

Universidad Tecnológica Nacional

Proyecto Final

Telegestión de Luminarias

Autores:

- Carnevale Yonzo, Saverio
- Grassi Bonci, Martin

Director:

*Proyecto final presentado para cumplimentar los requisitos académicos
para acceder al título de Ingeniero Electrónico.*

en la

Facultad Regional Paraná

Fecha: Noviembre de 2018

Declaración de autoría:

Nosotros declaramos que el Proyecto Final “Telegestión de luminarias” y el trabajo realizado son propios. Declaramos:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero Electrónico, en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

-
-
-

Fecha:

Agradecimientos:

Agradecemos a la institución por la excelente formación y capacitación que nos brindó. Y a la familia de ambos, que nos apoyaron siempre para poder lograr el objetivo de presentar este proyecto y alcanzar el título de ingeniero.

Carnevale Yonzo, Saverio

Grassi Bonci, Martin

Universidad Tecnológica Nacional

Abstract

Facultad Regional Paraná

Ingeniero en Electrónica

Telegestión de Luminarias

Carnevale Yonzo, Saverio

Grassi Bonci, Martin

Abstract:

The project consists on LED luminaires wireless management for public lighting, which includes the variables measurement, the on / off control and programming, and the luminosity intensity variation in each luminaire, all accessible through a graphic interface and with the possibility of remotely accesitivity through internet connection.

Arduino boards and Xbee modules were used for luminaries control and wireless communication. A Raspberry Pi 3 board was used to host the graphical interface made with Qt software, as well as to achieve the connection to the internet and to enable the remote access.

It was obtained, a management system, capable of wirelessly communicating an approximate operating capacity of 100 luminaires per coordinator, with the possibility of remotely accessing through internet, where the measured data are visualized and the variables are controlled.

Keywords:

Management, Control, Measurement, Zigbee Protocol, LED luminaires, Raspberry, Arduino.

Resumen:

El proyecto consiste en la gestión inalámbrica de luminarias LED para alumbrado público, que incluye la medición de las variables, el control y programación del encendido/apagado, y la variación de la intensidad de luminosidad en cada luminaria, todo accesible a través de una interfaz gráfica y con la posibilidad de acceder a la misma de forma remota a través de internet.

Se utilizaron placas Arduino y módulos Xbee para el control y comunicación inalámbrica de las luminarias. Se utilizó una placa Raspberry Pi 3 para albergar la interfaz gráfica realizada con el software Qt como también para lograr la conexión a internet y posibilitar el acceso remoto.

Se obtuvo un sistema de telegestión capaz de comunicar inalámbricamente una capacidad operativa aproximada de 100 luminarias por coordinador, con posibilidad de acceder al mismo remotamente mediante internet, con el cual se visualizan los datos medidos y se controlan las variables.

Palabras Clave:

Gestión, Control, Medición, Protocolo Zigbee, Luminarias LED, Raspberry, Arduino.

Reconocimientos:

Un agradecimiento especial para el ingeniero Carlos Vandevoorde y sus colaboradores por recibirnos en alumbrado público y aportarnos información respecto a la iluminación LED, destacando su generosidad en prestarnos una luminaria LED, la cual nos permitió desarrollar el proyecto.

Por otro lado, un agradecimiento al ingeniero Fabio Vincitorio, que fue para nosotros una fuente de consulta, nos facilitó la utilización de su laboratorio para realizar mediciones y también nos prestó una luminaria LED para realizar pruebas.

Índice:

Capítulo 1: Introducción	1
Descripción del proyecto:	1
Estudio de mercado:.....	3
Capítulo 2: Desarrollo	5
Diagrama de Bloques.	5
Lógica de Funcionamiento	5
Estudio de los distintos protocolos para la comunicación entre luminarias	6
Elementos a utilizar en esta etapa.....	9
Módulo Xbee	10
(Digi, 2018) Capas de pila ZigBee	10
Tipos de dispositivos	12
Topologías de red.....	12
Red de malla	12
Series de XBee.....	13
Cómo se comunican los dispositivos XBee	14
Comunicación inalámbrica	14
Modos de funcionamiento	15
Modo de funcionamiento transparente	15
Modo de funcionamiento API.....	15
Seguridad y cifrado.....	16
Xbee Explorer USB:	17
Arduino Uno R3	18
Librerías.....	21
Pretenciones a desarrollar en este proyecto	21
Prueba de rango.....	21
Prueba de la red	24
Adquisición de datos y fabricación de prototipo	26
Teorema de Nyquist	27
Pasos para convertir la señal analógica en digital.....	27
Sensor de efecto Hall	28
Sensor de efecto Hall de salida digital.....	29
Conversor ADC de Arduino	29
Conversión de señales del driver	30

Control autónomo del driver led.....	42
Manipulación, programación y monitoreo de datos	45
Raspberry Pi.....	45
Modelos.....	45
Raspberry Pi 1 Modelo A.....	45
Raspberry Pi 1 Modelo B y B+	46
Raspberry Pi 2 Modelo B.....	46
Raspberry Pi 3 Modelo B.....	47
Raspberry Pi 3 Model B+.....	47
Software	47
Hardware	49
QT	50
Propósitos y características.....	50
Plataformas	50
Implementación	51
Tramas	52
Comunicación.....	54
Programación	56
Coordinador.....	56
Luminaria.....	57
Acceso remoto.....	59
Remote Desktop Protocol (RDP).....	59
VNC (Virtual Network Computing).....	60
Funcionamiento.....	61
Implementación	62
Diseño completo	68
Exterior	68
Interior	69
Software	70
Controlador de luminaria	74
Placa de control.....	74
Esquemático.....	75
PCB.....	77
Placa de potencia.....	78
Esquemático.....	78

PCB	79
Exterior	80
Interior	81
Controlador de segmento y controladores de luminarias	83
Capítulo 3: Resultados	84
Clase del instrumento	84
Capítulo 4: Análisis de costo	90
Introducción	90
Costos de desarrollo de los prototipos:	90
Ejemplo de proyecto futuro.....	91
Costos de la mano de obra (M.O.)	91
Costo de los materiales (M).....	92
Costo y mantenimiento del lugar de trabajo (L.T.).....	93
Costos totales.....	93
Comparativa en base al mercado actual	93
Tecnología SODIO	94
Tecnología LED.....	94
Comparativa entre estas tecnologías	94
Ahorro en corto y mediano plazo.....	95
Equipos con telegestión de luminaria	95
SODIO vs LED vs LED + TELEGESTION.....	96
Comparación tecnologías (consumo, IRC, eficiencia lumínica y flujo luminoso)	98
Capítulo 5: Discusión y Conclusión.	102
Las prestaciones del sistema de telegestión desarrollado son las siguientes:	102
Los problemas hallados fueron:.....	103
Las mejoras propuestas son:.....	103
Capítulo 6: Literatura Citada.	104

Lista de Figuras:

Figura 1 – (1) Ahorro energético por uso de nuevas tecnologías	1
Figura 2 – (Lutemberg, 2016) Niveles de telegestión	2
Figura 3 – Diagrama en bloques.....	5
Figura 4 – Comparación de velocidad efectiva entre ZigBee y Bluetooth en base a largo de paquete.....	7
Figura 5 – Modelo de Pila de ZigBee.....	11
Figura 6 – Modulo Xbee S2C.....	14
Figura 7 – Xbee Explorer USB.....	17
Figura 8 – XCTU.....	18
Figura 9 – Arduino Uno R3	18
Figura 10 – Xbee Shield para Arduino	19
Figura 11 – IDE de Arduino	20
Figura 12 – Diagrama de flujo, Programa de prueba de Rango	22
Figura 13 – Google Earth – Distancia de Prueba de Rango.....	22
Figura 14 – Configuración de la Seguridad del Xbee mediante XCTU	23
Figura 15 – Grafica de Potencia de Señal Recibida	23
Figura 16 – Muestra de paquetes erróneos y perdidos en la prueba de Rango	23
Figura 17 – Distancia en Google Earth con y sin seguridad	24
Figura 18 – Grafico de XCTU de la Red	25
Figura 19 – Dispositivos que XCTU reconoció mediante RF	25
Figura 20 – Dispositivos que XCTU de la nueva Red	26
Figura 21 – Recepción de datos en Arduino.....	26
Figura 22 – (Mecanica Web, 2017)Respuesta de sensor de efecto Hall	28
Figura 23 – (Mecanica Web, 2017) Sensor de efecto Hall por dentro	29
Figura 24 –Curva de Votaje vs Flujo Magnetico	30
Figura 25 – Grafico de Arduino luego de la digitalización de corriente	32
Figura 26 – Grafico de Arduino luego de la digitalización tensión y corriente.....	32
Figura 27 – Imagen de la pinza SCT-013	33
Figura 28 – Circuito interno de la Pinza SCT-013.....	33
Figura 29 – Diseño de la pinza de corriente	34
Figura 30 – Circuito para referenciar la señal.....	35
Figura 31 – Imagen de los valores digitalizados en Arduino con la mayor velocidad	36
Figura 32 – Circuito de proteccion	36
Figura 33 – Conversion AC-DC y fuente de 12v.....	37
Figura 34 – Circuito de Arduino con la adaptación de señal.....	37
Figura 35 – Imagen del RTC DS1307.....	38
Figura 36 – Diagrama de flujo de la luminaria	39
Figura 37 – Circuito adaptador de tensión alterna	39
Figura 38 – Circuito de la Fuente completa de 12 [V] y -12 [V].....	40
Figura 39 – Circuito de adaptación 0-5 [V] a 0-10 [V]	42
Figura 40 – Circuito de la etapa de potencia	43
Figura 41 – Imagen de la luminaria usada para las pruebas	44
Figura 42 –Imagen que muestra de cerca el tipo de driver LED.....	44
Figura 43 –Raspberry 1 model A	46
Figura 44 –Raspberry 1 model B	46
Figura 45 –Raspberry Pi 2 model B (2015).....	46
Figura 46 –Raspberry Pi 3 model B.....	47
Figura 47 –Raspberry Pi 3 model B+	47
Figura 48 –Pines GPIO.....	50
Figura 49 –Ventana Principal del Programa	51

Figura 50 –Ventana del Historico.....	52
Figura 51 –Vista general del Sistema completo en base al Software	55
Figura 52 –Diagrama de Flujo del coordinador en Arduino.....	56
Figura 53 –Diagrama de Flujo de la Luminaria en Arduino.....	58
Figura 54 –Esquema de RDP	60
Figura 55 –Ventana de linea de comando Raspberry.....	63
Figura 56 –Configuración de la IP	63
Figura 57 – Imagen de Configuración del Router	65
Figura 58 –Configuración de IP	65
Figura 59 –Ventana del Escritorio Remoto	66
Figura 60 –Ventana de RDP en Raspberry	66
Figura 61 –Vista de la ventana de Raspberry.....	67
Figura 62 –Vista Superior del Coordinador.....	68
Figura 63 –Vista Lateral del Coordinador	68
Figura 64 –Vista Frontal del Coordinador	69
Figura 65 –Vista Completa Coordinador-Router.....	69
Figura 66 –Vista Interior del Controlador de segmento.....	70
Figura 67 –Ventana de Programa Principal Raspberry	70
Figura 68 –Ventana de Programa Principal Raspberry – Obtener datos.....	71
Figura 69 –Ventana de Programa Principal Raspberry – Estado de luminaria.....	71
Figura 70 –Ventana de Programa Principal Raspberry – Intensidad de luz	72
Figura 71 –Ventana de Programa Principal Raspberry – Datos mostrados.....	72
Figura 72 –Ventana de Programa Principal Raspberry – Configuración horario	73
Figura 73 –Ventana de Programa Secundaria Raspberry – Historico	74
Figura 74 –Fuente y circuito de protección	75
Figura 75 – Conexiones a Arduino	75
Figura 76 – Circuito adaptador de tension	76
Figura 77 – Circuito conversor PWM a 0-10 [V].....	76
Figura 78 –Circuito adaptador de señal de corriente	76
Figura 79 –Circuito adaptador de NTC	77
Figura 80 – Vista del PCB en Eagle	77
Figura 81 –Foto de la placa lista para soldar	77
Figura 82 – Circuito de potencia.....	78
Figura 83 –Vista del pcb en Eagle	79
Figura 84 –Vista de la placa ya soldada	79
Figura 85 –Vista Superior del controlador de luminaria 1	80
Figura 86 –Vista Superior del controlador de luminaria 2	80
Figura 87 –Vista Frontal de la luminaria 1	81
Figura 88 –Vista Lateral del controlador de luminaria 1.....	81
Figura 89 –Vista Completa del controlador de luminaria y la luminaria ya conectada	81
Figura 90 –Vista Superior interna del controlador de luminaria 1	82
Figura 91 –Vista Superior interna del controlador de luminaria 2	82
Figura 92 –Vista de los tres equipos armados.....	83
Figura 93 –Señal de dimerización al 100% - tiempo.....	87
Figura 94 –Señal de dimerización al 100% - FFT	87
Figura 95 –Señal de dimerización al 50% - tiempo.....	88
Figura 96 –Señal de dimerización al 50% - FFT	88
Figura 97 –Señal de dimerización al 10% - tiempo.....	89
Figura 98 –Señal de dimerización al 10% - FFT	89
Figura 99 –Foto de información desglosada de boleta de electricidad - ENERSA	90
Figura 100 –Grafico de barras – Tecnología vs Consumo.....	97
Figura 101 –Grafico de barras – Tecnología vs Ahorro	97

Figura 102 –Grafico de barras – Tecnología vs Amortización en años	97
Figura 103 –Grafico de barras – Tecnología vs Consumo.....	98
Figura 104 –Grafico de barras – Tecnología vs IRC.....	98
Figura 105 – Iluminación LED vs Sodio en el mismo lugar.....	99
Figura 106 –Grafico de barras – Tecnología vs Eficiencia Luminica	99
Figura 107 –Grafico de barras – Tecnología vs Flujo luminoso.....	100
Figura 108 –Foto de zona iluminada por Lámpara de Sodio a las 19 horas.....	100

Lista de Tablas:

Tabla 1 – Costo estimado del proyecto.....	4
Tabla 2 - Comparación ZigBee Bluetooth	6
Tabla 3 - Características de velocidad de transferencia de los paquetes Bluetooth	7
Tabla 4 – Descripción de partes del modelo ZigBee	12
Tabla 5 – Ventajas y desventajas entre modo API y transparente	16
Tabla 6 – Sensado de Tensión – Corriente de Arduino vs Generador de Señales.....	41
Tabla 7 – Sensado de Coseno Fi de Arduino vs Generador de Señales.....	41
Tabla 8 – Trama de Configuración de Horario Unicast.....	53
Tabla 9 – Trama de Configuración de Horario Broadcast.....	53
Tabla 10 – Trama de Obtención de Datos	53
Tabla 11 – Trama de Envío de PWM Unicast.....	53
Tabla 12 – Trama de envío de PWM Broadcast.....	54
Tabla 13 – Trama de envío de Coordinador a Raspberry	54
Tabla 14 – Características del Controlador de Segmento	68
Tabla 15 – Características del Controlador de Luminaria.....	74
Tabla 16 – Mediciones de tensión	84
Tabla 17 – Mediciones de Corriente	85
Tabla 18 – Mediciones del coseno fi.....	86
Tabla 19 – Costos de materiales para los prototipos	91
Tabla 20 – Costo de materiales – ejemplo.....	92
Tabla 21 – Costo y mantenimiento de lugar de trabajo	93
Tabla 22 – Costo del controlador de segmento	93
Tabla 23 – Costo de materiales – tecnología SODIO	94
Tabla 24 – Costo de materiales – tecnología LED.....	94
Tabla 25 – Tecnología vs Consumo anual para 35.000 luminarias	96
Tabla 26 – Tecnología vs Ahorro y amortización en años	97
Tabla 27 – Tecnología vs Consumo – IRC – Eficiencia lumínica – Flujo luminoso.....	98

Dedicado a:

Capítulo 1: Introducción

Descripción del proyecto:

En la actualidad, es cada vez más imperiosa la necesidad de implementar sistemas de iluminación inteligentes para las grandes ciudades, ya que esto ayuda al ahorro de energía de las mismas, así como también para poder cargar programas especiales a pedido de cada zona. A continuación se verá un gráfico de Philips donde se muestra el consumo en el alumbrado público según qué tecnología se aplique:

Alumbrado público

Ahorros energéticos por uso de nueva tecnología

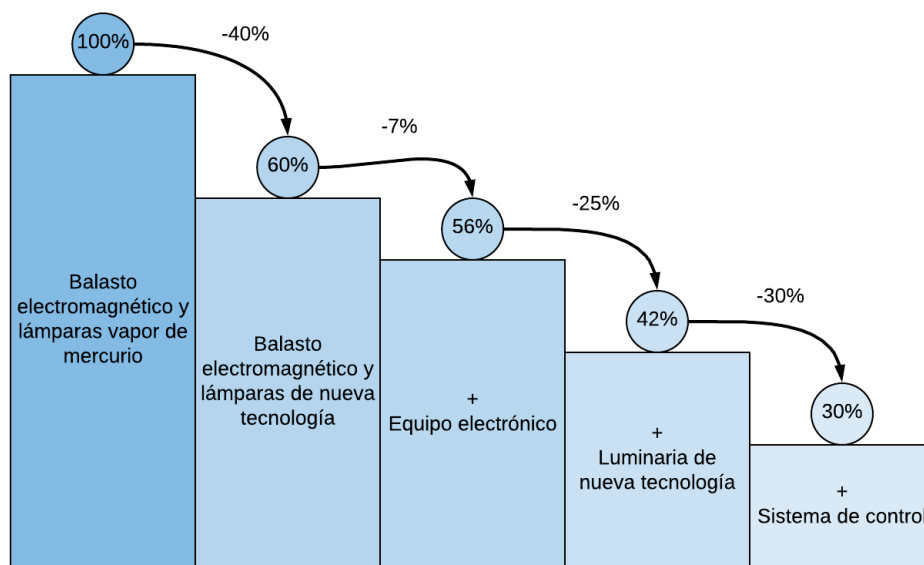


Figura 1 – Ahorro energético por uso de nuevas tecnologías

Es por esto, que este proyecto tuvo como fin la realización de sistemas de luminarias dirigidas por telexmando, donde se hicieron tanto los sistemas de control inteligentes de cada luminaria como así también los sistemas que controlan a distancia las mismas. Esto se hizo en base a uno o más protocolos de comunicación, donde se estudiaron las ventajas y desventajas de usar los protocolos basados en IEEE 802.15.4 e IEEE 802.11. A continuación se observa un cuadro donde se observan los distintos niveles dentro de la arquitectura de telegestión de luminarias.

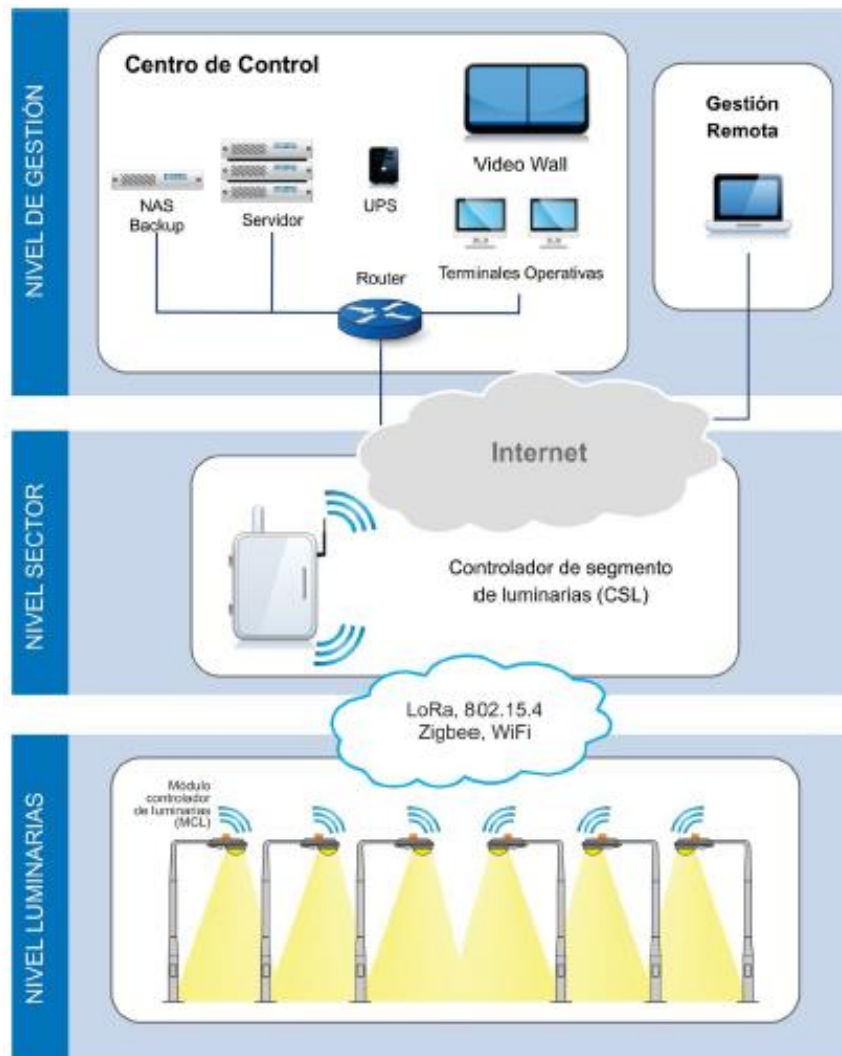


Figura 2 – (Lutemberg, 2016) Niveles de telegestión

De los 3 niveles se implementaron los niveles inferiores, el de nivel luminarias, donde se desarrolló el controlador de luminaria, y nivel sector, donde aquí se trabajó en el controlador de segmento. En el caso de nivel de gestión se utilizó una PC para acceder remotamente a través de internet al controlador de segmento.

Se estudiaron las características que CADIEEL considera importantes para un sistema de telegestión, basándonos en el “Documento de Acuerdo entre los Socios de CADIEEL respecto al Sistema de Telegestión de Luminarias”.

(CADIEEL, 2016) Las especificaciones técnicas de este acuerdo son las siguientes:

- Implementar el enlace de área local entre el Nivel de Sector y el Nivel de Luminarias en base al estándar 802.15.4 en frecuencia 2.4 GHz de nivel físico y de control de acceso al medio.
- Utilizar ZigBee Pro con un protocolo de aplicación a ser definido por CADIEEL adaptado a las necesidades locales para implementar la transferencia de datos y comandos sobre el enlace de área local entre el Nivel de Sector y el Nivel de Luminarias.
- Utilizar un protocolo a ser definido por CADIEEL adaptado a las necesidades locales para implementar el enlace de red privada virtual entre el Nivel de Gestión y el Nivel de Sector.

- El modulo Controlador de Luminarias debe ser capaz de medir potencia activa con una tolerancia menor o igual al 5% y reportar al Controlador de Segmento de Luminarias.
- El zocalo a utilizar para las lámparas será el definido por el estándar ANSI C136.41-2013.
- El modulo controlador de luminarias debe ser capaz de controlar la intensidad de iluminación mediante drivers de tipo 1-10VDC.
- El modulo controlador de luminarias debe ser capaz de detectar y reportar la salida de servicio en las luminarias.
- El modulo controlador de luminarias tendrá un contacto seco de libre disponibilidad en el zocalo de la lámpara tal que cada fabricante pueda utilizarlo para implementar funcionalidades adicionales.
- El modulo controlador de luminarias debe ser capaz de apagar completamente la luminaria.
- En el caso que se pierda la comunicación entre el modulo controlador de luminaria y el Controlador de segmento de luminarias, el modulo controlador de luminaria debe ser capaz de responder al nivel real de iluminación a través del sensado de la luz ambiente.
- El modulo controlador de luminarias debe funcionar en el rango de tensión de alimentación de 220 VAC +/- 10% de 50 Hz.

La idea de la realización de este proyecto se da porque es una tecnología nueva que se está desarrollando en la actualidad, y recién está implementándose en las ciudades más importantes del mundo.

Este proyecto se estudia para ser implementado en pequeñas o medianas ciudades.

Se desarrollaron una serie de controladores de luminaria donde cada una tiene:

- 1- Una serie de sensores para poder medir el funcionamiento del driver, la entrega de corriente y tensión, como así también sensores para el funcionamiento autónomo de la luminaria (foto célula).
- 2- Comunicación inalámbrica para poder comunicarse con el controlador de segmento y reportar el funcionamiento de la luminaria además de reportar fallas del driver. Además poder crear una red de comunicación entre luminarias donde cada una es un nodo de la red.
- 3- El controlador de luminaria puede suministrar al driver una tensión continua variable entre 1 [V] y 10 [V] para el control de la intensidad de potencia que entrega el driver.

Se hizo un conjunto de 2 controladores de luminaria que sirvió para estudiar el funcionamiento real de la red.

La luminaria tiene internamente un reloj en tiempo real y su propio módulo de comunicación para reportar el funcionamiento al centro de control.

Además tiene la suficiente inteligencia para poder solucionar problemas de red.

Estudio de mercado:

El objetivo principal de este proyecto está orientado a los municipios y comunas de todo el país.

En estos momentos no existe ni un precio establecido para este tipo de sistemas ni una empresa que monopolice el mercado, un acérrimo competidor será Philips, que está desarrollando el producto "starsense wireless".

Este producto es el futuro de la luminaria inteligente en todo el mundo, la utilidad del mismo está íntimamente relacionada con la estandarización de los equipos y protocolos de comunicación, como así también de la competitividad en costos y la flexibilidad de funcionamiento en diversos ambientes.

Por nuestra parte, se realizaron pruebas de campo tanto en la comunicación de las luminarias como en el funcionamiento de un driver real de luminaria LED.

La ventaja que tiene nuestro proyecto en comparación con los demás es que esta específicamente adecuado a las normas argentinas, por lo que sumado a un bajo costo es altamente competitivo en este país, y potencialmente competitivo en el mundo.

Para la realización del proyecto se necesitó conocer sobre ciertos aspectos como adquisición de datos, procesamiento digital, conexiones inalámbricas, programación e integración de todo el conjunto de componentes del sistema.

Fue necesario contar con una serie de elementos, desde los controladores de luminarias LED, los controladores de segmento y una computadora que funcione como servidor. El costo del proyecto dependerá de la cantidad de luminarias que se abarcaran y de la cantidad de controladores de segmento. Para nuestro caso y de forma de ilustrar el funcionamiento e implementación solo se utilizaron 2 luminarias, un controlador de segmento y el servidor. Aclarado esto deducimos que el costo del producto es el siguiente:

	Producto	Cantidad	Precio
Controlador de luminaria LED	Arduino	2	\$300
	Driver de LED		\$5000
	Sensores		\$300
	Modulo inalámbrico		\$250
	Módulo Xbee		\$2800
Controlador de segmento	Raspberry Pi 3	1	\$1500
	Arduino		\$300
	Modulo inalámbrico		\$250
	Módulo Xbee		\$2800
Servidor	PC de escritorio	1	\$10000
Precio total			\$32150

Tabla 1 – Costo estimado del proyecto

El precio que se obtuvo es un estimativo de lo que cuesta cada parte que compone el sistema y depende de ciertos parámetros y del producto que se utilice, que hasta el momento, no se había confirmado exactamente que se iba a utilizar, debido a que se hizo un análisis de que producto era más útil y cual ofrecía la mejor relación precio calidad. De todas maneras el precio estimado como se ve a continuación y pasado a dólares, es aproximadamente USD 869.

El ciclo de vida del producto depende principalmente del driver de la luminaria led ya que es este el que tiene mayor probabilidad de falla. Otro problema que puede existir es la falla en la comunicación, pero esta no perturba el funcionamiento de la luminaria ya que la misma sigue funcionando y solo debe solucionarse el problema realizando el mantenimiento necesario.

Capítulo 2: Desarrollo

Diagrama de Bloques.

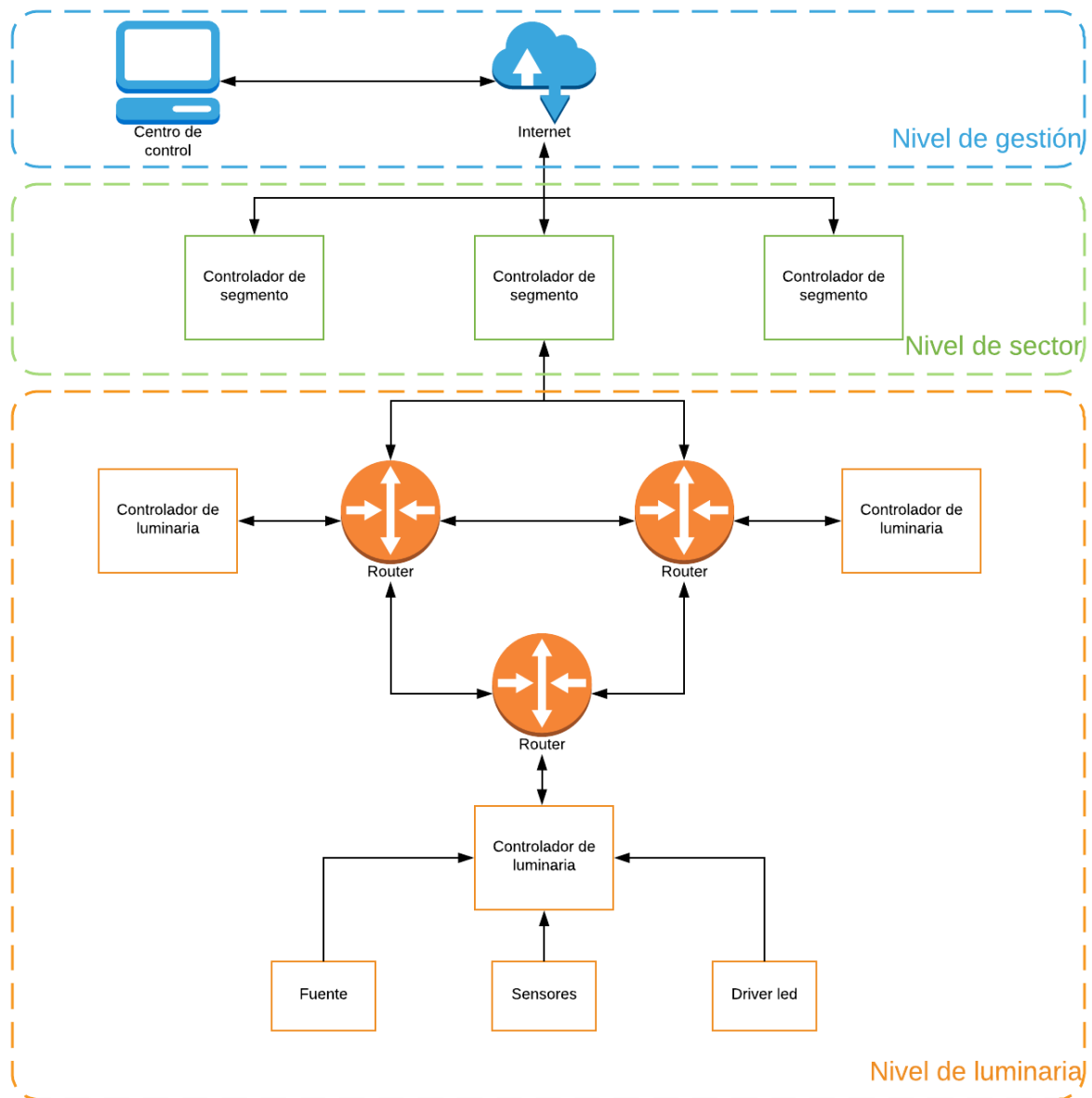


Figura 3 – Diagrama en bloques

Lógica de Funcionamiento

El nivel de luminarias se separa en dos etapas. La etapa de dispositivos finales y la etapa de routers. La etapa de dispositivos finales no se implementó en la realidad del proyecto, ya que se necesitaba una red tipo malla con una interconexión entre todas las luminarias, es por esto que solo se usó dispositivos tipo routers.

El nivel de sector posee un controlador de segmento que se encarga de comunicar los routers de una misma red entre sí, este controlador es el que envía información a todos los nodos (broadcast), o puede elegir a que dispositivo en particular (luminaria) enviarle un mensaje. En este caso los dispositivos routers re transmitirán la información para que llegue a destino.

Los puntos finales o nodos son los que adquieren los datos de cada driver LED, como el estado del mismo, el nivel de potencia que está entregando, y el consumo de energía que

se calcula en base a sensores, un sensor de efecto hall o transformador de corriente y un muestreo de tensión de alimentación. Estos datos son enviados periódicamente al router, para que lo entregue al controlador de segmento, el cual interpreta la información, la archiva y en caso necesario crea un informe de la misma.

El dispositivo final o nodo puede trabajar en línea (in-line) o fuera de línea (off-line). Hemos explicado el caso in line. En el caso que trabaje off-line tiene una fotocélula que indica al nodo el momento en que debe encenderse y apagarse la luminaria. Además almacena el promedio de consumo de energía para que en el momento en que este en línea reporte al controlador de segmento. A esto hay que sumarle que el controlador del driver debe generar una señal de tensión de 1 [v] a 10 [v] para controlar la potencia que el driver entrega.

El dispositivo final tiene también un reloj RTC para poder ser configurado con las características que el cliente desee dependiendo la hora.

El controlador de segmento se conecta a internet todo el tiempo, para el acceso remoto al mismo y de este modo se puede configurar toda la red desde internet. El controlador tiene un software que es el encargado de procesar los requerimientos del cliente y es el mismo que envía la información a los nodos correspondientes. Además en este programa se ven los consumos de los distintos nodos de la red y el consumo general.

(Dignani, 2011) Estudio de los distintos protocolos para la comunicación entre luminarias

ZigBee vs. Bluetooth

Es inevitable la comparación entre dos estándares de redes WPAN. Bluetooth es un protocolo que se ha popularizado en los últimos años. Su objetivo de diseño fue la eliminación de cables de interconexión de datos entre equipos de consumo masivo como: teléfonos celulares, notebooks, netbooks, pdas, equipos de audio, televisión. Así es posible escuchar temas musicales por la radio del auto transmitidos desde el teléfono celular o conectar un GPS a una netbook.

ZigBee está especialmente diseñado para el manejo de controles, recolección de datos de sensores, preparado para trabajar con muchos (centenas hasta miles) dispositivos de tipo portátiles en donde la duración de las baterías es un factor crítico.

Interface de comunicación radial.

Ambas usan la técnica de espectro extendido. Bluetooth usa FHSS y ZigBee usa DSSS. En la tabla siguiente se muestra una comparación de ZigBee y Bluetooth.

Característica	ZigBee	Bluetooth
Modulación	DSSS	FHSS
	11 chips/símbolo	1600 saltos/segundo
	62.5 Ksímbolo/segundo	1 Msímbolo/segundo
	4 bits/símbolo	1 bit/símbolo
Tasa de información pico	~127 Kbit/segundo	~108-723 Kbit/segundo
Tiempo de enumeración	30 ms típico	>3 s, 20 segundos típico
Tiempo de transición de dormido a activo	15 ms típico	3 s típico
Tiempo de un esclavo activo de acceso al canal	15 ms típico	2 ms típico

Tabla 2 - Comparación ZigBee Bluetooth

Comparación en gasto de batería.

El consumo de energía es directamente proporcional al tiempo que los dispositivos estén en transmisión/recepción y esto está relacionado al tamaño del paquete ya que cuanto más pequeño es el paquete antes los dispositivos entrarán en modo dormir. Por otro lado, cuanto más grande sea el paquete más se acerca la tasa efectiva de datos a la velocidad plana de la interface.

Bluetooth es un protocolo que se maneja en base a ranuras de tiempo. La comunicación puede ocurrir en 1 ranura = 625 [μs], 3 ranuras= 1875 [μs] ó 5 ranuras = 3125 [μs].

Considerando que después de cada transmisión llega un ACK en la próxima ranura, se muestran en la Tabla 3 los tiempos en función de los tipos de paquetes.

Tipo de paquete	N° de ranura	Carga máxima de información [bytes]	Tiempo total [us]	Tasa efectiva [Kb/s]
DH1	1	27	1250	172.8
DH3	3	183	2500	585.6
DH5	5	339	3750	723.2

Tabla 3 - Características de velocidad de transferencia de los paquetes Bluetooth

En la figura que sigue se puede ver cómo aparecen picos en la tasa efectiva de datos de Bluetooth. La carga se manda en ranuras y se debe usar una ranura tanto para enviar 2 bytes como para 5, 8 ó 27.

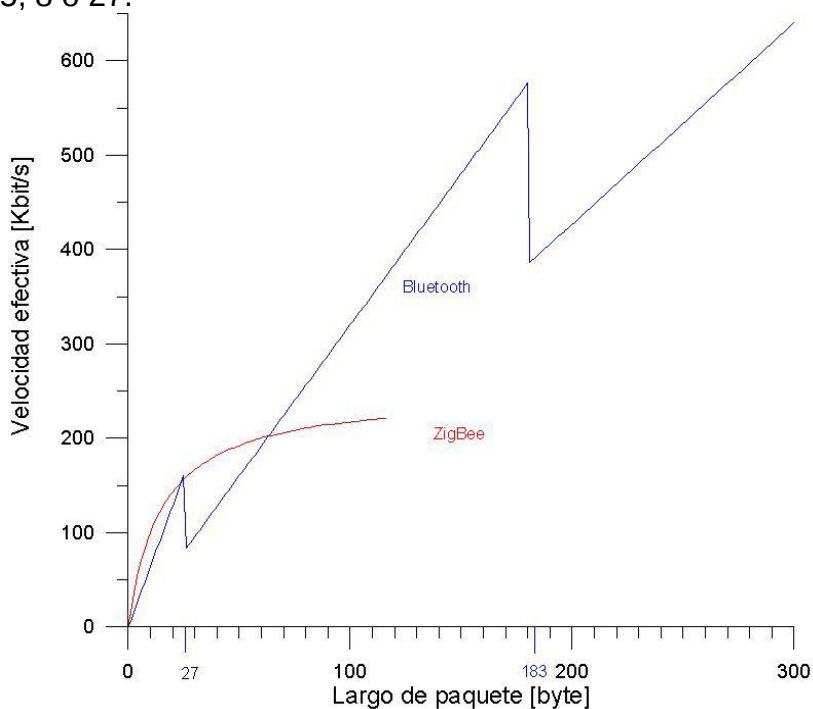


Figura 4 – Comparación de velocidad efectiva entre ZigBee y Bluetooth en base a largo de paquete

ZigBee fue diseñado pensando en paquetes pequeños y se puede observar que para paquetes de menos de 75 bytes, ZigBee tiene, a pesar de su baja velocidad física de transmisión de datos, una velocidad efectiva mayor que Bluetooth. Por lo tanto, para pequeños paquetes Bluetooth va a gastar mayor energía debido a que necesita mayor tiempo de transmisión/recepción. La gráfica se basa en los protocolos Bluetooth v1.1 y el IEEE 802.15.4 en condiciones ideales, esto es: el canal está vacío, no hay errores y no se necesita retransmisión.

Otra diferencia notable es en el tiempo de enumeración. En Bluetooth se requieren típicamente 20 segundos contra 15 [ms] de ZigBee. En conclusión como ZigBee está muy bien preparado para realizar rápidamente el proceso de conexión y desconexión a la red, esto determina que se requiera la centésima parte de la energía que requiere Bluetooth para esta misma tarea.

Además, el protocolo Bluetooth se basa en encuestar a cada dispositivo esclavo y en cambio ZigBee se basa en CSMA-CA, que debe esperar a tener el canal libre. Esto no representa un problema porque en redes de muy bajo tráfico como redes de sensores, no hay competencia por el canal.

Como corolario, el bajo consumo energético de ZigBee, permite usar pilas alcalinas tipo AA por más de 6 meses (hasta 2 años típico). Lo típico en Bluetooth es usar baterías recargables con periodicidad de carga de algunos días. ZigBee y Bluetooth son dos soluciones pensadas para aplicaciones diferentes. Bluetooth es apto para aplicaciones de baja latencia como audio y video. Es un protocolo maestro-esclavo para unos pocos dispositivos. ZigBee está diseñado para uso de sensores y controles que usan mensajes cortos. Está preparado para redes tipo estrella o entre pares de cientos de dispositivos.

Estas diferencias entre ZigBee y Bluetooth se originan en las arquitecturas de cada uno de estos protocolos y no parece probable que los cambios de versiones modificaría la situación.

Otras tecnologías WPAN

Existen varios estándares además de ZigBee que usan el IEEE802.15.4 como base de su protocolo. Los más conocidos son 6LoWPAN y WirelessHART. También existen otros protocolos que no usan IEEE802.15.4. Se destacan entre estos Z-wave, Bluetooth y ULP Bluetooth.

El protocolo 6LoWPAN

La filosofía de este protocolo es poder transmitir paquetes de tipo Ipv6 para simplificar la interface entre redes de sensores e Internet. La ventaja natural es la facilidad de conexión de la red WPAN a Internet. La desventaja es que los nodos de sensores tienen limitaciones en capacidad de procesamiento. IPV6 requiere soporte de paquetes más grande que lo que brinda IEEE 802.15.4. El payload máximo de IEEE 802.15.4 es de 128 bytes contra 1280 bytes requeridos por IPV6 por lo que requiere una fragmentación. Para eso se agrega una capa en 6LOWPAN llamada capa de adaptación que fragmenta y rearma los paquetes.

La comparación con ZigBee indica que para aplicaciones de paquetes pequeños con baja interacción con dispositivos IP es más eficiente ZigBee. La interacción con Internet la puede hacer un dispositivo puente.

WirelessHart

HART es un protocolo usado en la industria para control de procesos, diagnóstico y control. Es un protocolo para usar en redes cableadas. WirelessHart es la extensión a redes inalámbricas que usa la banda de 2.4 GHz y para seguridad aplica AES-128 tal como ZigBee. Si bien usa la misma base de IEEE 802.15.4 que ZigBee, utiliza potencias más elevadas, y programa la capa física para poder hacer saltos de canal paquete a paquete. Tanto ZigBee como WirelessHart se usan en ambientes industriales.

La ventaja de este último protocolo es la compatibilidad hacia atrás con las redes HART.

Z-wave

Fue desarrollado por la Alianza Z-wave. No adopta IEEE 802.15.4. Usa la banda de 900MHz en un sistema de banda angosta. Usa FSK con velocidad de datos de 40kbps. Z-Wave soporta direccionamiento de 8 bits contra los 16 bits de direccionamiento de ZigBee. Usa una variante de DES (Data Encryption Standard) como método de seguridad. Para algunas aplicaciones esta seguridad puede ser insuficiente ya que usa una clave de solo 56 bits. ZigBee ofrece más velocidad, seguridad y mayor cantidad de nodos en una red. Z-wave es más simple y consecuentemente tiene más bajo costo por nodo.

ULP Bluetooth

Ya se comparó ZigBee con Bluetooth y se remarcó el gran gasto energético de este último. ULP Bluetooth es un estándar desarrollado para comunicaciones con bajo ciclo efectivo con el objeto de reducir el consumo energético. ULP no soporta redes tipo malla. ULP compite con ZigBee en aplicaciones punto a punto. Hay algunos dispositivos Bluetooth llamados de modo dual que pueden hacer de interface entre los dispositivos ULP y los Bluetooth clásicos ya que trabajan en los dos modos.

La alianza ZigBee compuesta por importantes firmas desarrolladoras de hardware y software ha impuesto su "estándar" frente a otras alianzas que desarrollaron protocolos para un uso más específico. Hay muchos protocolos propietarios que fundamentalmente en redes pequeñas pueden ser más eficaces pero la estandarización de productos hace que sea más simple y seguro adoptar un estándar ampliamente difundido.

Hay todo un campo específico de aplicación en sistemas de baja transferencia de datos y baja energía y bajo costo en el cual ZigBee compite favorablemente. A pesar de su relativa simplicidad comparada con otros estándares, ZigBee provee confiabilidad, flexibilidad y escalabilidad.

Para desarrollar su estándar, la alianza ZigBee se ha valido de otros estándares: IEEE802.15.4 para sus capas inferiores y algoritmos clásicos para ruteo y seguridad.

Con los perfiles que los fabricantes usan en sus dispositivos es fácil para el usuario configurar una red.

Es por estas razones que se decidió la implementación de la red utilizando dispositivos de la Alianza ZigBee.

Elementos a utilizar en esta etapa

Para el controlador se utilizó una Raspberry Pi 3, para los nodos se usó un Arduino UNO y para la comunicación en general se utilizó el protocolo ZigBee, en particular módulos Xbee de end devices y router.

Para empezar se puede mencionar que para la primera parte, que es la comunicación RF con los módulos Xbee, se tiene hardware y software, los cuales se mencionaran a continuación:

Hardware:

- Arduino Uno R3
- Xbee Shield para Arduino Uno R3
- Módulo Xbee S2C
- Xbee Explorer USB

Software:

- Ide de Arduino
- XCTU

Donde se pueden separar los dispositivos que son programables de los no programables. Los dos dispositivos que son programables y de hecho se programaron son el Arduino Uno R3 y el Modulo Xbee S2C, cada uno con el software específico de cada dispositivo.

Módulo Xbee

Para saber cómo funcionan los módulos Xbee es importante entender de donde han surgido, es por esto que empezaremos por explicar que es el protocolo ZigBee.

ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (wireless personal area network, WPAN). Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

En principio, el ámbito donde se prevé que esta tecnología cobre más fuerza es en domótica, como puede verse en los documentos de la ZigBee Alliance, en las referencias bibliográficas que se dan más abajo en el documento «ZigBee y Domótica». La razón de ello son diversas características que lo diferencian de otras tecnologías:

- Su bajo consumo.
- Su topología de red en malla.
- Su fácil integración (se pueden fabricar nodos con muy poca electrónica).

(Digi, 2018) Capas de pila ZigBee

La mayoría de los protocolos de red utilizan el concepto de capas para separar diferentes componentes y funciones en módulos independientes que se pueden ensamblar de diferentes maneras.

ZigBee se construye sobre la capa física (PHY) y la capa secundaria de control de acceso al medio (MAC) definida en el estándar IEEE 802.15.4. Estas capas manejan operaciones de red de bajo nivel tales como direccionamiento y transmisión / recepción de mensajes.

La especificación ZigBee define la capa de Red (NWK) y el marco para la capa de aplicación. La capa de red se encarga de la estructura de la red, el enrutamiento y la seguridad. El marco de la capa de aplicación consiste en la subcapa de soporte de aplicaciones (APS), los objetos de dispositivo de ZigBee (ZDO) y las aplicaciones definidas por el usuario que le dan al dispositivo su funcionalidad específica.

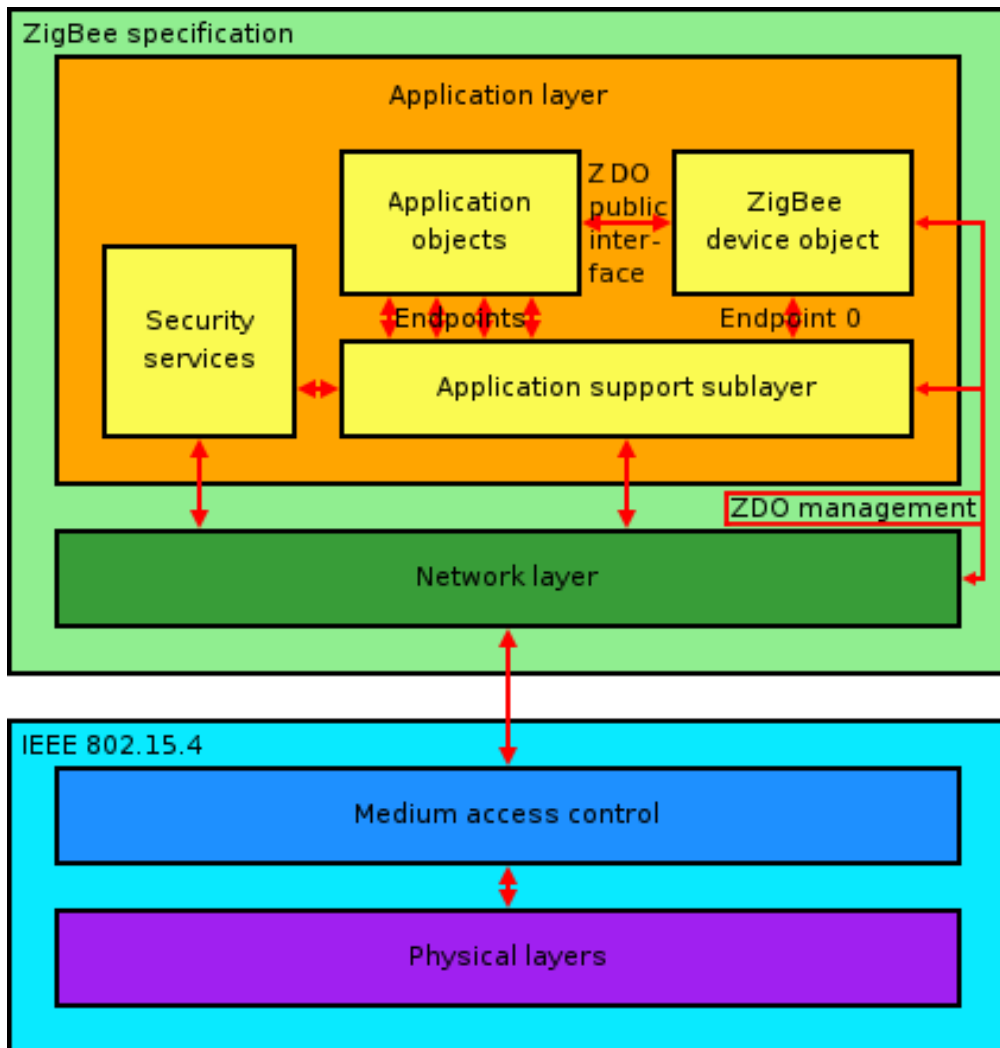


Figura 5 – Modelo de Pila de ZigBee

Cada red de ZigBee debe estar formada por uno y sólo un coordinador y por lo menos otro dispositivo (enrutador o dispositivo final).

Capa ZigBee	Descripción
PHY	Define el funcionamiento de la capa física del dispositivo ZigBee incluyendo: sensibilidad del receptor, degeneración del canal, potencia de salida, número de canales, modulación de chip y especificaciones de la tasa de transmisión. La mayoría de las aplicaciones ZigBee trabajan en la banda reservada ISM 2,4 [GHz], a una tasa de datos de 250 [Kbps]
MAC	Maneja las transacciones de datos de radiofrecuencia entre dispositivos cercanos, incluyendo servicio tales como el reintento de transmisión y gestión de reconocimiento, junto con técnicas para evitar colisiones como CSMA-CA
Red	Añade capacidades de enrutamiento que permiten los paquetes de datos de radiofrecuencia atravesar varios dispositivos para enrutar la información desde la fuente hacia el destino

APS	Capa de aplicación que define varios objetos de direccionamiento, incluidos perfiles, grupos, y los puntos finales
ZDO	Capa de aplicación que proporciona las características del dispositivo y de descubrimiento de servicio y capacidades avanzadas de administración de red

Tabla 4 – Descripción de partes del modelo ZigBee

Tipos de dispositivos

Se definen tres tipos distintos de dispositivo ZigBee según su papel en la red:

- Coordinador ZigBee (ZigBee Coordinator, ZC). El tipo de dispositivo más completo. Debe existir uno por red. Sus funciones son las de encargarse de controlar la red y los caminos que deben seguir los dispositivos para conectarse entre ellos.
- Router ZigBee (ZigBee Router, ZR). Interconecta dispositivos separados en la topología de la red, además de ofrecer un nivel de aplicación para la ejecución de código de usuario.
- Dispositivo final (ZigBee End Device, ZED). Posee la funcionalidad necesaria para comunicarse con su nodo padre (el coordinador o un router), pero no puede transmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías. Un ZED tiene requerimientos mínimos de memoria y es por tanto significativamente más barato.

Topologías de red

ZigBee permite tres topologías de red:

- Topología en estrella: el coordinador se sitúa en el centro.
- Topología en árbol: el coordinador será la raíz del árbol.
- Topología de malla: al menos uno de los nodos tendrá más de dos conexiones.

La topología más interesante (y una de las causas por las que puede triunfar ZigBee) es la topología de malla. Ésta permite que si, en un momento dado, un nodo del camino falla y se cae, pueda seguir la comunicación entre todos los demás nodos debido a que se rehacen todos los caminos. La gestión de los caminos es tarea del coordinador.

Red de malla

Una red de malla es una topología en la que cada nodo de la red está conectado a otros nodos a su alrededor. Cada nodo coopera en la transmisión de información. Las redes de malla proporcionan tres beneficios importantes:

- Enrutamiento. Con esta técnica, el mensaje se propaga a lo largo de una ruta saltando de nodo a nodo hasta que alcanza su destino final.
- Creación de redes ad-hoc. Este es un proceso automatizado que crea una red entera de nodos sobre la marcha, sin ninguna intervención humana.
- Auto-sanación. Este proceso calcula automáticamente si falta uno o más nodos en la red y reconfigura la red para reparar cualquier ruta rota.

Las redes de malla utilizan más ancho de banda para administración y por lo tanto tienen menos disponible para cargas útiles. También pueden ser más complejos de configurar y depurar en algunos casos.

Con redes de malla, la distancia entre dos nodos no importa, siempre y cuando haya suficientes nodos entre ellos para transmitir el mensaje. Cuando un nodo desea comunicarse con otro, la red calcula automáticamente la mejor ruta.

Una red de malla también es confiable y ofrece redundancia. Si un nodo ya no puede funcionar, por ejemplo porque se ha eliminado de la red o porque una barrera bloquea su capacidad de comunicación, el resto de los nodos pueden comunicarse entre sí, ya sea directamente o a través de nodos intermedios.

Series de XBee

Dentro de los módulos Xbee hay varias series que las describiremos a continuación:

- XBee Series 1 (también llamados XBee 802.15.4) – Son la serie más fácil para trabajar, no necesitan ser configurados, pero incluso así se pueden obtener beneficios de estos módulos. Debido a que son fáciles para trabajar, son los más recomendables especialmente si se está empezando. Para comunicaciones Punto-a-Punto, estos módulos trabajan tan bien como los de la Serie 2, pero sin todo el trabajo de pre configuración previa. El hardware de las Series 1 y las Series 2/2.5/ZB no son compatibles.
- XBee Znet 2.5 (Formalmente Series 2) Retirado – Los módulos Serie 2 deben ser configurados antes de ser usados. Pueden funcionar en modo Transparente o por medio de comandos API, pero todo esto depende de que firmware se configure en los módulos. También pueden funcionar en una red mesh. Son más difíciles que usar que los de la Serie 1. No existe una forma en que estos módulos sean compatibles con los de la Serie 1. Los módulos Znet 2.5 ya no se venden, pero han sido reemplazados con módulos ZB más compatibles.
- XBee ZB (el actual módulo Series2) – Básicamente es el módulo Znet 2.5, pero con un nuevo firmware. Esto significa que también funcionan en modo transparente o por medio de comandos API. También funcionan en redes mesh. Estos a menudo son llamados módulos de Serie 2, por lo que si escuchas a alguien hablar sobre esta serie, probablemente estén hablando de estos módulos. Puede que no sea el término correcto, pero se hace distinción de estos con los módulos de la Serie 1, los cuales son los más populares.
- XBee 2B (el más actual módulo Series2) – Son nuevos módulos que poseen mejoras en el hardware respecto de los de la Serie 2, básicamente son los mismo que los anteriores pero con un firmware más nuevo, mejorando por ejemplo el uso de la potencia. Funcionan con el Firmware del módulo ZB, pero debido al cambio de hardware, ya no pueden funcionar con el firmware del módulo Znet 2.5. Por lo que ten cuidado si agregas uno de estos módulos a una red ya existente que utilice módulos Znet 2.5. Actualmente algunas tarjetas son 2B y otras son ZB.
- 900 [MHz] vs 2.4 [GHz] – La mayoría de los módulos XBee operan a 2.4 [GHz], pero hay unos pocos que operan a 900 MHz. Básicamente los de 900 [MHz] pueden llegar muy lejos con una antena de alta ganancia (hasta casi 24 [Km]). Además a menor frecuencia, la señal posee mayor penetración. Otro punto importante es que los módulos de 900 [MHz] no están permitidos en algunos países, Digi tiene versiones de 868 [MHz] que sí está permitido en la mayoría de los países.

En nuestro caso se utilizó los Xbee S2C, se muestra una fotografía a continuación:



Figura 6 – Modulo Xbee S2C

Cómo se comunican los dispositivos XBee

Los dispositivos XBee se comunican entre sí a través del aire, enviando y recibiendo mensajes inalámbricos. Los dispositivos solo transfieren esos mensajes inalámbricos; no pueden administrar los datos recibidos o enviados. Sin embargo, pueden comunicarse con dispositivos inteligentes a través de la interfaz en serie.

Los dispositivos XBee transmiten datos provenientes de la entrada en serie a través del aire, y envían todo lo recibido de forma inalámbrica a la salida en serie. Ya sea para fines de comunicación o simplemente para configurar el dispositivo, una combinación de ambos procesos hace posible la comunicación con XBee. De esta forma, los dispositivos inteligentes como los microcontroladores o las PC pueden controlar lo que el dispositivo XBee envía y administrar los mensajes inalámbricos entrantes.

Con esta información, se identifican los dos tipos de transmisión inalámbrica de datos en un proceso de comunicación XBee:

1. Comunicación inalámbrica: esta comunicación tiene lugar entre los módulos XBee. Los módulos que se supone que funcionan juntos deben formar parte de la misma red y deben usar la misma frecuencia de radio. Todos los módulos que cumplen estos requisitos pueden comunicarse de forma inalámbrica entre sí.

2. Comunicación en serie: esta comunicación tiene lugar entre el módulo XBee y el dispositivo inteligente conectado a él a través de la interfaz en serie.

Comunicación inalámbrica

Se puede hacer una comparación entre la comunicación inalámbrica XBee con una llamada telefónica. Si tú levantas el teléfono y llamas a uno de tus amigos; aquí está la analogía como se ve en el mundo de XBee:

- El que llama (usted) es el XBee que está enviando datos.
- La persona que recibe la llamada (su amigo) es el módulo de destino. La línea telefónica es el enlace inalámbrico establecido entre los dos XBee.
- Y la "conversación" entre usted y su amigo es la información transmitida.

La "línea telefónica" en la comunicación inalámbrica XBee se establece por el aire. Los módulos transmiten y reciben información a través de la modulación de ondas en el espectro electromagnético. Para transmitir datos de un XBee a otro, ambos módulos deben estar en la misma frecuencia y en la misma red. Esto está determinado por el valor de dos parámetros:

1. El canal (CH) le dice al XBee la frecuencia que debe usar para comunicarse; es decir, el canal dentro de su red. Como cuando sintonizas una radio para buscar tu estación favorita, XBee en una red debe usar el mismo canal para "escuchar" el resto.

2. El identificador de red de área personal (ID) dirige a los XBee para que hablen entre sí estableciendo, a través de este identificador único, que todos son parte de la misma red.

Modos de funcionamiento

Los dispositivos XBee pueden usar su conexión serie local de maneras muy diferentes. El "modo de operación" establece la forma en que el dispositivo host se comunica con un módulo XBee a través de la interfaz en serie.

Los módulos XBee admiten dos modos de funcionamiento diferentes:

- Aplicación transparente ("modo transparente")
- Interfaz de programación de aplicaciones ("modo API")

Modo de funcionamiento transparente

Este modo se llama "transparente" porque la radio pasa la información exactamente como la recibe. Todos los datos en serie recibidos por el módulo de radio se envían de forma inalámbrica a un módulo XBee de destino remoto. Cuando el otro módulo recibe los datos, se envía a través del puerto serie exactamente como se recibió. El modo transparente tiene una funcionalidad limitada pero es una manera fácil de comenzar con los dispositivos XBee.

Modo de funcionamiento API

El modo de funcionamiento de la interfaz de programación de aplicaciones (API) es una alternativa al modo transparente. En el modo API, un protocolo determina la forma en que se intercambia la información. Los datos se comunican en paquetes (comúnmente llamados marcos o frames de API). Este modo le permite formar redes más grandes y es más apropiado para crear redes de sensores para realizar tareas tales como recopilar datos de múltiples ubicaciones, controlar dispositivos de forma remota o automatizar su hogar.

Modo de funcionamiento de la API	Modo de funcionamiento transparente
<p>Cuándo usar: Envía datos inalámbricos a múltiples destinos. Configura dispositivos XBee remotos en la red. Recibe paquetes de datos inalámbricos desde múltiples dispositivos Xbee, y la aplicación necesita identificar qué dispositivos envían cada paquete. Recibe muestras de E/S de dispositivos Xbee remotos. Debe admitir múltiples puntos finales, clústeres y / o perfiles (para módulos ZigBee). Utiliza los servicios ZigBee Device Object (ZDO) (para Módulos ZigBee).</p>	<p>Cuándo usar: Las condiciones para usar el modo API no aplican.</p>

<p>Ventajas: Puede establecer o leer la configuración de dispositivos Xbee remotos en la red. Puede transmitir datos a uno o múltiples destinos; esto es mucho más rápido que el modo transparente donde la configuración debe actualizarse para establecer un Nuevo destino. Los datos recibidos incluyen la dirección del remitente. Los datos recibidos incluyen detalles de transmisión y razones para el éxito o el fracaso Varias funciones avanzadas, como diagnósticos avanzados de red y actualizaciones de firmware.</p>	<p>Ventajas: Proporciona una interfaz simple que hace que sea fácil comenzar con dispositivos Xbee. Fácil para una aplicación para apoyo; lo que envía es exactamente qué otros módulos obtienen, y viceversa. Funciona muy bien para comunicación bidireccional entre dispositivos Xbee.</p>
<p>Desventajas: La interfaz es más compleja; los datos están estructurados en paquetes con un formato específico. Más difícil de mantener; las transmisiones son estructurado en paquetes que necesitan ser analizados (para obtener datos) o creado (para transmitir datos). Los datos enviados y los recibidos no son idénticos; paquetes recibidos incluyen algunos datos de control e información extra.</p>	<p>Desventajas: No se puede configurar o leer la configuración de dispositivos Xbee remotos en la red. Primero debe actualizar la configuración para establecer un nuevo destino y transmitir datos. No se puede identificar la fuente de datos recibidos, ya que no incluye la dirección del remitente. Los datos recibidos no incluyen detalles de la transmisión o las razones para el éxito o el fracaso No ofrece el características avanzadas del modo API, incluido diagnóstico de redes avanzadas y actualizaciones de firmware</p>

Tabla 5 – Ventajas y desventajas entre modo API y transparente

Seguridad y cifrado

En ocasiones, la información transmitida en una red XBee necesita protección. Por ejemplo, una red XBee que transfiera información financiera debe estar cuidadosamente protegida contra agentes externos. Los módulos XBee se pueden configurar para una comunicación segura mediante claves de cifrado.

Cuando habilita la seguridad en un dispositivo, la información que transmite se cifra antes de enviarla. Del mismo modo, la información que reciba primero debe descifrarse para que sea legible. Este proceso de cifrado / descifrado puede generar un ligero aumento tanto en la latencia como en el tamaño del paquete.

Xbee Explorer USB:

El XBee explorer USB permite conectar y utilizar cualquier módulo XBee directamente mediante un puerto USB. Se debe conectar un módulo XBee, conectar un cable mini USB al PC y tendremos acceso a los pines TX/RX del XBee y estará listo para funcionar.

Es ideal para establecer una base inalámbrica desde un ordenador y así poder conectar sin cables a una placa que utilice un módulo XBee. Esto nos permite comunicarnos con nuestro Xbee de manera muy sencilla.

Se utilizó uno como el siguiente:



Figura 7 – Xbee Explorer USB

Para poder programar el módulo Xbee a través del Xbee Explorer USB se debe utilizar un programa llamado XCTU que es propiedad de Digi.

XCTU es una aplicación gratuita multiplataforma diseñada para permitir a los desarrolladores interactuar con módulos Digi RF a través de una interfaz gráfica fácil de usar. Incluye nuevas herramientas que facilitan la configuración, configuración y prueba de los módulos XBee RF.

XCTU incluye todas las herramientas que un desarrollador necesita para comenzar a trabajar rápidamente con XBee. Funciones únicas como la vista de red gráfica, que representa gráficamente la red XBee junto con la potencia de la señal de cada conexión, y el generador de cuadros XBee API, que ayuda a construir e interpretar marcos API para XBees que se usan en modo API, se combinan para hacer el desarrollo en la plataforma XBee es más fácil que nunca.

Un ejemplo de la ventana principal se ve a continuación:

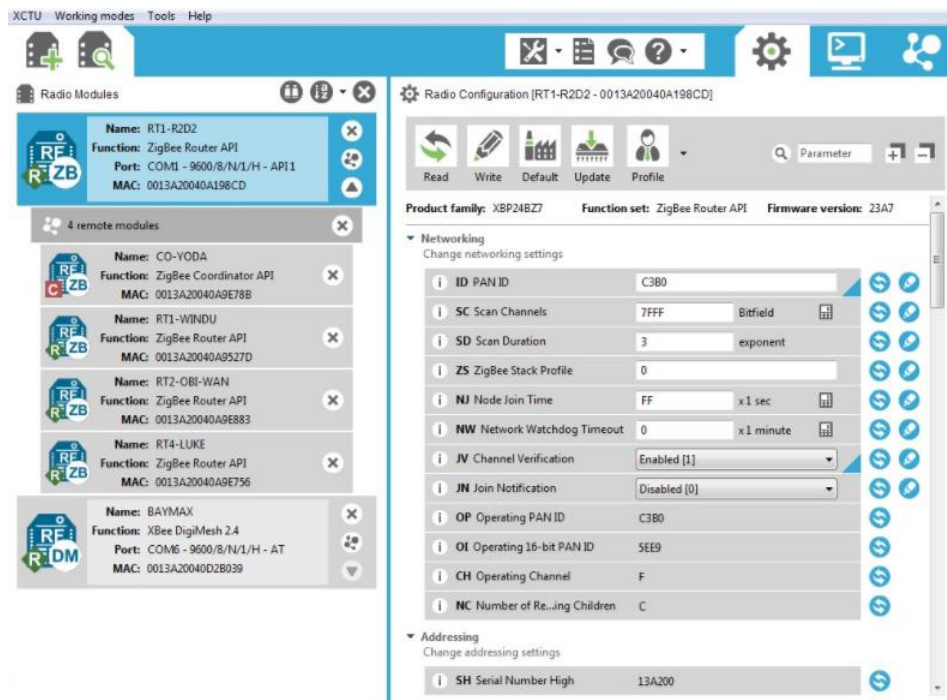


Figura 8 – XCTU

Arduino Uno R3

El Arduino es una plataforma computacional física open-source basada en una simple tarjeta de I/O y un entorno de desarrollo que implementa el lenguaje Processing/Wiring. El Arduino Uno R3 puede ser utilizado para desarrollar objetos interactivos o puede ser conectado a software de tu computadora (por ejemplo, Flash, Processing, MaxMSP). El IDE open-source puede ser descargado gratuitamente (actualmente para Mac OS X, Windows y Linux).

Características:

- Microcontrolador ATmega328.
- Voltaje de entrada 7-12 [V].
- 14 pines digitales de I/O (6 salidas PWM).
- 6 entradas análogas.
- 32k de memoria Flash.
- Reloj de 16 [MHz] de velocidad.

El arduino uno R3 que se utilizo es como el que sigue:



Figura 9 – Arduino Uno R3

Para la conexión del Arduino UNO con el módulo Xbee S2C se utilizó un adaptador que se llama Xbee Shield para Arduino Uno R3.

Este XBee Shield es un adaptador de USB a serial equipado con headers especiales para XBee (20pin 2.0 [mm]). Cuenta con el IC FT232RL integrado, puede ser usado para programar o comunicarse con tarjetas como Arduino pero sin interfaz USB. Por otro lado puedes conectar tu PC a varias aplicaciones inalámbricas a través de los módulos XBee compatibles.

Especificaciones:

- Salida de energía dual de 3.3 [V] y 5 [V]
- IO compatible de 3.3 [V] y 5 [V]
- Protocolo USB 2.0
- Modo BitBang listo
- Fácilmente conectable a una PC a través de cable mini USB
- La configuración XBee soporta software X-CTU
- Permite que la PC se conecte directamente vía cable USB con adaptadores XBee

Una imagen de uno como el que utilizamos en este proyecto sería como se ve a continuación:

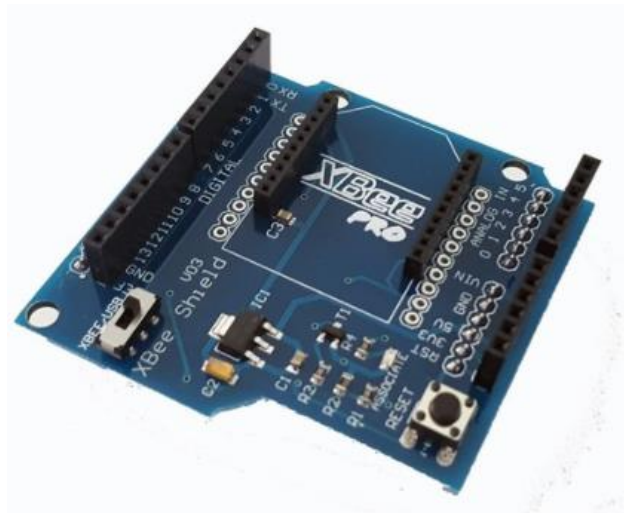


Figura 10 – Xbee Shield para Arduino

El entorno de desarrollo integrado también llamado IDE (sigla en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios lenguajes. En nuestro caso se utilizó la programación en C.

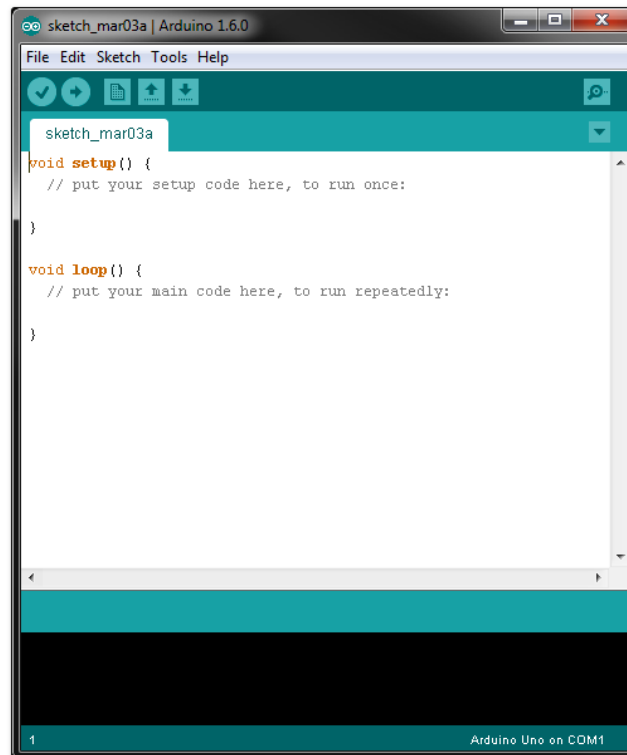


Figura 11 – IDE de Arduino

Los fundamentos por los cuales se utilizó Arduino son los siguientes:

Podríamos mencionar que hay soluciones comerciales que facilitan el trabajo de programar un micro controlador y poder interactuar con ellos como podría ser Parallax Basic Stamp, BX-24 de Netmedia, Phidgets o HandyBoard del MIT, por citar algunos. Arduino además de simplificar este proceso intenta ofrecer otras ventajas:

- Asequible: las placas Arduino son más asequibles comparadas con otras plataformas de micro controladores. La versión más cara de un módulo de Arduino puede ser montada a mano e incluso ya armada cuesta aproximadamente \$2400.
- Multi plataforma: el software Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los entornos para micro controladores están limitados a Windows.
- Entorno de programación simple y directo: el entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados.
- Software ampliable y código abierto: el software Arduino está publicado bajo una licencia libre, y preparado para ser ampliado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C/C++, y si está interesado en profundizar en los detalles técnicos se puede dar el salto a la programación en el lenguaje C, en el que está basado. De igual modo se puede añadir directamente código en C en los programas.
- Hardware ampliable y de código abierto: Arduino está basado en los micro controladores ATMEGA 168, ATMEGA 328, y ATMEGA 1280. Los planos de los módulos están publicados bajo licencia de Creative Commons, por lo que los diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. Incluso usuarios relativamente inexpertos pueden contruir la versión para placa de desarrollo para entender cómo funciona y ahorrar algo de dinero.

Librerías

Cuando se tuvo que escribir una aplicación para que permita que un dispositivo inteligente monitoree y administre una red Xbee, se escribió un código propio para trabajar con el modo API, y también se aprovecharon las bibliotecas de software existentes que ya analizan los marcos de la API. Dependiendo del lenguaje de programación preferido y del dispositivo inteligente conectado a la interfaz en serie del XBee, se puede elegir entre una variedad de bibliotecas disponibles:

1. XBee mbed Library es una extensión de mbed lista para importar para desarrollar proyectos XBee en las plataformas mbed.
2. La biblioteca Digi XBee Ansi C es una colección de código ANSI C portátil para comunicarse con los módulos XBee en modo API.
3. XBee-arduino es una biblioteca Arduino para comunicarse con XBees en modo API.
4. XBee Java Library es una biblioteca fácil de usar desarrollada en Java que le permite interactuar con módulos XBee que funcionan en modo API.

En nuestro caso se utilizaron las librerías de Xbee-arduino para desarrollar el programa que genere la trama API y analice los demás datos que se verán más adelante.

Previsiones a desarrollar en este proyecto

La idea fue programar con el Xbee explorer USB conectado al módulo Xbee S2C el coordinador de la red mediante XCTU. Una vez hecho esto se conectó el Xbee coordinador al Arduino UNO mediante el Xbee Shield para Arduino y en con el IDE de Arduino se programó una secuencia para que explore e investigue si hay dispositivos disponibles, en caso de haberlo lo configura para que sea un router y transmita información al coordinador. Es por esto que además del coordinador se tienen dos dispositivos Xbee S2C que se conectaron al Arduino UNO mediante el Shield Xbee para Arduino y se hizo que el coordinador configure vía remota (RF) los dos dispositivos que reconoció. Una vez que se configuraron espera los datos que cada uno le envió y el coordinador los procesa y los muestra a través de la comunicación serie con el IDE de Arduino.

También se hizo una prueba de rango real entre un coordinador y un router con cifrado de los paquetes enviado por RF y utilizando el modo API en una topología determinada como mesh, esto se hizo para saber cuáles son los límites de distancia reales de una red que será exigida en el campo real en un interior y en un exterior.

Prueba de rango

En la primera prueba de rango que se realizó no se tuvo en cuenta la seguridad de la red, esto quiere decir que el alcance de los radios será mayor que el que se utilice en la implementación, ya que al agregarle la seguridad se tendrá una reducción de la distancia real en la comunicación.

Además no se está generando un bucle cerrado sino que es solo unidireccional la comunicación, lo que no es específicamente como debe realizarse la prueba de rango, pero es importante saber estimativamente cuál es la distancia máxima a la que llegar la comunicación con línea de vista y ambas en el modo API 1.

A continuación se verá un diagrama de flujo de cómo es el mecanismo de transmisión y recepción de señal para saber si minimamente hay comunicación:



Figura 12 – Diagrama de flujo, Programa de prueba de Rango

Se debe tener en cuenta que es difícil encontrar una zona en la ciudad donde se puedan probar los dispositivos con línea de vista y una distancia mayor a 1 Km. La distancia a la que se llegó es de 690 metros. Se puede ver una imagen de Google Earth en donde se hizo la prueba.

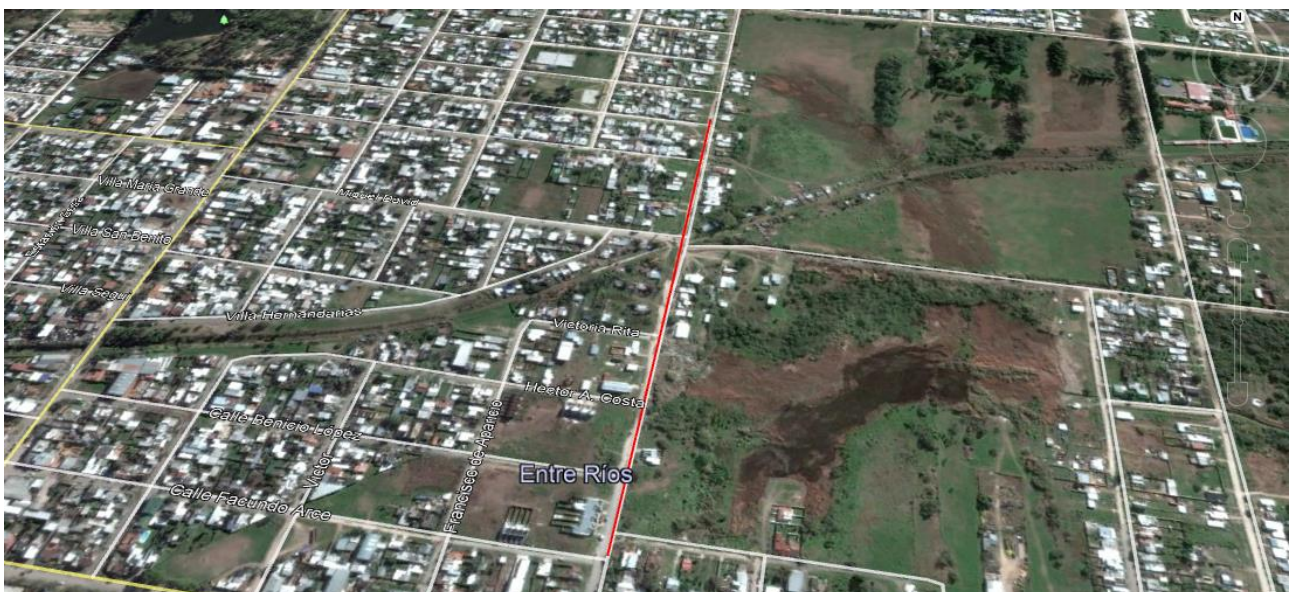


Figura 13 – Google Earth – Distancia de Prueba de Rango

En realidad la distancia puede ser mayor, pero al no tener línea de vista en este punto, y además haber recorrido más de 20 luminarias en línea recta, fue suficiente para el trabajo que debe realizar el coordinador de la red.

En la segunda prueba de rango lo que se hizo fue incorporar la seguridad de la red, como veremos a continuación en una imagen del programa XCTU, donde se preconfiguro las contraseñas del coordinador y la contraseña KY del router. En el router no se configura la NK ya que depende del coordinador:

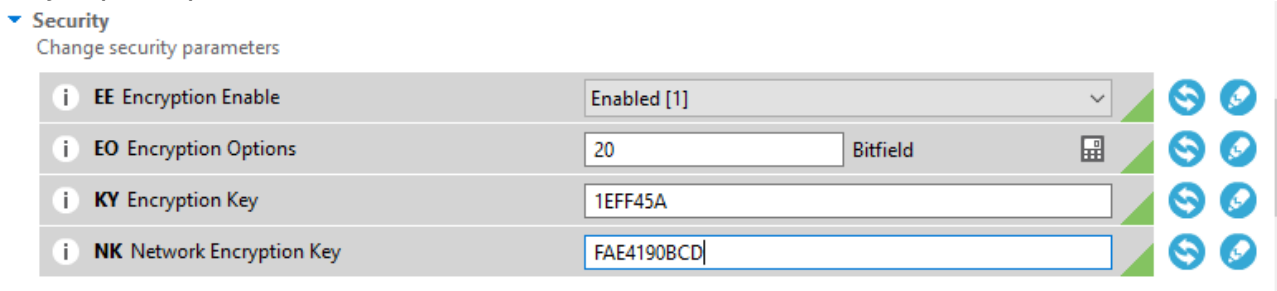


Figura 14 – Configuración de la Seguridad del Xbee mediante XCTU

Esta imagen muestra las contraseñas utilizadas y la habilitación de la encriptación. Para realizar la prueba de rango, se utilizó el software de ZigBee, XCTU. Este programa tiene la opción de hacer su propia prueba de rango, enviando información y viendo si tiene respuesta. Además lee el RSSI del dispositivo local y el remoto y lo imprime en la pantalla.

De aquí que se puede ver la imagen a continuación de la degradación de la señal a medida que nos alejamos de dispositivo local y luego volvemos a acercarnos:

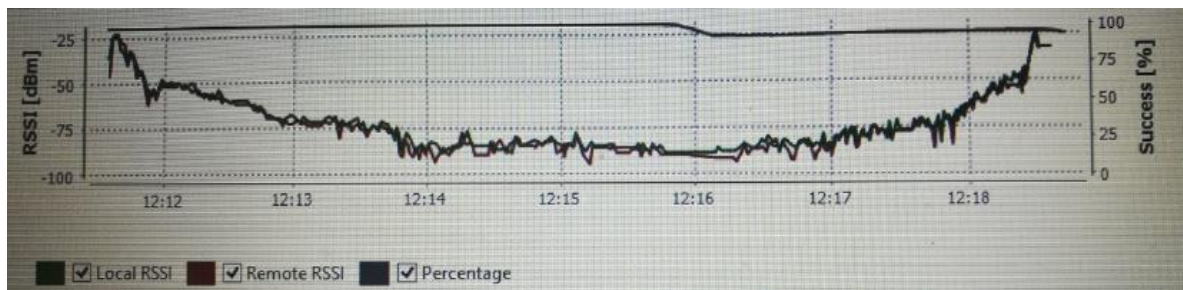


Figura 15 – Grafica de Potencia de Señal Recibida

Además cuenta cuantos paquetes se enviaron y cuantos de estos tienen error. Cuando los paquetes empezaron a perderse, se determinó la distancia máxima:

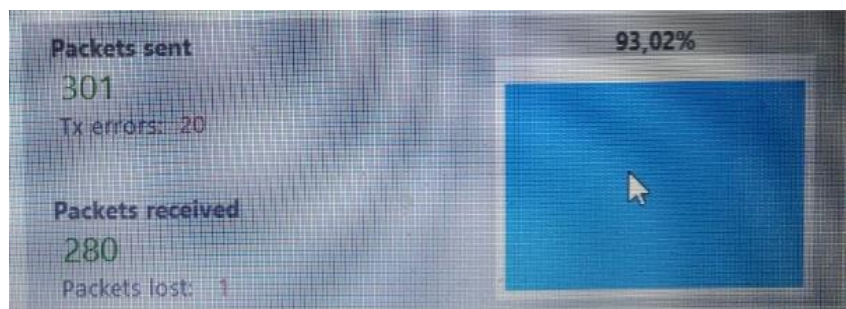


Figura 16 – Muestra de paquetes erróneos y perdidos en la prueba de Rango

El valor en distancia a la que llego la comunicación es de 210 metros.
Se puede ver en Google Earth cual fue el recorrido de la prueba de rango:

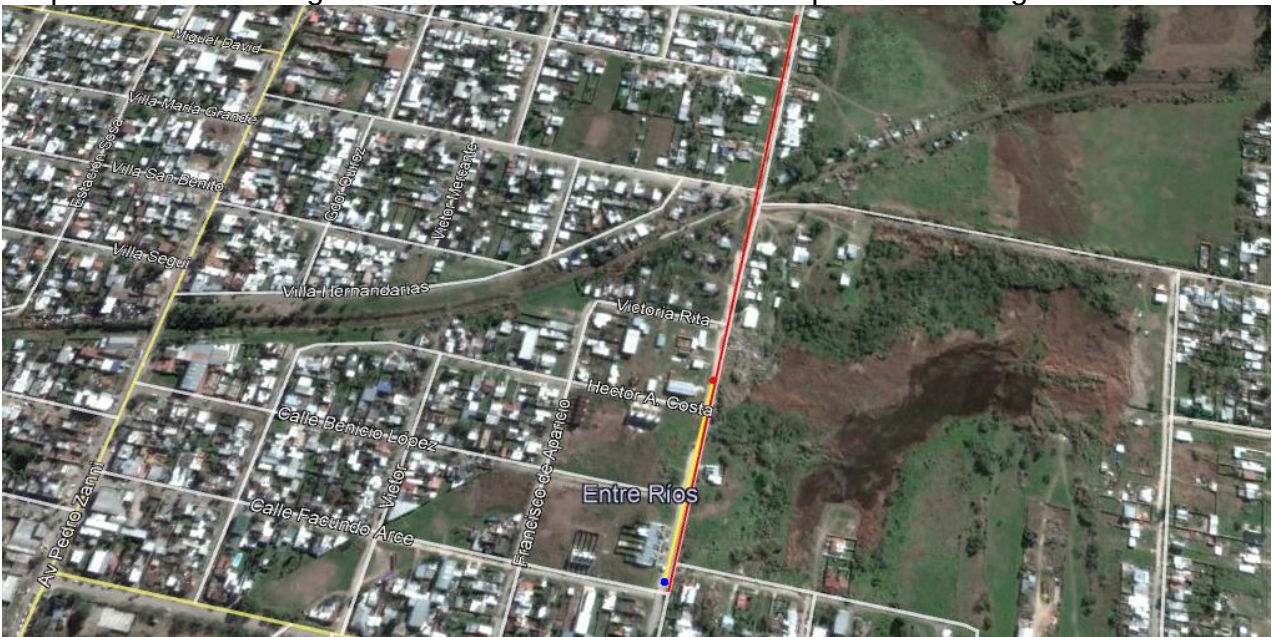


Figura 17 – Distancia en Google Earth con y sin seguridad

En amarillo se observa que la distancia es mucho menor con respecto a la roja. La misma es la distancia maxima a la que se llego con cifrado seguro, mientras que la roja fue sin seguridad.

Se estimo que el alcance en linea de vista a la altura real de la luminaria sera mayor, de aproximadamente 300 [m] pero con lluvia (que no se probó), se cree que el alcance puede decrecer a 200 [m], por lo tanto parecio interesante tomar 210 [m] como lo tipico. Esto servira para saber si es coherente el uso de este tipo de dispositivo, ya que el coordinador debe abarcar por lo menos 14 dispositivos en un radio de 210 [m].

Prueba de la red

Las pruebas realizadas para evaluar la forma en que se programaron los módulos Xbee se debió realizar disminuyendo el valor de la potencia transmitida al mínimo posible por cada módulo, ya que sino el área de prueba de tres módulos seria casi de 1 kilómetro. Es por esto que se disminuyó la potencia y se agregaron obstáculos metálicos para saber:

- 1- Si la red era una malla
- 2- Si al cortarse un camino los mensajes pueden llegar de un dispositivo a otro mediante saltos entre los router de una red.

Para esto se programaron dos módulos como router, y uno como coordinador. Los dos router deben enviar información al módulo coordinador, para esto se pone una dirección especifica de 64 bits que son todos cero, esto indica que la información debe ir si o si al coordinador.

Por otro lado el coordinador de la red debe recibir los datos, que eran valores tomados del ADC de cada módulo Arduino y debe identificar que Xbee lo envió mediante un estudio de la trama recibida, separando y mostrando la MAC del dispositivo que envía los datos, e imprimiéndolos en pantalla para que el operador pueda saber que módulo envió la información y que valor le entrego.

Para poder hacer todo esto, primero se evaluó la red en un lugar pequeño, con los módulos cerca y sin interferencia. En este caso la red debía comunicarse como una malla donde todos se comunican con todos, esto sucedió así y lo pudimos observar claramente

en el XCTU. Ya que el mismo tiene una opción para estudiar la forma en que está conectada la red. Así que lo que vemos en el XCTU es lo siguiente:

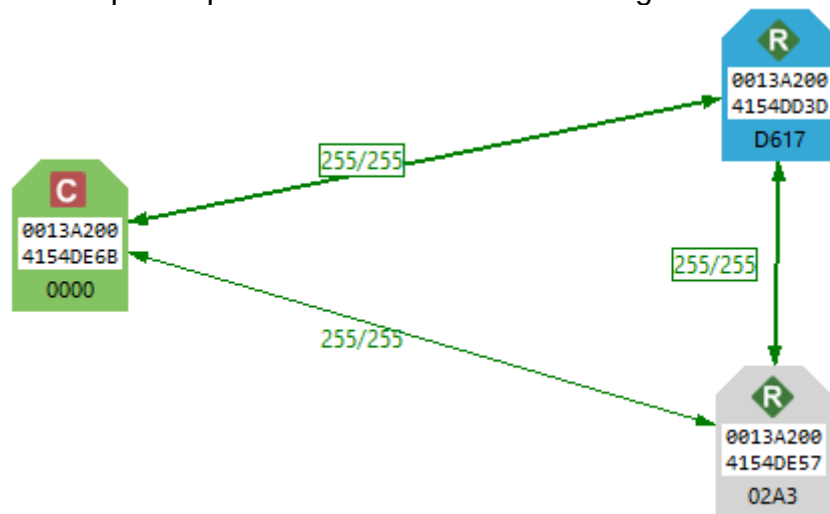


Figura 18 – Grafico de XCTU de la Red

En los módulos descubiertos se pueden observar los nombres de cada uno así como la dirección MAC:

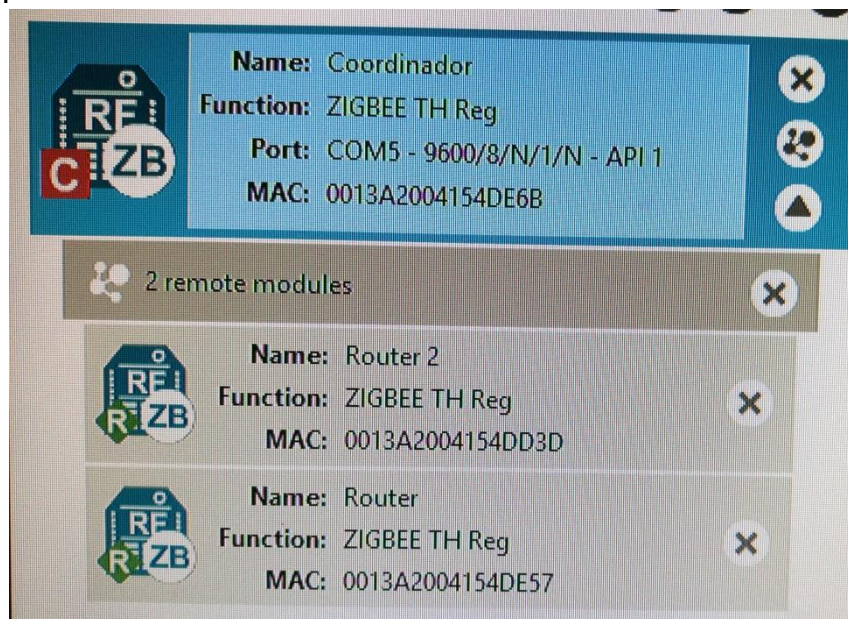


Figura 19 – Dispositivos que XCTU reconoció mediante RF

Luego de verificar este tipo de red se decidió conectar el coordinador al módulo Arduino para ver como llegaban las tramas y mostraba los valores por pantalla.

Esto se verifico rápidamente ya que es lo que se venía probando ampliamente.

Luego se disminuyó la potencia de los tres módulos al mínimo, esto se hace por medio de los atributos:

- PL (TX Power Level): tiene valor 0 que indica -5 [dBm].
- PM (Power Mode): esta deshabilitado, lo que quiere decir que no aumenta la sensibilidad de recepción ni la potencia de transmisión.

Una vez configurado esto, se distribuyeron los módulos en un espacio grande y con paredes de cemento y estructuras metálicas de por medio, esto, junto a otras placas metálicas sirvieron para evitar la comunicación en forma de red, y poder crear una comunicación "lineal". En el XCTU se observó lo siguiente:



Figura 20 – Dispositivos que XCTU de la nueva Red

Una vez que se logró esta configuración se decidió conectar el modulo coordinador al Arduino y estudiar si los datos de ambos router llegaban a destino, esto se comprobó al ver la impresión en pantalla de los datos de ambos router recogidos por el coordinador en esta red:

```
Starting up!
013A20 4154DE57 - Dato: 1 2C
013A20 4154DD3D - Dato: 0 C6
013A20 4154DE57 - Dato: 1 2B
013A20 4154DD3D - Dato: 0 CA
```

Figura 21 – Recepción de datos en Arduino

Gracias a la imagen anterior y a la imagen donde se muestra la red con la dirección MAC de cada dispositivo se puede observar que la MAC de ambos routers logra llegar a destino. La red se programó para que pudieran hacerse aproximadamente 8 saltos, siendo en este caso probado que un salto es posible por lo tanto la programación de los saltos está bien lograda.

Adquisición de datos y fabricación de prototipo

En esta etapa se trabajó en torno a la obtención de los datos de cada luminaria (consumo, tensión, corriente y estado de la luminaria).

Antes de comenzar es importante aclarar las bases teóricas para la adquisición de datos. (Pallas Areny, 2003) Se denomina normalmente transductor o sensor a todo dispositivo que convierte una señal de una forma física en una señal correspondiente pero de otra forma física distinta. Es, por tanto, un dispositivo que convierte un tipo de energía en otro. (Mora , 2011) En la actualidad, la mayor parte de los sensores:

- Generan una salida en tensión o corriente, o bien,
- Modifican una propiedad que puede ser evaluada de forma eléctrica.

De esta manera, y con el debido acondicionamiento, la señal de salida puede ser tratada por un equipo automático de adquisición de datos.

Las señales del mundo real son, en general, analógicas y varían de manera continua en el tiempo, para que un computador sea capaz de procesarla se debe convertir a datos digitales.

Cada uno de estos sensores tiene unas características propias y genera una tensión o intensidad determinada, por lo que estas señales tienen que ser adaptadas para ser tratadas posteriormente.

Para obtener datos digitales a partir de señales analógicas, la señal debe ser muestreada. Esto significa tomar el valor instantáneo de la señal en un momento determinado. Para

una señal continua, las muestras se toman a intervalos regulares, generalmente con un periodo de muestreo fijo entre medidas.

Para recoger información útil, un factor clave es el ritmo o frecuencia con la que se toman las medidas.

En una aplicación de procesamiento de señal en la que tenemos que muestrear una señal continua ¿cómo sabemos qué frecuencia de muestreo debemos utilizar? El teorema que define la mínima frecuencia requerida para representar de una manera precisa una señal analógica se denomina Teorema de Nyquist.

Teorema de Nyquist

El Teorema de Nyquist indica que la frecuencia de muestreo mínima que tenemos que utilizar debe ser mayor que $2 \cdot f_{\max}$, donde f_{\max} es la frecuencia máxima de la señal. Si utilizamos esa frecuencia de muestreo, podremos reproducir posteriormente la señal a partir de las muestras tomadas.

Pasos para convertir la señal analógica en digital

Un conversor o convertidor de señal analógica a digital (Conversor Analógico Digital, CAD; Analog-to-Digital Converter, ADC) es un dispositivo electrónico capaz de convertir una señal analógica, ya sea de tensión o corriente, en una señal digital mediante un cuantificador y codificándose en muchos casos en un código binario en particular. Donde un código es la representación unívoca de los elementos, en este caso, cada valor numérico binario hace corresponder a un solo valor de tensión o corriente.

Para convertir una señal analógica en digital, el primer paso consiste en realizar un muestreo (sampling) de ésta, o lo que es igual, tomar diferentes muestras de tensiones o voltajes en diferentes puntos de la señal. La frecuencia a la que se realiza el muestreo se denomina razón, tasa o también frecuencia de muestreo y se mide en kilo Hertz [kHz].

Una vez realizado el muestreo, el siguiente paso es la cuantización de la señal analógica. Para esta parte del proceso los valores continuos de la señal se convierten en series de valores numéricos decimales discretos correspondientes a los diferentes niveles o variaciones de voltajes que contiene la señal analógica original.

En la cuantificación de la señal se produce pérdida de la información que no puede ser recuperada en el proceso inverso, es decir, en la conversión de señal digital a analógica, y esto es debido a que se truncan los valores entre 2 niveles de cuantificación, mientras mayor cantidad de bits mayor resolución y por lo tanto menor información pérdida.

Después de realizada la cuantización, los valores de las tomas de voltajes se representan numéricamente por medio de códigos y estándares previamente establecidos. Lo más común es codificar la señal digital en código numérico binario.

Para trabajar sobre el consumo de la luminaria es imperiosa la necesidad de sensar tanto la tensión como la corriente del driver LED, así como también la fase de ambos a fin de determinar que potencia es activa y que parte es reactiva.

Para esto se necesita un sensor de efecto hall o transformador de corriente y una adaptación de la señal de tensión para medirla desde el dispositivo medidor, en nuestro caso Arduino.

Sensor de efecto Hall

(Pallas Areny, 2003) El efecto Hall fue descubierto por E. H. Hall en 1879 y consiste en la aparición de una diferencia de potencial transversal en un conductor o semiconductor, por el que circula corriente, cuando hay un campo magnético aplicado en dirección perpendicular a esta.

En la práctica la tensión Hall depende de otros factores como son la tensión mecánica o presión y la temperatura.

La dependencia de la presión (efecto piezorresistivo) es un factor a ser considerado sobre todo por el fabricante al encapsular el componente, puesto que para el usuario es fácil adoptar precauciones al respecto.

La temperatura tiene un efecto doble. Por una parte, afecta a la resistencia que presenta el elemento, por lo que si se alimenta a una tensión constante la corriente de polarización, I , variara con la temperatura y con ella la tensión de salida V_H . Por esta razón es preferible alimentar a corriente constante que a tensión constante.

Por otra parte, la temperatura afecta a la movilidad de los portadores mayoritarios, y por lo tanto, a la sensibilidad. Dado que estos dos efectos tienen signo opuesto, es posible su compensación con un circuito adecuado. En cualquier caso es conveniente limitar el valor de la corriente de control para evitar auto calentamientos.

Otra limitación importante en aplicaciones de precisión es la presencia de una tensión de desequilibrio (offset), que es la tensión obtenida con campo magnético nulo, a pesar de tener los electrodos bien centrados en las caras. Se debe a inexactitudes físicas y no uniformidades del material y puede ser de hasta 100 [mV] cuando se alimenta a 12 [V].

Debido a que la tensión generada por este efecto es muy baja (aproximadamente 30 [$\mu\text{V}/\text{G}$]), es necesario implementar un sistema acondicionador de señal para poder ser interpretada por un circuito electrónico o un módulo de control; y con respecto al tipo de acondicionador surgen dos tipos de sensores, los de salida lineal y los de salida digital

Sensores de efecto Hall de salida lineal

Este elemento proporciona una tensión de salida que es proporcional al campo magnético al que está expuesto; cuando se detecta un campo magnético positivo, la salida aumenta por encima de la tensión nula y a la inversa, cuando se detecta un campo magnético negativo, la salida disminuye por debajo de la tensión nula, pero sigue siendo positiva.

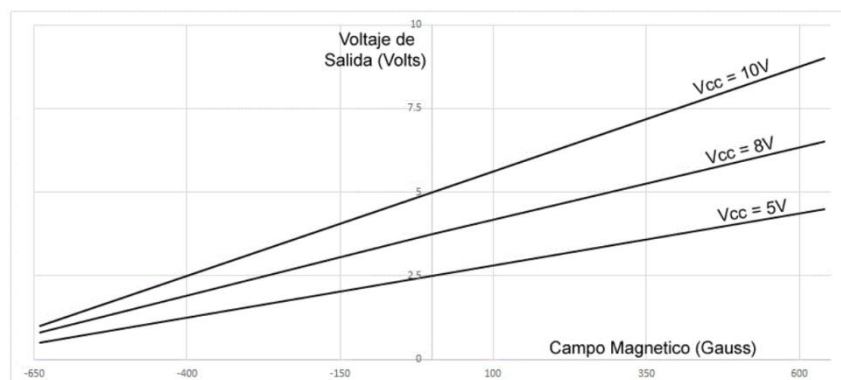


Figura 22 – (Mecanica Web, 2017) Respuesta de sensor de efecto Hall

Básicamente es un sensor que envía una señal de voltaje variable proporcionalmente a la cantidad de campo magnético, este voltaje es procesado por un ADC, dejándolo tomar decisiones en base a la información proporcionada.

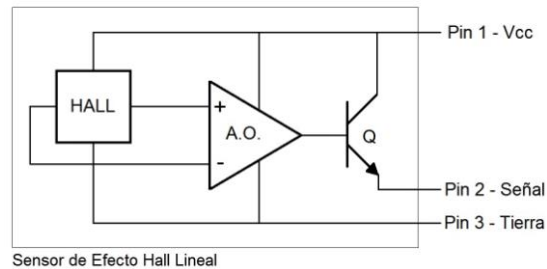


Figura 23 – (Mecanica Web, 2017) Sensor de efecto Hall por dentro

Sensor de efecto Hall de salida digital

La descripción del sensor de salida lineal describe un sensor que emite una señal analógica. En este sensor que se acondiciona para trabajar con dos estados, estado alto y bajo (on / off), agregando un disparador Schmitt se integra una característica extra de histéresis para prevenir ruido y cambios de estado indeseado.

Convertor ADC de Arduino

(Crespo Moreno, 2014) En el caso de un arduino Uno, el valor de 0 voltios analógico es expresado en digital como binario: 000000000 (0) y el valor de 5V analógico es expresado en digital como binario: 111111111 (1023). Por lo tanto todo valor analógico intermedio es expresado con un valor entre 0 y 1023, es decir, sumo 1 en binario cada 4,883 [mV].

Arduino Uno tiene una resolución de 10 bits, es decir, unos valores entre 0 y 1023.

Los microcontroladores de Arduino contienen en la placa un convertor analógico a digital de 6 canales.

También se debe tener cuidado de la frecuencia máxima de trabajo del ADC, este valor se especifica en la ficha técnica y es de 200 [kHz], este es el valor del reloj interno de la circuitería del ADC y se genera dividiendo el reloj principal ATmega, que en el caso del UNO es 16 [MHz], este divisor del reloj se realiza mediante pre-escaladores y sólo hay un rango limitado de valores, por lo que la frecuencia máxima que se puede utilizar y estar dentro de la frecuencia máxima de trabajo es 125 [kHz]. El siguiente pre-escalador supone usar el ADC a 250 [kHz], en este caso no se puede garantizar la resolución de 10 bits, pero si una resolución de 8 bits. De todas formas en caso de necesitar un ADC más rápido se podría usar uno externo.

El ADC puede trabajar en dos modos: single conversión mode y free running mode. En modo single conversión el ADC hace una sola conversión y para, pero en modo free running el ADC está continuamente convirtiendo, es decir, hace una conversión y luego comienza con la siguiente.

El ADC en microcontroladores AVR utiliza una técnica conocida como aproximación sucesiva mediante la comparación de la tensión de entrada con la mitad de la tensión de referencia generada internamente. La comparación continúa dividiendo de nuevo la tensión y actualizando cada bit del registro ADC a 1 si el voltaje es HIGH en la comparación o 0 en el otro caso. Este proceso se realiza 10 veces (por cada bit de resolución del ADC) y genera como resultado la salida binaria.

Conversión de señales del driver

Por lo tanto, para reconstruir con precisión la forma de onda, la velocidad de muestreo f_s debe ser mayor que dos veces el componente de interés de frecuencia más alto en la señal medida. Se desea muestrear cinco veces más alto que la frecuencia de la señal.

Pero eso no es todo, es importante saber el desfase entre la tensión y la corriente además del valor real y preciso tanto de la tensión como de la corriente. Es por esto que se debe muestrear la tensión y la corriente a la vez. Esto quiere decir que el ADC debe estar alternándose entre la toma de señal de corriente y la toma de señal de tensión y ambas almacenarlas en un vector propio para luego ser procesadas.

Haciendo los cálculos se sabe que la frecuencia de red es alterna entre 50 [Hz] y 60 [Hz]. Es necesario, por lo tanto, tomar 60 [Hz] como la frecuencia de trabajo para garantizar el correcto funcionamiento de la conversión.

Además se quiere muestrear por lo menos 5 veces más rápido y las 2 señales a la vez, por lo que la frecuencia de muestreo para cada canal debe ser de:

$$f_s = 2 * F_{max} * 5 * 2 = 2 * 60 * 5 * 2 = 1200 [Hz]$$

Teniendo en cuenta, como se observó anteriormente, que la frecuencia máxima de trabajo del ADC de Arduino es muy superior a este valor, se intentó aumentar esta frecuencia de muestreo hasta ver que esto no comprometa el trabajo del dispositivo pero aumente la calidad de señales de tensión y corriente que deben estudiarse.

Teniendo en cuenta todo esto se empezó a trabajar en la conversión en sí. Como se convierte la señal de salida en una tensión entre 0 [V] y 5 [V] y el sensor 49E tiene el punto medio en 2.5 [V] aproximadamente, se tiene un valor aproximado de salida del ADC de 512. El sensor tiene la siguiente curva:

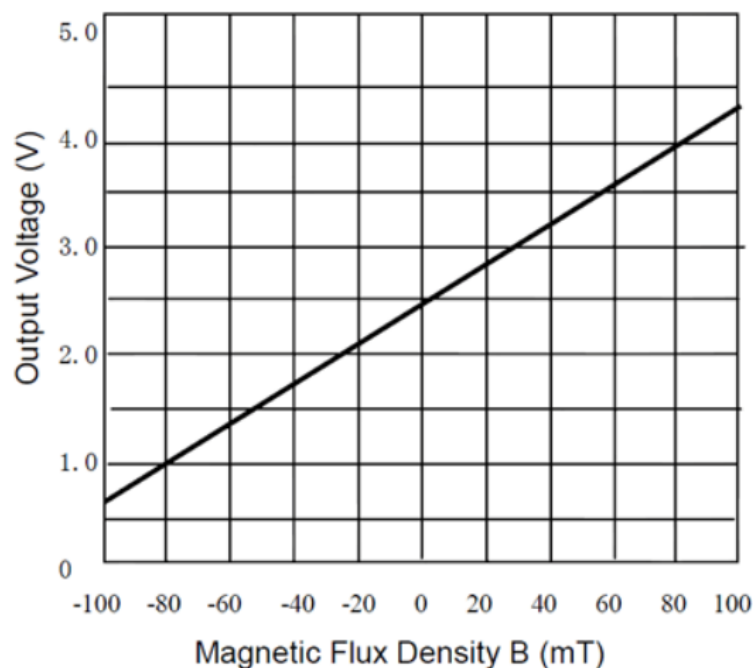


Figura 24 –Curva de Voltaje vs Flujo Magnético

El valor medio del sensor varía de cada sensor en sí. Es por esto que se modificó el punto medio sacando un promedio antes de empezar la conversión.

Una vez que se tiene el valor medio “se centraron” las mediciones venideras en este valor que se da sin campo magnético.

El siguiente paso fue pasar el valor de 0 a 1023 a un valor de tensión, dado por la ecuación:

$$Tension = VALOR_MEDIDO * 5000.0/1024$$

Luego lo se pasó a un valor de campo magnético, el sensor tiene una ecuación de la forma:

$$Campo\ Magnético = 53.33 * Tension - 133.3$$

En este punto es importante mencionar que se calcula la corriente que circula en base al campo magnético generado por el mismo, es por esto que se usara la ley de Biot-Savart.

La ley de Biot-Savart, relaciona los campos magnéticos con las corrientes que los crean. De una manera similar a como la ley de Coulomb relaciona los campos eléctricos con las cargas puntuales que las crean. La obtención del campo magnético resultante de una distribución de corrientes, implica un producto vectorial.

Utilizamos la ley de Biot para calcular el campo magnético B producido por un conductor rectilíneo indefinido por el que circula una corriente de intensidad i.

El campo magnético B producido por el hilo rectilíneo en el punto P tiene una dirección que es perpendicular al plano formado por la corriente rectilínea y el punto P, y sentido el que resulta de la aplicación de la regla de la mano derecha al producto vectorial $u_t \times u_r$

Para calcular el módulo de dicho campo es necesario realizar una integración.

$$B = \frac{\mu_0 * i}{4 * \pi} \int_{-\infty}^{\infty} \frac{\text{sen } \theta}{r^2} dx = \frac{\mu_0 * i}{4 * \pi * R} \int_0^{\pi} \text{sen } \theta d\theta = \frac{\mu_0 * i}{2 * \pi * R}$$

De esta ecuación se puede despejar la corriente, además de agregarle la permeabilidad magnética relativa del cobre $\mu_r = 1388$.

De esta manera se obtiene no solo el valor real de la corriente sino también la forma de onda aproximada de la misma.

Cuando se tiene una potencia de salida aproximada de 200 watts se obtiene una corriente de:

$$i = \frac{200 [W]}{220 [V]} = 0.91 [A]$$

El campo magnético resultante máximo con un cable de radio 1.5mm será de:

$$B = \frac{\mu_0 * i}{2 * \pi * R} = \frac{4 * \pi * 1 * 10^{-7} * 1388 * 0.91}{2 * \pi * 0.0015} = 0.1684 [G]$$

Al estar trabajando en un rango tan pequeño del sensor se tiene un error muy grande que se torna insalvable.

En la práctica se logró determinar no solo los valores picos de corriente, sino que también se graficó la corriente. Pero el error es grande al igual que el procesamiento de la señal para un microcontrolador de 8 bits.

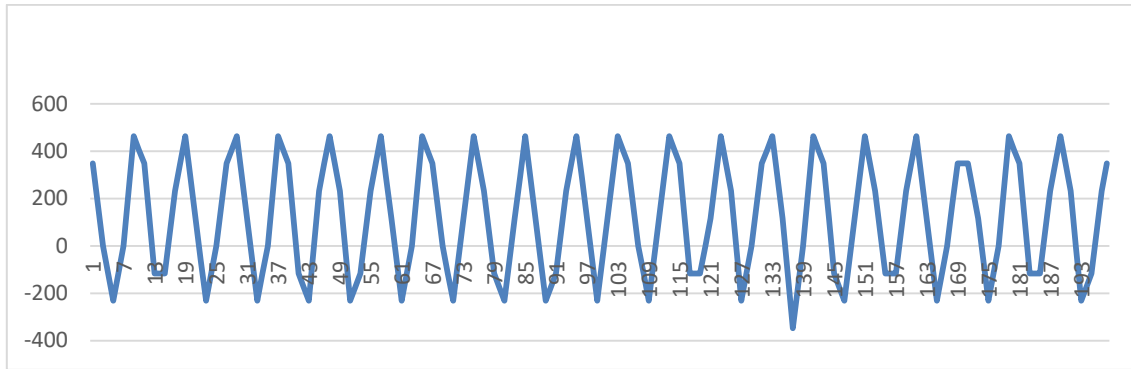


Figura 25 – Gráfico de Arduino luego de la digitalización de corriente

Esta es la corriente sensada de un ventilador que tiene un consumo máximo de 130 [W]. En este caso estaba al máximo, entregaba una corriente de 0.51 [A], y la tensión de alimentación era de 227 [V]. De aquí se supo la potencia que consumía, que era:

$$P = V * I = 227 * 0.51 * \sqrt{2} = 82 [W]$$

De hecho, se logró sensar la tensión y la corriente con bastante precisión, pero la tensión sufre un desfase producto de un transformador reductor que está conectado para reducir y aislar la parte lógica de la parte de potencia. Por otra parte, parte del desfase se produce por la naturaleza inductiva de la máquina que está funcionando, en nuestro caso un motor que es altamente inductivo.

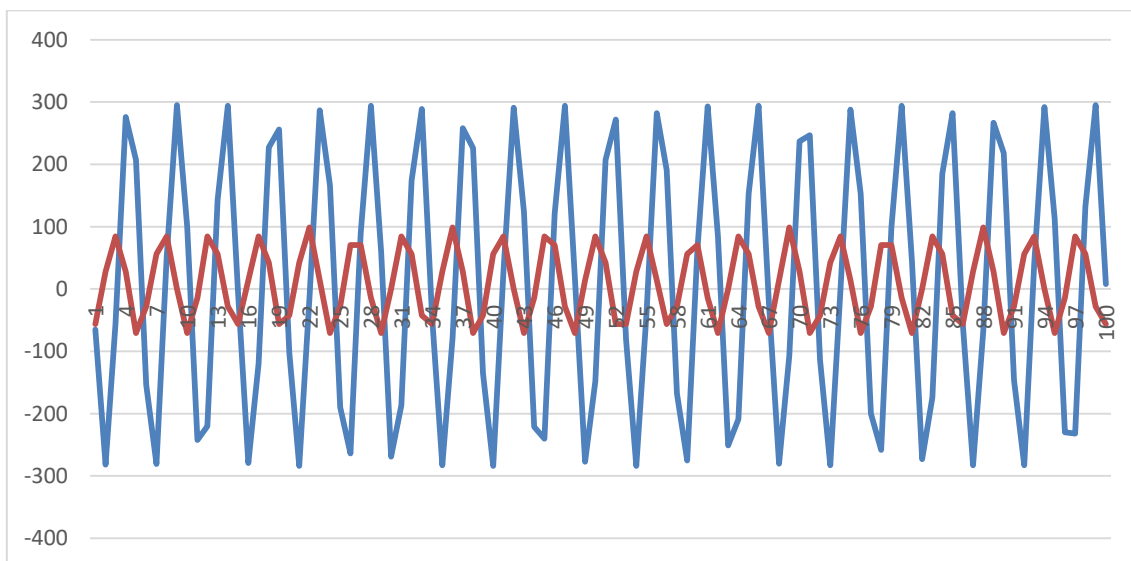


Figura 26 – Gráfico de Arduino luego de la digitalización tensión y corriente

La corriente sensada está en [mA]. Además las formas de onda son bastante deformes ya que no se había logrado mejorar la frecuencia de sensado, que para este gráfico es de $f_s=5000$ [Hz]. Se optó por realizar algunas modificaciones.

Primero se cambió la forma de sensar la corriente. Se pasó de utilizar un sensor de efecto hall 49E, a una pinza que posee un transformador para que sea más sensible a corrientes pequeñas.

La pinza que se utilizó es la SCT-013-000:



Figura 27 – Imagen de la pinza SCT-013

(Llamas, 2017) La familia SCT-013 son sensores de corrientes no invasivos que permiten medir la intensidad que atraviesa un conductor sin necesidad de cortar o modificar el conductor. Se pueden emplear estos sensores con un procesador como Arduino para medir la intensidad o potencia consumida por una carga.

Los sensores SCT-013 son transformadores de corriente, dispositivos de instrumentación que proporcionan una medición proporcional a la intensidad que atraviesa un circuito. La medición se realiza por inducción electromagnética.

Los sensores SCT-013 disponen de un núcleo ferromagnético partido (como una pinza) que permite abrirlo para arrollar un conductor de una instalación eléctrica sin necesidad de cortarlo.

Dentro de la familia SCT-013 existen modelos que proporcionan la medición como una salida de intensidad o de tensión.

Hay 10 modelos diferentes para estas pinzas. Se eligió la única que transforma corriente en corriente, donde 100 [A] sobre el cable sensado nos devuelve 50 [mA] a la salida del sensor.

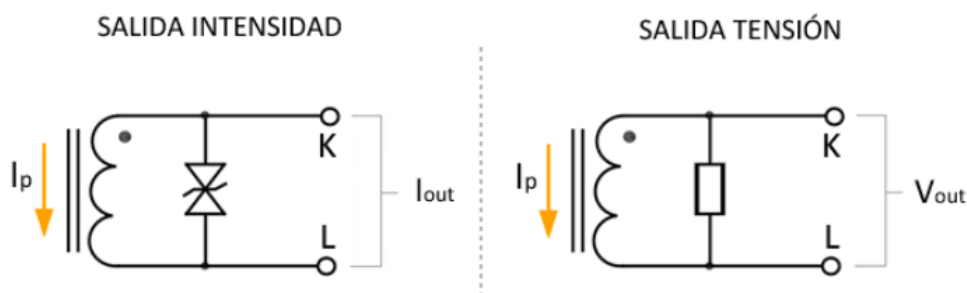


Figura 28 – Circuito interno de la Pinza SCT-013

La precisión del sensor puede ser de 1-2 [%], pero para ello es muy importante que el núcleo ferromagnético se cierre adecuadamente. Hasta un pequeño hueco de aire puede introducir desviaciones del 10 [%].

Como desventaja, al ser una carga inductiva, el SCT-013 introduce una variación del ángulo de fase cuyo valor es función de la carga que lo atraviesa, pudiendo llegar a ser de hasta 3°.

Los transformadores de intensidad son componentes frecuentes en el mundo industrial y en distribución eléctrica, permitiendo monitorizar el consumo de puntos de consumo

donde sería imposible otro tipo de medición. También forman parte de múltiples instrumentos de medición, incluso en equipos portátiles como en pinzas perimétricas o analizadores de red.

La familia de sensores SCT-013 son pequeños transformadores de corriente o CT (Current transformer). Los transformadores de corriente son instrumentos ampliamente empleados como elementos de medición.

Un transformador de corriente es similar a un transformador de tensión y está basado en los mismos principios de funcionamiento (de hecho, formalmente es idéntico). Sin embargo persiguen objetivos diferentes y, en consecuencia, están diseñados y construidos de forma distinta.

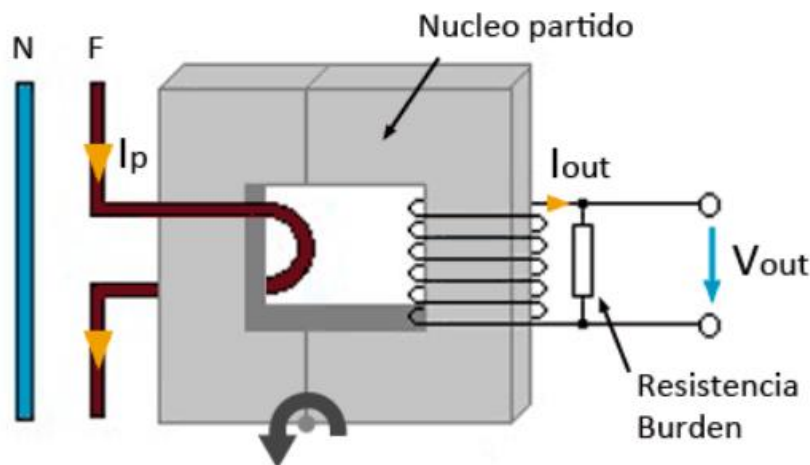


Figura 29 – Diseño de la pinza de corriente

Se puede emplear esto para construir un sensor de corriente no invasivo. En un sensor de corriente el núcleo ferromagnético puede estar dividido de forma que pueda abrirse y arrollar un conductor.

De esta forma, se tiene un transformador en el que:

- El cable por el que circula la intensidad a medir constituye un devanado primario
- La “pinza” es el núcleo magnético
- El devanado secundario está integrado como parte de la sonda.

Cuando la corriente alterna circula por el conductor se genera un flujo magnético en el núcleo ferromagnético, que a su vez genera una corriente eléctrica en el devanado secundario.

La relación de transformación de intensidad depende de la relación entre el número de espiras.

$$\frac{I_s}{I_p} = \frac{V_p}{V_s} = \frac{N_p}{N_s}$$

El primario generalmente está formado por una única espira formada por el conductor a medir. Aunque es posible enrollar el conductor haciendo que pase más de una vez por el interior de la “pinza”. El número de espiras del secundario, integrado en la sonda, varía 1000-2000 según modelos del SCT-013.

A diferencia de los transformadores de tensión, en un transformador de intensidad el circuito secundario nunca debería estar abierto, porque las corrientes inducidas podrían

llegar a dañar el componente. Por ese motivo, los sensores de SCT-130 disponen de protecciones (resistencia burden en los sensores de salida por tensión, o diodos de protección en los sensores de salida por corriente).

Excepto el modelo SCT-013-000, también llamado SCT-013-100, todos los demás modelos tienen una resistencia de burden interna para que la salida sea una señal de tensión de 1 [V]. Por lo cual ni siquiera tendremos que preocuparnos por ello.

Únicamente en el caso del SCT-013-100, carece de resistencia burden interna, por lo que la salida es una señal de $\pm 50\text{mA}$. Una resistencia de 33Ω en paralelo con el sensor será suficiente.

Otro problema que tenemos que resolver es que estamos midiendo corriente alterna, y la intensidad inducida en el secundario es igualmente alterna. Tras el paso por la resistencia burden (interna o externa) la salida de tensión también es alterna.

Se añadió un offset en DC mediante el uso de dos resistencias y un condensador que proporcionen un punto medio entre GND y V_{cc} . Mucho mejor si además se añade un amplificador operacional como seguidor de tensión.

Se debe tener en cuenta que al hablar de tensión alterna normalmente se emplean valores RMS. Recordando las ecuaciones de tensión pico, y pico a pico.

$$V_{pico} = \sqrt{2} \cdot V_{RMS} = R \cdot I_{pico} = \sqrt{2} \cdot R \cdot I_{RMS}$$

$$V_{picoapico} = 2 \cdot V_{pico} = 2 \cdot \sqrt{2} \cdot V_{RMS} = R \cdot I_{picoapico} = 2 \cdot \sqrt{2} \cdot R \cdot I_{RMS}$$

Se utilizó una Resistencia de 330 [ohm] , ya que no se sensaron corrientes mayores a los 10 [A] que sería (por regla de tres) 5 [mA] de corriente de salida. Lo que se tuvo fue:

$$V_{picoapico} = 2 \cdot \sqrt{2} \cdot 330 \cdot 0.005 = 4.66\text{ [V]}$$

Además se tratara de centrarlo en 2.5 [V] , lo que daría una señal entre 4.83 [V] y 0.17 [V] .

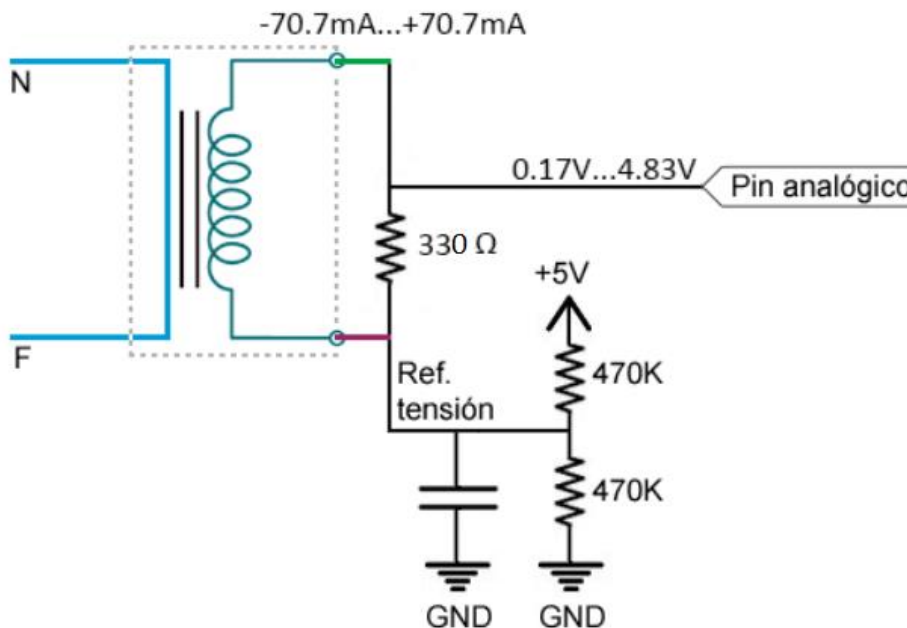


Figura 30 – Circuito para referenciar la señal

Luego del montaje del circuito se probó el sensado de corriente con el mismo dispositivo de antes, un ventilador. Además se logró mejorar la frecuencia de muestreo del ADC del Arduino UNO, pasando de una frecuencia de 500 [Hz] a una frecuencia aproximada de 20000 [Hz]. Lo que se observó fue lo siguiente:

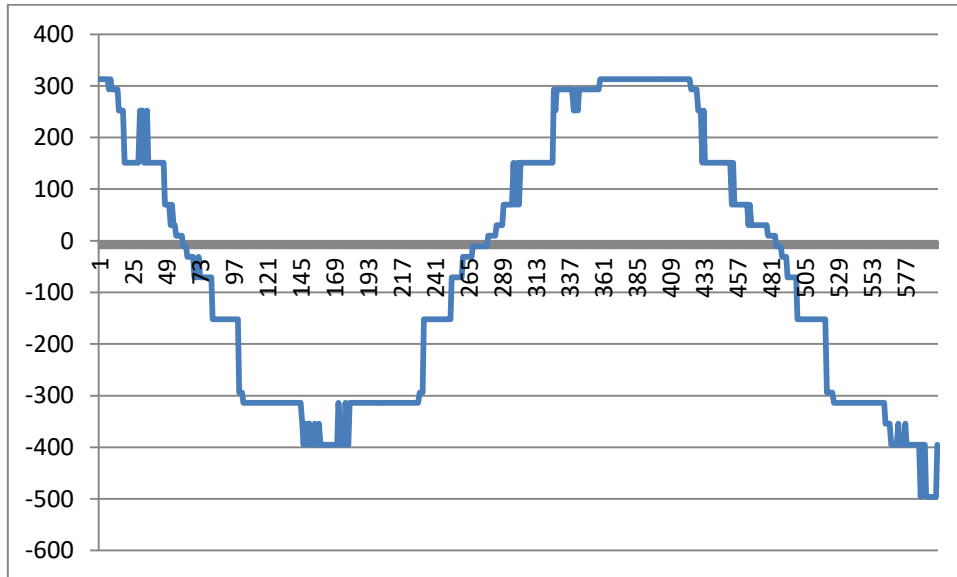


Figura 31 – Imagen de los valores digitalizados en Arduino con la mayor velocidad

Teniendo en cuenta que la corriente pico debía estar entre 450 [mA] y 500 [mA], se puede observar que el valor no se aleja demasiado.

Pero se optó por sensar el pico de corriente por un lado, luego el pico de tensión y por otro lado tratar de medir el desfase entre las señales por medio de la función micros(), que posee el Arduino, ya que al calcular la diferencia del tiempo t_s de cruce por cero de las señales de tensión y corriente podremos saber el factor de potencia.

Al medir la corriente pico I_p se puede saber la corriente RMS tomando por sentado que la señal será senoidal. Lo mismo con la tensión, tomamos V_p y sacamos el valor RMS.

Luego se multiplica para sacar la potencia aparente. De aquí sale la ecuación:

$$\theta = 2 * \pi * t_s[ms]/10[ms]$$

$$P = V_p * 0.707 * I_p * 0.707 * \cos \theta$$

Luego del censado, se pasó a diseñar la placa donde se instalara el Arduino junto con el resto de los componentes.

El esquemático se puede dividir en tres partes. Primero la etapa de conexión de 220 [v] de alterna con su circuito de seguridad:

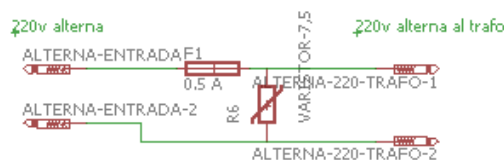


Figura 32 – Circuito de protección

La segunda parte es la fuente de 9 [v] de continua realizado con un regulador lineal, capacitores de filtro y una señal luminosa que indica su funcionamiento:

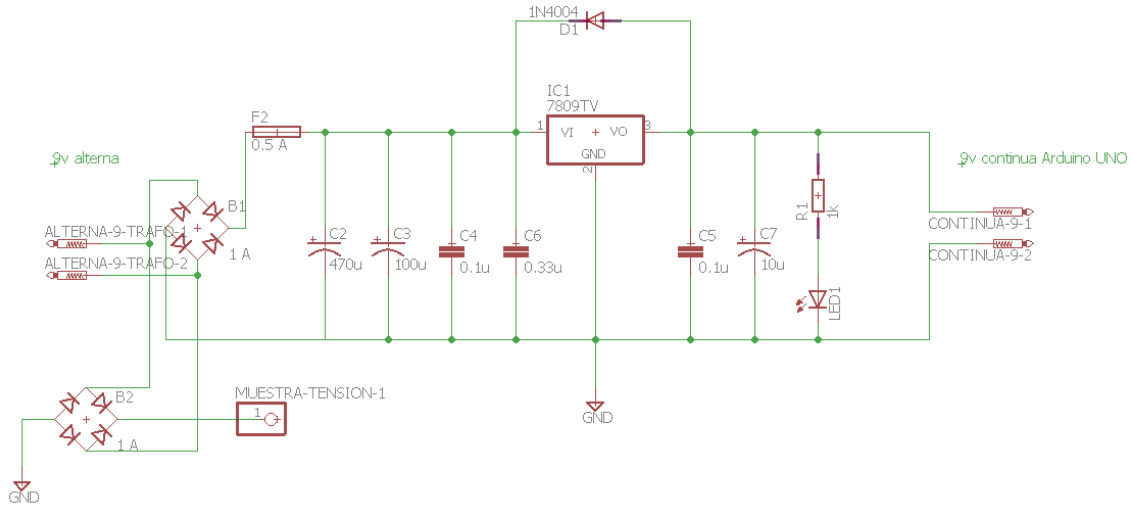


Figura 33 – Conversión AC-DC y fuente de 12v

La tercera parte es la de acondicionamiento de la señal para el ADC del Arduino:

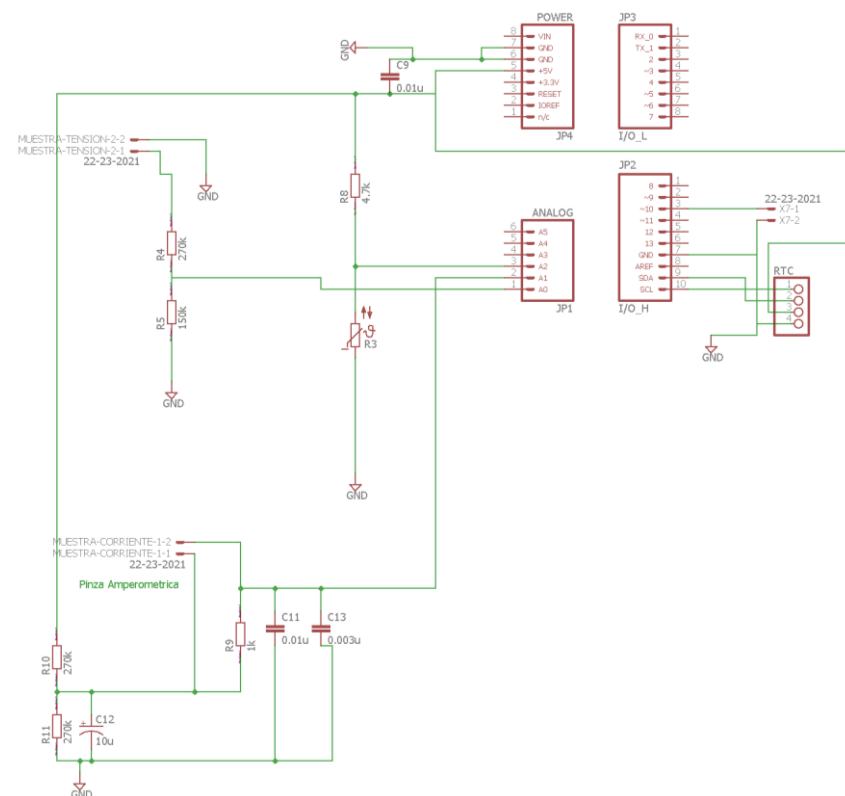


Figura 34 – Circuito de Arduino con la adaptación de señal

Una vez que se implementó, se notaron deficiencias en el diseño que sirvió para modificar el diseño del PCB a la espera del diseño final.

Una de las características que se debe mencionar, es que a la empresa a la que se le realizaran las instalaciones de los equipos (hipotéticamente), necesitarían saber la energía consumida, al igual que les interesaría, a la empresa de energía, saber cuánta energía consume cada luminaria.

Es por esto que se debió implementar un reloj en tiempo real (RTC), ya que no se quiso sobrecargar el microcontrolador con procesos que no son del todo fiables, como un reloj en tiempo real por software.

Para eso se utilizó una placa que posee el integrado DS1307 de Maxim/Dallas que es capaz de almacenar la fecha y la hora de manera autónoma. Posee un zócalo para una pila CR2032 y sus propios circuitos de precisión incluido un cristal de 32768 [Hz], lo que da como resultado que, aunque se corte el suministro eléctrico, la fecha y la hora no se pierda.

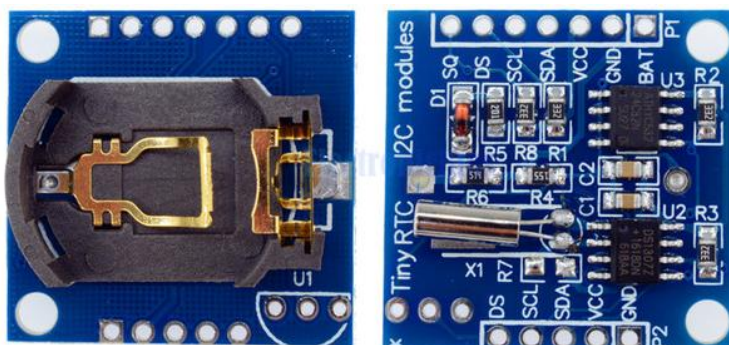


Figura 35 – Imagen del RTC DS1307

Además utiliza un protocolo de comunicación físico denominado I2C mediante el cual se comunicó el Arduino con el RTC. El bus I2C tiene dos líneas, datos y reloj. Ambas del tipo colector abierto. Por lo que se necesitaron resistencias de pull up para generar un estado lógico alto.

El DS1307 codifica en BCD los datos de fecha y hora.

Es importante aclarar que la placa necesita dos fuentes de alimentación para funcionar. Por una parte requiere 5 [V] que opera mientras el circuito está encendido y funcionando y otra fuente de poder que proviene de una batería de litio que mantiene funcionando el reloj/calendario mientras la alimentación principal no está disponible. El cambio de ambas fuentes lo gestiona el DS1307 de manera autónoma.

Llegados a este punto, se logró ensamblar tanto la placa desarrollada por nosotros, como así también el arduino y el módulo Xbee.

Se utilizó para sensar la temperatura ambiente (de manera aproximada) un dispositivo denominado termistor.

Un termistor es un sensor de temperatura por resistencia. Su funcionamiento se basa en la variación de la resistividad que presenta un semiconductor con la temperatura.

Se utilizó un termistor NTC. Para los termistores NTC, al aumentar la temperatura, aumentará también la concentración de portadores, por lo que la resistencia será menor, de ahí que el coeficiente sea negativo. La variación de la resistencia con la temperatura no es lineal. Para un termistor NTC, la característica es hiperbólica. Para pequeños incrementos de temperatura, se darán grandes incrementos de resistencia.

Se adquirió un NTC de 5 [KOhm] a 25 [°C]. Es por esto que se hizo un divisor resistivo con una resistencia de 10 [KOhm]

Se senso tensión, corriente y temperatura con un NTC que nos entrega la temperatura aproximada que hay en el ambiente donde se encuentra la placa.

La tensión que se senso y la corriente, es pico. Luego, se calibro para los valores RMS y se midió por otro lado el factor de potencia.

Por último, se calculó el tiempo que permanece encendido el dispositivo que consume y con esto se determinó la energía activa en [Watt / Hora].

El diagrama general del programa dedicado a la luminaria es:

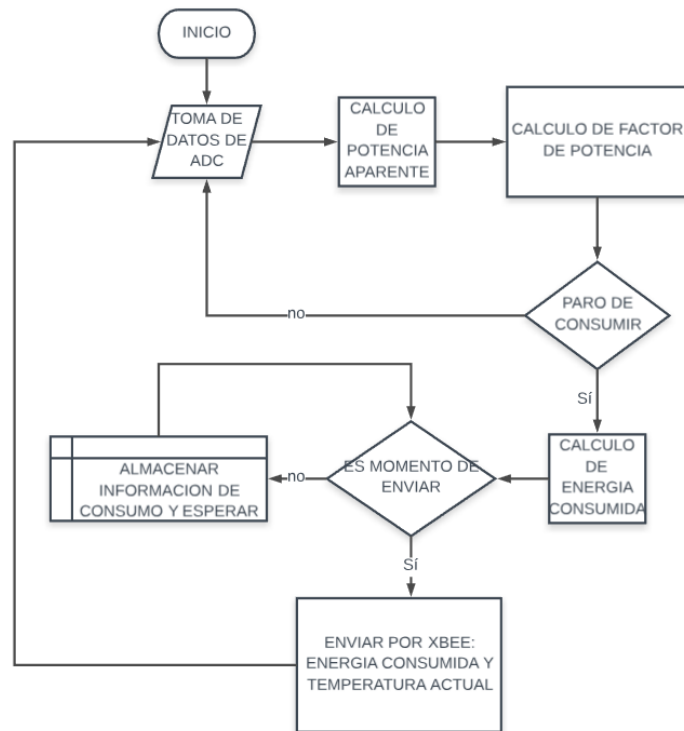


Figura 36 – Diagrama de flujo de la luminaria

Para la medición de tensión con el Arduino, se debía realizar un circuito de adaptación. De esta manera se realizaron diversos circuitos para lograrlo, teniendo resultados poco satisfactorios para los niveles de precisión que se deseaban. Luego de varias pruebas se logró hacer un circuito que tenía buenas prestaciones, podremos observarlo a continuación:

Circuito adaptador de tensión

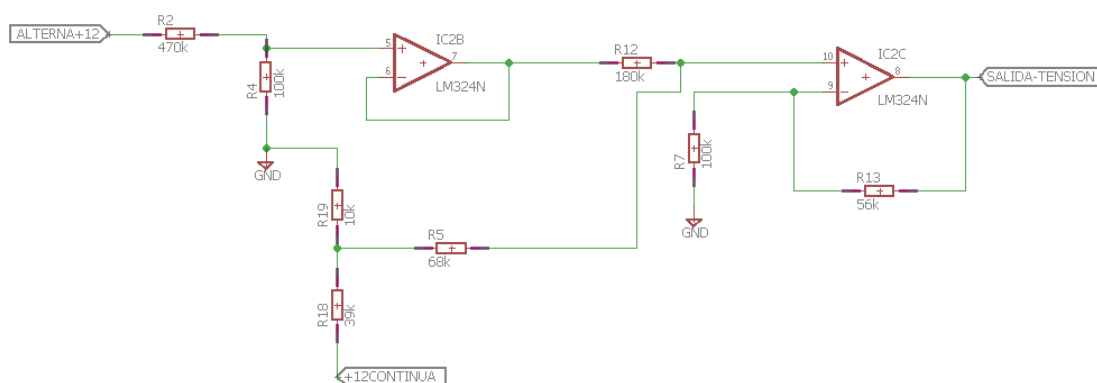


Figura 37 – Circuito adaptador de tensión alterna

Con este circuito se logró reducir el valor de la señal de 12 [V] alterna, además de sumarle un valor de continua de 2,5 [V] sin exigirle corriente al transformador que es el que provee la tensión al resto del circuito.

El inconveniente de este circuito es la utilización de fuente partida. Es por esto que se tuvo que cambiar el transformador a uno de punto medio y además se debió colocar +-12 [V]. El resultado es una señal de alterna de entre 0.7 [V] y 4.5 [V] montada sobre una continua de 2.5 [V].

Se utilizó la librería Emolib para los cálculos de tensión, corriente, potencia aparente, potencia activa y coseno fi.

La fuente partida fue implementada de la siguiente manera:

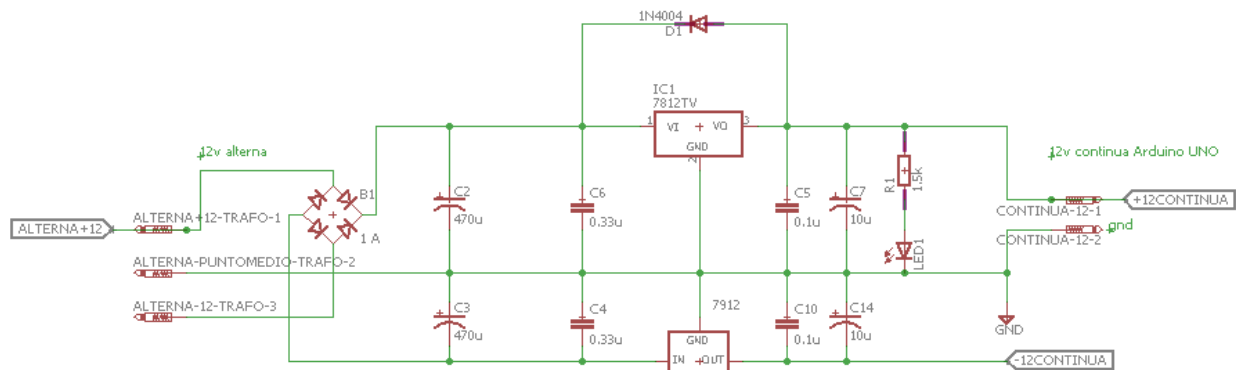


Figura 38 – Circuito de la Fuente completa de 12 [V] y -12 [V]

Se realizaron varias pruebas con un generador de señales, en donde se ingresaron diferentes señales de distinta amplitud y fase para estudiar el comportamiento de las mediciones de potencia y coseno fi.

Se confecciono una tabla donde, luego de calibrar el sensor, vimos cuales son los valores máximos de tensión y corriente que acepta el Arduino.

Tensión [mVpp]	Arduino tensión	Corriente [mApp]	Arduino corriente
2	0,43	2	0
100	6,45	100	0,08
200	12,85	200	0,16
300	19,34	300	0,25
400	25,73	400	0,33
500	32,11	500	0,41
600	38,63	600	0,5
700	45,05	700	0,58
800	51,51	800	0,66
900	57,95	900	0,74
1000	64,37	1000	0,83
1100	70,65	1100	0,91
1200	77,2	1200	0,99
1300	83,61	1300	1,07
1400	90,05	1400	1,16
1500	96,49	1500	1,24
1600	102,95	1600	32
1700	109,51	1700	1,41
1800	115,99	1800	1,49
1900	122,56	1900	1,57
2000	129	2000	1,66
2100	135,4	2100	1,74
2200	141,87	2200	1,82
2300	148,2	2300	1,9
2400	154,65	2400	1,98

2500	160,99	2500	2,06
2600	167,5	2600	2,15
2700	173,9	2700	2,23
2800	180,4	2800	2,31
2900	186,85	2900	2,4
3000	193,44	3000	2,48
3100	200,05	3100	2,56
3200	206,43	3200	2,65
3300	212,76	3300	2,73
3400	219,05	3400	2,81
3500	225,4	3500	2,89
3600	232,25	3600	2,97
3700	238,7	3700	3,05
3800	244,9	3800	3,14
3900	251,4	3900	3,22
4000	257,7	4000	3,31

Tabla 6 – Sensado de Tensión – Corriente de Arduino vs Generador de Señales

Lo que se observa en la tabla en las columnas 1 y 3 es el valor en tensión con el que se ingresa a la entrada analógica del Arduino, dado por el generador de señales, y en las columnas 2 y 4, el valor medido en el Arduino utilizando la librería Emonlib

Luego se realizó la misma prueba pero se midió el factor de potencia, utilizando la misma librería Emonlib. Se fueron dando desfases a las señales de tensión y corriente y se obtuvo lo siguiente:

Desfase [°]	Factor de potencia teórico	Factor de potencia medido
10	0,98	0,96
20	0,94	0,9
30	0,87	0,81
40	0,77	0,7
50	0,64	0,57
60	0,50	0,41
70	0,34	0,25
80	0,17	0,08
90	0,00	-0,1
100	-0,17	-0,27
110	-0,34	-0,43
120	-0,50	-0,58
130	-0,64	-0,71
140	-0,77	-0,82
150	-0,87	-0,91
160	-0,94	-0,97
170	-0,98	-1
180	-1,00	-1

Tabla 7 – Sensado de Coseno Φ de Arduino vs Generador de Señales

Control autónomo del driver led

En este punto se trabajó sobre el protocolo del driver de la luminaria para su funcionamiento en todos los casos.

El dispositivo final o nodo puede trabajar en línea (in-line) o fuera de línea (off-line). En el caso que trabaje off-line tiene una fotocélula que indica al nodo el momento en que debe encenderse y apagarse la luminaria. Además almacena el promedio de consumo de energía para que en el momento en que este en línea lo reporte al controlador de segmento. A esto se le suma que el controlador del driver debe generar una señal de tensión de 1 [v] a 10 [v] para controlar la potencia que el driver entrega.

Lo que se realizó en esta etapa es primero diseñar un circuito que pudiera entregar una señal de 1 [v] a 10 [v] de continua, haciendo la transformación desde una señal de PWM de 0 [v] a 5[v]. El circuito que se diseñó y se llevó a cabo es el que se ve a continuación:

Circuito adaptador para PWM

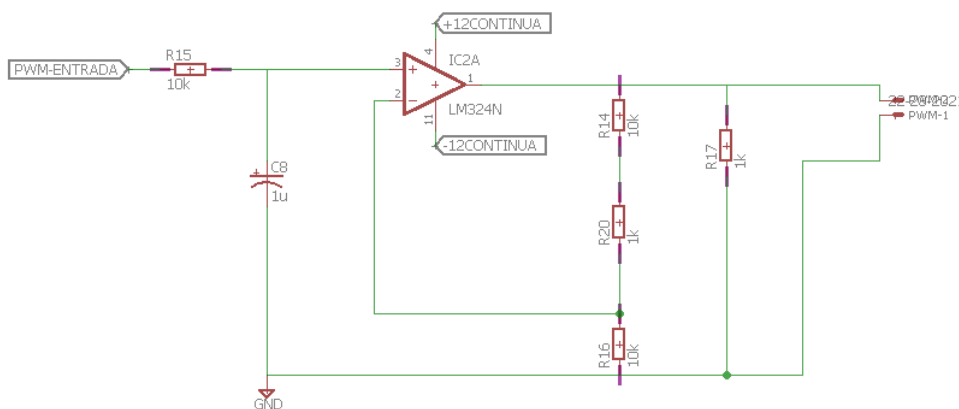


Figura 39 – Circuito de adaptación 0-5 [V] a 0-10 [V]

Este circuito entrega a la salida 10 [v] cuando el PWM tiene cargado un valor de 255 y 1 [v] cuando el PWM tiene un valor de 26. De este modo se puede utilizar el protocolo 1-10, como así también el 0-10.

Una de las cosas más importantes que se notó, es que actualmente en las conexiones hechas por los técnicos de alumbrado público, en la iluminación led, la fotocélula está conectada directamente al driver led, por lo tanto cuando se hace de noche la fotocélula alimenta el driver y este a los led, y cuando es de día los apaga. Por lo tanto la fotocélula solo sirve como una llave electrónica que se activa por incidencia de la luz.

Se observó que el protocolo de señal 1-10 solo sirve para la dimerización del driver, no del encendido o apagado del mismo. Es por esto que se debe encender y apagar el driver con un circuito de potencia y además por otro lado dimerizarlo con tensión de 1 a 10 [v].

Por lo tanto se procedió a diseñar una placa de potencia que active el driver led. La misma está separada de la placa de control, ya que de este modo, en el eventual caso de rotura del mismo, se puede cambiar rápidamente sin tocar el resto del equipo.

El circuito de potencia es el que se ve a continuación:

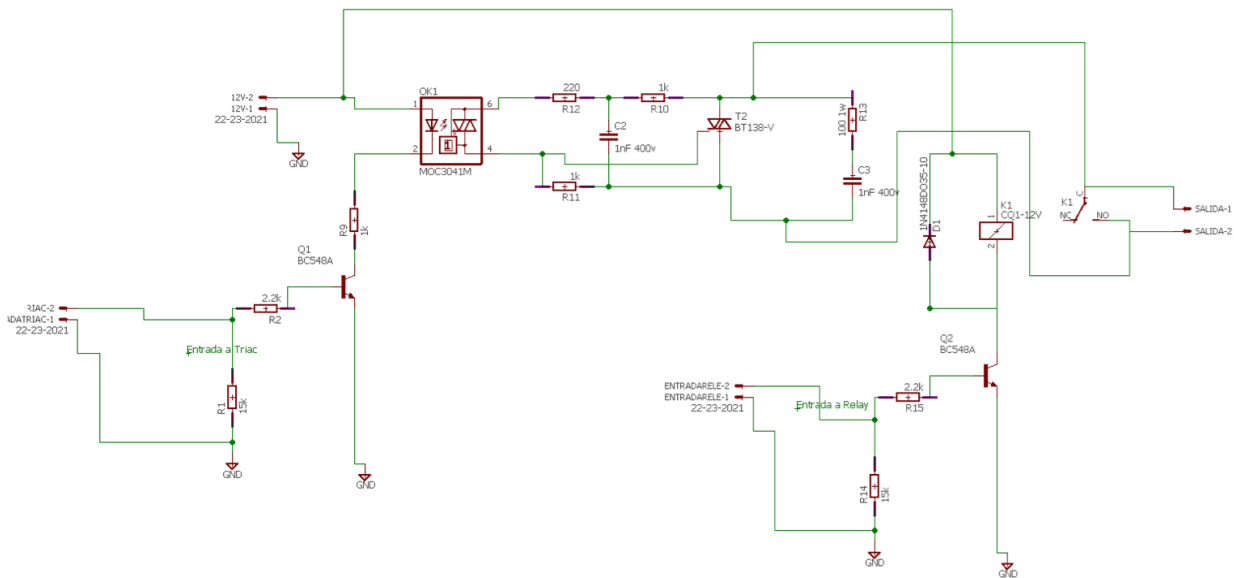


Figura 40 – Circuito de la etapa de potencia

Como puede observarse, posee un triac y un relé. La razón de esta redundancia es por una cuestión de robustez, ya que si queda “enganchado” el triac, se puede desconectar directamente con el relé que está en serie. La secuencia de encendido es, primero el relé, luego el triac. La secuencia de apagado es primero el triac y luego el relé.

De todos modos es importante saber que en la programación el encendido y el apagado es dimerizado, o sea, que la luminaria no se enciende ni se apaga a máxima potencia, sino que se incrementa o decremента lentamente. De este modo no se le exige a la placa de encendido del driver que entregue los picos de corriente que necesita el driver LED en el caso de encendido a máxima potencia.

En el circuito de encendido del driver led se separó la parte de control de la parte de potencia con un MOC, para evitar problemas en la parte más importante del proyecto.

Se realizaron pruebas de encendido y apagado del driver led utilizando la dimerización para encender y apagar, y no se tuvo ningún inconveniente. La prueba se realizó con dos drivers de General Electric (GE150/1050/MV-GL) conectados en paralelo y sus respectivos leds conectados a ellos.

Fotos luminaria LED

A continuación se podrán ver fotos de la luminaria LED utilizada y su respectivo driver (GE150/1050/MV-GL).

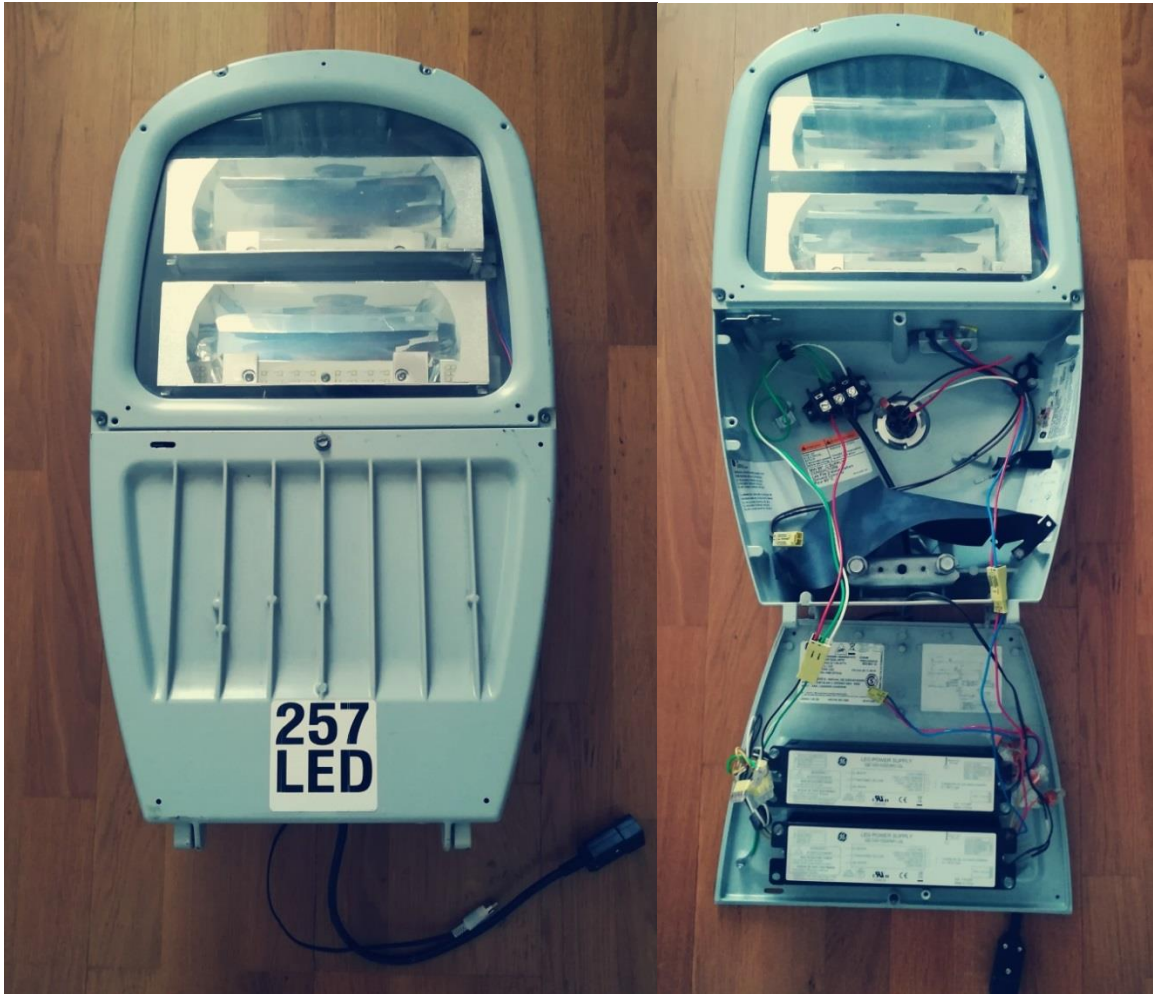


Figura 41 – Imagen de la luminaria usada para las pruebas

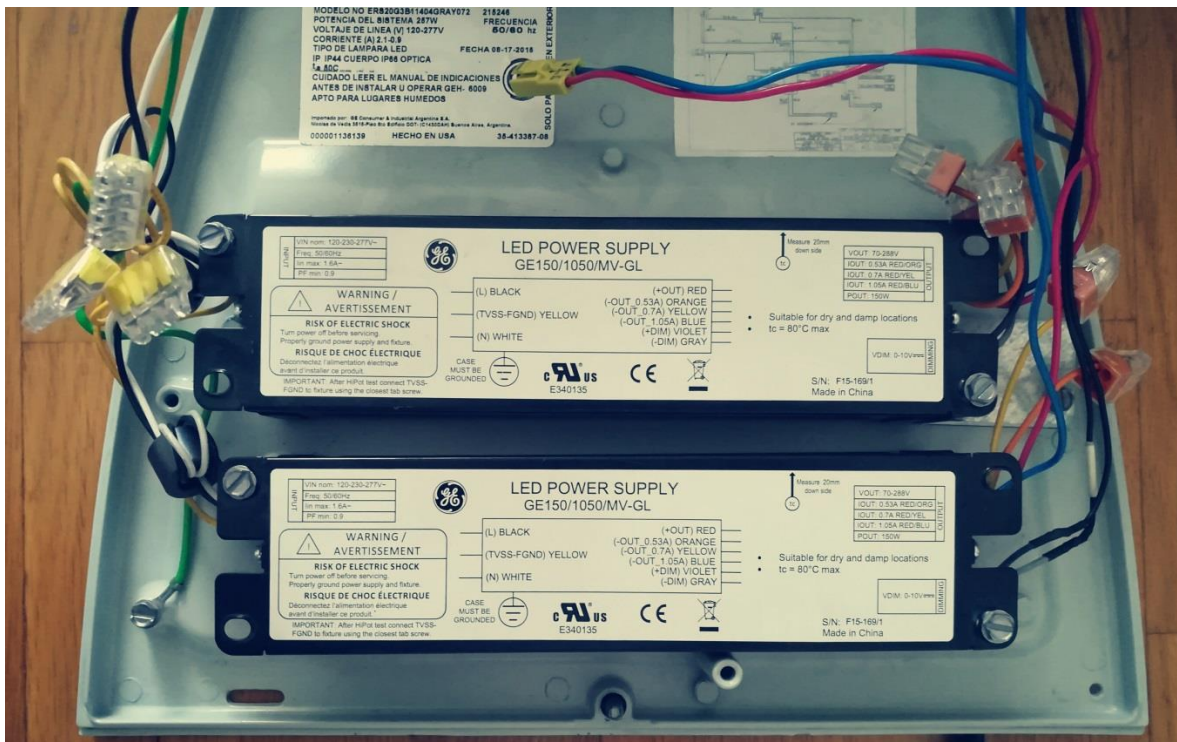


Figura 42 – Imagen que muestra de cerca el tipo de driver LED

Manipulación, programación y monitoreo de datos

En este paso se trabajó sobre el controlador de segmento (Raspberry pi 3), en la cual se reciben todos los datos de los nodos, se procesa la información y se utiliza un software para el control de todos los datos. El controlador es capaz de enviar datos a todos los nodos para realizar las tareas solicitadas. También se detalla la programación realizada tanto en el coordinador como en las luminarias.

Para comenzar con el desarrollo de esta etapa, se realizara una descripción del controlador de segmento que se utilizara, en este caso la Raspberry Pi 3. De esta forma se conocerá el funcionamiento, la potencialidad y el porqué de utilizar este dispositivo como controlador de segmento.

Raspberry Pi

(Raspberry Pi Foundation, 2016) Es un computador de placa reducida, computador de placa única o computador de placa simple (SBC) de bajo costo desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

Aunque no se indica expresamente si es hardware libre o con derechos de marca, en su web oficial explican que disponen de contratos de distribución y venta con dos empresas, pero al mismo tiempo cualquiera puede convertirse en revendedor o redistribuidor de las tarjetas Raspberry Pi, por lo que da a entender que es un producto con propiedad registrada, manteniendo el control de la plataforma, pero permitiendo su uso libre tanto a nivel educativo como particular.

En cambio el software sí es open source, siendo su sistema operativo oficial una versión adaptada de Debian, denominada Raspbian, (el utilizado en nuestro caso para el desarrollo del proyecto), aunque permite usar otros sistemas operativos. En un principio se decidió instalar Ubuntu, pero al ser un sistema que no está pensado para ser utilizado con Raspberry, la performance de la misma no era muy buena, por lo que se optó por utilizar Raspbian, que funciona perfectamente. Otro sistema que soporta es por ejemplo una versión de Windows 10. En todas sus versiones incluye un procesador Broadcom, una memoria RAM, una GPU, puertos USB, HDMI, Ethernet (El primer modelo no lo tenía), 40 pines GPIO y un conector para cámara. Ninguna de sus ediciones incluye memoria, siendo esta en su primera versión una tarjeta SD y en ediciones posteriores una tarjeta MicroSD.

Modelos

(Raspberry Pi Foundation, 2016) Los esquemas del modelo A y el modelo B fueron lanzados el 20 de abril de 2012 por la fundación.

Raspberry Pi 1 Modelo A

Este fue el primer modelo de Raspberry, sus ventas comenzaron en el año 2012. Carecía de puerto Ethernet, por lo que para su conexión a Internet requería de un adaptador Wi-Fi por USB. Poseía 26 conectores GPIO, salida de vídeo vía HDMI y Video RCA, un conector Jack de 3.5 [mm], un único conector USB, MicroUSB (De alimentación) y un conector de cámara. Su procesador fue un Broadcom BCM2835, Single-Core a 700 [MHz]. También tuvo 256 [MB] de RAM y una gráfica Broadcom VideoCore IV. Requería de una fuente de alimentación de 5 voltios y 2 amperios, elemento común al resto de versiones. Tuvo un coste inicial de 40 euros.



Figura 43 –Raspberry 1 model A

Raspberry Pi 1 Modelo B y B+

También del año 2012, es una variante del Modelo A, trajo consigo diversas mejoras, la inclusión del doble de memoria RAM, pasando de 256 [MB] a 512 [MB]. Trajo consigo un puerto USB más y, por fin, un conector Ethernet (RJ-45). Se mantuvo tanto su tamaño como su coste. No hubo variaciones ni en el procesador ni en la parte gráfica. Tiempo después se lanzó el Modelo B+, que incluyó 4 puertos USB y pasó de usar una SD a una MicroSD.



Figura 44 –Raspberry 1 model B

Raspberry Pi 2 Modelo B

Lanzada en 2014 es el primer modelo que no incluye el mismo procesador usado en los tres anteriores: se sustituye por uno de la misma marca, pero de modelo BCM2836. Pasa de ser de un núcleo a cuatro, y de 700 [MHz] a 900 [MHz]. No obstante emplea la misma gráfica, la VideoCore IV. Dobra la cantidad de memoria RAM, pasando de 512 [MB] a 1 [GB] (Algo menos en realidad) esta memoria está compartida con la gráfica. También incluye 40 pines GPIO, y mantiene los cuatro puertos USB. Suprime la conexión RCA.



Figura 45 –Raspberry Pi 2 model B (2015)

Raspberry Pi 3 Modelo B

Sacada a la luz en el año 2016, renueva procesador, una vez más de la compañía Broadcom, una vez más un Quad-Core, pero pasa de 900MHz a 1.20 [GHz]. Mantiene la RAM en 1 [GB]. Su mayor novedad fue la inclusión de Wi-Fi y Bluetooth (4.1 Low Energy) sin necesidad de adaptadores.

Es el modelo que se utilizó para la elaboración del proyecto.

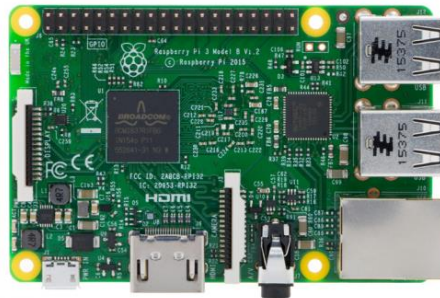


Figura 46 –Raspberry Pi 3 model B

Raspberry Pi 3 Model B+

La Raspberry Pi 3 B+ apareció en Marzo del 2018 para actualizar el modelo anterior la Raspberry Pi 3 Model B y entre sus mejoras cuenta con un nuevo procesador y mejor conectividad, nuevo procesador así que pasa de tener 1.2 [Ghz] a tener 1.4 [Ghz] y en cuanto a la conectividad inalámbrica ahora incorpora doble banda a 2,4 [GHz] y 5 [GHz], y su nuevo puerto Ethernet se triplica, pasa de 100 [Mbits/s] en el modelo anterior a 300 [Mbits/s] en el nuevo modelo, también contará con Bluetooth 4.2 y Bluetooth BLE.



Figura 47 –Raspberry Pi 3 model B+

Software

(Wikipedia, 2018) El Raspberry Pi usa mayormente sistemas operativos GNU/Linux. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi, se lanzó durante julio de 2012 y es la distribución recomendada por la fundación para iniciarse.

A la GPU se accede mediante una imagen del firmware de código cerrado (un *blob* binario, en inglés), que se carga dentro de la GPU al arrancar desde la tarjeta SD. El archivo está asociado a los controladores del núcleo Linux que también son de código cerrado. Las aplicaciones hacen llamadas a las bibliotecas de tiempo de ejecución que son de código abierto, y las mismas hacen llamadas a unos controladores de código abierto en el núcleo Linux. La API del controlador del núcleo es específica para estas bibliotecas. Las aplicaciones que usan vídeo hacen uso de OpenMAX, las aplicaciones tridimensionales usan OpenGL ES y las aplicaciones 2D usan OpenVG;

OpenGL ES y OpenVG hacen uso de EGL y este último, del controlador de código abierto del núcleo.

El 19 de febrero de 2012, la fundación lanzó un prototipo de imagen de tarjeta SD que almacenaba un sistema operativo y que podía ser cargado en una tarjeta SD. La imagen se basaba en Debian 6.0 (Squeeze), con el escritorio LXDE y el navegador Midori, más algunas herramientas de programación. La imagen funcionaba bajo QEMU permitiendo que el Raspberry Pi pudiera ser emulado en otros sistemas.

El 8 de marzo de 2012, la fundación lanzó Raspberry Pi Fedora Remix (actualmente llamada Pidora), que en el momento de era la distribución recomendada por la fundación, y fue desarrollada en la universidad de Séneca, en Canadá. También se propuso crear una tienda de aplicaciones para que la gente intercambiara programas.

El 24 de octubre de 2012, Alex Bradbury, director de desarrollo Linux de la fundación, anunció que todo el código del controlador de la GPU Videocore que se ejecuta en ARM sería de código abierto, mediante licencia BSD modificada de 3 cláusulas. El código fuente está disponible en un repositorio de la fundación en GitHub.

El 5 de noviembre de 2012, Eben Upton anunció el lanzamiento del sistema operativo RISC OS 5 para Raspberry Pi a la comunidad, pudiéndose descargar la imagen de forma gratuita desde la web de la fundación. . Además se incluyen manuales para crear aplicaciones en BASIC para el sistema operativo.

El 24 de noviembre de 2012, se anunció en la Minecon de París, el juego *Minecraft: Pi Edition* para Raspberry Pi, basado en la versión *Minecraft: Pocket Edition* para teléfonos inteligentes y tabletas. La descarga se hizo disponible de forma oficial y gratuita por primera vez el 12 de febrero de 2013 desde el blog del juego, como versión 0.1.1 alpha, junto a instrucciones para ejecutarlo en Raspbian Wheezy. Una de las características principales de este lanzamiento es poder interactuar con el juego mediante programación, con la intención de motivar a los niños a aprender a programar.

El 25 de mayo de 2013, la fundación informó de que se estaba trabajando en una versión del servidor gráfico Wayland para Raspberry Pi, para sustituir al sistema de ventanas X. Con este cambio se lograría suavidad al usar la interfaz gráfica del escritorio, ya que el procesamiento lo realizaría el núcleo de video de la GPU y no la CPU, sin interferir en el renderizado 3D.

El 3 de junio de 2013, fue lanzado en la web de la fundación para su descarga la aplicación NOOBS (New Out of Box Software), utilidad que facilita la instalación de diferentes sistemas operativos para Raspberry Pi. Se distribuye en forma de archivo zip que se copia descomprimido a una tarjeta SD de 4 o más GB, y una vez arrancada la placa con la tarjeta por primera vez, aparece un menú en que se da la opción de instalar una de las diferentes distribuciones en el espacio libre de la tarjeta de memoria, o acceder a internet con el navegador Arora integrado. Más adelante si se desea, es posible acceder a este menú apretando la tecla shift durante el arranque para reinstalar el sistema operativo, elegir otro, o editar el archivo config.txt. NOOBS contiene las distribuciones GNU/Linux de carácter general Raspbian, Arch Linux ARM y Pidora; las distribuciones para mediacenter con Kodi Openelec y RaspBMC; y el sistema operativo Risc OS 5.

El 26 de septiembre de 2013, se añadió a los repositorios de Raspbian una versión oficial de Oracle Java JDK ARM con soporte para coma flotante por hardware, que ofrece bastante más rendimiento que la versión OpenJDK ARM ya existente hasta ese momento y más compatibilidad con aplicaciones. También se anunció que esta versión de Oracle Java JDK se incluiría dentro de la distribución en futuras versiones de Raspbian.

Algo que no se menciona en esta evolución e incorporación de nuevas características, es la posibilidad y compatibilidad de utilizar el software QT con el cual se desarrolló la interfaz gráfica para el control de las luminarias. Se ampliara esto más adelante.

Hardware

Raspberry Pi 3 model B

Se detallara solamente el modelo 3 B que fue el utilizado para el proyecto.

(Foundation Raspberry Pi, 2018) La misma cuenta con las siguientes especificaciones:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1 [GB] RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port.
- Upgraded switched Micro USB power source up to 2.5 [A]

(Comohacer.eu, 2018) Las características de los pines GPIO son las siguientes:

GPIO (General Purpose Input/Output) es, como su propio nombre indica, un sistema de E/S (Entrada/Salida) de propósito general, es decir, una serie de conexiones que se pueden usar como entradas o salidas para usos múltiples. Estos pines están incluidos en todos los modelos de Raspberry Pi.

Todos los pines son de tipo “unbuffered”, es decir, no disponen de buffers de protección, así que se debe tener cuidado con las magnitudes (voltajes, intensidad,...) cuando se conectan componentes a ellos para no dañar la placa.

- **Pines de alimentación:** pines de 5 [v], 3,3 [v] (limitados a 50 [mA]) y tierra (GND o Ground), que aportan alimentación. Pueden servir como una fuente de alimentación, aunque también se puede utilizar otras fuentes (pilas, fuentes de alimentación externas, etc.). Recordando que son unbuffered y se debe tener cuidado en no dañar la placa.
- **DNC (Do Not Connect):** son pines que por el momento no tienen función, pero en futuras implementaciones serán utilizados para otros fines. Por eso solo se encuentran en modelos más primitivos de la Raspberry Pi. En las actuales placas han sido marcados como GND.
- **GPIO normales:** son conexiones configurables que se pueden programar.
- **GPIO especiales:** dentro de éstos se encuentran algunos pines destinados a una interfaz UART, con conexiones TXD y RXD que sirven para comunicaciones en

serie. También podemos ver otros como SDA, SCL, MOSI, MISO, SCLK, CE0, CE1, etc.

Model B (Rev 1.0)

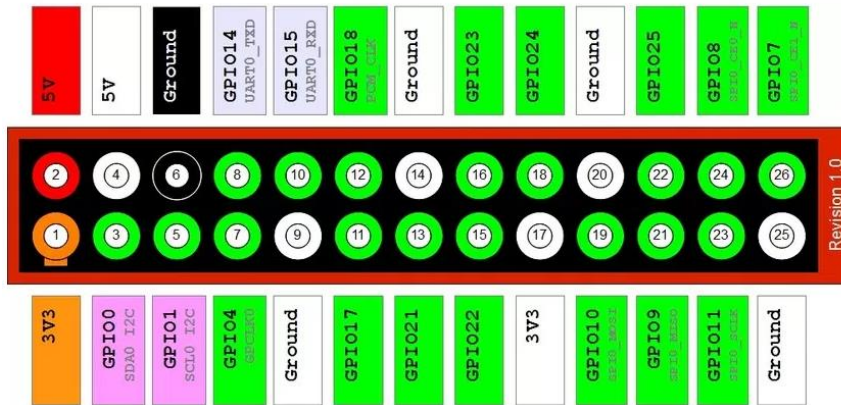


Figura 48 –Pines GPIO

Para el desarrollo de la interfaz gráfica se utilizó el software QT. Se detallara el mismo a continuación.

QT

(Blanchette & Summerfield, 2006) Qt es un framework multiplataforma orientado a objetos ampliamente usado para desarrollar programas (software) que utilicen interfaz gráfica de usuario, así como también diferentes tipos de herramientas para la línea de comandos y consolas para servidores que no necesitan una interfaz gráfica de usuario.

Qt es desarrollado como un software libre y de código abierto a través de Qt Project, donde participa tanto la comunidad, como desarrolladores de Nokia, Digia y otras empresas. Anteriormente, era desarrollado por la división de software de Qt de Nokia, que entró en vigor después de la adquisición por parte de Nokia de la empresa noruega Trolltech, el productor original de Qt, el 17 de junio de 2008. Qt es distribuida bajo los términos de GNU Lesser General Public License y otras. Por otro lado, Digia está a cargo de las licencias comerciales de Qt desde marzo de 2011.

Qt es utilizada en KDE, entorno de escritorio para sistemas como GNU/Linux o FreeBSD, entre otros.

Propósitos y características

Qt utiliza el lenguaje de programación C++ de forma nativa, adicionalmente puede ser utilizado en varios otros lenguajes de programación a través de *bindings*. También es usada en sistemas informáticos embebidos para automoción, aeronavegación y aparatos domésticos como frigoríficos.

Funciona en las principales plataformas y tiene un amplio apoyo. El API de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML, gestión de hilos, soporte de red, una API multiplataforma unificada para la manipulación de archivos y una multitud de otros métodos para el manejo de ficheros, además de estructuras de datos tradicionales.

Plataformas

Qt se encuentra disponible para sistemas tipo Unix con el servidor gráfico X Windows System (Linux, BSDs, Unix), para Apple Mac OS X, para sistemas Microsoft Windows,

para Linux embebido (en inglés *Embedded Linux*), para sistemas embebidos como PDA, Smartphone, etc. y para dispositivos que utilizan Windows CE.

Implementación

Utilizando el software QT se implementó la interfaz gráfica para la interacción con el usuario. Esta interfaz se encuentra alojada en el nivel de segmento, en nuestro caso utilizando la Raspberry Pi 3.

Se optó por usar este software debido a todo lo mencionado anteriormente, sobre todo la posibilidad que nos brinda de ser multiplataforma.

La interfaz consta de una ventana en donde el usuario podrá visualizar los datos de las luminarias, encender/apagar o variar la intensidad de luz, como también configurar horarios de encendido y apagado.

A continuación veremos una captura de la interfaz gráfica.



Figura 49 –Ventana Principal del Programa

Entrando más en detalles se pasara a explicar la funcionalidad de cada botón.

Empezando por el botón “Buscar”, el mismo realiza la función de buscar si hay un coordinador conectado. Si bien esto lo realiza al momento de la ejecución del programa, puede ocurrir que el coordinador sea desconectado de la Raspberry en plena ejecución del programa. En caso de volverlo a conectar, se deberá presionar este botón para tener comunicación nuevamente con el coordinador.

El botón “Obtener datos” envía un comando al coordinador para que este se comunique con las luminarias y que las mismas le envíen los datos sensados. Paso siguiente el coordinador enviara estos datos por el puerto serie a la Raspberry y se podrán observar en cada “display”, el factor de potencia, la energía medida en Watt por hora, el tiempo que estuvo encendida la luminaria en horas y la temperatura en grados Celsius. También se tiene la opción de elegir en el “combobox” la luminaria de la que se desea conocer los datos, la cual va a estar identificada por su MAC en el “label” que se ve en la captura. También se tiene la opción broadcast, en donde se observa un promedio de los datos de todas las luminarias sensadas.

Los botones “Encender”, “Apagar” y “Enviar PWM” realizan básicamente la misma función. Que se trata en enviar al coordinador el valor del PWM que se desea setear en la luminaria elegida en el “combobox”, o bien a todas las luminarias si elegimos la opción broadcast. “Encender” envía el valor de PWM 255, el cual sería el valor de encendido al máximo de intensidad. “Apagar” envía el valor de PWM 25, que sería el valor que apaga la luminaria por completo. Por último, “Enviar PWM” envía el valor seteado en la barra de desplazamiento, que va desde “0”, valor 25 de PWM hasta “100”, valor 255 de PWM. El coordinador luego se encarga de enviar estos valores a la luminaria correspondiente.

El botón “Enviar configuración”, envía al coordinador el horario seteado por el usuario en “Horario encendido” y “horario apagado”. Luego el coordinador enviara esta configuración a la luminaria correspondiente.

Se debe tener en cuenta que si no se setea un horario, o bien el horario de encendido es el mismo que el horario de apagado, la luminaria o las luminarias funcionarán automáticamente con la fotocélula.

El “label” “Estado” permite observar el estado actual de la luminaria, que puede ser “encendida” o “apagada”.

Para finalizar, el botón “Ver histórico” nos permite ver en un archivo de texto todo lo realizado por el usuario, identificando que función se presionó, fecha y hora del momento en que se hizo.

A continuación se ve un ejemplo.

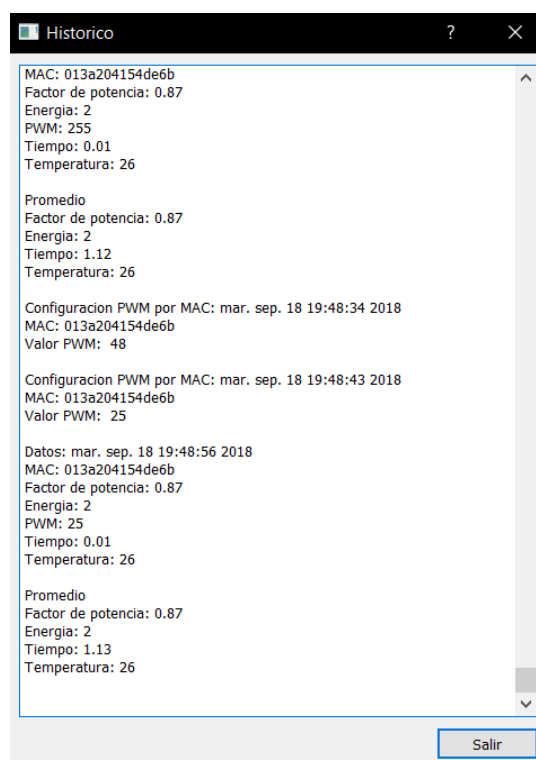


Figura 50 –Ventana del Histórico

Tramas

Para la comunicación entre la Raspberry y el coordinador se establecieron tramas de un tamaño fijo para poder hacer el tratamiento y separar los diferentes datos.

Hay dos tipos de tramas, las que se envían desde la Raspberry al coordinador, y la que se envía desde el coordinador a la Raspberry.

Para el primer caso las tramas son de un tamaño de 40 caracteres (320 bits o 40 bytes) y serían las siguientes:

Opción 1: Botón “Enviar configuración” unicast.

Inicio de trama	Horario encendido	Separador	Horario apagado	Separador	MAC	Separador	Opción	Final de trama
1 byte	8 bytes	1 byte	8 bytes	1 byte	14 bytes	1 byte	4 bytes	2 bytes
*	01:00:00	-	02:00:00	-	013a 2041 54de 6b	-	XXX1	#\n

Tabla 8 – Trama de Configuración de Horario Unicast

Opción 2: Botón “Enviar configuración” broadcast.

Inicio de trama	Horario encendido	Separador	Horario apagado	Separador	Bits de relleno	Separador	Opción	Final de trama
1 byte	8 bytes	1 byte	8 bytes	1 byte	17 bytes	1 byte	1 byte	2 bytes
*	01:00:00	-	02:00:00	-	XXXXXXX XXXXXXX XXX	-	2	#\n

Tabla 9 – Trama de Configuración de Horario Broadcast

Opción 3: Botón “Obtener datos”

Inicio de trama	Bits de relleno	Separador	Opción	Final de trama
1 byte	35 bytes	1 byte	1 byte	2 bytes
*	XX	-	3	#\n

Tabla 10 – Trama de Obtención de Datos

Opción 4: Enviar PWM unicast

Inicio de trama	Valor PWM	Separador	MAC	Separador	Bits de relleno	Separador	Opción	Final de trama
1 byte	3 bytes	1 byte	14 bytes	1 byte	16 bytes	1 byte	1 byte	2 bytes
*	100 a 255	-	013a2041 54de6b	-	XXXXXXX XXXXXXX XX	-	4	#\n

Inicio de trama	Valor PWM	Separador	MAC	Separador	Bits de relleno	Separador	Opción	Final de trama
1 byte	2 bytes	1 byte	14 bytes	1 byte	17 bytes	1 byte	1 byte	2 bytes
*	25 a 99	-	013a2041 54de6b	-	XXXXXXX XXXXXXX XXX	-	4	#\n

Tabla 11 – Trama de Envío de PWM Unicast

Opción 5: Enviar PWM broadcast

Inicio de trama	Valor PWM	Separador	Bits de relleno	Separador	Opción	Final de trama
1 byte	3 bytes	1 byte	31 bytes	1 byte	1 byte	2 bytes
*	100 a 255	-	XXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXX	-	4	#\n

Inicio de trama	Valor PWM	Separador	Bits de relleno	Separador	Opción	Final de trama
1 byte	2 bytes	1 byte	32 bytes	1 byte	1 byte	2 bytes
*	25 a 99	-	XXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXX	-	4	#\n

Tabla 12 – Trama de envío de PWM Broadcast

Para el caso del envío de tramas desde el coordinador a la Raspberry el tamaño de trama es de 37 bytes

Inicio de trama	MAC	Separador	Factor de potencia	Separador	Energía	Separador	Tiempo	Separador
1 byte	14 bytes	1 byte	1 a 3 bytes	1 byte	1 a 4 bytes	1 byte	1 a 4 bytes	1 byte
*	013a204154de6b	-	1 a 100	-	1 a 9999	-	1 a 9999	-

PWM	Separador	Temperatura	Final de trama
2 a 3 bytes	1 byte	1 a 2 bytes	1 a 11 bytes
25 a 255	-	1 a 99	# o X*cantidad de bits faltantes + #

Tabla 13 – Trama de envío de Coordinador a Raspberry

Comunicación

Como vimos anteriormente existe una interacción entre el nivel de gestión (Raspberry), el coordinador y las luminarias.

Entre la Raspberry y el coordinador la comunicación se establece mediante un cable USB, utilizando un protocolo serie, el cual se implementó utilizando los siguientes valores:

- Bits de datos = 8 bits
- Velocidad = 9600 baudios
- Paridad = Sin bits de paridad
- Bit de parada = 1 bit
- Control de flujo = Sin control de flujo

Entre el coordinador y las luminarias se utilizó el protocolo Xbee, ya mencionado sus características en las etapas anteriores del proyecto.

A continuación podremos ver como interaccionan estos dispositivos entre sí. Se realizó un esquema en donde podremos ver las diferentes opciones de la interfaz gráfica, como se comunica con el coordinador y como este último lo realiza con las luminarias.

El esquema de comunicación entre los 3 niveles es el siguiente:

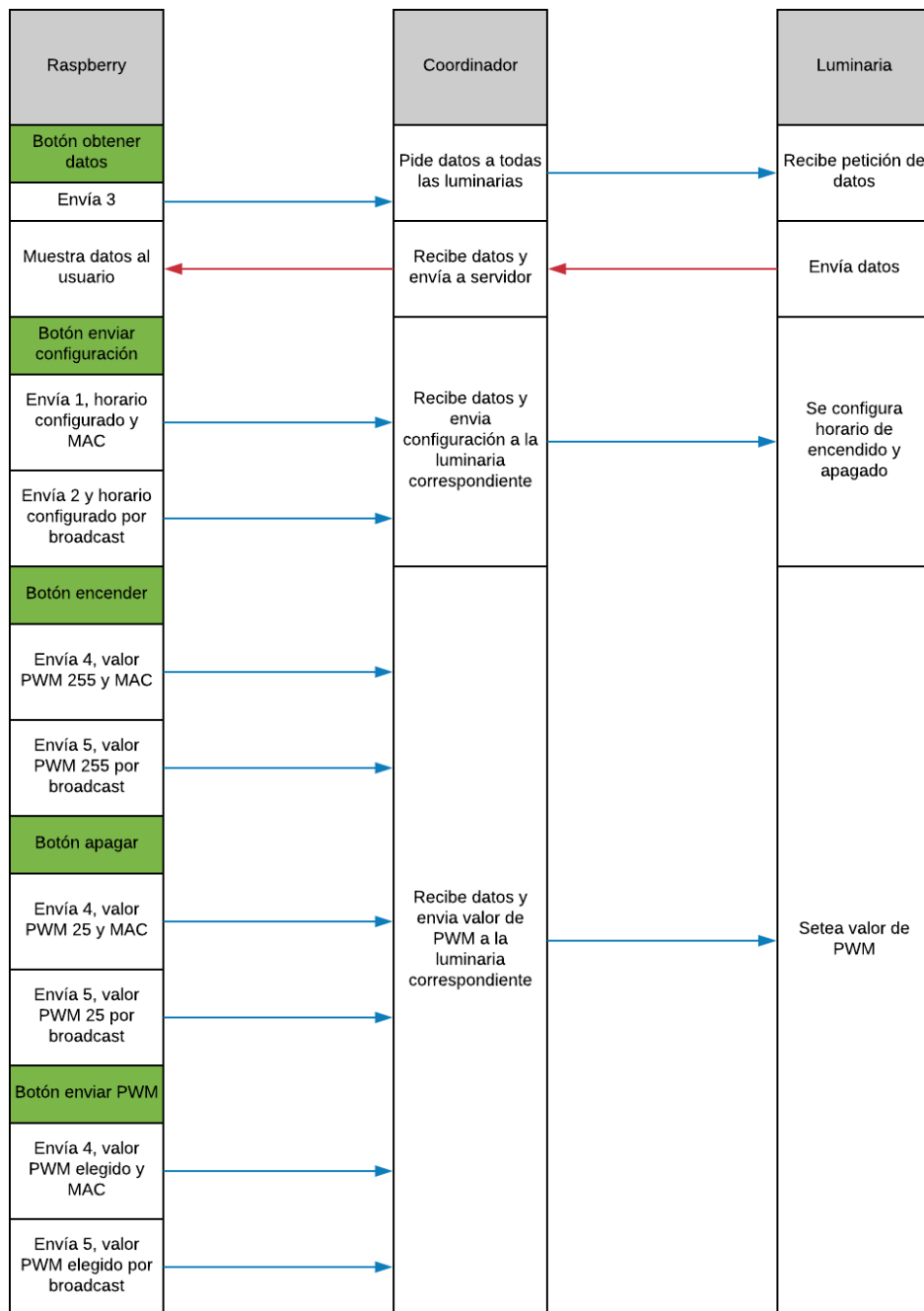


Figura 51 –Vista general del Sistema completo en base al Software

Como se puede visualizar en el esquema, solo hay una opción que recibe datos tanto en el coordinador como en la Raspberry, y es la opción 3 de obtener datos. En las demás opciones no se espera ninguna respuesta por parte de la luminaria, por lo cual, en el caso de que querer conocer el estado de la luminaria luego de haber enviado un comando de encender o apagar, se deberá presionar el botón “Obtener datos” y así se verá el estado actual.

Programación

Ya visto el esquema general de funcionamiento, se continuara explicando más detalladamente la programación del coordinador y de la luminaria.

Coordinador

A continuación se puede ver un diagrama de flujos de la programación del coordinador.

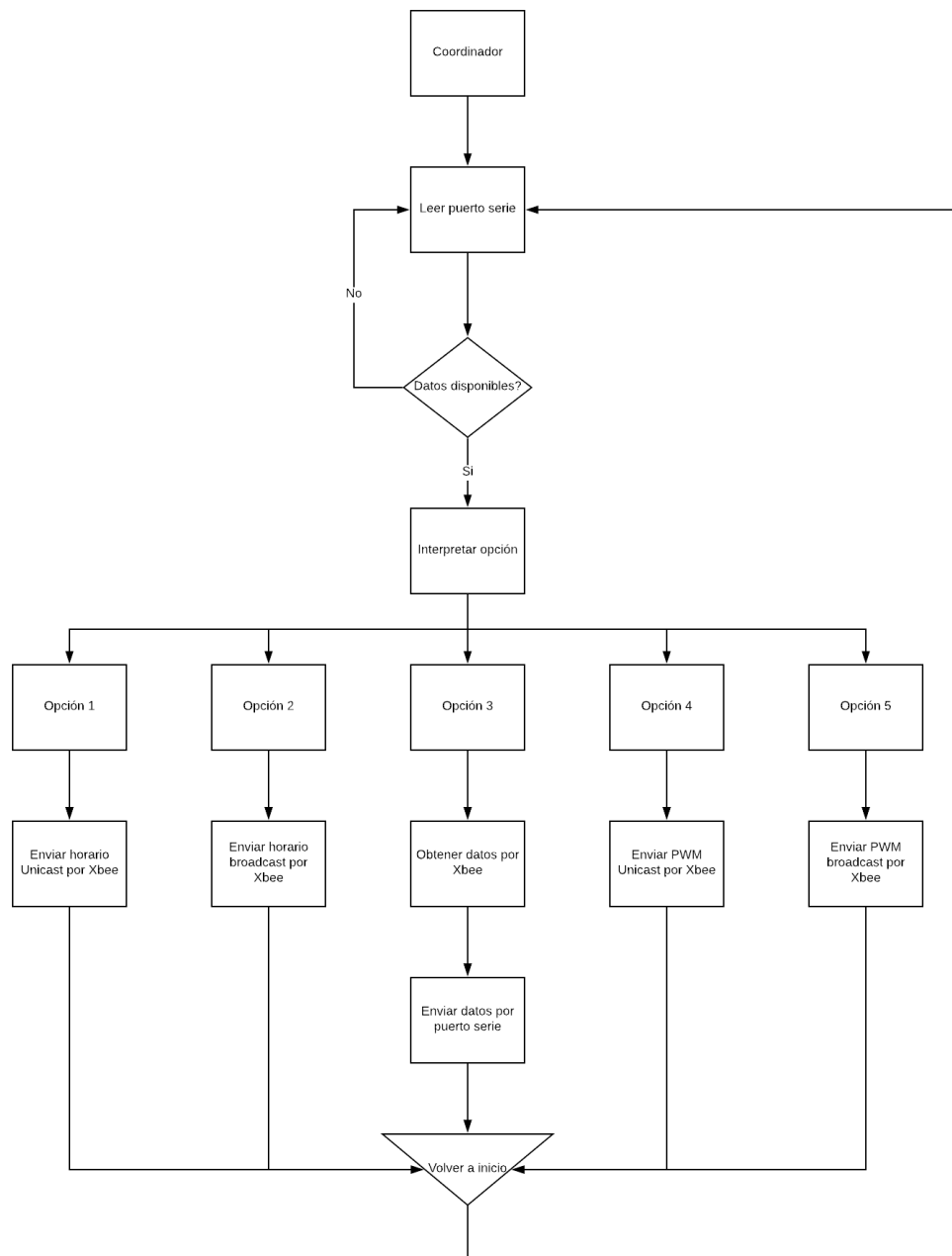


Figura 52 –Diagrama de Flujo del coordinador en Arduino

Básicamente, el coordinador, más allá de que en un principio lo que va a hacer junto con la luminaria es asociarse a nivel de protocolo Xbee, lo cual es transparente para nosotros, solamente estará esperando que le llegue una instrucción del usuario por el puerto serie. En cuanto se recibe una trama por el puerto serie, el coordinador en un primer lugar, va a separar el campo de “opción” y va a interpretar en que opción se desea ingresar.

En el caso en que la opción elegida sea la 3 (“Obtener datos”), el coordinador envía esta opción a la luminaria con la cual esta envía los datos al coordinador. El coordinador recibe estos datos y los reenvía por puerto serie a la Raspberry. Luego vuelve al inicio y espera hasta que una nueva orden sea enviada.

Para la opción 1 y 2, el coordinador separara los campos de la trama para obtener el horario de encendido y apagado, y en caso de la opción 1, también obtendrá la MAC de la luminaria a la cual se desea enviar el comando. De esta forma se procede a enviar el horario configurado por Xbee, a la luminaria correspondiente en caso de que sea la opción 1, o a todas las luminarias en caso de ser la opción 2.

Para la opción 4 y 5, nuevamente el coordinador separara los campos de la trama para obtener el valor de PWM, en caso de la opción 4, obtendrá la MAC de la luminaria a la cual se desea enviar el comando. De esta forma al igual que el caso anterior, se procede a enviar el PWM configurado por Xbee, a la luminaria correspondiente en caso de que sea la opción 4, o a todas las luminarias en caso de ser la opción 5.

Luminaria

El diagrama de flujos de la programación de la luminaria es el siguiente.

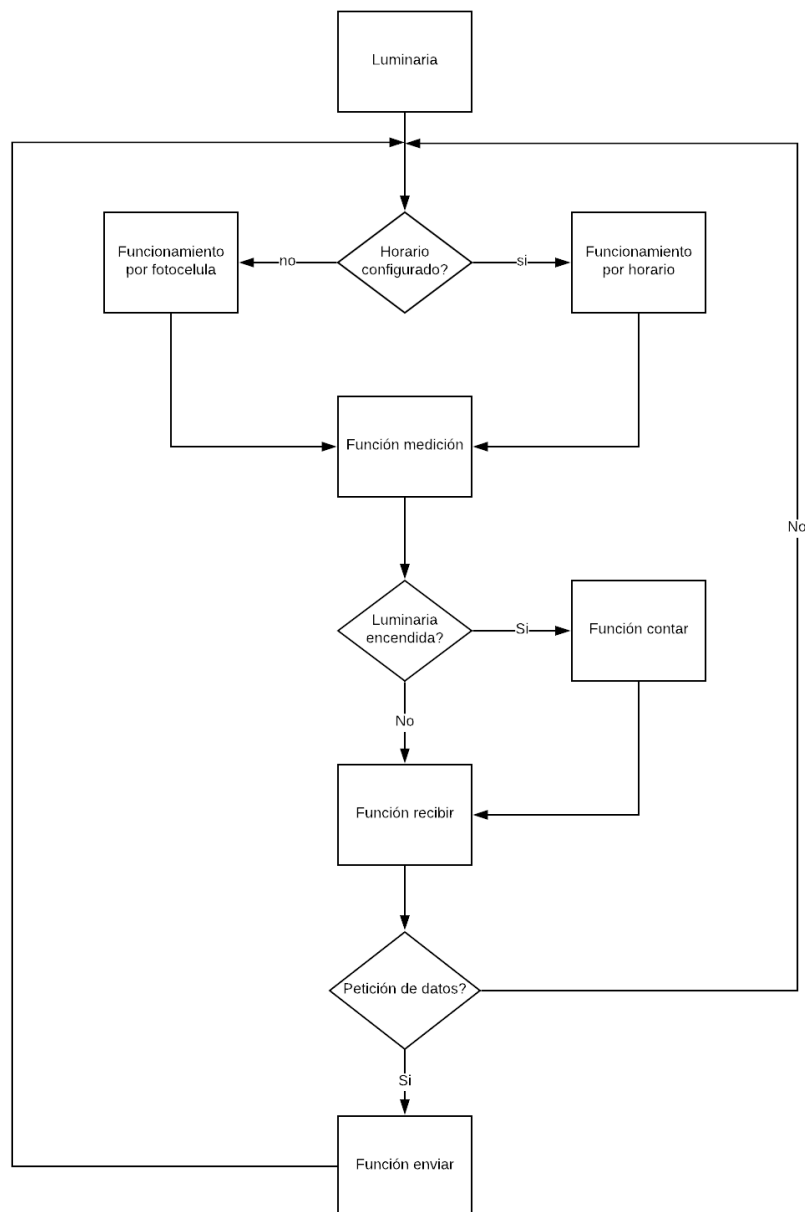


Figura 53 –Diagrama de Flujo de la Luminaria en Arduino

En la programación de la luminaria el primer paso es revisar si existe un horario configurado, en caso de que no, funcionara por medio de la fotocélula, esto quiere decir que sensa el nivel de luminosidad y dependiendo del mismo la luminaria se enciende o apaga automáticamente.

En caso de que el coordinador le haya enviado un horario de configuración, la luminaria ignora la fotocélula y solamente enciende y apaga en los horarios configurados.

Cualquiera sea el caso, la luminaria estará continuamente midiendo las variables (corriente, tensión, factor de potencia, potencia activa, potencia aparente y temperatura).

En caso de que exista un consumo de corriente, esto quiere decir que la luminaria se encendió, se ejecuta la función “contar”, que básicamente lo que hace es ir acumulando los valores medidos desde que la luminaria se encendió hasta que la luminaria se apagó. De esta forma se puede obtener los valores acumulados de energía consumida promedio, el factor de potencia promedio y tiempo total que estuvo encendida la luminaria.

Ya sea que la luminaria haya estado encendida o no, el siguiente paso es ejecutar la función recibir, que es la encargada de obtener los datos que el coordinador le envía. Si el coordinador envió la opción 3 "Obtener datos" la luminaria ejecuta la función enviar, que envía los datos guardados de energía, factor de potencia, PWM actual, tiempo y temperatura.

Acceso remoto

En este punto se estableció una conexión remota mediante internet para acceder al controlador de segmento y poder monitorear el mismo desde cualquier sitio y en cualquier momento.

Se debe definir antes que todo, que se entiende por acceso remoto.

(Perez Porto & Merino, 2017) La idea de acceso remoto se emplea en el terreno de la informática para nombrar a la posibilidad de realizar ciertas tareas en una computadora (ordenador) sin estar físicamente en contacto con el equipo. Esto es posible gracias a programas informáticos que permiten trabajar con la computadora a distancia.

El acceso remoto, por lo tanto, consiste en acceder a una computadora a través de otra diferente. De este modo, las acciones que se llevan a cabo en una computadora también se ejecutan en la otra.

Es importante tener en cuenta que el acceso a distancia exige que los dos equipos cuenten con el mismo software de administración remota. Esto, sumado a un sistema de permisos y autorizaciones, hace que la tarea pueda concretarse y que resulte segura.

Remote Desktop Protocol (RDP)

(Microsoft, 2018) Para el desarrollo del acceso remoto se optó por utilizar RDP que es un protocolo propietario desarrollado por Microsoft que permite la comunicación en la ejecución de una aplicación entre una terminal (mostrando la información procesada que recibe del servidor) y un servidor Windows (recibiendo la información dada por el usuario en el terminal mediante el ratón o el teclado).

El modo de funcionamiento del protocolo es sencillo. La información gráfica que genera el servidor es convertida a un formato propio RDP y enviada a través de la red al terminal, que interpretará la información contenida en el paquete del protocolo para reconstruir la imagen a mostrar en la pantalla del terminal. En cuanto a la introducción de órdenes en el terminal por parte del usuario, las teclas que pulse el usuario en el teclado del terminal así como los movimientos y pulsaciones de ratón son redirigidos al servidor, permitiendo el protocolo un cifrado de los mismos por motivos de seguridad. El protocolo también permite que toda la información que intercambien cliente y servidor sea comprimida para un mejor rendimiento en las redes menos veloces.

Este servicio utiliza por defecto el puerto TCP 3389 en el servidor para recibir las peticiones. Una vez iniciada la sesión desde un punto remoto el ordenador servidor mostrará la pantalla de bienvenida de Windows, no se verá lo que el usuario está realizando de forma remota.

(Microsoft Support, 2018) El protocolo de escritorio remoto se basa y es una extensión de la familia de T-120 de los estándares de protocolo. Un protocolo compatible con multicanal permite canales virtuales independientes para transportar datos de presentación, comunicación de dispositivos serie, información de licencia, datos cifrados altamente (teclado, actividad del mouse) y así sucesivamente. Como una extensión del

núcleo del protocolo T.Share RDP, varias otras capacidades se conservan como parte de RDP, como los elementos arquitectónicos necesarios para soportar multipunto (sesiones con varios participantes). Entrega de datos multipunto permite que los datos de una aplicación se entreguen en "tiempo real" en varias partes sin tener que enviar los mismos datos para cada sesión de forma individual (por ejemplo, pizarras virtuales).

Además, RDP está diseñado para admitir muchos tipos diferentes de topologías de red (como ISDN (RDSI), POTS y muchos protocolos de LAN como IPX, TCP/IP, NetBIOS y así sucesivamente). Sólo se ejecutará la versión actual de RDP sobre TCP/IP.

RDP fue desarrollada para ser completamente independiente de su pila de transporte subyacente, en este caso TCP/IP. RDP, siendo completamente independiente de su pila de transporte, significa que se pueden agregar otros controladores de transporte para otros protocolos de red a medida que crecen las necesidades de los clientes para ellos, con pocos o ningún cambio significativo en las partes fundamentales del protocolo. Éstos son elementos clave para el rendimiento y la capacidad de ampliación de RDP en la red.

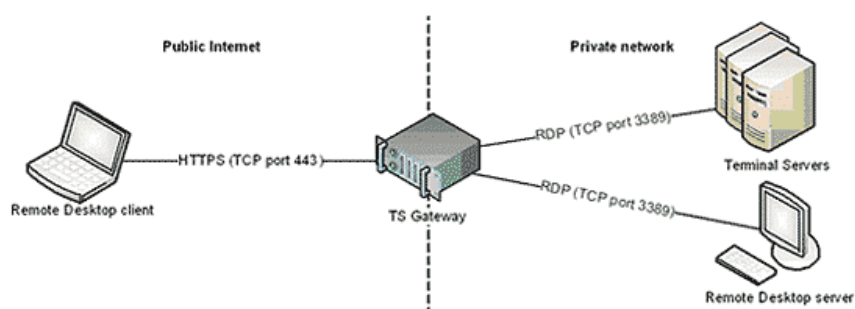


Figura 54 –Esquema de RDP

VNC (Virtual Network Computing)

(GEOTEK DATENTECHNIK, 2018) **VNC** es un programa de software libre basado en una estructura cliente-servidor que permite observar las acciones del ordenador servidor remotamente a través de un ordenador cliente. VNC no impone restricciones en el sistema operativo del ordenador servidor con respecto al del cliente: es posible compartir la pantalla de una máquina con cualquier sistema operativo que admita VNC conectándose desde otro ordenador o dispositivo que disponga de un cliente VNC portado.

La versión original del **VNC** se desarrolló en el Reino Unido, concretamente en los laboratorios AT&T Olivetti Research Laboratory, en Cambridge. El programa era de código abierto, por lo que cualquiera podía modificarlo, y existen hoy en día varios programas para el mismo uso. Muchos derivados modernos de él son software libre con licencia GPL.

VNC es independiente de la plataforma, un cliente VNC de un sistema operativo pueden conectarse a un servidor VNC del mismo sistema operativo o de cualquier otro. Hay clientes y servidores tanto para muchos sistemas operativos basados en GUI como para java. Varios clientes pueden conectarse a un servidor VNC al mismo tiempo. Los usos populares de esta tecnología incluyen ayuda técnica remota y acceso a los archivos presentes en el ordenador del trabajo desde la computadora de la casa o viceversa.

Hay una serie de variantes de VNC, que ofrecen, aparte de las funciones típicas de VNC, funciones particulares; por ejemplo, algunos están optimizados para Microsoft Windows;

otros disponen de transferencia de archivos, (que no es propiamente parte de VNC), etc. Muchos son compatibles (sin funciones adicionales) con el propio VNC en el sentido de que un usuario de una variante de VNC puede conectar con un servidor de otra, mientras que otros se basan en el código fuente de VNC, pero no son compatibles con el estándar VNC.

Funcionamiento

Un sistema de VNC se compone de un cliente, un servidor, y un protocolo de comunicación.

- El VNC servidor es el programa en el equipo que comparte su pantalla. El servidor de forma pasiva permite al cliente tomar el control de la misma.
- El VNC cliente (o espectador) es el programa que vigila, controla e interactúa con el servidor. El cliente controla al servidor.
- El VNC protocolo (RFB) es muy simple, basado en una primitiva gráfica del servidor al cliente ("Put a rectangle of pixel data at the specified X,Y position", en español "Póngase un rectángulo de datos de píxel en la posición X,Y especificada) y mensajes de eventos desde el cliente al servidor.

En el método normal de operación, un visor (espectador) se conecta a un puerto en el servidor (puerto por defecto 5900). Alternativamente, un navegador puede conectarse al servidor (dependiendo de la implementación) (puerto por defecto 5800). Y un servidor puede conectarse a un espectador en "modo de escucha" en el puerto 5500. Una de las ventajas del modo de escucha es que el sitio del servidor no tiene que configurar su cortafuegos para permitir el acceso en el puerto 5900 (o 5800), la responsabilidad recae en el espectador, lo cual es útil si el sitio del servidor no tiene conocimientos informáticos, mientras que del visor usuario se espera que sea más sabio.

El servidor envía pequeños rectángulos de la framebuffer para el cliente. En su forma más simple, el protocolo VNC puede utilizar una gran cantidad de ancho de banda, por lo que han sido diseñados varios métodos para reducir la sobrecarga de comunicación. Por ejemplo, hay varias *codificaciones* (métodos para determinar la manera más eficiente de transferencia de estos rectángulos). El protocolo VNC permite que el cliente y el servidor negocien la codificación que se utilizará. La forma más simple de codificación, que es apoyada por todos los clientes y servidores, es la codificación *cruda (raw)*, donde los datos se envían en píxeles en orden scanline de izquierda a derecha, y después de haberse transmitido la pantalla completa original, sólo se transfieren los rectángulos que cambien. Esta codificación funciona muy bien si sólo una pequeña porción de la pantalla cambia de un fotograma a otro (como un puntero del ratón se mueve en un escritorio, o el texto que se escriben en el cursor), pero las demandas de ancho de banda crecen radicalmente si una gran cantidad de píxeles cambia al mismo tiempo, como al desplazarse por una ventana o visualizar un vídeo a pantalla completa.

VNC por defecto usa puerto TCP $5900+N$, donde N es el número de la pantalla (por lo general: 0 para una pantalla física). Varias implementaciones también inician un servidor básico HTTP en el puerto $5800+N$ para proporcionar un visor VNC como applet Java, que permite la conexión fácil a través de cualquier navegador web con Java activado. Se puede utilizar distintas asignaciones de puerto siempre y cuando el cliente y el servidor estén configurados para ello.

El uso de VNC a través de Internet funciona bien si el usuario tiene una conexión de banda ancha en ambos extremos. Sin embargo, puede requerir avanzada NAT, cortafuegos así como configuración del router, como el reenvío de puertos para el paso de la conexión entrante y saliente a través.

La máquina donde se ejecuta el servidor VNC no necesita tener una pantalla física. *Xvnc* es el servidor Unix VNC server, que se basa en el estándar X server y es justamente este protocolo, el utilizado en la Raspberry Pi 3, para la realización del escritorio remoto en este proyecto.

Para aplicaciones *Xvnc* es un X "servidor" (es decir, muestra ventanas del cliente), y para los usuarios remotos de VNC es un servidor VNC. Las aplicaciones pueden mostrarse en *Xvnc* como si fueran una pantalla X normal, pero van a aparecer en cualquier conexión VNC espectadores más que en una pantalla física.

Además, la pantalla que muestra VNC no es necesariamente la misma pantalla vista por un usuario en el servidor. En computadores Unix/Linux que soporten múltiples sesiones simultáneas X11, VNC puede ser configurado para servir a una sesión particular existente de X11, o para iniciar una propia. También es posible ejecutar múltiples sesiones de VNC desde el mismo ordenador. En Microsoft Windows la sesión VNC servida (proporcionada) es siempre la sesión del usuario actual.

Por defecto, VNC no es un protocolo seguro. Como las contraseñas se envían en texto plano (como en telnet), el intento de romper o agrietar (cracking) la contraseña puede tener éxito si tanto la clave de cifrado y la contraseña cifrada es capturada desde una red. Por esta razón se recomienda utilizar una contraseña de al menos 8 caracteres. Por otro lado, también existe un límite de 8 caracteres en algunas versiones de VNC; si se envía una contraseña de más de 8 caracteres, los caracteres sobrantes se retiran y la cadena truncada es comparada con la contraseña.

Sin embargo, VNC puede ser tunelado a través de una conexión SSH o VPN que añada una capa extra de seguridad con un cifrado más seguro. Están disponibles Clientes SSH para todas las plataformas principales (y muchas plataformas más pequeñas también); se pueden crear túneles SSH a partir de clientes UNIX, Microsoft Windows, Macintosh (incluyendo Mac OS X y System 7 en adelante) - y muchos otros. Hay aplicaciones freeware que crean al instante túneles VPN entre ordenadores.

UltraVNC soporta el uso de un plugin de código abierto que cifra toda la sesión de VNC incluyendo autenticación de contraseña y transferencia de datos. También permite a la autenticación realizarse sobre la base de cuentas de usuario NTLM y Active Directory. Sin embargo, el uso de plugins de cifrado como lo hacen incompatible con otros programas VNC. RealVNC ofrece un alto nivel de cifrado como parte de su paquete comercial. Workspot ha publicado parches para VNC de cifrado AES.

Implementación

Para la implementación del escritorio remoto y el acceso desde internet se tuvo que realizar lo siguiente:

1. Instalar Xrdp en Raspberry Pi 3.
2. Configurar una IP de Lan estática.
3. Crear un dominio virtual.
4. Instalar el paquete del dominio virtual en Raspberry Pi 3.

5. Realizar un script para ejecutar dominio virtual en cada encendido de Raspberry Pi 3.
6. Configurar router.

1- Instalar Xrdp en Raspberry Pi 3

Para instalar Xrdp lo primero que se debe hacer es abrir un terminal en la Raspberry Pi 3.

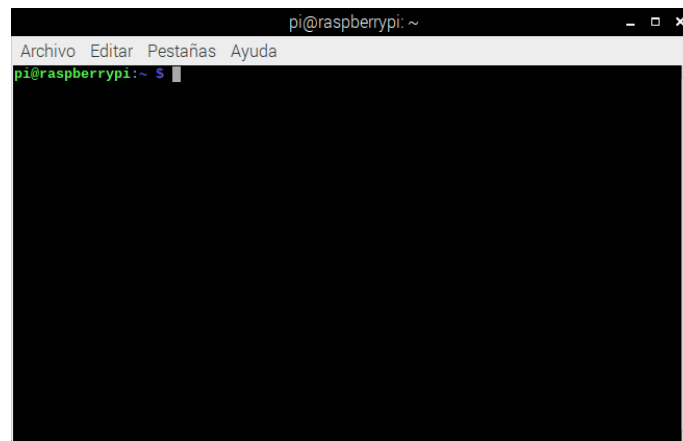


Figura 55 –Ventana de línea de comando Raspberry

Ingresamos los siguientes comandos:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install tightvncserver
```

2 - Configurar una IP de Lan estática

Se configura la IP estática de la siguiente manera:

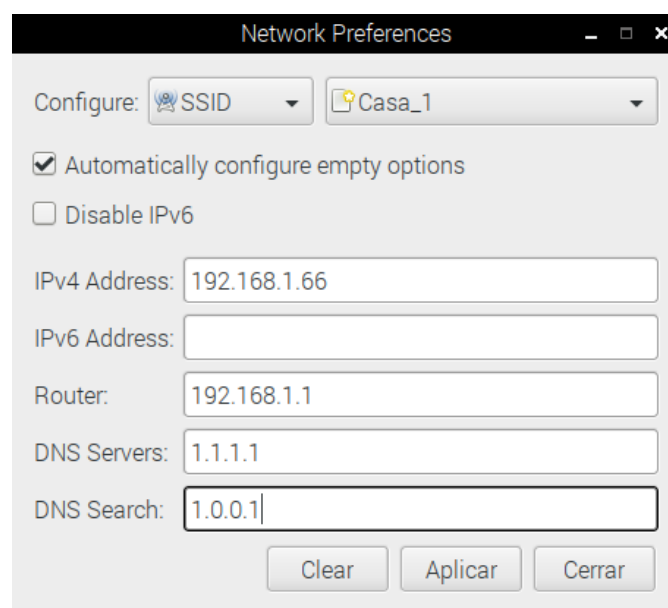


Figura 56 –Configuración de la IP

3- Crear un dominio virtual

La mayoría de los proveedores de internet no provee una dirección de IP estática, es por esto que se necesita alguna manera conocer la dirección IP de nuestro modem, de tal manera de poder lograr el acceso.

Existe un servicio gratuito (www.no-ip.com), que permite crear un dominio virtual y este asignarlo a una dirección IP donde está conectado el servidor, en nuestro caso la Raspberry Pi 3. Por lo tanto, lo primero que se debe hacer es crear una cuenta nueva en este servicio y escoger un subdominio, se creó el dominio gestionluminaria.hopto.org.

Una vez que se tiene creada y configurada la cuenta, se pasa a trabajar en la Raspberry. La idea es que, periódicamente, la Raspberry detecte la dirección IP pública a la que está conectado el modem, y la envíe al servicio no-ip para actualizar el dominio virtual. Para esto, se debe instalar el paquete no-ip en la Raspberry.

4- Instalar el paquete del dominio virtual en Raspberry Pi 3

Se abre un terminal y se ejecutan los siguientes comandos:

```
mkdir no-ip
cd no-ip
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
tar -zxvf noip-duc-linux.tar.gz
cd noip-2.1.9-1/
make
sudo make install
```

Al empezar la instalación, solicita los datos de la cuenta de no-ip, verifica el dominio virtual creado y pregunta el tiempo de actualización de la dirección ip externa. Se pueden dejar todos los valores por defecto tal cual, y continuar hasta finalizar la instalación.

Se puede iniciar el servicio ejecutando "sudo /usr/local/bin/noip2" pero lo ideal es que inicie automáticamente en el arranque de la Raspberry.

5- Realizar un script para ejecutar dominio virtual en cada encendido de Raspberry Pi 3

Primero se debe crear el archivo /etc/init.d/noip2 con el siguiente comando:

```
sudo nano /etc/init.d/noip2
```

Dentro del archivo se escribe el siguiente script:

```
#!/bin/bash
### BEGIN INIT INFO
# Provides: Servicio No-IP
# Required-Start: $syslog
# Required-Stop: $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: arranque automático para no-ip
# Description:
#
### END INIT INFO
sudo /usr/local/bin/noip2
```

Lo siguiente será guardar el archivo, dar permisos de ejecución y colocarlo en la cola de arranque.

```
sudo chmod +x /etc/init.d/noip2
sudo update-rc.d noip2 defaults
```

De esta manera se garantiza que el dominio virtual creado en no-ip tenga siempre asignada nuestra dirección IP externa del modem.

6- Configurar router

Para finalizar se debe configurar el servidor virtual en el router.

Los servidores virtuales se pueden usar para configurar servicios públicos en nuestra LAN. Un servidor virtual se define como un puerto de servicio, y todas las solicitudes desde Internet a este puerto de servicio se redirigirán a la computadora especificada por la IP del servidor. Cualquier PC que se use para un servidor virtual debe tener una dirección IP estática o reservada porque su dirección IP puede cambiar cuando se usa la función DHCP.

Como se vio, los puertos VNC se encuentran en los 5900+N en donde N es la cantidad de pantallas. Se debe también habilitar el puerto 80 (HTTP) para que la Raspberry pueda conectarse con el servicio no-ip. Por lo tanto, en el router se configura lo siguiente:

ID	Service Ports	IP Address	Protocol	Status	Modify
1	5900-5910	192.168.1.66	ALL	Enabled	Modify Delete
2	80	192.168.1.66	TCP	Enabled	Modify Delete

Figura 57 – Imagen de Configuración del Router

Por último se debe configurar un servidor DMZ. La función de host DMZ permite que un host local se exponga a Internet para un servicio de propósito especial. El host de DMZ reenvía paquetes de todos los servicios. Al igual que en el servidor virtual la PC configurada como host DMZ debe tener desactivada su función de cliente DHCP y debe tener asignada una nueva dirección IP estática, ya que su dirección IP puede cambiar al usar la función DHCP.

La configuración DMZ se realizó de la siguiente manera:

Figura 58 – Configuración de IP

De esta manera se tiene acceso por escritorio remoto vía internet, como se ve en las siguientes capturas:

Primero se debe ingresar a la aplicación de Windows “Escritorio remoto” y se coloca la IP pública de nuestro modem, que de acuerdo al procedimiento realizado puede obtenerse de nuestro server virtual ingresando a la página www.no-ip.com.

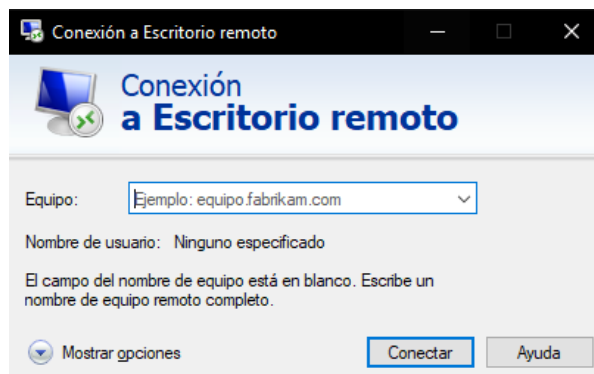


Figura 59 –Ventana del Escritorio Remoto

En la siguiente ventana, se ingresa el usuario y contraseña de nuestra Raspberry.

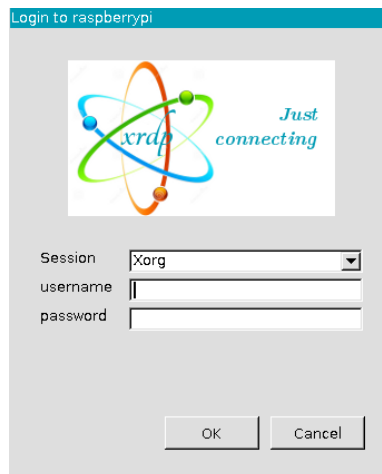


Figura 60 –Ventana de RDP en Raspberry

Y finalmente, se logra el acceso a nuestra Raspberry y se puede hacer uso del software de gestión.

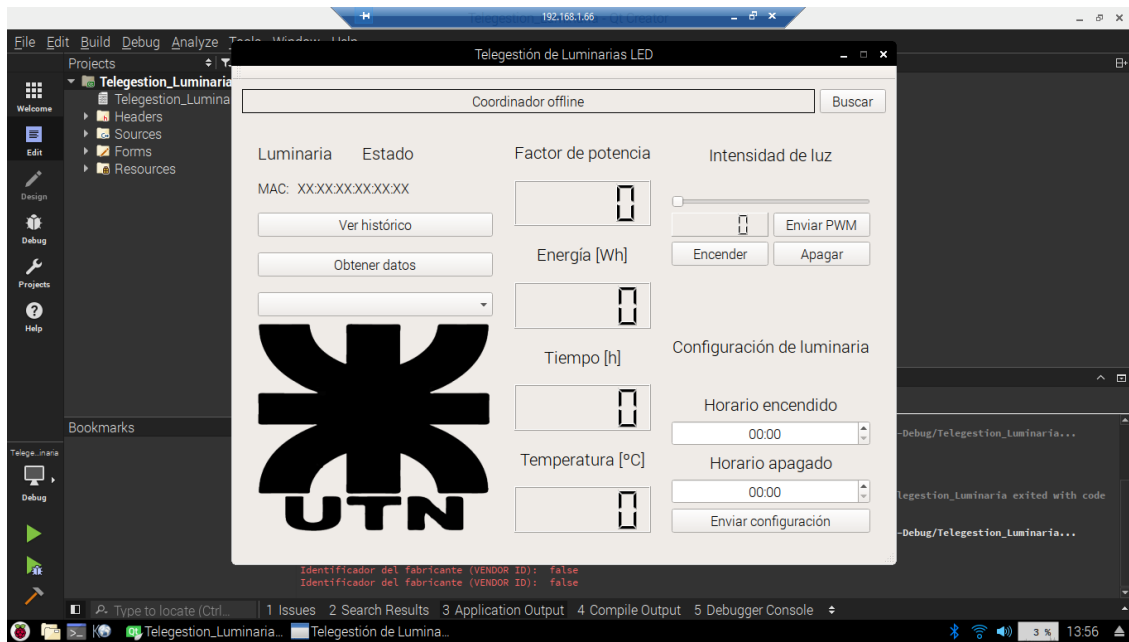


Figura 61 –Vista de la ventana de Raspberry

Aclaración

Cabe destacar que la creación del dominio virtual se aplica en el caso de que el proveedor de internet no seas capaz de proporcionar una IP estática para nuestro sistema. En caso de que poder contar con una IP estática no será necesaria la creación de un dominio virtual. Simplemente se configura el router y la IP en la Raspberry, y ya se obtiene el acceso.

Diseño completo

Controlador de segmento

Como ya se ha mencionado anteriormente el controlador de segmento está formado por una Raspberry Pi 3 que alberga el software de control y permite el acceso remoto, y un Arduino UNO que posibilita la comunicación inalámbrica mediante el protocolo Xbee.

Los equipos se colocaron dentro de una caja estanca en donde se agregaron 2 conexiones, una ficha hembra del tipo Plug hueca para alimentar la Raspberry Pi 3 y una ficha hembra RJ45 para conectar este mismo equipo a un router, modem o switch.

Las especificaciones son las siguientes:

Consumo	2,25 [Wh]
Medidas	115x165x80 [mm]
Peso	384 [g]

Tabla 14 – Características del Controlador de Segmento

Exterior

A continuación se encuentran fotos del controlador de segmento en donde se pueden observar las conexiones de alimentación y de red.



Figura 62 – Vista Superior del Coordinador



Figura 63 – Vista Lateral del Coordinador



Figura 64 –Vista Frontal del Coordinador



Figura 65 –Vista Completa Coordinador-Router

Interior

Se procederá a observar cómo se encuentran los equipos en el interior de la caja estanca. En la parte inferior (no visible en la foto) se encuentra la Raspberry Pi 3. Se utilizó una placa para poder separar y atornillar la Raspberry Pi 3 con el Arduino. También se puede visualizar la conexión de alimentación para la Raspberry Pi 3 y la conexión Ethernet.

El Arduino se conecta por USB a la Raspberry Pi 3, de esta forma se logra la comunicación entre ambos dispositivos, así como también, el Arduino obtiene su alimentación.

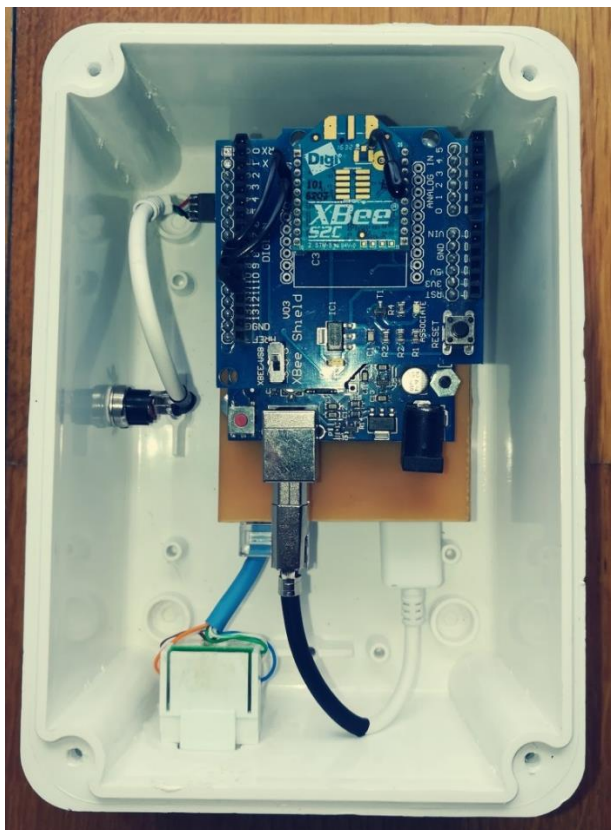


Figura 66 –Vista Interior del Controlador de segmento

Software

La interfaz gráfica se realizó con el software Qt, a continuación se podrá ver el procedimiento de utilización del mismo mediante capturas de pantalla.

1 - Al ejecutar el programa se puede ver la siguiente pantalla



Figura 67 –Ventana de Programa Principal Raspberry

2 – Si se presiona el botón “Obtener datos” se creara una lista con las luminarias que se encuentran online.

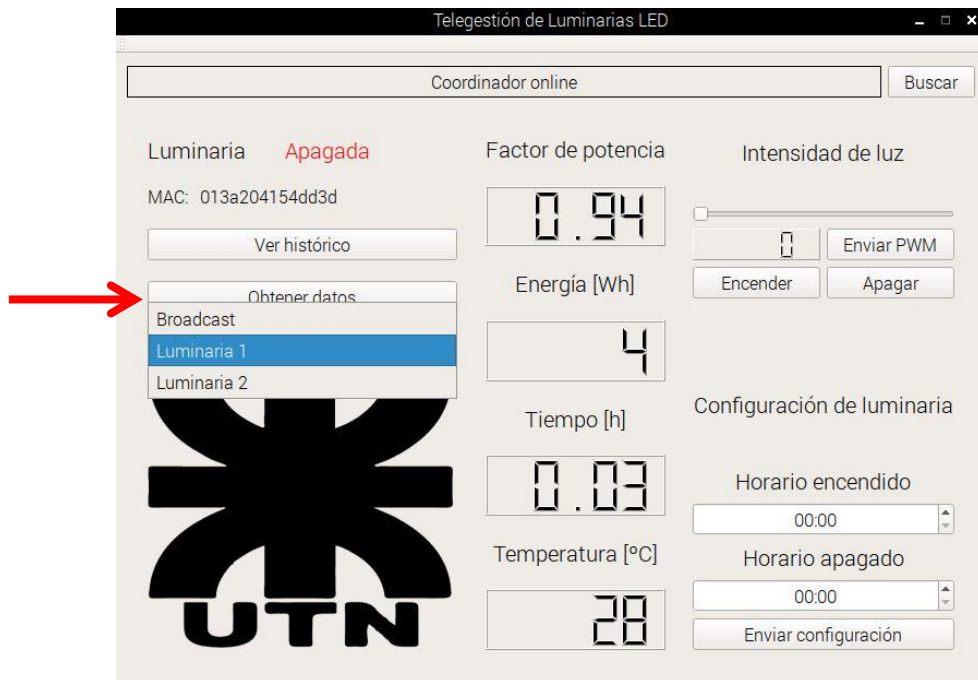


Figura 68 –Ventana de Programa Principal Raspberry – Obtener datos

3 – En la parte superior izquierda se pueden observar dos labels, en donde uno indica si la luminaria se encuentra encendida o apagada y el otro nos permite conocer la MAC del dispositivo Xbee con el cual se identifica a la luminaria. En este caso se puede ver que la luminaria de MAC 013a204154dd3d se encuentra encendida.



Figura 69 –Ventana de Programa Principal Raspberry – Estado de luminaria

4 – En la parte superior derecha del programa se encuentra una barra deslizadora que permite variar el valor de la intensidad de luz, abarcando los valores de 1 a 100 %, en donde 1 % es el mínimo de intensidad y 100 % el máximo.

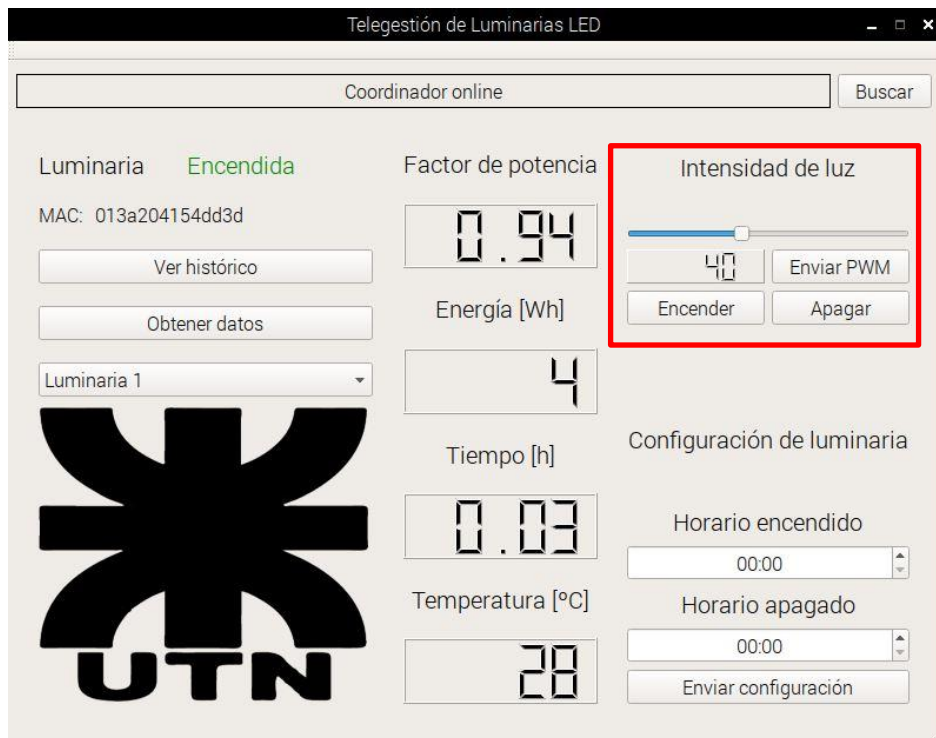


Figura 70 – Ventana de Programa Principal Raspberry – Intensidad de luz

5 – En el centro del programa se pueden ver 4 displays que proporcionan los datos de las mediciones (factor de potencia, energía, tiempo en que estuvo encendida y temperatura). En caso de seleccionar una luminaria específica se podrán ver los datos de esa luminaria en particular, y en caso de seleccionar la opción “Broadcast” se obtendrá un promedio de los datos de todas las luminarias.

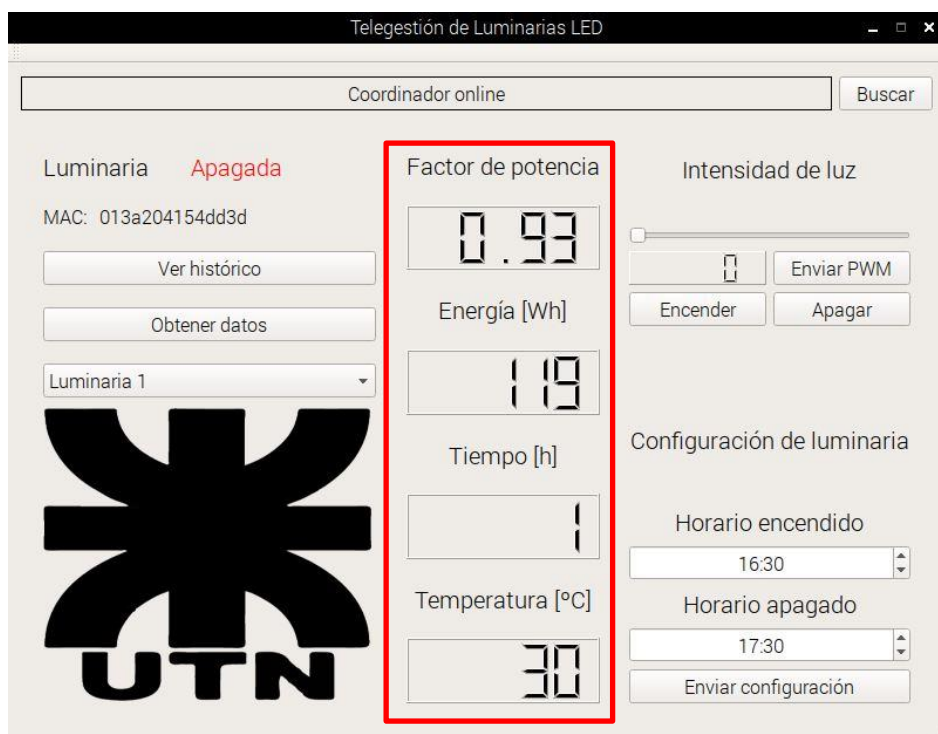


Figura 71 – Ventana de Programa Principal Raspberry – Datos mostrados

6 – En la esquina inferior derecha del programa se encuentra la configuración del horario de encendido y apagado de las luminarias. En caso de elegir una luminaria específica se configurara el horario enc/apa a esa luminaria en particular, y en caso de seleccionar la opción broadcast el horario de configuración se aplicara a todas las luminarias que se encuentren en línea en ese momento.

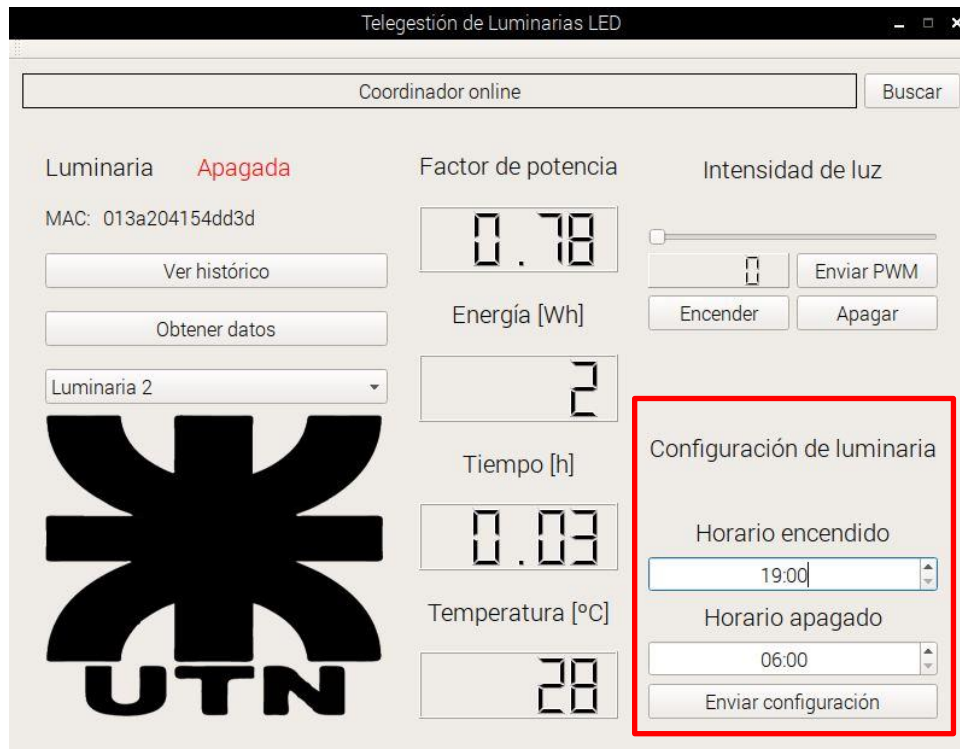


Figura 72 – Ventana de Programa Principal Raspberry – Configuración horario

7 – Si se presiona el botón “Ver histórico” se abrirá una ventana que contiene un archivo de texto con una lista que muestra con hora y fecha todo lo realizado con el programa.

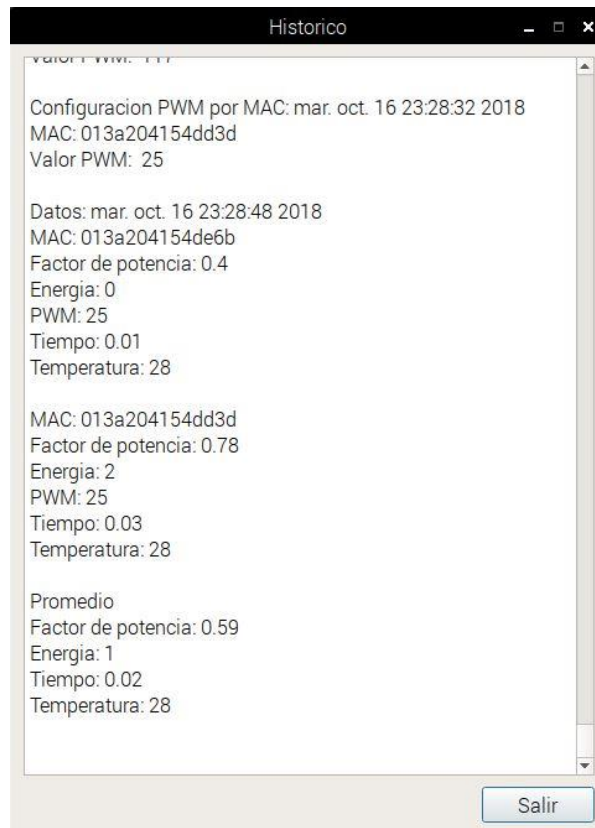


Figura 73 –Ventana de Programa Secundaria Raspberry – Histórico

Controlador de luminaria

Como ya fue descrito anteriormente el controlador de luminaria está formado por dos placas, en donde en una de ellas se encuentra la etapa de control y comunicación, y en la otra, la etapa de potencia. Ambas fueron colocadas dentro de una caja estanca con las siguientes especificaciones.

Consumo	6,35 [Wh]
Medidas	165x210x110 [mm]
Peso	1,22 [Kg]

Tabla 15 – Características del Controlador de Luminaria

Placa de control

A continuación se observara el diseño final del circuito de control, tanto el esquemático como el PCB.

Esquemático

El esquemático de la placa de control se dividió en etapas para que de esta forma sea más comprensible y se visualice mejor.

Fuente de alimentación

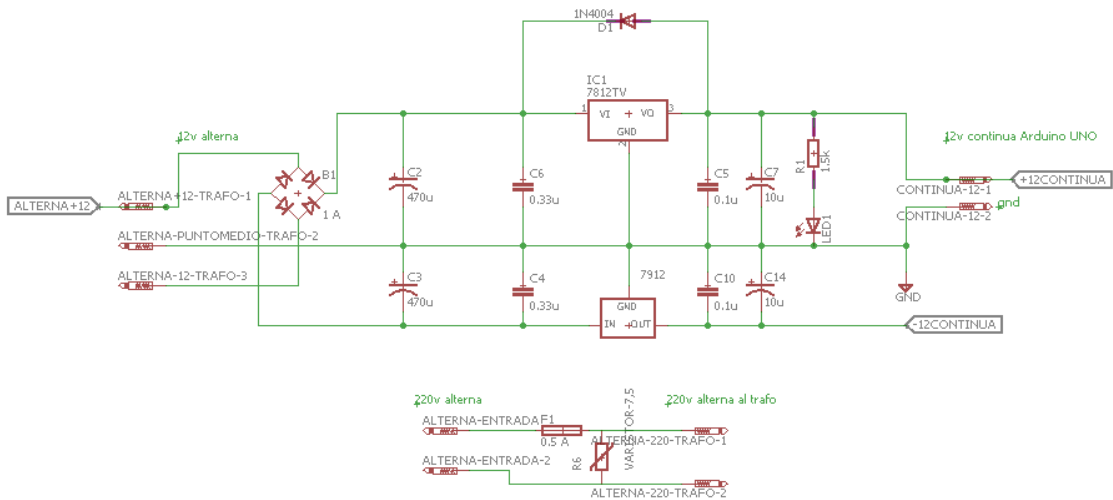


Figura 74 –Fuente y circuito de protección

Conexiones Arduino

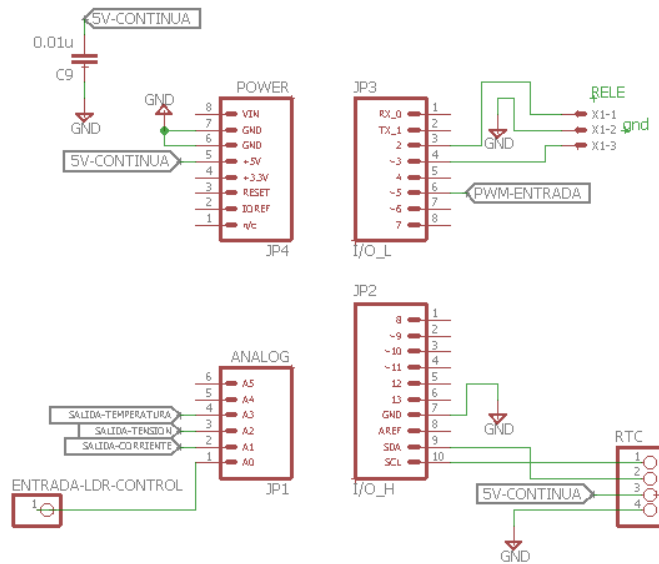


Figura 75 – Conexiones a Arduino

Circuito adaptador de tensión

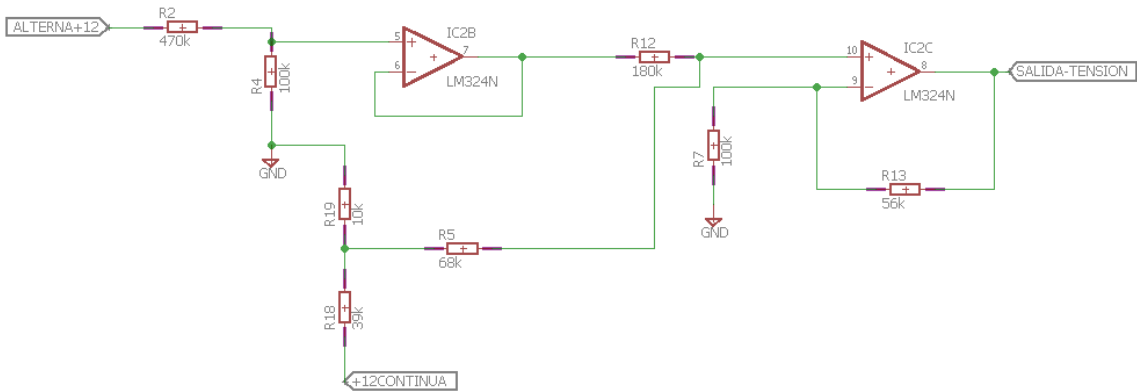


Figura 76 – Circuito adaptador de tensión

Circuito adaptador para PWM

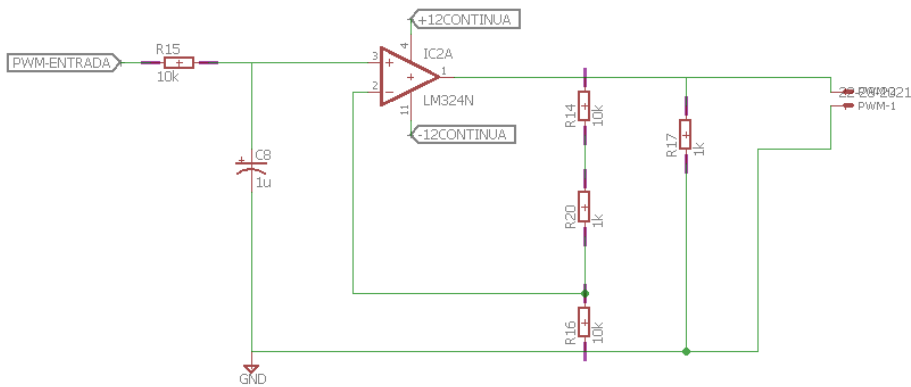


Figura 77 – Circuito conversor PWM a 0-10 [V]

Circuito medición de corriente

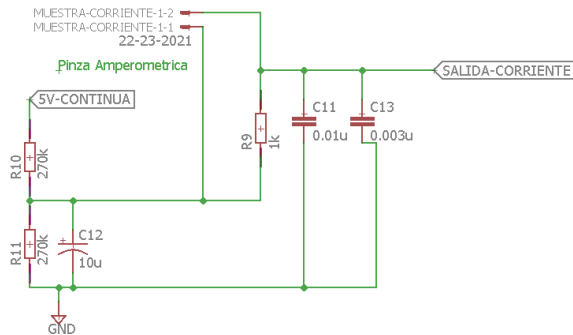


Figura 78 – Circuito adaptador de señal de corriente

Circuito sensado de temperatura

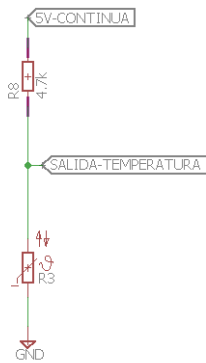


Figura 79 –Circuito adaptador de NTC

PCB

El PCB realizado es el siguiente:

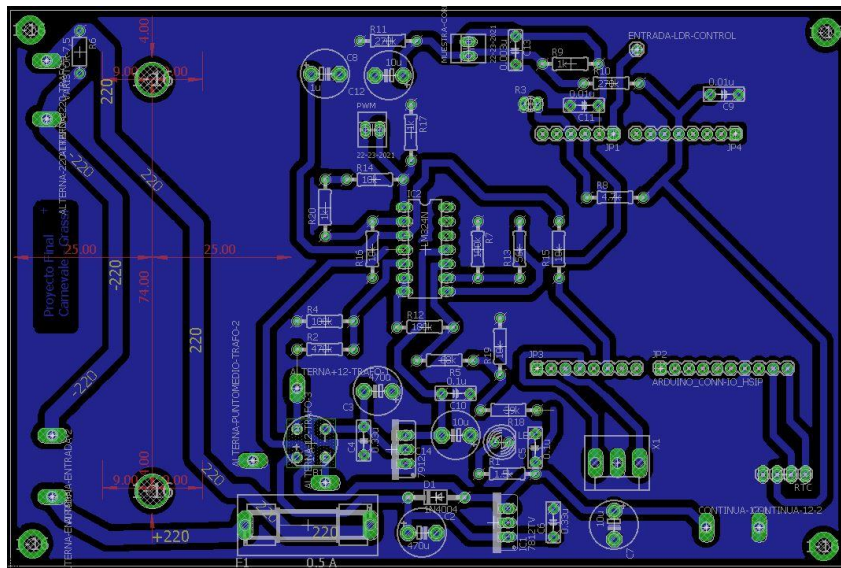


Figura 80 – Vista del PCB en Eagle

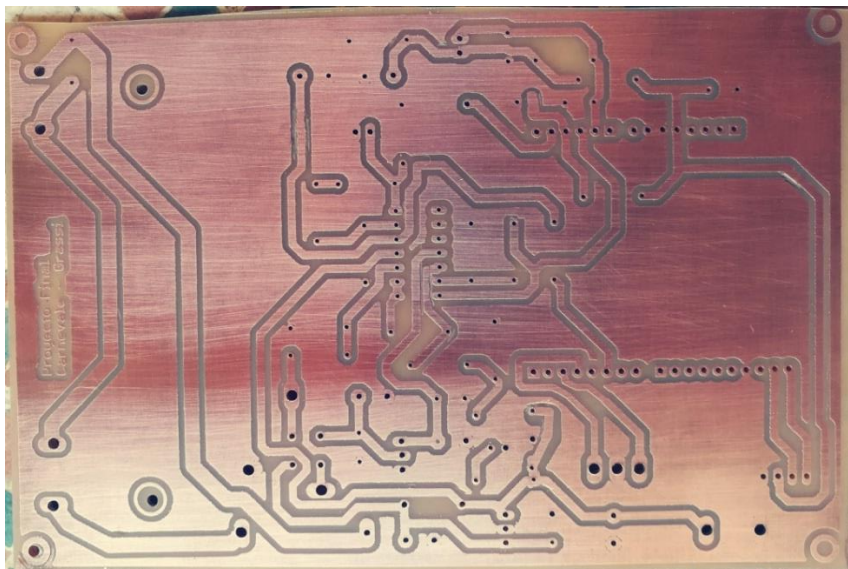


Figura 81 –Foto de la placa lista para soldar

PCB

El PCB diseñado se observa debajo.

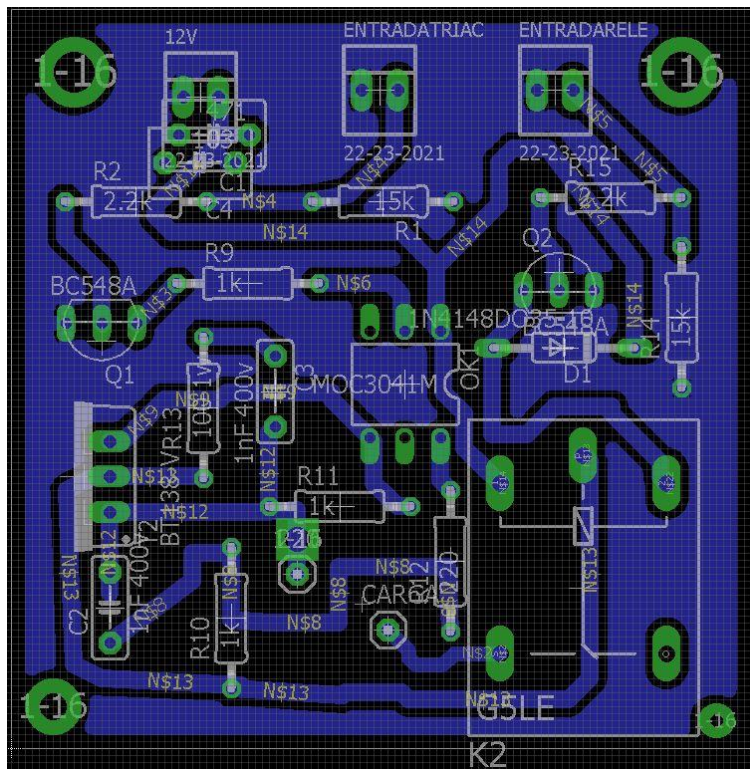


Figura 83 –Vista del pcb en Eagle

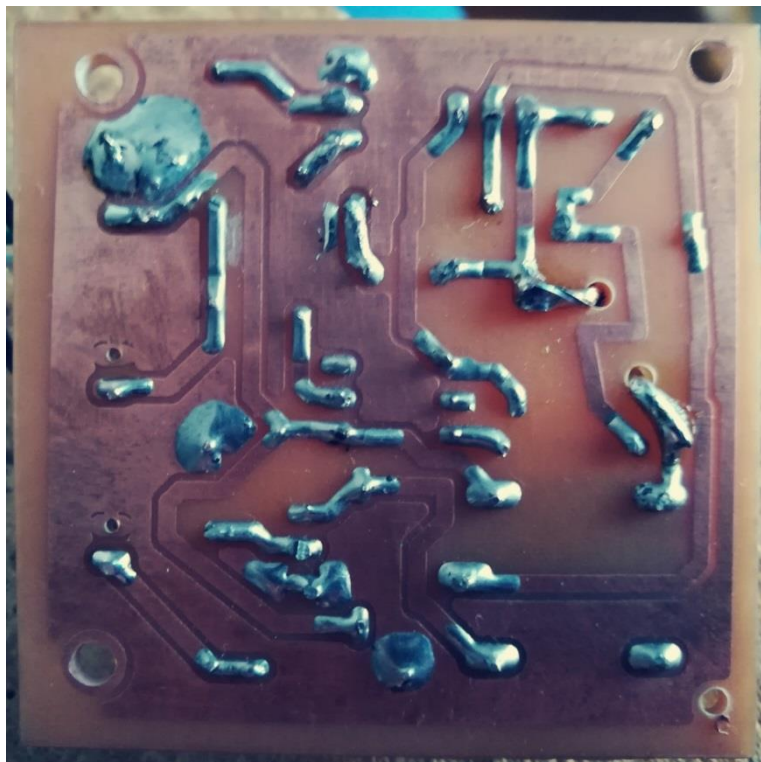


Figura 84 –Vista de la placa ya soldada

Exterior

A continuación se observara el exterior del controlador de luminaria y podrán visualizarse las siguientes conexiones:

- Conector alimentación 220 [V] macho del tipo Interlock, para alimentación de todo el circuito.
- Conector alimentación 220 [V] hembra del tipo IEC, para alimentación de la luminaria LED.
- Conector RCA hembra, para conexión del PWM de la luminaria LED.
- Conectar Plug estéreo, para conexión de la fotocélula.

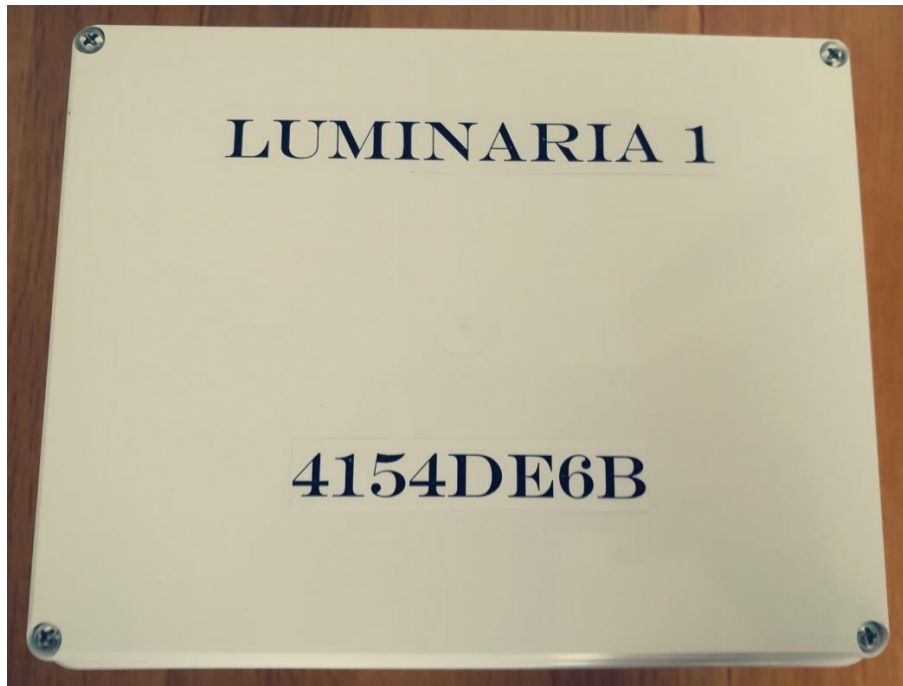


Figura 85 –Vista Superior del controlador de luminaria 1

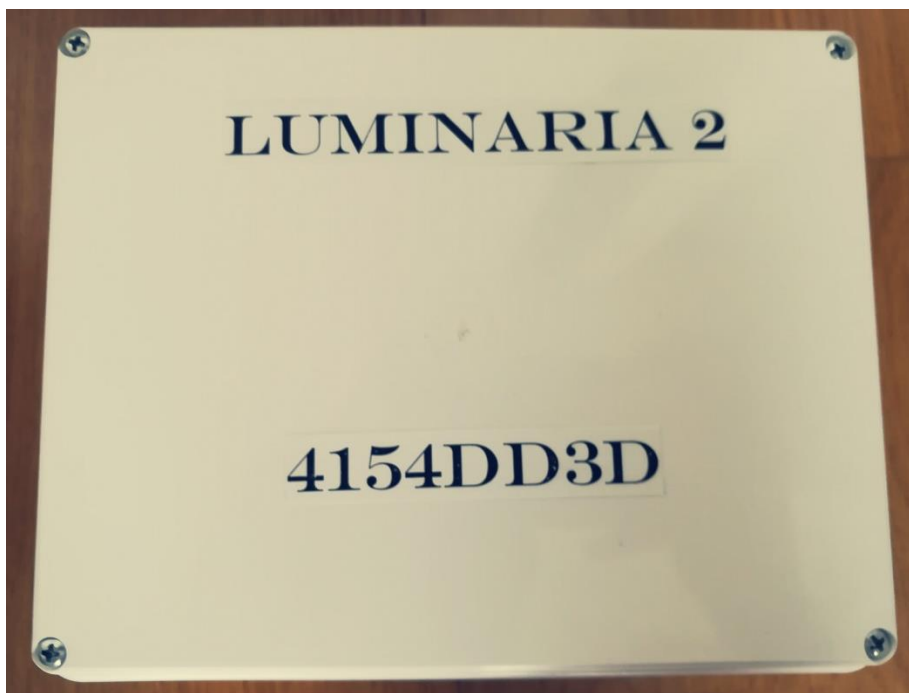


Figura 86 –Vista Superior del controlador de luminaria 2



Figura 87 –Vista Frontal de la luminaria 1



Figura 88 –Vista Lateral del controlador de luminaria 1



Figura 89 –Vista Completa del controlador de luminaria y la luminaria ya conectada

Interior

En el interior de la caja estanca se podrán observar las conexiones entre la placa de control y la placa de potencia. También puede observarse la disposición del transformador para la medición de corriente de la luminaria LED.

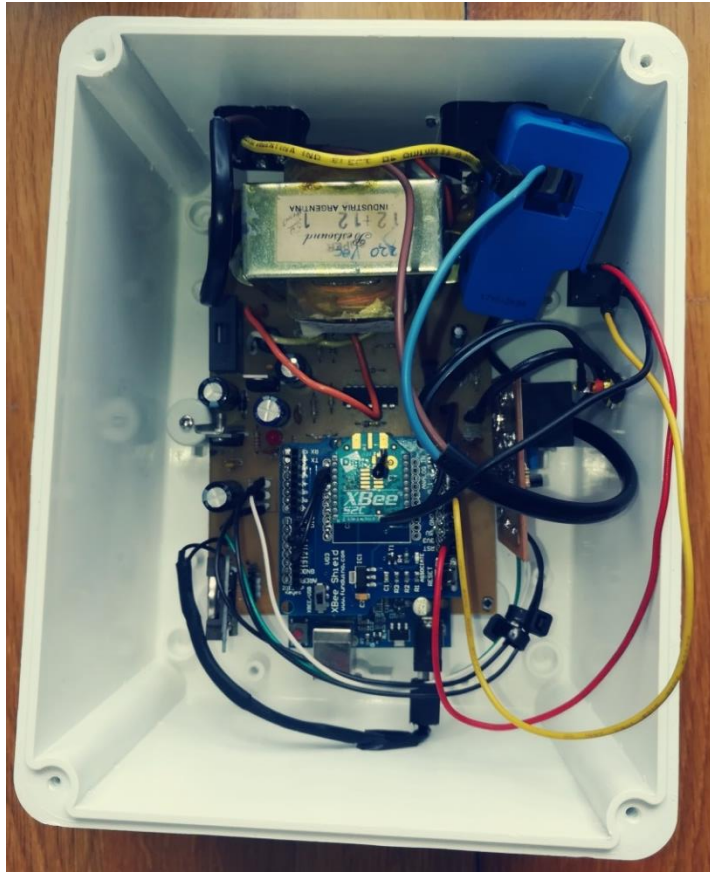


Figura 90 –Vista Superior interna del controlador de luminaria 1

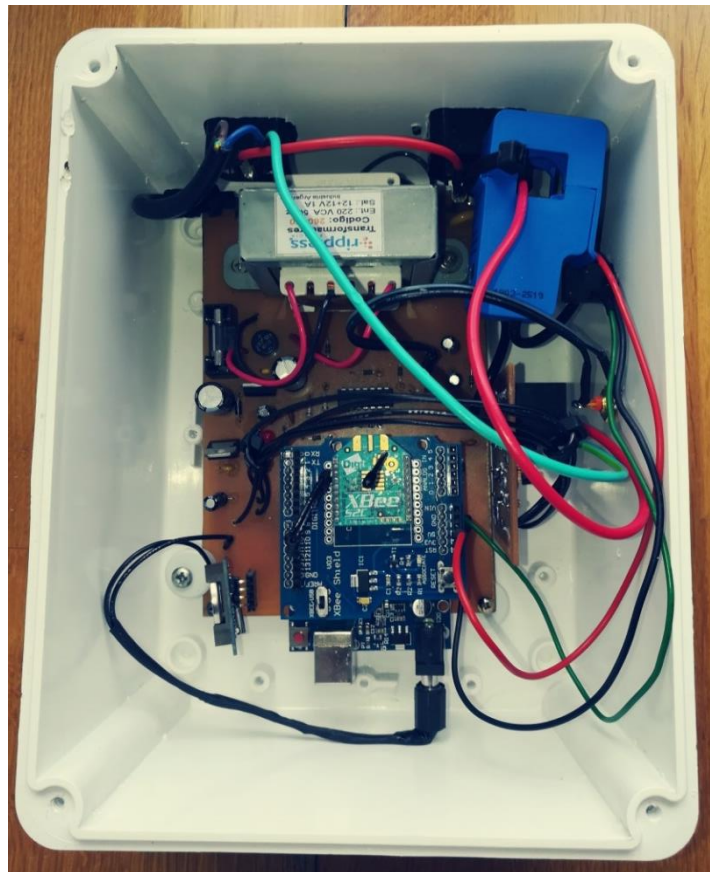


Figura 91 –Vista Superior interna del controlador de luminaria 2

Controlador de segmento y controladores de luminarias

A continuación se pueden observar los equipos desarrollados en conjunto.



Figura 92 –Vista de los tres equipos armados

Capítulo 3: Resultados

Clase del instrumento

Debido a que una de las características del proyecto realizado es la medición del consumo en Watts por hora, se realizó un estudio de la exactitud de las mediciones realizadas. Se midió corriente, tensión y desfase, y se compararon estas mediciones con un instrumento patrón (Analizador de energía MPR-63).

Cabe destacar que no hubo mayores inconvenientes en las mediciones de corriente y tensión, pero si en la medición de desfase entre ambas señales.

Para las mediciones lo que se realizó fue conectar la luminaria LED a uno de los controladores y mediante el software de control se fue variando el valor del PWM, desde el 100% hasta el 10% con saltos de 10%, lo que nos dio como resultado 10 mediciones de corriente, tensión y desfase. Este procedimiento se realizó 3 veces para poder calcular valores medios.

Se podrá observar en las mediciones que los valores de desfase sufren un error significativo con porcentajes de PWM que van de entre el 10 y el 50 %, y esto se debe a que en estos niveles de corrientes la señal se distorsiona drásticamente. Se detallara este fenómeno en las siguientes páginas.

A continuación podremos ver las tablas confeccionadas con las mediciones y la clase obtenida para cada caso.

TENSION														
PWM [%]	Tension patron [V] (1)	Tension patron [V] (2)	Tension patron [V] (3)	Valor medio [V]	Desviacion estandar [V]	Tension medida [V] (1)	Tension medida [V] (2)	Tension medida [V] (3)	Valor medio [V]	Desviacion estandar [V]	Error absoluto	Error relativo	Error porcentual	Clase del instrumento
10	215	215	214	214.667	0.577	214	213	212	213.000	1.000	-1.667	-0.008	-0.776	-0.940
20	215	216	214	215.000	1.000	212	214	213	213.000	1.000	-2.000	-0.009	-0.930	
30	216	216	213	215.000	1.732	215	213	211	213.000	2.000	-2.000	-0.009	-0.930	
40	216	215	214	215.000	1.000	214	213	212	213.000	1.000	-2.000	-0.009	-0.930	
50	214	213	214	213.667	0.577	213	211	211	211.667	1.155	-2.000	-0.009	-0.936	
60	214	214	214	214.000	0.000	212	213	212	212.333	0.577	-1.667	-0.008	-0.779	
70	214	214	215	214.333	0.577	212	213	213	212.667	0.577	-1.667	-0.008	-0.778	
80	215	214	212	213.667	1.528	214	213	209	212.000	2.646	-1.667	-0.008	-0.780	
90	214	215	214	214.333	0.577	213	212	212	212.333	0.577	-2.000	-0.009	-0.933	
100	215	215	214	214.667	0.577	213	213	212	212.667	0.577	-2.000	-0.009	-0.932	
Valor medio											-1.867	-0.009	-0.870	
Desviacion estandar											0.172	0.001	0.079	

Tabla 16 – Mediciones de tensión

Debido a que la variación del PWM no afecta directamente a la tensión medida, esta permanece prácticamente constante, por lo que la variación de tensión va a estar dada únicamente por variaciones en la tensión de la línea de 220 [V]. Podemos observar por lo

tanto, que el instrumento responde bastante bien para este tipo de mediciones, dando como resultado una clase aproximada de 0,940.

CORRIENTE														
PWM [%]	Corriente patron [mA] (1)	Corriente patron [mA] (2)	Corriente patron [mA] (3)	Valor medio [mA]	Desviacion estandar [mA]	Corriente medida [mA] (1)	Corriente medida [mA] (2)	Corriente medida [mA] (3)	Valor medio [mA]	Desviacion estandar [mA]	Error absoluto	Error relativo	Error porcentual	Clase del instrumento
10	165	166	165	165.333	0.577	190	190	190	190.000	0.000	24.667	0.149	14.919	4.384
20	224	223	223	223.333	0.577	240	240	240	240.000	0.000	16.667	0.075	7.463	
30	278	278	278	278.000	0.000	300	300	300	300.000	0.000	22.000	0.079	7.914	
40	339	338	337	338.000	1.000	360	360	360	360.000	0.000	22.000	0.065	6.509	
50	398	400	404	400.667	3.055	420	420	420	420.000	0.000	19.333	0.048	4.825	
60	465	462	467	464.667	2.517	490	490	490	490.000	0.000	25.333	0.055	5.452	
70	533	532	527	530.667	3.215	550	550	550	550.000	0.000	19.333	0.036	3.643	
80	594	597	598	596.333	2.082	620	620	630	623.333	5.774	27.000	0.045	4.528	
90	650	647	645	647.333	2.517	680	670	680	676.667	5.774	29.333	0.045	4.531	
100	649	647	645	647.000	2.000	680	670	680	676.667	5.774	29.667	0.046	4.585	
Valor medio											23.533	0.064	6.437	
Desviacion estandar											4.409	0.033	3.287	

Tabla 17 – Mediciones de Corriente

Para el caso de la corriente, por el contrario, se puede observar que los diferentes valores de PWM afectan directamente a la corriente medida y hace que esta varíe. Se puede observar que para valores de PWM de entre el 10 y el 40 % el error cometido en la medición de corriente es levemente más elevado que para valores de PWM de entre el 50 y 100%. Esto se debe a la distorsión armónica que la corriente sufre con valores pequeños de PWM. Por lo tanto, la clase para este caso nos da un valor de 4,384.

FACTOR DE POTENCIA														
PWM [%]	Factor de potencia patron (1)	Factor de potencia patron (2)	Factor de potencia patron (3)	Valor medio	Desviacion estandar	Factor de potencia medido (1)	Factor de potencia medido (2)	Factor de potencia medido (3)	Valor medio	Desviacion estandar	Error absoluto	Error relativo	Error porcentual	Clase del instrumento
10	0.76	0.77	0.76	0.76	0.01	0.5	0.51	0.51	0.507	0.006	-0.257	-0.336	-33.624	-26.460
20	0.86	0.86	0.87	0.86	0.01	0.69	0.7	0.71	0.700	0.010	-0.163	-0.189	-18.919	
30	0.91	0.91	0.91	0.91	0.00	0.79	0.79	0.8	0.793	0.006	-0.117	-0.128	-12.821	
40	0.93	0.93	0.92	0.93	0.01	0.85	0.85	0.85	0.850	0.000	-0.077	-0.083	-8.273	
50	0.95	0.95	0.95	0.95	0.00	0.9	0.89	0.89	0.893	0.006	-0.057	-0.060	-5.965	
60	0.96	0.96	0.96	0.96	0.00	0.92	0.93	0.92	0.923	0.006	-0.037	-0.038	-3.819	
70	0.97	0.96	0.97	0.97	0.01	0.94	0.94	0.94	0.940	0.000	-0.027	-0.028	-2.759	
80	0.97	0.97	0.97	0.97	0.00	0.96	0.96	0.96	0.960	0.000	-0.010	-0.010	-1.031	
90	0.97	0.97	0.97	0.97	0.00	0.97	0.97	0.97	0.970	0.000	0.000	0.000	0.000	
100	0.97	0.97	0.97	0.97	0.00	0.97	0.97	0.97	0.970	0.000	0.000	0.000	0.000	
Valor medio											-0.074	-0.087	-8.721	
Desviacion estandar											0.083	0.106	10.644	

Tabla 18 – Mediciones del coseno fi

Por último, para el caso de la medida de desfase entre las señales de corriente y tensión, se puede ver que se produce un error muy significativo entre los valores de PWM desde el 10 al 50%, que se debe a la distorsión armónica que sufre la señal de corriente, como se mencionó anteriormente. Por lo que para este caso y lo que terminara por definir la clase del instrumento ya que es la medida más inexacta, da un valor de 26,46.

Es un valor elevado de error, por lo que se propone como objetivo a mejorar en versiones posteriores.

Una posible solución sería corregir estos valores por software, ya que se conocen exactamente las condiciones en donde se producen.

A continuación se puede observar capturas realizadas con el osciloscopio (OWON SDS7102E), en donde se puede observar la degradación de la señal a medida que los valores de corriente disminuyen.

Con el PWM al 100 % se puede ver que los armónicos no son tan significativos y la señal no sufre tanta deformación.

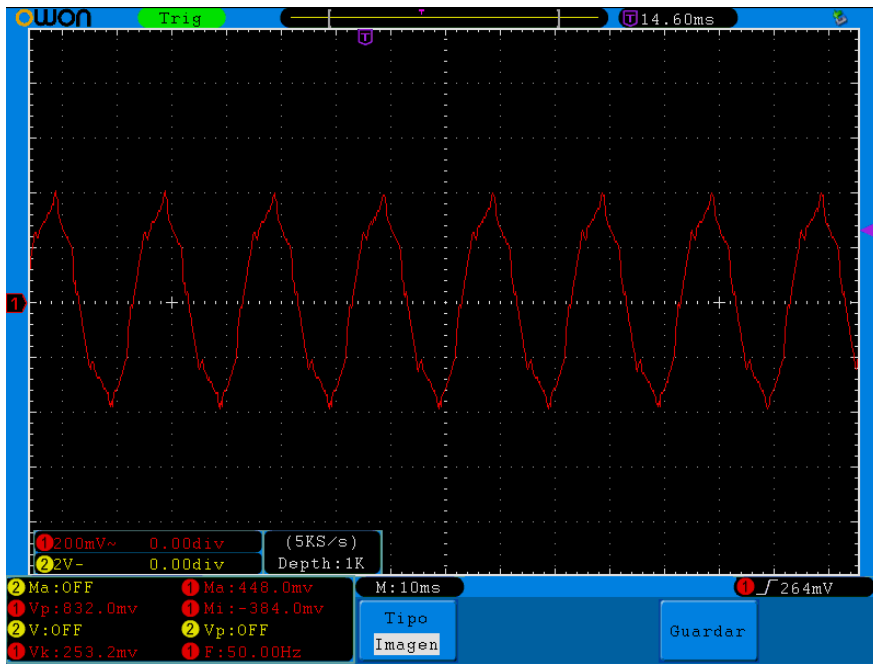


Figura 93 –Señal de dimerización al 100% - tiempo

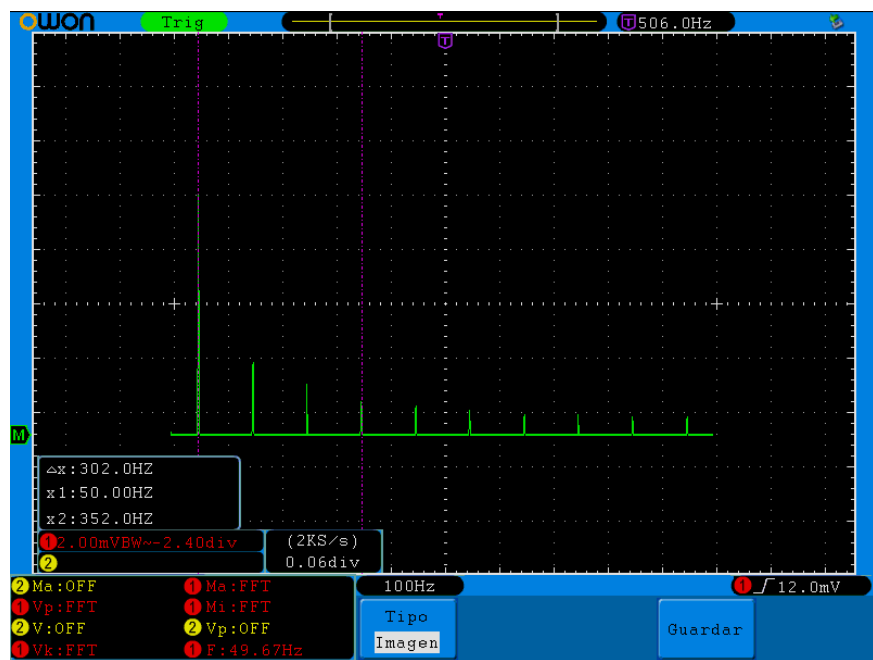


Figura 94 –Señal de dimerización al 100% - FFT

Con un PWM al 50 % se podrá ver como las armónicas empiezan a tener más influencia y la señal comienza a distorsionarse.

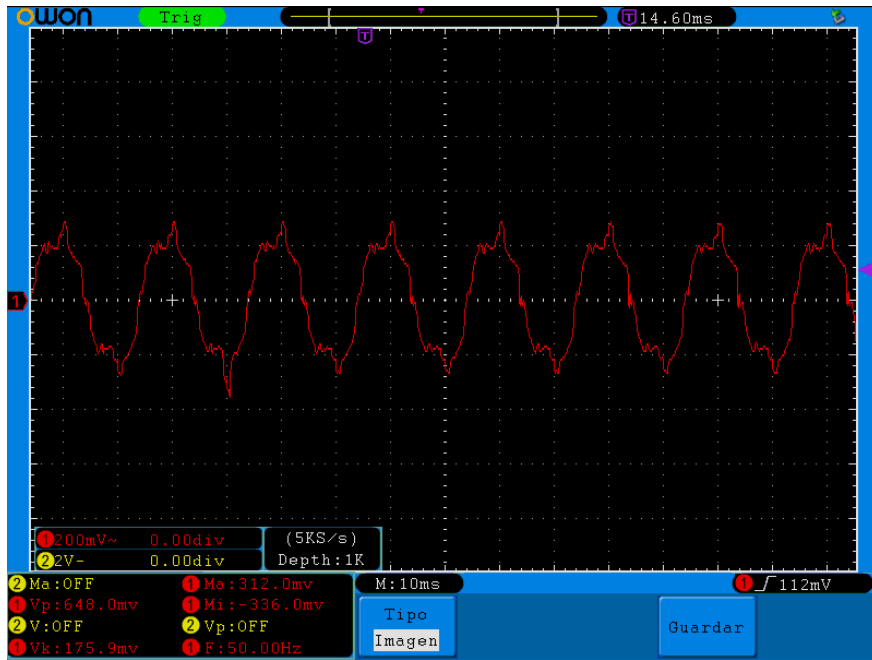


Figura 95 –Señal de dimerización al 50% - tiempo

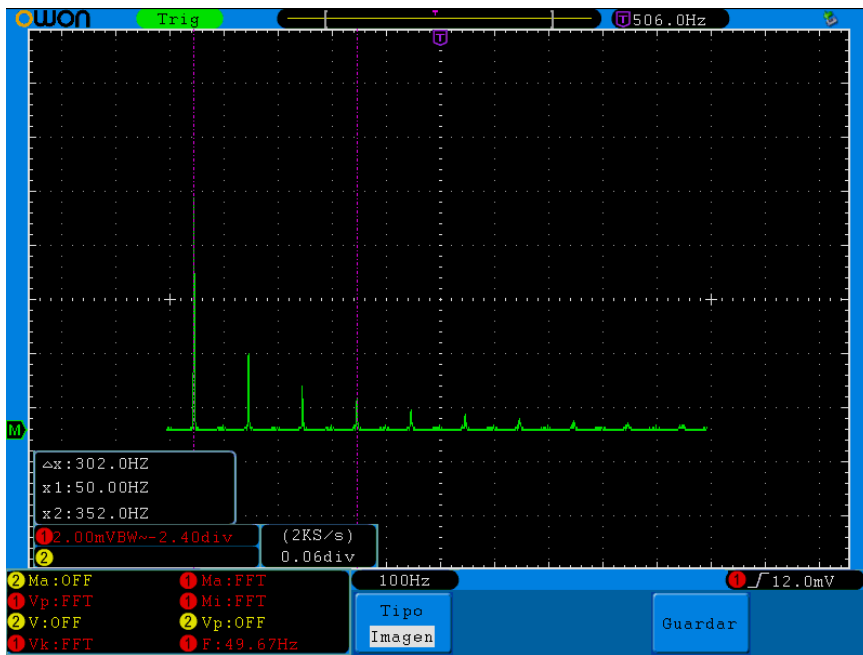


Figura 96 –Señal de dimerización al 50% - FFT

Con el PWM al 10% las armónicas pasan a ser muy significantes y la señal se distorsiona aún más y es por esta razón que las medidas de corriente y sobre todo la de desfase contienen un error mucho mayor a estos niveles.

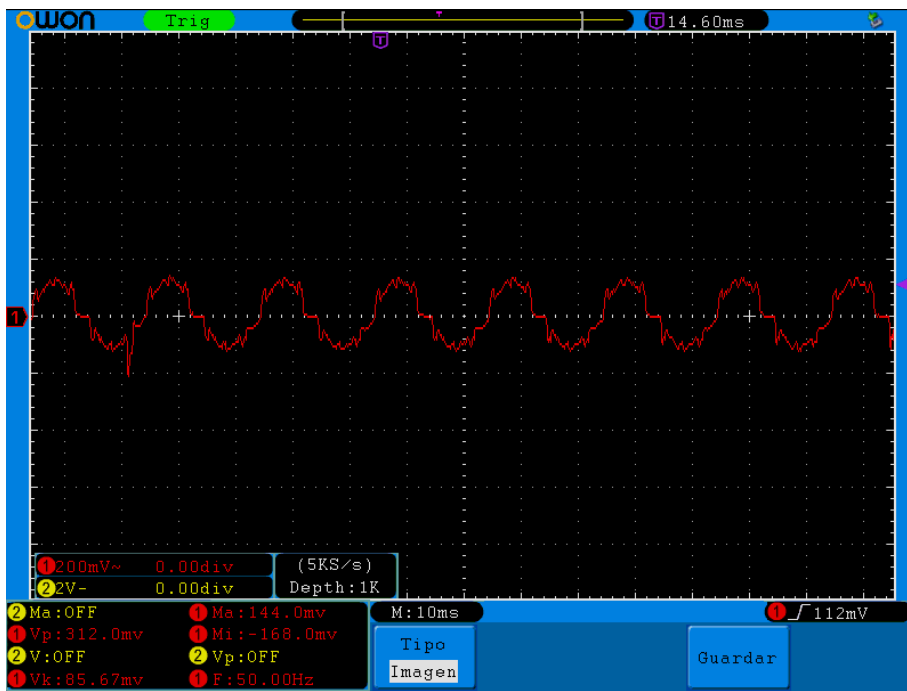


Figura 97 –Señal de dimerización al 10% - tiempo

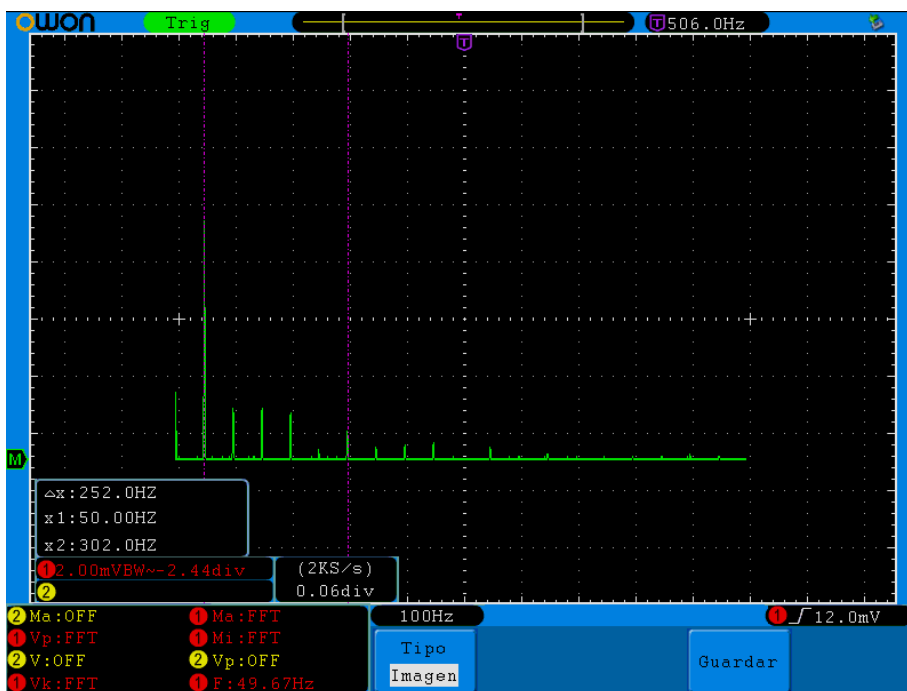


Figura 98 –Señal de dimerización al 10% - FFT

Capítulo 4: Análisis de costo

Introducción

Primero se detallara el tiempo y los recursos que se utilizaron para la fabricación de estos prototipos, luego se estudiara un ejemplo de proyecto a realizar a futuro, y por último la comparativa entre los sistemas actuales de distintas tecnologías de alumbrado público.

Dentro del análisis de costo se tendrá en cuenta el precio del KWh como fijo, y con un valor que corresponde a la provincia de Entre Ríos en los 200KWh iniciales que se consumen para uso residencial. El costo es de 2,6774 \$/KWh.

Vemos a continuación una fotografía de la información desglosada del precio del KWh para un residente de la ciudad de Paraná, Entre Ríos:



Información Desglosada	
Cargo Fijo Bimestral:	$(62 \times 109,3400) / 62 = 109,34$
Primeros 200 kWh:	$200 \times (62 \times 2,6774) / 62 = 535,48$
Siguientes 200 kWh:	$200 \times (62 \times 3,0760) / 62 = 615,20$
Siguientes 200 kWh:	$200 \times (62 \times 4,2610) / 62 = 852,20$
Excedentes 106 kWh:	$106 \times (62 \times 4,8643) / 62 = 515,62$

Figura 99 –Foto de información desglosada de boleta de electricidad - ENERSA

Además, los precios de los equipos se mostraran en valores finales. Esto quiere decir, el precio original más el IVA y la ganancia del vendedor.

Costos de desarrollo de los prototipos:

El desarrollo de los equipos fue un arduo trabajo que empezó por entender y realizar programas donde se envíen y reciban paquetes de información por medio de Arduino y XBee. El estudio de este protocolo, el envío, recepción, tratamiento de las tramas y estudio de las librerías que nos proporcionó Arduino para simplificar el trabajo, nos llevó dos meses de trabajo aproximadamente.

El siguiente paso fue el desarrollo del hardware correcto, lo que fue una evolución entre software de Arduino y hardware, teniendo que modificar muchas veces los prototipos, ya sea de componente, como de esquemático de circuito. El incursionar en los sensores de efecto Hall y luego cambiar de tecnología para sensar corriente demandó un tiempo prolongado. El problema principal es la falta de proveedores de componentes electrónicos en Entre Ríos, pero lo suplementamos mediante las compras de internet. El problema era que los tiempos de espera de los componentes era superior a los 7 días hábiles siempre, llegando a ser de hasta 14 días hábiles. Por lo tanto el desarrollo de la placa principal nos llevó cuatro meses de trabajo.

En el momento de realizar la placa que maneja la potencia del driver se tuvo en cuenta los tiempos necesarios para realizar las compras, es por esto que el tiempo se redujo y el desarrollo de la placa solo llevo dos semanas.

Para el desarrollo en Raspberry no había que esperar nada más que programar, es por esto que con muchas horas de desarrollo se logró la comunicación exitosa entre Raspberry/Arduino/XBee. Esto demandó aproximadamente dos meses más.

Por último, el ensamble de todos los prototipos que son, dos controladores de luminaria y un coordinador, conllevó dos semanas más.

En conclusión el desarrollo total se logró en un tiempo aproximado de nueve meses de trabajo.

Los costos de los materiales fueron los siguientes:

Producto	Precio (USD)	Cantidad	Precio total	Precio total (\$)
Capacitor multicapa 1uF	0,05859	6	0,35154	13,288212
Capacitor Electrolítico 470uF	0,0524	3	0,1572	5,94216
Resistencia metalfilm 1% 1/2w	0,02893	30	0,8679	32,80662
triac bt137	0,83526	1	0,83526	31,572828
relay tyco 8A 230v	5,9076	1	5,9076	223,30728
lm7812	0,45045	1	0,45045	17,02701
lm7912	0,5	1	0,5	18,9
bc548	0,04963	5	0,24815	9,38007
lm324	0,18	1	0,18	6,804
pin macho hembra	0,04	60	2,4	90,72
caja estanca 165x210x110	6,613	2	13,226	499,9428
caja estanca 115x165x80	5,29	1	5,29	199,962
Arduino Uno R3	5,29	3	15,87	599,886
Shield Arduino - Xbee	5,29	3	15,87	599,886
Xbee S2C Pro	74,1	3	222,3	8402,94
Raspberry Pi 3	26,45	1	26,45	999,81
Transformador 12+12	6,61	2	13,22	499,716
PCB FR4 100x150	3,3	3	9,9	374,22
RTC DS1302	2,96	2	5,92	223,776
Pila, estaño, cables conectores	5,22	3	15,66	591,948
Pinza SCT001	9,26	3	27,78	1050,084
TOTAL				14491,91898

Tabla 19 – Costos de materiales para los prototipos

En total se trabajó 22 días al mes a un promedio de 6 horas por día durante 9 meses, lo que da un total de 1188 horas. La hora se estimó en \$400, por lo tanto se obtiene un total de \$ 475.200. Equivale a que cada uno cobre \$26.400 por mes.

Por lo tanto, el costo total del diseño e implementación del proyecto, resulta en un total de \$ 489.692.

Ejemplo de proyecto futuro

En este apartado se lleva a cabo la elaboración de un presupuesto para la fabricación de 1.000 unidades del prototipo en una línea de fabricación.

Costos de la mano de obra (M.O.)

Para poder fabricar el producto se necesitara un Ingeniero Electrónico, un Ingeniero Industrial y un Técnico Electrónico, que desempeñarán las siguientes tareas:

El Ingeniero Industrial será el responsable de la supervisión de los trabajos y del control y suministro del material necesario para la fabricación y montaje de las placas.

El Ingeniero Electrónico se encargará de la fabricación de las PCBs y de realizar el control de calidad antes de que se monten los componentes sobre estas y después de la fabricación de cada producto.

El Técnico Electrónico se encargará del montaje de los componentes y de comprobar el funcionamiento del producto final.

Los costos laborales se calculan en base al mes de Octubre de 2018, donde tendrán una obra social, jubilación y trabajo de 8 horas diarias.

Para los dos ingenieros, el costo total para el empleador será alrededor de \$37.220,52 (USD 984), pero el empleado recibirá en limpio \$24.000 (USD 635). Mientras que el técnico tendrá un sueldo aproximado de \$31.200 (USD 825,4), y el empleado recibirá \$18.000 (USD 476,2)

Esto quiere decir que el costo por hora del ingeniero es de: \$ 211,5 (USD 5,6). En cambio para el técnico es de \$177,27 (USD 4,69).

El ingeniero industrial tendrá un tiempo de trabajo de 400 horas, mientras que el ingeniero electrónico y el técnico electrónico tendrán un tiempo de trabajos estimado en 1.500 horas.

Por lo tanto, en mano de obra habrá un costo de:

Ingeniero Industrial = 400 [h] x 211,5 [\$/h] = \$ 84.600

Ingeniero electrónico = 1500 [h] x 211,5 [\$/h] = \$ 317.250

Técnico electrónico = 1500 [h] x 177,27 [\$/h] = \$ 265.905

El costo total es de \$ 667.755 (USD 17.665), y el tiempo estimado de realización de estos 1.000 productos es de 1 año.

Costo de los materiales (M)

Material	Precio por unidad (\$)	Precio por cantidad (\$)
PCB FR4 1 faz 10x15	125	125.000
Componentes electrónicos	400	400.000
Transformador	250	250.000
Arduino Uno	200	200.000
Shield XBee	200	200.000
RTC DS1302	112	112.000
Xbee S2C	2.800	2.800.000
Caja estanca	250	250.000
Varios (cables y conectores)	120	120.000
estaño	2,35	2.350
Pila CR2032	15	15.000
Total	4.474,35	4.474.350

Tabla 20 – Costo de materiales – ejemplo

Como se puede observar el costo de materiales para la fabricación de un controlador de luminaria es de \$ 4.474,35 (USD 118,37). Lo que da un costo total de \$4.474.350 (USD 118.369).

Costo y mantenimiento del lugar de trabajo (L.T.)

Herramientas y servicios	Precio unitario	Precio total
Electricidad	4.000 (al mes)	48.000
PC	13.500	13.500
Estación de soldado	4.400	8.800
Repuesto para estación de soldado	1.000	2.000
Osciloscopio Hantek 20MHz	6.200	6.200
Reflector LED 50w	700	2.800
Xbee explorer USB	2.100	4.200
Mesa de trabajo	3.500	15.000
Total	35.400	157.000

Tabla 21 – Costo y mantenimiento de lugar de trabajo

Por lo tanto el costo total de todas las herramientas y el consumo eléctrico tendrán un total de \$ 157.000 (USD 4153,44). Además habrá gastos para alimentar al personal, teniendo en cuenta los días hábiles de trabajos de cada uno, tendrán una comida al día, lo que da un total de 425 almuerzos. Lo que será, a un costo de \$400 cada almuerzo, un total de \$170.000.

Costos totales

Los costos totales para la fabricación de 1.000 controladores de luminarias es de:

$$CT = M.O. + M. + L.T. + víveres = \$667.755 + \$4.474.350 + \$157.000 + \$170.000 = \$ 5.469.105$$

El costo por unidad será entonces de \$5.469,105 (USD 145).

Este costo unitario puede reducirse si se trabaja en cantidades mayores y con una mentalidad a gran escala, donde se tendrán más máquinas y herramientas como así también más personal, lo que producirá un costo total mayor, obviamente, pero por unidad se verá reducido.

El costo del controlador de segmento será desglosado a continuación:

Controlador de segmento	
Dispositivo	Precio (\$)
Arduino Uno R3	200
Shield XBee	200
XBee S2C Pro	2800
Raspberry Pi 3	2500
Caja estanca	150
Conectores y cables	100
Total	5950

Tabla 22 – Costo del controlador de segmento

Comparativa en base al mercado actual

La comparativa que se hará, será en base a los costos de las lámparas de alumbrado público basados en la tecnología del sodio, contra una tecnología más nueva del tipo LED, y la diferencia entre la tecnología LED con y sin telegestión de las luminarias.

Tecnología SODIO

Se usara de referencia una ciudad mediana de 35.000 luminarias, todas ellas en la antigua tecnología del sodio. Se tiene 35.000 postes metálicos o “jirafas” donde se colocaran las lámparas. Además se tiene 35.000 lámparas y 35.000 fotocélulas que serán las encargadas de mandar la señal de encendido y apagado de las lámparas.

Costo de materiales – tecnología del tipo SODIO		
Material	Precio por unidad (\$)	Precio por 35.000 (\$)
Columnas metálicas	6400	224.000.000
Lámparas (250 w)	4000	140.000.000
Fotocélulas	150	5.250.000

Tabla 23 – Costo de materiales – tecnología SODIO

El costo total solo de materiales es de \$ 369.250.000. No se tiene en cuenta la mano de obra ya que será lo mismo para ambas tecnologías, de este modo se quita del análisis de costo.

Tecnología LED

Ahora se verá el precio de la instalación de los equipos para la tecnología LED, recordemos que las columnas metálicas utilizadas por la tecnología LED no necesita tener un ángulo de 15° respecto a la superficie horizontal que debe iluminar, por lo que difiere un poco de las columnas para la vieja tecnología, sin embargo el costo de la jirafa es el mismo.

Costo de materiales – tecnología del tipo LED		
Material	Precio por unidad (\$)	Precio por 35.000 (\$)
Columnas metálicas	6400	224.000.000
Lámparas (250 w)	7500	262.500.000
Fotocélulas	150	5.250.000

Tabla 24 – Costo de materiales – tecnología LED

El costo total de los materiales necesarios para la instalación de esta nueva tecnología en una ciudad mediana asciende a \$ 491.750.000, que expresado en dólares es USD 13.000.000.

Comparativa entre estas tecnologías

Al comparar este número de USD 13.000.000 de la tecnología LED contra los USD 9.768.518 de la tecnología SODIO se ve que la instalación de la nueva tecnología lleva a un incremento del 33,1% respecto de la vieja. Pero, lo que normalmente se hace en las ciudades es hacer un cambio de tecnología, por lo que las columnas metálicas no se cambian, ni tampoco las fotocélulas, solo se cambia la lámpara.

La lámpara es básicamente la carcasa metálica, el conjunto de led y el driver led. En este caso, el cambio de tecnología tendrá un costo total de USD 6.944.444.

Si la ciudad se está “diseñando”, como suele suceder, hay que tener en cuenta que las lámparas de sodio están obsoletas, ya que están dejando de fabricarse. Además tiene otro problema basado en la distorsión armónica y el bajo factor de potencia, por lo que al transcurrir el tiempo es más económica la tecnología LED. Se podrá ver esto más adelante.

Si se compara la cantidad de lumen por watt de los driver LED y la tecnología de SODIO de Philips, entonces se verá que, un driver LED de 250 [W] entrega entre 125 [Lm/W] y 130 [Lm/W]. Y la lámpara de sodio SON PLUS de 250 [W] E40 ARGENTA entrega 120 [Lm/W]. La diferencia está en que la luminaria de la tecnología del sodio consume 280 [w] de promedio por los circuitos auxiliares, mientras que la LED consume 240 [w]. Además hay que tener en cuenta que el factor de potencia para lámparas de sodio puede llegar a ser de 0.46 y de LED puede llegar a 0.95.

Para la empresa de energía es más provechosa claramente la tecnología LED. Por otra parte la luz LED es blanca y muestra los colores de las cosas que ilumina como son, no como la luz de la lámpara de sodio, esto se entiende estudiando el IRC de las lámparas. Por otra parte la luminaria LED tiene una vida útil de 50.000 horas contra las 5.000 horas de la lámpara de sodio.

El índice de reproducción cromática (IRC) es una medida de la capacidad que una fuente luminosa tiene para reproducir fielmente los colores de varios objetos en comparación con una fuente de luz natural o ideal. Es un valor de 0 a 100.

- Excelente: por encima de 90
- Muy bueno: de 80 a 90
- Bueno: de 60 a 80

El IRC del SON PLUS E40 ARGENTA es de 23, en comparación con Autobahn Series ATB2 que tiene como IRC por lo menos 70.

Ahorro en corto y mediano plazo

Si una ciudad de 35.000 luminarias decide hacer una inversión y pasar a la nueva tecnología LED, tendrá un gasto de USD 6.944.444 según lo que vimos anteriormente.

Hay que calcular cuánto se ahorra con esta tecnología respecto a la anterior hasta que se amortiza el gasto. El gasto fue de \$ 262.500.000. Por día podemos contar que la luminaria funciona 10 horas en promedio por día durante todo el año, ya que en invierno funciona un promedio de 12 horas y en verano de 8 horas.

Al tener 35.000 lámparas de tecnología de SODIO de 250 [w], con un consumo de 280 [Wh] cada una, se tiene un consumo anual de 35.770.000.000 [Wh/año].

En cambio con la tecnología LED el consumo de luminarias de 250 [W] que tienen un consumo real de 240[Wh], tendremos un consumo total anual de 30.660.000.000 [Wh/año].

Como el consumo de SODIO es 35.770 [MWh/año] y el LED es 30.660 [MWh/año], el ahorro por año teniendo en cuenta el precio por KWh que se menciona anteriormente, es de \$ 13.681.514.

Por lo tanto, la amortización de los equipos se hará en aproximadamente 20 años. Por supuesto que este valor es en realidad mucho mayor al que será en la práctica, ya que como se menciona anteriormente los equipos de sodio tienen menos vida útil, mas fallas ya que posee varios componentes internos, etc. Por lo tanto se sospecha que la amortización de los equipos se hará en la mitad del tiempo, o sea en 10 años. Como no se puede aseverar se tomara el tiempo de 20 años para compararlo con el siguiente nivel: tecnología LED con telegestión para el ahorro de la electricidad.

Equipos con telegestión de luminaria

Los equipos de telegestión de luminaria tienen un costo mayor, ya que como se estudió anteriormente necesitan de un controlador por luminaria. A la inversión que se necesita para pasar de la tecnología del sodio a la LED hay que agregarle el controlador por

luminaria y los coordinadores de red cada 100 luminarias. Esto quiere decir, que para una ciudad mediana, el costo de la inversión se incrementa en:

$$(35.000 \text{ luminarias} \times \$ 5.469,105) + (350 \text{ coordinadores} \times \$ 5950) = \$ 191.418.675 + \$ 2.082.500 = \$ 193.501.175.$$

Por lo que el costo total de la inversión con el sistema de telegestión es de \$ 456.001.175. Ahora bien, el gasto en electricidad que tendrá la ciudad se verá reducido enormemente por el plan de trabajo de que este sistema de telegestión puede implementar.

Por ejemplo, el sistema de mayor ahorro sería: de 19:00 a 20:00 la potencia del alumbrado público será del 30%, luego se incrementa de 20:00 a 22:00 la potencia del alumbrado público será del 50%, de 22:00 a 02:00 será del 100%, luego bajara al 50% desde las 02:00 hasta las 04:00. En este momento se reduce a 30% hasta que la luz de la ciudad sea suficiente para que la luminaria se apague, en verano se apagará directamente pero en invierno estará encendida hasta las 07:00 aproximadamente.

La potencia total de la ciudad se reducirá, siguiendo esta lógica un 36,7%. De este modo se calculara en cuanto tiempo se hará la amortización del gasto para toda la ciudad, tomado como referencia el precio de la antigua tecnología del SODIO y la tecnología LED pero sin telegestión.

SODIO vs LED vs LED + TELEGESTION

Como se mencionó anteriormente, con tecnología de SODIO, habrá un consumo anual de 35.770.000.000 [Wh/año].

En cambio con la tecnología LED, el consumo total anual será de 30.660.000.000 [Wh/año].

Por ultimo con tecnología LED y telegestión se tendrá un consumo anual de 19.407.780.000 [Wh/año].

Como el consumo de SODIO es 35.770 [MWh/año], el LED es 30.660 [MWh/año] y el LED más telegestión es de 19.407,780 [MWh/año], el ahorro por año teniendo en cuenta el precio por KWh que mencionamos anteriormente con respecto al SODIO, es de \$ 43.808.207,83, mientras que el ahorro en comparación a la tecnología LED es de \$ 30.126.693,83.

Por lo tanto, pasar de tecnología del SODIO a tecnología LED mas telegestión, que tiene un gasto de \$ 456.001.175, tendrá que amortizarse a $(456.001.175 / \$ 43.808.207,83) = 10,4$ años. Por otra parte, de la tecnología LED a LED más telegestión, habrá un gasto de \$ 193.501.175, con un ahorro de \$ 30.126.693,83, se amortiza el gasto a 6,4 años.

A continuación se verán gráficos representando estos valores.

Tecnología	Consumo [MW/año]
Sodio 250W	35770
LED	30660
Telegestión LED	19407

Tabla 25 – Tecnología vs Consumo anual para 35.000 luminarias

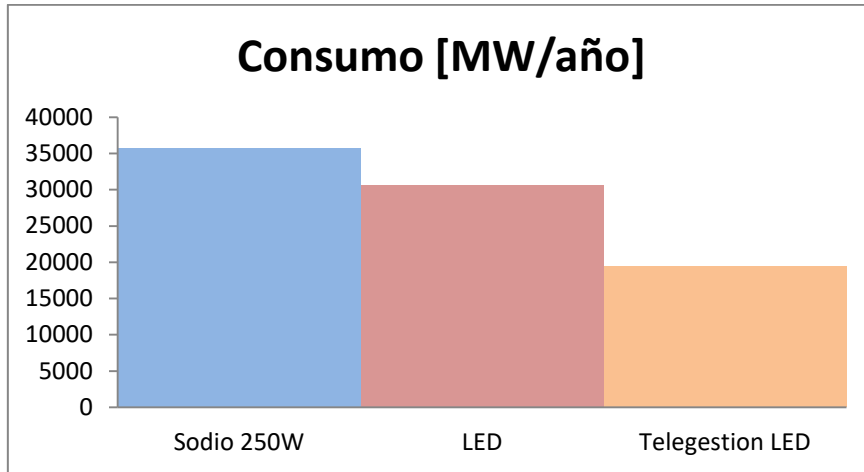


Figura 100 –Gráfico de barras – Tecnología vs Consumo

Tecnología	Ahorro en \$	Amortización aproximada en años
De sodio 250 W a LED	\$13,681,514	20
De LED a telegestión LED	\$30,126,693	6
De Sodio 250 W a telegestión LED	\$43,808,207	10

Tabla 26 – Tecnología vs Ahorro y amortización en años

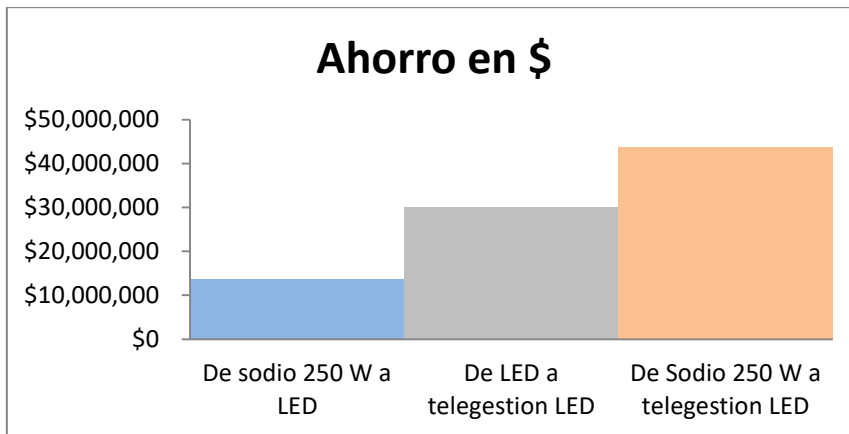


Figura 101 –Gráfico de barras – Tecnología vs Ahorro

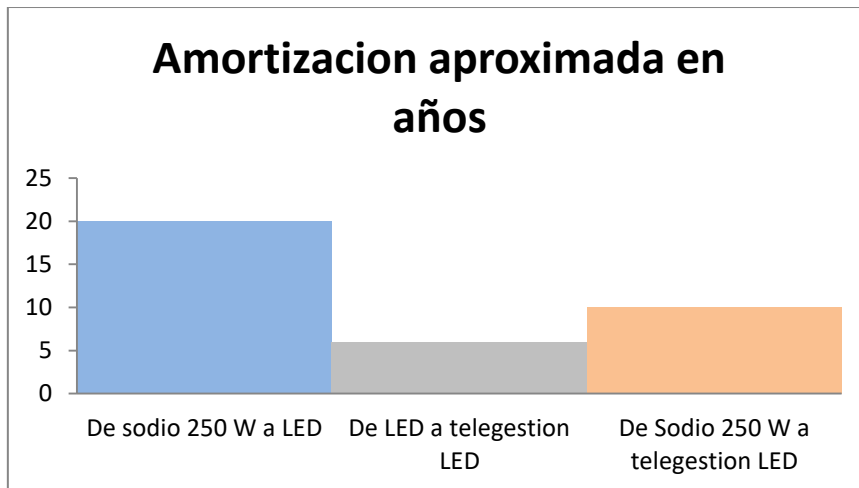


Figura 102 –Gráfico de barras – Tecnología vs Amortización en años

Comparación tecnologías (consumo, IRC, eficiencia lumínica y flujo luminoso)

A continuación se podrá observar una tabla con los datos de consumo por hora, índice de reproducción cromática (IRC), eficiencia lumínica y flujo luminoso.

Tecnología	Consumo [W/h]	IRC	Eficiencia lumínica [Lm/W]	Flujo luminoso [Lm]
Sodio 250W	280	23	120	28800
Sodio 400W	440	23	135	55000
LED	240	70	130	31200
Telegestión LED	88	70	130	11440

Tabla 27 – Tecnología vs Consumo – IRC – Eficiencia lumínica – Flujo luminoso

En forma gráfica se obtiene lo siguiente:

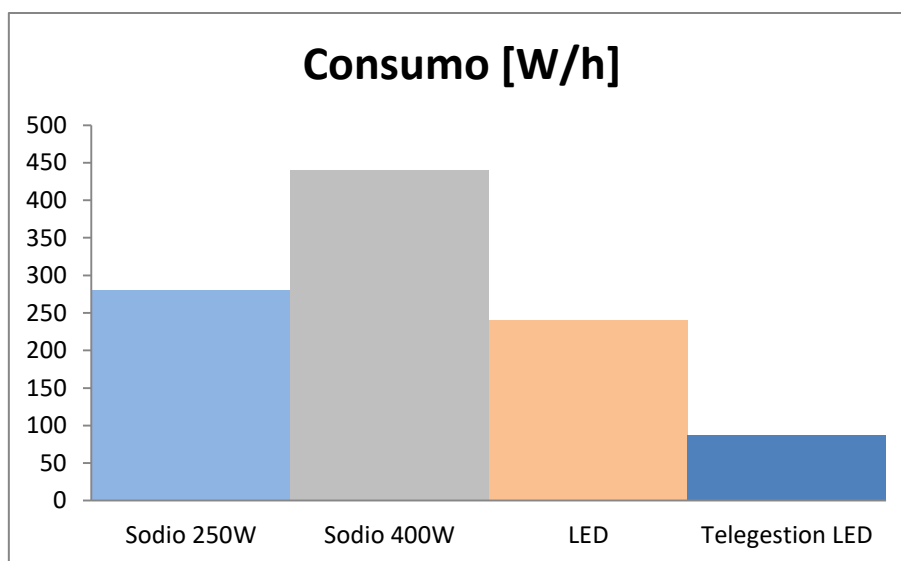


Figura 103 – Gráfico de barras – Tecnología vs Consumo

Lo que se puede analizar en este gráfico, es la diferencia de consumo entre las diferentes tecnologías. Como ya hemos visto, utilizar tecnología LED y sobre todo utilizar telegestión, conlleva un ahorro de consumo muy significativo contra la tecnología de Sodio, entre un 68 a un 80 %.

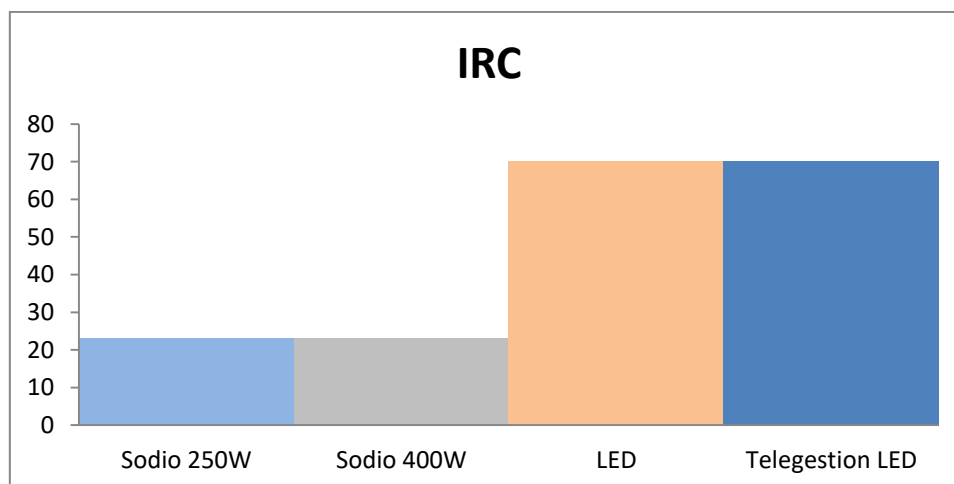


Figura 104 – Gráfico de barras – Tecnología vs IRC

Se observa que el índice de reproducción cromática en la tecnología LED es muy superior a la de sodio, y que es indistinto si se utiliza o no telegestión. Este valor genera que la capacidad de visión que una lámpara LED proporciona es mucho mejor que la de una lámpara de sodio, independientemente de su potencia.

A continuación se podrá ver una foto comparando IRC de ambas tecnologías.



Figura 105 – Iluminación LED vs Sodio en el mismo lugar

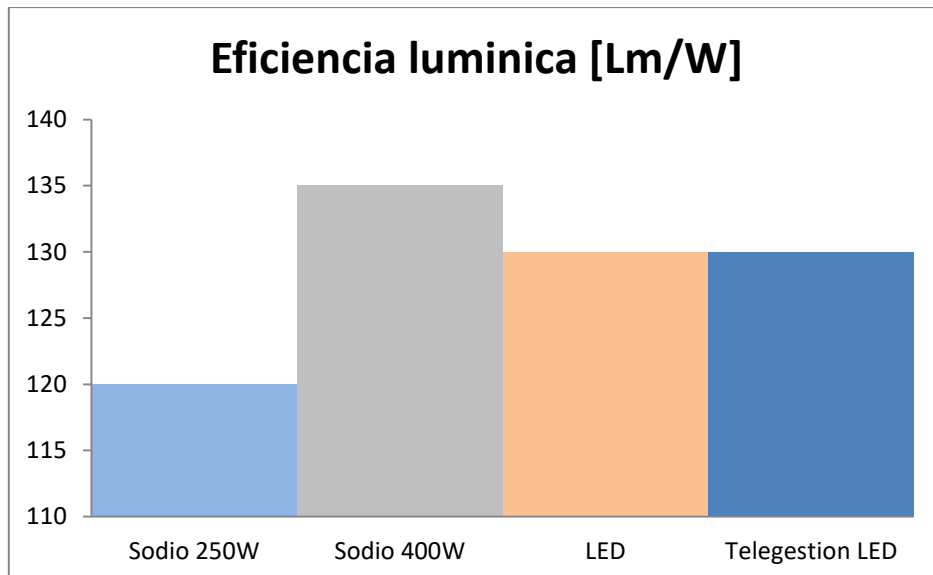


Figura 106 –Gráfico de barras – Tecnología vs Eficiencia Lumínica

En cuanto a la eficiencia lumínica se puede ver que para una luminaria de Sodio de 400 [W] este valor es superior contra la luminaria de LED de 250 [W], solo se gana un 3,7% de eficiencia lumínica pero se consume casi un 80% más de energía, por lo que resulta conveniente la tecnología LED.

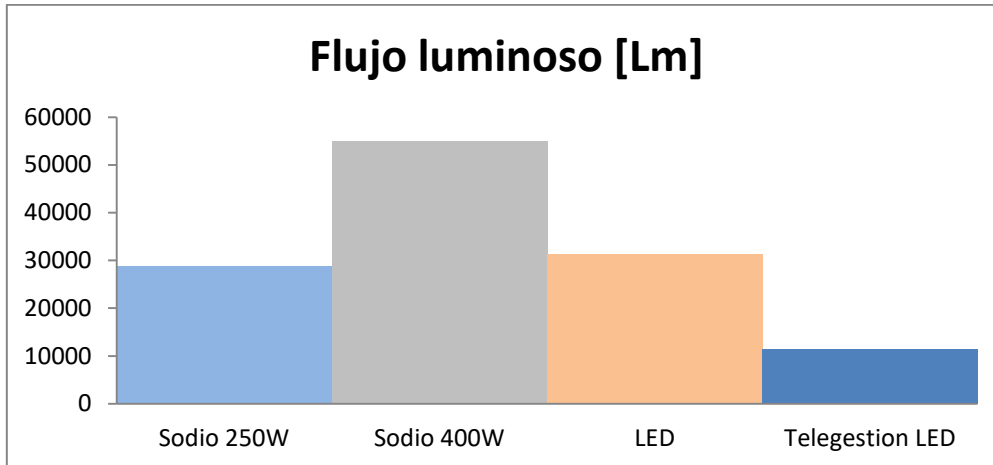


Figura 107 –Grafico de barras – Tecnología vs Flujo luminoso

Para el caso del flujo luminoso es prácticamente la misma situación que el caso anterior, la lámpara de sodio de 400 W supera en un 43 % aproximadamente a la tecnología LED, pero en su contra parte consume un 80 % más de energía.

Cabe destacar en este caso, que al parecer la telegestión LED es la peor en tanto a flujo luminoso, pero este dato puede no ser del todo real.

Como ya se mencionó, la utilización de la telegestión implica que se varíe la intensidad lumínica en ciertos horarios donde existe aún luz solar, con lo que implica un menor consumo y la luminosidad total sumando la luz solar sigue siendo aceptable. Para estos casos utilizando cualquier tipo de lámpara sin telegestión, puede ser posible que las mismas estén encendidas al 100% cuando todavía existe luz solar, lo que conlleva un consumo excesivo sin necesidad.

La siguiente foto fue tomada a las 19:25 hs. Y podemos ver esta situación, la luminosidad total, sumando la luz solar es aceptable, pero las luminarias, en este caso de Sodio, ya están encendidas.



Figura 108 –Foto de zona iluminada por Lámpara de Sodio a las 19 horas

Es por esto que la telegestión es tan importante a la hora de hablar de ahorro energético y cuidado del medio ambiente.

Capítulo 5: Discusión y Conclusión.

En conclusión podemos decir cómo se mencionó anteriormente, que la utilización de un sistema de telegestión de luminarias LED, aporta una mejora en el consumo eléctrico y en la visibilidad total, dado por su mejor índice de reproducción cromática, que tiene un impacto directo, no solo en ahorro energético y económico sino que también, medio ambiental, ya que según un estudio de la escuela politécnica de ingeniería de minas y energía de la universidad de Cantabria, asegura que: “Las lámparas LED contribuyen a reducir las emisiones de CO₂ en 10Kg por año y azufre a la atmósfera, al consumir menos que las lámparas convencionales, lo que supone un ahorro energético y por lo tanto una disminución de la huella de carbono. Además, en su fabricación no se emplean productos tóxicos como el mercurio. Su proceso de fabricación es más eficiente, por lo que también contribuye a la reducción de la huella de carbono.”

Además la utilización de iluminación LED tiene un impacto social importante, ya que al ser su IRC mayor y de un color más blanco que en las lámparas de Sodio, provee una sensación de seguridad superior a los peatones y una mejor visibilidad a los automovilistas.

Cabe destacar que las luminarias LED tienen una vida útil muy superior a las luminarias de Sodio, lo que representa otro importante factor económico.

El objetivo propuesto en el presente proyecto fue: “La realización de sistemas de luminarias dirigidas por telemando, donde se harán tanto los sistemas de control inteligentes de cada luminaria como así también los sistemas que controlan a distancia las mismas”.

Los resultados hallados fueron satisfactorios, ya que se pudo cumplir con este objetivo logrando la realización de un sistema de telegestión, en donde tanto los sistemas de control como la comunicación inalámbrica funcionaron correctamente y de acuerdo a lo establecido.

Como se mencionó en la introducción de este proyecto no existe ni un precio establecido para este tipo de sistemas ni una empresa que monopolice el mercado, únicamente Philips, está desarrollando el producto “starsense wireless”. Como únicas diferencias con nuestro sistema, Phillips incorpora las siguientes características: manejo del protocolo DALI, detección de fallas y menor peso y tamaño del equipo.

Las prestaciones del sistema de telegestión desarrollado son las siguientes:

- Comunicación inalámbrica tipo Mesh 2.4 [GHz]
- Ancho de banda = 250 [KBit/s]
- Distancia máxima entre nodos = 300 [m]
- Numero de nodos (luminarias) por controlador de segmento = 100.
- Medición de energía, factor de potencia, tiempo, temperatura y estado de la luminaria.
- Control de encendido/apagado, dimerización y configuración de horario de encendido/apagado.
- Interfaz de dimerización 0-10 [V] o 1-10 [V].
- Interfaz gráfica para monitoreo y control.
- Gestión remota conexión Ethernet (RJ 45).

El costo del sistema desarrollado depende de la cantidad de luminarias que se desean controlar. Para el caso del presente trabajo, en donde se utilizó un controlador de segmento y dos controladores de luminarias, el costo total es \$16888.

Los problemas hallados fueron:

- En primer lugar surgió que a la hora de la utilización del módulo XBee se configuró el mismo en el modo transparente, lo que generó problemas de comunicación. La solución surgió en el momento en que se cambió este modo al API.
- El segundo problema fue al utilizar el modo API ciertos caracteres no eran recibidos correctamente. La solución fue utilizar el modo API 2 o con escape.
- Otro problema fue en la utilización de los sensores de efecto Hall para la toma de señal y sensado de corriente. Se tuvo inconvenientes debido a que el sensor de efecto Hall necesitaba una posición fija respecto al cable medido lo que resultaba dificultoso de implementar. Además resultó difícil encontrar una relación Gauss-Volt. Se resolvió cambiando el dispositivo medidor a un transformador de corriente que brindaba la facilidad de elegir la escala en que se deseaba medir, aparte de que solo se necesitaba pasar el cable por el centro de la pinza sin importar la posición de esta.
- Otro problema que surgió, fue la medición del factor de potencia, ya que se debía implementar por software utilizando las señales medidas de tensión y corriente. No se logró encontrar una manera efectiva de realizar esta programación. Por lo que la solución fue utilizar una librería de Arduino denominada "EmonLib", la cual ya estaba preparada para calcular este valor.
- Por último se detectó un problema en la programación del Arduino lo que provocaba una inestabilidad del software y un envío de datos erróneos. Se solucionó optimizando el programa y liberando memoria que variables globales están ocupando.

Las mejoras propuestas son:

- Disminución del peso, tamaño y precio del equipo.
- Que la interfaz gráfica sea capaz de recibir datos sin necesidad de hacer una petición.
- Mejorar la medición de factor de potencia.
- Programar detección de fallas.
- Posibilidad de configurar módulos RTC a distancia.
- Implementar la posibilidad de realizar planes de iluminación.

De cara a un futuro estudio, con el fin de obtener unos resultados más acertados, podría analizarse más en profundidad los conceptos de luminotecnía aplicado a la iluminación LED y a la telegestión. Además sería un importante aporte, un estudio más completo sobre el impacto social que la iluminación LED tiene sobre la seguridad, en tanto vial como en criminalidad.

Capítulo 6: Literatura Citada.

- Blanchette, J., & Summerfield, M. (2006). *C++ GUI Programming with QT4*. Prentice Hall.
- CADIEEL. (2016). *Documento de acuerdo entre los socios de CADIEEL respecto al sistema de telegestión de luminarias*. Ciudad Autónoma de Buenos Aires.
- Comohacer.eu. (2018). *Proyecto Raspberry Pi*. Recuperado el 15 de Octubre de 2018, de Todo sobre los GPIO Raspberry Pi: <https://comohacer.eu/gpio-raspberry-pi/>
- Crespo Moreno, E. (Diciembre de 2014). *Aprendiendo Arduino*. Recuperado el 15 de Octubre de 2018, de Entradas y Salidas Analógicas Arduino. PWM: <https://aprendiendoarduino.wordpress.com/tag/conversor-analogico-digital/>
- Digi. (2018). *XBee ZigBee Mesh Kit Radio Frequency (RF) Module*. Recuperado el 15 de Octubre de 2018, de Digi: <https://www.digi.com/resources/documentation/digidocs/pdfs/90001942-13.pdf>
- Dignani, J. (2011). *Análisis del Protocolo ZigBee*. La Plata: Universidad Nacional de La Plata.
- Foundation Raspberry Pi. (2018). *Raspberry Pi 3 model b*. Recuperado el 15 de Octubre de 2018, de Specification: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- GEOTEK DATENTECHNIK. (2018). *IP Address Info*. Recuperado el 15 de Octubre de 2018, de The VNC family of Remote Control Applications: http://ipinfo.info/html/vnc_remote_control.php
- Llamas, L. (2017). *Ingeniería, Informática y Diseño*. Recuperado el 15 de Octubre de 2018, de Sensor de corriente eléctrica no invasivo con arduino y SCT-013: <https://www.luisllamas.es/arduino-sensor-corriente-sct-013/>
- Lutemberg, A. (Octubre de 2016). Arquitectura del sistema de telegestión de alumbrado público [Figura]. *Documento de acuerdo entre los socios de CADIEEL respecto al sistema de Telegestión de luminarias*. Ciudad Autónoma de Buenos Aires, Buenos Aires, Argentina.
- Mecanica Web. (2017). *mecanicappweb*. Recuperado el 15 de Octubre de 2018, de Sensores de Efecto Hall [Figura]: <http://mecanicappweb.com/sensores-de-efecto-hall/>
- Microsoft. (30 de Mayo de 2018). *Microsoft*. Recuperado el 15 de Octubre de 2018, de Remote Desktop Protocol: <https://docs.microsoft.com/en-us/windows/desktop/termserv/remote-desktop-protocol>
- Microsoft Support. (18 de Abril de 2018). *Soporte Técnico de Microsoft*. Recuperado el 15 de Octubre de 2018, de Descripción del protocolo de escritorio remoto (RDP): <https://support.microsoft.com/es-ar/help/186607/understanding-the-remote-desktop-protocol-rdp>
- Mora, H. (2011). *Sistemas de Adquisición y Procesamiento de datos*. Alicante, España: Repositorio Institucional de la Universidad de Alicante.
- Pallas Areny, R. (2003). *Sensores y Acondicionadores de Señal (4ª ed)*. Barcelona, España: MARCOMBO S.A.
- Perez Porto, J., & Merino, M. (2017). *Definición.de*. Recuperado el 15 de Octubre de 2018, de Definición de Acceso Remoto: <https://definicion.de/acceso-remoto/>
- Raspberry Pi Foundation. (2016). *Raspberry Pi*. Recuperado el 15 de Octubre de 2018, de Putting the power of digital making into the hands of people all over the world: <https://static.raspberrypi.org/files/about/RaspberryPiFoundationStrategy2016-18.pdf>
- Wikipedia. (7 de Octubre de 2018). *Wikipedia*. Recuperado el 15 de Octubre de 2018, de Raspberry Pi: https://es.wikipedia.org/wiki/Raspberry_Pi

Universidad Tecnológica Nacional
Facultad Regional Paraná

Se certifica que , DNI: ha
realizado la dirección del Proyecto Final:

.....
.....
.....

De los alumnos:

-
-
-

Realizada durante el ciclo lectivo:, obteniendo el grupo un calificación final
de:

A fin de ser emitida la correspondiente certificación por el departamento de electrónica, se
extiende la siguiente constancia.

Pañoni Sergio

Ramos Hector

Maggiolini Lucas