador I2C PCF8574, el cual es un Expansor de Entradas y Salidas digitales controlado por I2C. Por el diseño del PCB este módulo se usa especialmente para controlar un LCD Alfanumérico, así lo muestra la Figura. B.7.



Figura. B.7. LCD I2C

Esquema electrónico de módulos de adquisición de datos sin display en la Figura. B.8.

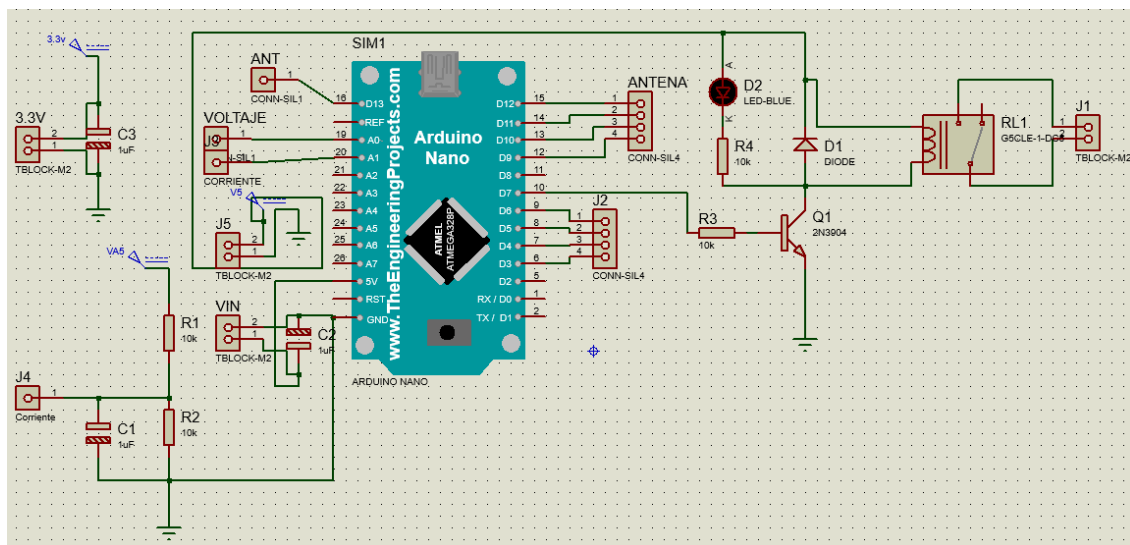


Figura. B.8. Esquema de módulos sin display

Diagrama electrónico del módulo central presentado en la Figura. B.9.

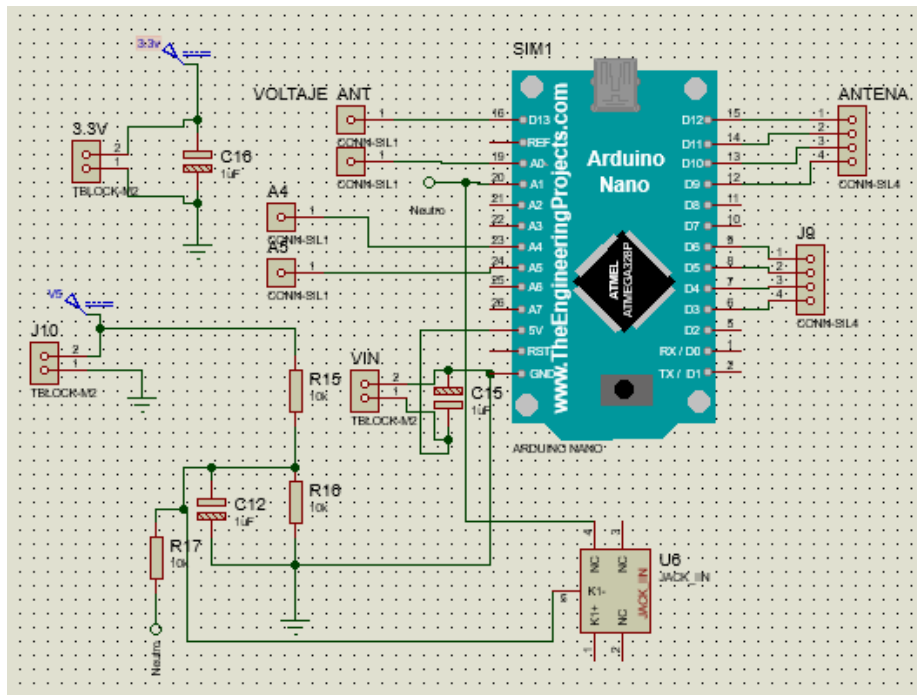


Figura. B.9. Diagrama del módulo central

La Disposición de pines módulo NRF 24L01+ se indica en la Figura. B.10.

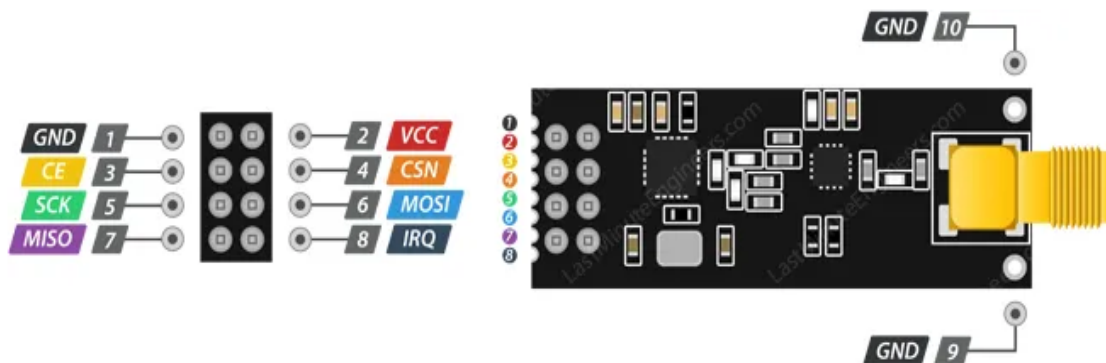


Figura. B.10. Pines del módulo NRF 24L01+

En la Figura. B.11. se muestra la distribución de pines GPIO en Raspberry pi

BOARD	GPIO		GPIO	BOARD
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)		DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)		(I ² C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Figura. B.11.Pines de Raspberry pi

Tabla. B.1. Resumen para la conexión de módulo de comunicación inalámbrica NRF 24101+

Numero de pines	Disposición de pines NRF 24L01+	Disposición de pines Arduino nano	Disposición de pines Raspberry Pi
1	GND	GND	6
2	VCC	3.3V	1
3	CE	D9	11
4	CSN	D10	24
5	SCK	D13	23
6	MOSI	D11	19
7	MISO	D12	21
8	IRQ	-	

Fuente: El autor

Para la configuración de cuenta en plataforma IoT, ThingSpeak , se crea una cuenta para la administración de datos. <https://thingspeak.com/>. Se observa en la Figura. B.12.

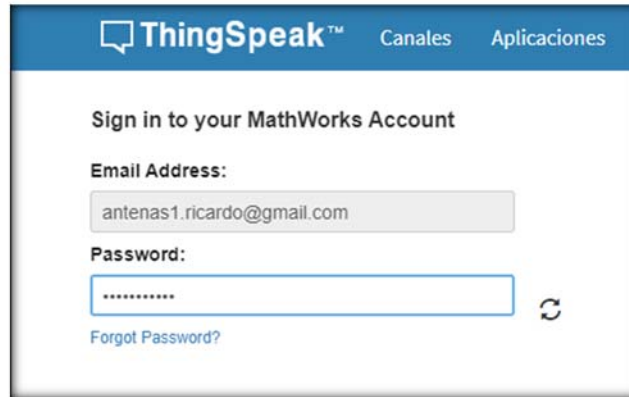


Figura. B.12. Ingreso a cuenta ThingSpeak

Luego se procede a configurar los diferentes canales y campos requeridos en la cuenta según la Figura. B.13.



Figura. B.13. Configuración de canales

Para acceder a la información del servidor IoT como la Figura B.14 se dispone de una API para escritura de datos: **MFWN7K1P1ARPQD4S** y una para lectura de datos: **OZ4AP9MLBCS3Y7KQ**

```

pi@raspberrypi: ~/Desktop/servidorRasPi/NRF24L01
GNU nano 3.2 nube.py
import sys
import urllib2
from time import sleep

myAPI='MFWN7K1P1ARPQD4S'

baseURL = 'https://api.thingspeak.com/update?api_key=%s' % myAPI

f=open('/home/pi/Desktop/NRF24L01/data.txt','r')
kws_nube=float(f.read())
f.close()
f2=open('/home/pi/Desktop/NRF24L01/data2.txt','r')
kws_nube2=float(f2.read())
f2.close()
f3=open('/home/pi/Desktop/NRF24L01/data3.txt','r')
kws_nube3=float(f3.read())
f3.close()
f4=open('/home/pi/Desktop/NRF24L01/data4.txt','r')
kws_nube4=float(f4.read())

```

Figura. B.14. Pantalla con protocolo API

En la aplicación manejo del entorno

En la pantalla principal de la Figura. B.15 se observa un número, el cual indica el estado de los dispositivos, al enviar la solicitud de estado de dispositivos se encuentra el resultado de los datos enviados desde el servidor a la aplicación e indica l como valor del estado.

Tabla. B.2. Resumen de valores sumatoria estado módulo 1, si A es 10 se realiza la suma caso contrario se realiza una multiplicación.

Estado	Valor	Estado	Valor	Resultado	ESTADO
encender		apagar			
A	10	A	0	10	Sin carga y relé apagado
A	0	B	20	0	Sin carga relé encendido
B	10	B	20	30	Con carga y relé encendido



Figura. B.15. Lectura de estados

Visualización de datos: Mediante los protocolos HTTP y con la clave API se accede a la lectura de información en la nube y muestra mediante la ayuda de gráficos el estado ON / OFF del switch, y si hay o no carga en el módulo, como se muestra en la Figura. B.16.



Figura. B.16. Lectura de información en la nube

ANEXO C

Características técnicas elementos electrónicos



nRF24L01+ Preliminary Product Specification

1.1 Features

Features of the nRF24L01+ include:

- Radio
 - Worldwide 2.4GHz ISM band operation
 - 126 RF channels
 - Common RX and TX interface
 - GFSK modulation
 - 250kbps, 1 and 2Mbps air data rate
 - 1MHz non-overlapping channel spacing at 1Mbps
 - 2MHz non-overlapping channel spacing at 2Mbps
- Transmitter
 - Programmable output power: 0, -6, -12 or -18dBm
 - 11.3mA at 0dBm output power
- Receiver
 - Fast AGC for improved dynamic range
 - Integrated channel filters
 - 13.5mA at 2Mbps
 - -82dBm sensitivity at 2Mbps
 - -85dBm sensitivity at 1Mbps
 - -94dBm sensitivity at 250kbps
- RF Synthesizer
 - Fully integrated synthesizer
 - No external loop filter, VCO varactor diode or resonator
 - Accepts low cost ± 60 ppm 16MHz crystal
- Enhanced ShockBurst™
 - 1 to 32 bytes dynamic payload length
 - Automatic packet handling
 - Auto packet transaction handling
 - 6 data pipe MultiCeiver™ for 1:6 star networks
- Power Management
 - Integrated voltage regulator
 - 1.9 to 3.6V supply range
 - Idle modes with fast start-up times for advanced power management
 - 26 μ A Standby-I mode, 900nA power down mode
 - Max 1.5ms start-up from power down mode
 - Max 130 μ s start-up from standby-I mode
- Host Interface
 - 4-pin hardware SPI
 - Max 10Mbps
 - 3 separate 32 bytes TX and RX FIFOs
 - 5V tolerant inputs
- Compact 20-pin 4x4mm QFN package

Características técnicas del módulo NRF 24L01 +

nRF24L01 Product Specification

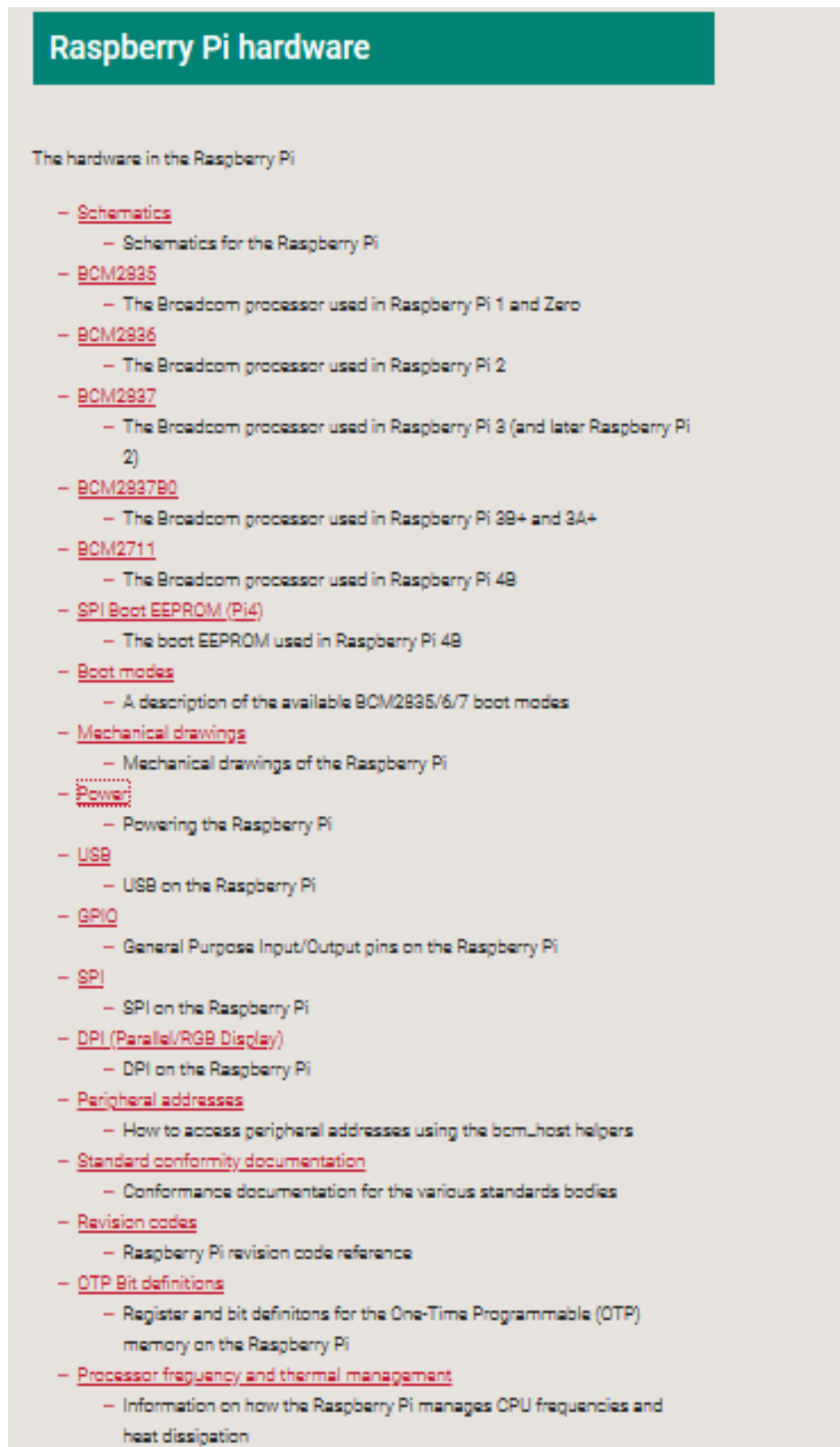


2.2 Pin functions

Pin	Name	Pin function	Description
1	CE	Digital Input	Chip Enable Activates RX or TX mode
2	CSN	Digital Input	SPI Chip Select
3	SCK	Digital Input	SPI Clock
4	MOSI	Digital Input	SPI Slave Data Input
5	MISO	Digital Output	SPI Slave Data Output, with tri-state option
6	IRQ	Digital Output	Maskable interrupt pin. Active low
7	VDD	Power	Power Supply (+1.9V - +3.6V DC)
8	VSS	Power	Ground (0V)
9	XC2	Analog Output	Crystal Pin 2
10	XC1	Analog Input	Crystal Pin 1
11	VDD_PA	Power Output	Power Supply Output(+1.8V) for the internal nRF24L01 Power Amplifier. Must be connected to ANT1 and ANT2 as shown in Figure 30 .
12	ANT1	RF	Antenna interface 1
13	ANT2	RF	Antenna interface 2
14	VSS	Power	Ground (0V)
15	VDD	Power	Power Supply (+1.9V - +3.6V DC)
16	IREF	Analog Input	Reference current. Connect a 22kΩ resistor to ground. See: Figure 30 .
17	VSS	Power	Ground (0V)
18	VDD	Power	Power Supply (+1.9V - +3.6V DC)
19	DVDD	Power Output	Internal digital supply output for de-coupling purposes. See: Figure 30 .
20	VSS	Power	Ground (0V)

Table 1. nRF24L01 pin function

Características técnicas Raspberry PI



Raspberry Pi hardware

The hardware in the Raspberry Pi

- [Schematics](#)
 - Schematics for the Raspberry Pi
- [BCM2835](#)
 - The Broadcom processor used in Raspberry Pi 1 and Zero
- [BCM2836](#)
 - The Broadcom processor used in Raspberry Pi 2
- [BCM2837](#)
 - The Broadcom processor used in Raspberry Pi 3 (and later Raspberry Pi 2)
- [BCM2837B0](#)
 - The Broadcom processor used in Raspberry Pi 3B+ and 3A+
- [BCM2711](#)
 - The Broadcom processor used in Raspberry Pi 4B
- [SPI Boot EEPROM \(Pi4\)](#)
 - The boot EEPROM used in Raspberry Pi 4B
- [Boot modes](#)
 - A description of the available BCM2835/6/7 boot modes
- [Mechanical drawings](#)
 - Mechanical drawings of the Raspberry Pi
- [Power](#)
 - Powering the Raspberry Pi
- [USB](#)
 - USB on the Raspberry Pi
- [GPIO](#)
 - General Purpose Input/Output pins on the Raspberry Pi
- [SPI](#)
 - SPI on the Raspberry Pi
- [DPI \(Parallel/RGB Display\)](#)
 - DPI on the Raspberry Pi
- [Peripheral addresses](#)
 - How to access peripheral addresses using the bcm_host helpers
- [Standard conformity documentation](#)
 - Conformance documentation for the various standards bodies
- [Revision codes](#)
 - Raspberry Pi revision code reference
- [OTP Bit definitions](#)
 - Register and bit definitions for the One-Time Programmable (OTP) memory on the Raspberry Pi
- [Processor frequency and thermal management](#)
 - Information on how the Raspberry Pi manages CPU frequencies and heat dissipation

ANEXO D

Transformador de voltaje AC-zmpt101B


Buscar

INICIO
ARDUINO
COMUNICACIÓN
SENSORES
ACTUADORES
BLOG
TUTORIALES

🏠
Sensores
Corriente y voltaje
Transformador de voltaje AC - ZMPT101B



Transformador de voltaje AC - ZMPT101B

Modelo SEN-VOLT-ZMPT101B

Util para medir voltaje de corriente alterna (AC) hasta 250VAC. Incluye un transformador para mayor seguridad en su uso.

Producto Agotado

tu@email.com

Avisame cuando esté disponible

 Imprimir

DESCRIPCIÓN

INFO

El módulo transformador de voltaje alterno ZMPT101B permite medir voltaje alterno como el que tenemos en nuestros hogares (en Perú: 220VAC-60Hz), este voltaje AC no puede ser medido directamente por el ADC de nuestro Arduino pues escapa al rango de entrada (0V a 5V). El módulo ZMPT101B soluciona el problema reduciendo el voltaje AC de entrada a un voltaje menor que pueda ser leído por el Arduino o cualquier otro microcontrolador.

El modulo está integrado por un transformador que cumple la función de aislamiento galvánico para mayor seguridad en el uso. El lado primario del transformador se conecta al voltaje alterno que deseamos medir, por ejemplo: la red eléctrica de nuestro hogar de 220VAC. En el lado secundario del transformador se encuentra un divisor de tensión y un circuito con amplificador operacional (OPAMP LM358) para adicionar un desplazamiento (offset) a la salida analógica.

Soporta voltajes de entrada de hasta 250VAC y entrega una onda senoidal de amplitud regulable por un potenciómetro en placa. La onda senoidal de salida está desplazada positivamente para que la onda no tenga voltajes negativos y así poder leer la onda completamente con el ADC. El desplazamiento depende del voltaje con el que alimentemos el módulo: si el voltaje de alimentación es de 5V el desplazamiento será de 2.5V y si alimentamos el módulo con 3.3V el desplazamiento será de 1.65V. El circuito de acondicionamiento de señal permite que el voltaje de salida del módulo pueda ser leído por cualquier microcontrolador con entrada analógica (ADC), de esta forma es posible leer el voltaje instantáneo y realizar cálculos de energía, como: voltaje pico a pico (Vpp) y voltaje eficaz (Vrms).

Ideal para aplicaciones de monitoreo de energía eléctrica, muy común en aplicaciones de domótica e IoT (Internet of Things) como: Medidores de energía conectados a internet por Wifi/Bluetooth/GSM/LoRa. Debido a la naturaleza de los transformadores solo puede medir voltaje AC.

NOTA: EL VOLTAJE DE 220V AC ES MUY PELIGROSO! MANEJADO INCORRECTAMENTE PUEDE CAUSAR LA MUERTE! POR ESO DEBEMOS SER MUY CUIDADOSOS AL REALIZAR LAS CONEXIONES. NAYLAMP MECHATRONICS NO SE RESPONSABILIZA POR LOS DAÑOS CAUSADOS POR EL MAL USO DE ESTE MODULO.

APLICACIONES

- Adecuado para la medición de voltaje AC de baja tensión como 220VAC
- Monitoreo remoto y protección de equipos AC
- Retroalimentación para sistemas de control de voltaje AC

ESPECIFICACIONES TÉCNICAS

- Voltaje de alimentación: 3.3V - 5VDC
- Voltaje alterno de entrada: 250VAC máx.
- Voltaje alterno de salida: Onda senoidal 5VAC máx.
- Señal de salida: analógica senoidal
- Dimensiones: 5 cm x 2 cm x 2.4 cm
- Propiedades del transformador:
 - Corriente nominal de entrada y salida: 2mA
 - Ratio entrada-salida 1000:1000
 - Diferencia de fase: <math><30^\circ</math> (a 50ohm)
 - Rango lineal: 0-3mA (a 50ohm)
 - Linealidad: 1%
 - Precisión: 0.2%
 - Aislamiento eléctrico entrada: hasta 3000V

LINKS

- [Esquemático - Mod. ZMPT101B](#)
- [Naylamp Sketchs - Arduino y ZMPT101B](#)
- [Datasheet - Transformador ZMPT101B](#)
- [Guía de uso del fabricante](#)
- [Paper - Calibración para medición precisa de cargas AC con Arduino](#)
- [Concepto - Corriente Alterna AC](#)
- [Concepto - Aislamiento galvanico](#)
- [Proyecto - OpenEnergyMonitor](#)

ANEXO E

Características del sensor de corriente SCT 0013

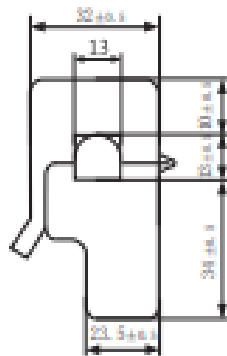
SPECIFICATION

Customer Title : XiDi Technology Product Name: Split-core current
 Manufacture Model : SCT-013-000 transformer

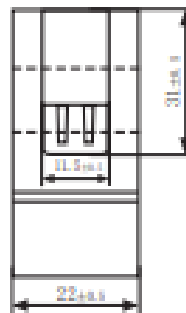
Characteristics: open size:13mm×13mm
 1m leading wire
 Core material:Ferrite
 Fire resistance property:in accordance with
 UL 94-V0
 Dielectric strength: 1000V AC/1min 5mA
 (between shell and output)



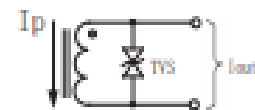
Outline size diagram: (in mm)



Front View



Side View



TVS: Transient-voltage Suppressor

Current output type

Schematic Diagram

Typical table of technical parameters:

input current	output voltage	non-linearity	build-in sampling resistance (R _s)
0-100A	0-50mV	±3%	Ω
turn ratio	resistance grade	work temperature	dielectric strength(between shell and output)
100A:0.05A	Grade B	-25℃~+70℃	1000V AC/1min 5mA

Customer Sign:

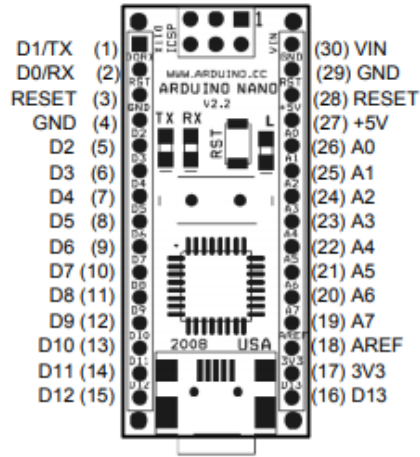
Beijing YaoHuadechang Electronic Co.,Ltd
 Phone: 0355-7929499-803
 Cell: 13693334514
 Contact Name: Engineer Chen

Approve Sign:Chenjianping

2011-7-26

Características técnicas Arduino Nano

Arduino Nano Pin Layout



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

Especificaciones técnicas de Arduino Nano

Microcontrolador	ATmega328P - Microcontrolador de la familia AVR de 8 bits
Tensión de funcionamiento	5V
Voltaje de entrada recomendado para pin Vin	7-12V
Pines de entrada analógica	6 (A0 - A5)
Pines de E / S digitales	14 (de los cuales 6 proporcionan salida PWM)
Corriente CC en pines de E / S	40 mA
Corriente DC en Pin de 3.3V	50 mA
Memoria flash	32 KB (2 KB se utilizan para el cargador de arranque)
SRAM	2 KB

ANEXO F

Lenguaje de programación

Código del programa Arduino IDE:

Módulos sin display

//SendReceive.ino

```
#include<SPI.h>
#include<RF24.h>
#include "printf.h"
#include "EmonLib.h"          // Include Emon Library
EnergyMonitor emon1;        // Create an instance
float KWs=0;
int c_rele=0;
// CE, CSN pins
RF24 radio (9, 10);

void setup(void){
  pinMode(7,OUTPUT);
  emon1.voltage(A0, 164.5, 0); // Voltage: input pin, calibration, phase_shift
  emon1.current(A1, 29);      // Current: input pin, calibration.
  printf_begin();
  while(!Serial);
  Serial.begin(9600);

  radio.begin();
  radio.setPALevel(RF24_PA_MAX);
  radio.setChannel(0x76);
  radio.openWritingPipe(0xF0F0F0F0E3LL);
  const uint64_t pipe = (0xE8E8F0F0E2LL);
  radio.openReadingPipe(1, pipe);

  radio.enableDynamicPayloads();
  radio.powerUp();
  radio.printDetails();
}
void loop(void){
  float starttime=millis()/1000;
  emon1.calcVI(20,2000);      // Calculate all. No.of half wavelengths (crossings), time-
  out
  //emon1.serialprint();      // Print out all variables (realpower, apparent power, Vrms,
  Irms, power factor)
  float realPower = emon1.realPower; //extract Real Power into variable
  float endtime=millis()/1000;
  KWs= realPower/1000*(endtime-starttime)+KWs;
  radio.startListening();
```

```

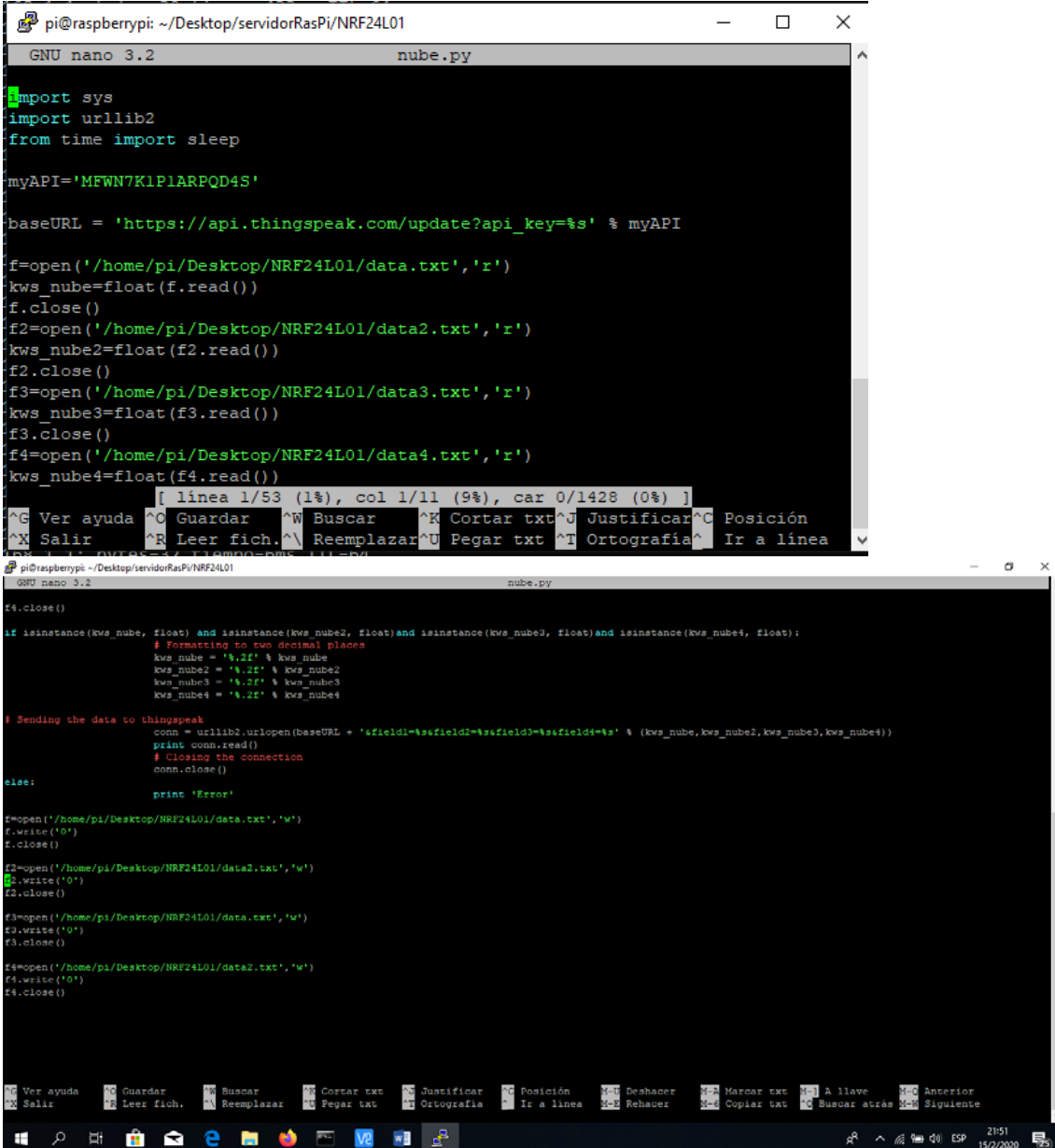
char receivedMessage[32] = {0};
if(radio.available()){
  radio.read(receivedMessage, sizeof(receivedMessage));
  Serial.println(receivedMessage);
  Serial.println("Turning off the radio.");
  radio.stopListening();
  String stringMessage(receivedMessage);
  if(stringMessage == "GETSTRING"){
    Serial.println("Looks like they want a string!");
    Serial.print("Consumo (Kws)=");
    Serial.println(KWs);
    const char text[8];
    dtostrf(KWs, 4, 2, text);
    radio.write(text, sizeof(text));
    Serial.println("We sent our message.");
    //KWs=0;
  }else if(stringMessage == "Encerar"){
    KWs=0;
    Serial.println("Encerado");
  }else if(stringMessage== "Prender"){
    Serial.println("Ok prender");
    digitalWrite(7,HIGH);
    c_rele=1;
  }else if(stringMessage== "Apagar"){
    Serial.println("Ok apagar");
    digitalWrite(7,LOW);
    c_rele=0;
  }else if(stringMessage== "Potencia"){
    Serial.print("P (W)=");
    Serial.println(realPower);
    const char text1[3];
    String t;
    if (c_rele==0){
      t="AA";
    }else if (c_rele==1 && abs(realPower)<5){
      t="AB";
    }else{
      t="BB";
    }
  }
  t.toCharArray(text1, 3);
  //dtostrf(realPower, 4, 2, text1);
  radio.write(text1, sizeof(text1));
  Serial.println("We sent our message.");
}
}
delay(100);

```

}

Código del programa Raspberry Python

La función del código es publicar información a la nube y envía información desde la nube a la Raspberry. (Nube)



```

import sys
import urllib2
from time import sleep

myAPI='MFWN7K1P1ARPQD4S'

baseURL = 'https://api.thingspeak.com/update?api_key=%s' % myAPI

f=open('/home/pi/Desktop/NRF24L01/data.txt','r')
kws_nube=float(f.read())
f.close()
f2=open('/home/pi/Desktop/NRF24L01/data2.txt','r')
kws_nube2=float(f2.read())
f2.close()
f3=open('/home/pi/Desktop/NRF24L01/data3.txt','r')
kws_nube3=float(f3.read())
f3.close()
f4=open('/home/pi/Desktop/NRF24L01/data4.txt','r')
kws_nube4=float(f4.read())

if isinstance(kws_nube, float) and isinstance(kws_nube2, float) and isinstance(kws_nube3, float) and isinstance(kws_nube4, float):
    # Formating to two decimal places
    kws_nube = '%.2f' % kws_nube
    kws_nube2 = '%.2f' % kws_nube2
    kws_nube3 = '%.2f' % kws_nube3
    kws_nube4 = '%.2f' % kws_nube4

# Sending the data to thingspeak
conn = urllib2.urlopen(baseURL + '%sfield1=%sfield2=%sfield3=%sfield4=%s' % (kws_nube,kws_nube2,kws_nube3,kws_nube4))
print conn.read()
# Closing the connection
conn.close()
else:
    print 'Error'

f=open('/home/pi/Desktop/NRF24L01/data.txt','w')
f.write('0')
f.close()

f2=open('/home/pi/Desktop/NRF24L01/data2.txt','w')
f2.write('0')
f2.close()

f3=open('/home/pi/Desktop/NRF24L01/data3.txt','w')
f3.write('0')
f3.close()

f4=open('/home/pi/Desktop/NRF24L01/data4.txt','w')
f4.write('0')
f4.close()

```

```
import sys
```

```
import urllib2
```

```
from time import sleep
```

```
myAPI='MFWN7K1P1ARPQD4S'
```

```
baseURL = 'https://api.thingspeak.com/update?api_key=%s' % myAPI
```

```

f=open('/home/pi/Desktop/NRF24L01/data.txt','r')
kws_nube=float(f.read())
f.close()
f2=open('/home/pi/Desktop/NRF24L01/data2.txt','r')
kws_nube2=float(f2.read())
f2.close()
f3=open('/home/pi/Desktop/NRF24L01/data3.txt','r')
kws_nube3=float(f3.read())
f3.close()
f4=open('/home/pi/Desktop/NRF24L01/data4.txt','r')
kws_nube4=float(f4.read())
f4.close()

if isinstance(kws_nube, float) and isinstance(kws_nube2, float) and isinstance(kws_nube3,
float) and isinstance(kws_nube4, float):
    # Formatting to two decimal places
    kws_nube = '%.2f' % kws_nube
    kws_nube2 = '%.2f' % kws_nube2
    kws_nube3 = '%.2f' % kws_nube3
    kws_nube4 = '%.2f' % kws_nube4
# Sending the data to thingspeak
    conn = urllib2.urlopen(baseUrl +
'&field1=%s&field2=%s&field3=%s&field4=%s' %
(kws_nube,kws_nube2,kws_nube3,kws_nube4))
    print conn.read()
    # Closing the connection
    conn.close()
else:
    print 'Error'
f=open('/home/pi/Desktop/NRF24L01/data.txt','w')
f.write('0')
f.close()
f2=open('/home/pi/Desktop/NRF24L01/data2.txt','w')
f2.write('0')
f2.close()

f3=open('/home/pi/Desktop/NRF24L01/data.txt','w')
f3.write('0')
f3.close()
f4=open('/home/pi/Desktop/NRF24L01/data2.txt','w')
f4.write('0')
f4.close()

```

```

pi@raspberrypi:~/Desktop/servidorRasPi/NRF24L01
GNU nano 3.2 receivevube.py

import xml.etree.ElementTree as ET
import sys
import urllib2
from time import sleep
import time
import RPi.GPIO as GPIO
from lib_nrf24 import NRF24
import spidev

GPIO.setmode(GPIO.BCM)

pipes = [[0xE8, 0xE8, 0xF0, 0xF0, 0xE2], [0xF0, 0xF0, 0xF0, 0xF0, 0xE3], [0xE8, 0xE8, 0xF0, 0xF0, 0xE3], [0xE8, 0xE8, 0xF0, 0xF0, 0xE4], [0xE8, 0xE8, 0xF0, 0xF0, 0xE5]]
#pipes = [[0xE8, 0xE8, 0xF0, 0xF0, 0xE2], [0xF0, 0xF0, 0xF0, 0xF0, 0xE2]]

radio = NRF24(GPIO, spidev.SpiDev())
radio.begin(0, 17)
radio.setPayloadSize(32)
radio.setChannel(0x76)
radio.setDataRate(NRF24.BR_1MBPS)
radio.setPALevel(NRF24.PA_MAX)
radio.setAutoAck(True)
radio.enableDynamicPayloads()
radio.enableAckPayload()
radio.openReadingPipe(1, pipes[1])
radio.printDetails()

myAPI='OZ4AP9MLBCS3Y7KQ'
message='Apagar'
baseURL = 'https://api.thingspeak.com/channels/951270/fields/5.xml?api_key=%s' % myAPI
while True:
    sleep(1)
    conn = urllib2.urlopen(baseURL + '&results=1')
    xml_data=conn.read()
    conn.close()
    doc=ET.fromstring(xml_data)
    rele=doc.find('Feed')
    rele2=rele.find('Feed')
    rele3=rele2.find('fields').text
    print rele3

```

```
import xml.etree.ElementTree as ET
```

```
import sys
```

```
import urllib2
```

```
from time import sleep
```

```
import time
```

```
import RPi.GPIO as GPIO
```

```
from lib_nrf24 import NRF24
```

```
import spidev
```

```
GPIO.setmode(GPIO.BCM)
```

```

pipes = [[0xE8, 0xE8, 0xF0, 0xF0, 0xE2], [0xF0, 0xF0, 0xF0, 0xF0, 0xE3], [0xE8, 0xE8,
0xF0, 0xF0, 0xE3], [0xE8, 0xE8, 0xF0, 0xF0, 0xE4], [0xE8, 0xE8, 0xF0, 0xF0, 0xE5]]
#pipes = [[0xE8, 0xE8, 0xF0, 0xF0, 0xE2], [0xF0, 0xF0, 0xF0, 0xF0, 0xE2]]

```

```
radio = NRF24(GPIO, spidev.SpiDev())
```

```
radio.begin(0, 17)
```

```
radio.setPayloadSize(32)
```

```
radio.setChannel(0x76)
```

```
radio.setDataRate(NRF24.BR_1MBPS)
```

```
radio.setPALevel(NRF24.PA_MAX)
```

```
radio.setAutoAck(True)
```

```
radio.enableDynamicPayloads()
```

```
radio.enableAckPayload()
```

```
radio.openReadingPipe(1, pipes[1])
```

```
radio.printDetails()
```

```
myAPI='OZ4AP9MLBCS3Y7KQ'
```

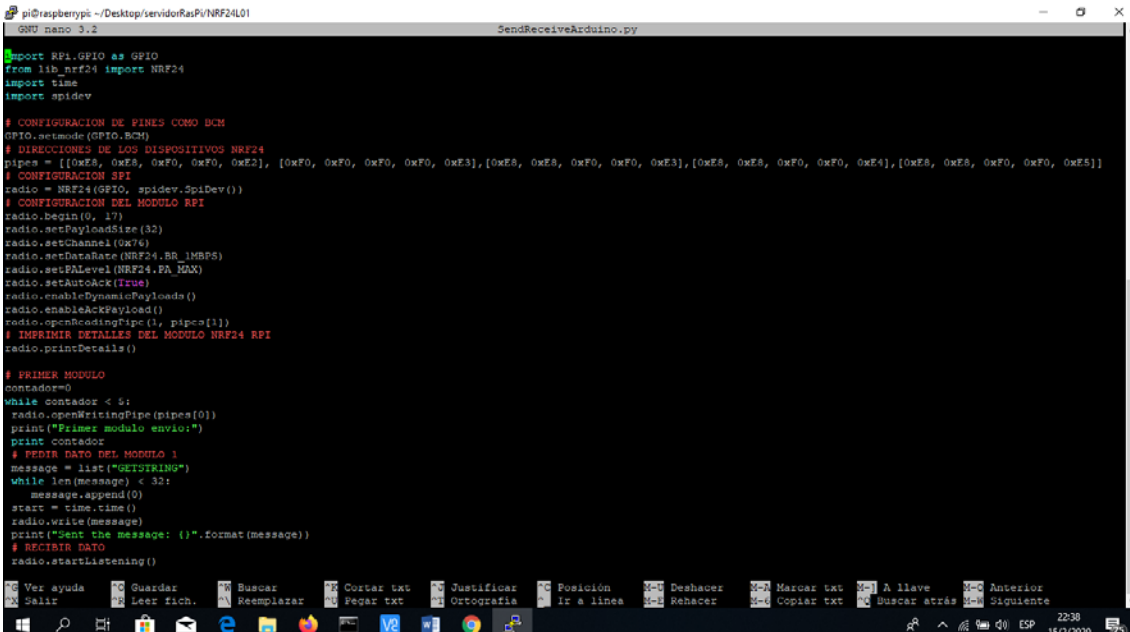
```
message='Apagar'
```

```

baseURL = 'https://api.thingspeak.com/channels/951270/fields/5.xml?api_key=%s' %
myAPI
while True:
    sleep(1)
    conn = urllib2.urlopen(baseURL + '&results=1')
    xml_data=conn.read()
    conn.close()
    doc=ET.fromstring(xml_data)
    rele=doc.find('feeds')
    rele2=rele.find('feed')
    rele3=rele2.find('field5').text
    print rele3

```

ENVIAR Y RECIBIR DATOS DESDE ARDUINO, HACIA LA RASPBERRY



```

pi@raspberrypi: ~/Desktop/servidor/RasPi/NRF24L01
GNU nano 3.2 SendReceiveArduino.py
import RPi.GPIO as GPIO
from lib_nrf24 import NRF24
import time
import spidev

# CONFIGURACION DE PINES COMO BCM
GPIO.setmode(GPIO.BCM)
# DIRECCIONES DE LOS DISPOSITIVOS NRF24
pipes = [[0xE8, 0xE8, 0xF0, 0xF0, 0xE2], [0xF0, 0xF0, 0xF0, 0xF0, 0xE3], [0xE8, 0xE8, 0xF0, 0xF0, 0xE3], [0xE8, 0xE8, 0xF0, 0xF0, 0xE4], [0xE8, 0xE8, 0xF0, 0xF0, 0xE5]]
# CONFIGURACION SPI
radio = NRF24(GPIO, spidev.SpiDev())
# CONFIGURACION DEL MODULO RPI
radio.begin(0, 17)
radio.setPayloadSize(32)
radio.setChannel(0x76)
radio.setDataRate(NRF24.BR_1MBPS)
radio.setPALevel(NRF24.PA_MAX)
radio.setAutoAck(True)
radio.enableDynamicPayloads()
radio.enableAckPayload()
radio.openReadingPipe(1, pipes[0])
# IMPRIMIR DETALLES DEL MODULO NRF24 RPI
radio.printDetails()

# PRIMER MODULO
contador=0
while contador < 5:
    radio.openWritingPipe(pipes[0])
    print("Primer modulo envio:")
    print contador
    # PEDIR DATO DEL MODULO 1
    message = list("GETSTRING")
    while len(message) < 32:
        message.append(0)
    start = time.time()
    radio.write(message)
    print("Sent the message: {}".format(message))
    # RECIBIR DATO
    radio.startListening()

```

```

import RPi.GPIO as GPIO
from lib_nrf24 import NRF24
import time
import spidev
# CONFIGURACION DE PINES COMO BCM
GPIO.setmode(GPIO.BCM)
# DIRECCIONES DE LOS DISPOSITIVOS NRF24
pipes = [[0xE8, 0xE8, 0xF0, 0xF0, 0xE2], [0xF0, 0xF0, 0xF0, 0xF0, 0xE3], [0xE8, 0xE8,
0xF0, 0xF0, 0xE3], [0xE8, 0xE8, 0xF0, 0xF0, 0xE4], [0xE8, 0xE8, 0xF0, 0xF0, 0xE5]]
# CONFIGURACION SPI
radio = NRF24(GPIO, spidev.SpiDev())
# CONFIGURACION DEL MODULO RPI
radio.begin(0, 17)
radio.setPayloadSize(32)
radio.setChannel(0x76)
radio.setDataRate(NRF24.BR_1MBPS)

```

```

radio.setPALevel(NRF24.PA_MAX)
radio.setAutoAck(True)
radio.enableDynamicPayloads()
radio.enableAckPayload()
radio.openReadingPipe(1, pipes[1])
# IMPRIMIR DETALLES DEL MODULO NRF24 RPI
radio.printDetails()

# PRIMER MODULO
contador=0
while contador < 5:
    radio.openWritingPipe(pipes[0])
    print("Primer modulo envio:")
    print contador
    # PEDIR DATO DE ENERGIA MODULO 1
    message = list("GETSTRING")
    while len(message) < 32:
        message.append(0)
    start = time.time()
    radio.write(message)
    print("Sent the message: {}".format(message))
    # RECIBIR DATO
    radio.startListening()
    while not radio.available(0):
        time.sleep(1 / 100)
        if time.time() - start > 3:
            print("Timed out.")
            contador=contador+1
            break
    receivedMessage = []
    radio.read(receivedMessage, radio.getDynamicPayloadSize())
    print("Received: {}".format(receivedMessage))
    print("Translating the receivedMessage into unicode characters")
    string = ""
    for n in receivedMessage:
        # DECODIFICA EL MENSAJE
        if (n >= 48 and n <= 57 or n == 46):
            contador=10
            string += chr(n)
    print("Out received message decodes to: {}".format(string))
    radio.stopListening()
    # CONFIRMAR DATO Y MANDAR A ENCERAR AL MODULO 1

```

```

time.sleep(1 / 100)
if time.time() - start > 3:
    print("Timed out.")
    contador=contador+1
    break
receivedMessage = []
radio.read(receivedMessage, radio.getDynamicPayloadSize())
print("Received: {}".format(receivedMessage))
print("Translating the receivedMessage into unicode characters")
string = ""
for n in receivedMessage:
    # DECODIFICA EL MENSAJE
    if (n >= 49 and n <= 57 or n == 46):
        contador=0
        string += chr(n)
print("Out received message decodes to: {}".format(string))
radio.stopListening()
# CONFIRMAR DATO Y MANDAR A ENCIERAR AL MODULO 1
if len(string)>0:
    message = list("Encerrar")
    while len(message) < 32:
        message.append(0)
    time.sleep(3)
    radio.write(message)
    print("Sent the message: {}".format(message))

#SEGUNDO MODULO
contador=0
while contador < 5:
    radio.openWritingPipe(pipes[2])
    print("Segundo modulo envio:")
    print contador
    # SOLICITAR DATO DE ENERGIA MODULO 2
    message = list("GETSTRING")
    while len(message) < 32:
        message.append(0)
    start = time.time()
    radio.write(message)
    print("Sent the message: {}".format(message))

```

if len(string)>0:

message = list("Encerrar")

while len(message) < 32:

message.append(0)

time.sleep(3)

radio.write(message)

print("Sent the message: {}".format(message))

#SEGUNDO MODULO

contador=0

while contador < 5:

radio.openWritingPipe(pipes[2])

print("Segundo modulo envio:")

print contador

SOLICITAR DATO DE ENERGIA MODULO 2

message = list("GETSTRING")

while len(message) < 32:

message.append(0)

start = time.time()

radio.write(message)

print("Sent the message: {}".format(message))

ESCUCHAR DATO ENVIADO POR MODULO 2

radio.startListening()

while not radio.available(0):

time.sleep(1 / 100)

if time.time() - start > 3:

print("Timed out.")

contador=contador+1

break

receivedMessage = []

```

radio.read(receivedMessage, radio.getDynamicPayloadSize())
print("Received: {}".format(receivedMessage))
print("Translating the receivedMessage into unicode characters")
string2 = ""
for n in receivedMessage:
    # DECODIFICA EL MENSAJE
if (n >= 48 and n <= 57 or n == 46):
    contador=10
    string2 += chr(n)
print("Out received message decodes to: {}".format(string2))
radio.stopListening()
# CONFIRMAR RECEPCION DE DATO Y ENCERAR MODULO 2
if len(string2)>0:
    message = list("Encerar")
    while len(message) < 32:
        message.append(0)
        time.sleep(3)
    radio.write(message)
    print("Sent the message: {}".format(message))

# TERCER MODULO
contador=0
while contador < 5:
    radio.openWritingPipe(pipes[3])
    print("Tercer modulo envio:")
    print contador
    # PEDIR DATO DE ENERGIA CON MODULO 3
    message = list("GETSTRING")
    while len(message) < 32:
        message.append(0)
    start = time.time()
    radio.write(message)
    print("Sent the message: {}".format(message))
    # ESUCHAR Y RECEPCION DEL DATO DE MODULO 3
    radio.startListening()
    while not radio.available(0):
        time.sleep(1 / 100)
        if time.time() - start > 3:
            print("Timed out.")
            contador=contador+1
            break
    receivedMessage = []
    radio.read(receivedMessage, radio.getDynamicPayloadSize())
    print("Received: {}".format(receivedMessage))
    print("Translating the receivedMessage into unicode characters")
    string3 = ""

```

```

for n in receivedMessage:
    # DECODIFA MENSAJE
    if (n >= 48 and n <= 57 or n == 46):
        contador=10
        string3 += chr(n)
print("Out received message decodes to: {}".format(string3))
radio.stopListening()
# CONFIRMAR Y ENCERAR DATO DEL MODULO 3
if len(string3)>0:
    message = list("Encerar")
    while len(message) < 32:
        message.append(0)
        time.sleep(3)
    radio.write(message)
    print("Sent the message: {}".format(message))
# CUARTO MODULO
contador=0
while contador < 5:
    radio.openWritingPipe(pipes[4])
    print("Cuarto modulo envio:")
    print contador
# PEDIR DATO DEL MODULO 4
message = list("GETSTRING")
while len(message) < 32:
    message.append(0)
start = time.time()
radio.write(message)
print("Sent the message: {}".format(message))
# ESCUCHAR Y RECEPCION DEL DATO DE ENERGIA DEL MODULO 4
radio.startListening()
while not radio.available(0):
    time.sleep(1 / 100)
    if time.time() - start > 3:
        print("Timed out.")
        contador=contador+1
        break
receivedMessage = []
radio.read(receivedMessage, radio.getDynamicPayloadSize())
print("Received: {}".format(receivedMessage))
print("Translating the receivedMessage into unicode characters")
string4 = ""
for n in receivedMessage:
    # DECODIFICA MENSAJE
    if (n >= 48 and n <= 57 or n == 46):
        contador=10
        string4 += chr(n)

```

```
print("Out received message decodes to: {}".format(string4))
radio.stopListening()
# CONFIRMAR Y ENCERAR EL MODULO 4
if len(string4)>0:
    message = list("Encerar")
    while len(message) < 32:
        message.append(0)
time.sleep(3)
radio.write(message)
print("Sent the message: {}".format(message))

if len(string)>0:
    print("Envio energia modulo 1")
    f=open('/home/pi/Desktop/NRF24L01/data.txt','r')
    Kws_ant=float(f.read())
    f.close()
    Kws=float(string)
    f=open('/home/pi/Desktop/NRF24L01/data.txt','w')
    f.write(str(Kws_ant+Kws))
    f.close()

if len(string2)>0:
    print("Envio energia modulo 2")
    f2=open('/home/pi/Desktop/NRF24L01/data2.txt','r')
    Kws_ant=float(f2.read())
    f2.close()
    Kws=float(string2)
    f2=open('/home/pi/Desktop/NRF24L01/data2.txt','w')
    f2.write(str(Kws_ant+Kws))
    f2.close()

if len(string3)>0:
    print("Envio energia modulo 3")
    f3=open('/home/pi/Desktop/NRF24L01/data3.txt','r')
    Kws_ant=float(f3.read())
    f3.close()
    Kws=float(string3)
    f3=open('/home/pi/Desktop/NRF24L01/data3.txt','w')
    f3.write(str(Kws_ant+Kws))
    f3.close()
if len(string4)>0:
    print("Envio energia modulo 4")
    f4=open('/home/pi/Desktop/NRF24L01/data4.txt','r')
    Kws_ant=float(f4.read())
```