



UNIVERSIDAD DE GUANAJUATO

Campus Irapuato-Salamanca

División de Ingenierías

**“Desarrollo de un sistema de control basado en
IoT utilizando la arquitectura Particle”**

TESIS PROFESIONAL PARA OBTENER EL TÍTULO DE

INGENIERO EN COMUNICACIONES Y ELECTRÓNICA

PRESENTA

Alexander Martínez Álvarez

ASESORES:

Dr. Oscar Gerardo Ibarra-Manzano

Dr. Jesús Ixbalank Torres-Zuñiga

SALAMANCA, GTO.

SEPTIEMBRE 2021

Índice general

1. Introducción	5
1.1. Fundamento del IoT	5
1.2. Motivación	8
1.3. Objetivos	10
1.4. Alcances de la tesis	11
2. Fundamentos de la arquitectura Particle	13
2.1. Fundamento del IoT (Internet of Things)	13
2.1.1. Machine to Machine (M2M)	17
2.2. La arquitectura Particle	18
2.2.1. Características	20
2.2.2. Módulo de poder	21
2.2.3. Antena	23
2.2.4. El Conector de Depuración	24
2.2.5. Las partes del microcontrolador	24
2.2.6. Especificaciones de Radiocomunicación	27
2.3. Entorno de desarrollo de los microcontroladores Particle	28
2.3.1. Entorno de la Aplicación de Particle	30

2.3.2. Entorno de Particle Web IDE	31
2.4. Aplicación	34
2.4.1. Descripción	36
2.4.2. Programación	37
2.4.3. Pruebas y resultados	39
3. Desarrollo de un sistema de control basado en IoT	44
3.1. Descripción general	44
3.2. Planteamiento del sistema de control	46
3.3. Desarrollo del Hardware	49
3.3.1. Lámpara LED de potencia	50
3.3.2. LED driver	52
3.4. Desarrollo del Software	55
4. Pruebas y resultados	60
4.1. Pruebas del sistema de control	60
4.1.1. Prueba 1	60
4.1.2. Prueba 2	61
4.1.3. Prueba 3	62
4.2. Resultados	63
4.2.1. Resultados forma local	63
4.2.2. Resultados forma remota	64
5. Conclusiones	68
5.1. Conclusiones generales de la tesis	68
5.2. Generalidades del IoT	69

Agradecimientos

Primero gracias a *Dios* por darme la *Gran Bendición* de la vida y la salud para cumplir mis sueños, agradezco también a mis padres **José Luis S. Martínez Gutiérrez** y **María del Carmen Álvarez Tolentino** por acompañarme en este camino de mi formación profesional, por brindarme su atención, cariño, esfuerzo y sobre todo amor hacia mí para concluir con esta bella e importante etapa de mi vida alentándome a superarme día con día.

A mis hermanos **José Luis Christian Martínez Álvarez** y **Jonathan Enrique Martínez Álvarez** por sus constantes consejos y apoyo brindados y ser buenos ejemplos a seguir y motivándome a concluir con este capítulo importante en mi vida.

A mi asesor de tesis el **Dr. Óscar Gerardo Ibarra Manzano** quien me ayudó a seleccionar este tema de tesis y me brindó constante apoyo a lo largo de todo el desarrollo de la misma.

A todos los profesores de la **División de Ingenierías Campus Irapuato-Salamanca** que son parte importante de los pilares de la Universidad ya que gracias a ellos obtuve esta formación académica, moral y ética para ser mejor persona por mis valores y mejor ingeniero, por el invaluable conocimiento aprendido a lo largo de mi carrera universitaria.

Y finalmente agradezco a la **Universidad de Guanajuato** por haberme dado tantas cosas a lo largo de este aprendizaje, por ser mi segundo hogar, una abeja más de la Gran Colmena, no cabe duda que siempre llevaré en mi corazón a mi *Alma Máter*, mi casa de estudios y siempre con mucho orgullo el haber aprendido y estudiado aquí.

”La verdad os hará libres”

Universidad de Guanajuato

Capítulo 1

Introducción

1.1. Fundamento del IoT

En la actualidad se sabe que la tecnología avanza a pasos enormes, gracias a la invención del transistor la electrónica pasó de los tubos al vacío a los materiales semiconductores que eran mucho más eficientes, por ejemplo su calentamiento resulta en menor grado en comparación con la antigua tecnología de los tubos al vacío, ocupan menos espacio y son más baratos que los bulbos. Después de la gran invención del transistor se abrió paso a la creación de circuitos integrados en el que dentro de éstos mismos, alojaban en un principio cientos de transistores dentro del circuito integrado, después fue incrementando la capacidad

de los circuitos integrados a mayores escalas y cada vez tenían la capacidad de alojar un número muy grande de transistores, hoy en día su número llega a millones de elementos dentro de un microprocesador, con esto logrando entrar a una revolución tecnológica jamás vista todo gracias al transistor, [16], este increíble material semiconductor. Con el tiempo llegó la era digital que ayudó en muchos sentidos a la vida cotidiana, científica e industrial. En los procesos, por ejemplo, que anteriormente se realizaron con electrónica analógica, la electrónica digital resultó ser más barata y mucho más rápida y eficiente en cuanto a su funcionamiento, por ello con estos avances han surgido cada día nuevas y mejores tecnologías para la vida en la sociedad, con la llegada de los dispositivos inteligentes, por ejemplo los *smartphones*, computadoras cada vez de mayor capacidad y procesamiento al igual que los sistemas de comunicación y automatización en la ingeniería. Sin lugar a dudas el *Internet de las Cosas* o IoT es una tecnología muy prometedora.

La terminología *Internet de las Cosas* es muy fácil de entender, como su nombre lo indica es interconectar las cosas o dispositivos, unos con otros mediante la red de redes, es decir,

es conectar los dispositivos inteligentes y comunicarlos unos con otros por medio de Internet para realizar procesos con mayor facilidad y en tiempo real, procesos que de hecho siempre se hacen en la vida cotidiana, simplificando la vida de los usuarios y realizando también labores en menor tiempo, [17]. Muchos de los conceptos que han surgido gracias al IoT son los *Hogares Inteligentes* (o también llamados casas del mañana), con el *Monitoreo Remoto* desde el celular conectado a Internet, la persona puede mantenerse en contacto con los dispositivos también conectados a la nube así como controlar todas las funciones que se le permitan por medio del IoT, como ejemplo de ello la iluminación del hogar puede ser controlada mediante el encendido y apagado de las lámparas LED inteligentes o lámparas LED que se encuentren conectadas con microcontroladores IoT. El IoT también ha tenido gran impacto en la industria, que es lo que se conoce con el famoso término de la *Industria 4.0* que más adelante se explicará con más detalle dicho concepto. Entonces como se observa, el IoT tiene un futuro prometedor ya que logrará conectar muchísimos dispositivos de forma más óptima, se seguirá escuchando aún mucho del IoT en los años siguientes, también

de hecho se está observando la implementación de *Ciudades Inteligentes*, [20], las cuales están desarrollándose lentamente en algunas ciudades del primer mundo. Una característica importante del IoT es su gran ventaja al tener acceso a la información en tiempo real y de forma remota.

Es fácil observar el potencial presente y futuro de esta nueva tecnología ya que técnicamente son más los beneficios que se obtienen en comparación de los perjuicios como lo pueden ser la vida útil que tienen los dispositivos para después desecharlos ya que la mayoría de los dispositivos ocupan poca energía para su uso y en caso que tengan baterías, éstas se reemplazan en un par de años o más incluso. La siguiente imagen lo dice todo, el alcance que esta teniendo y llegará a tener en un futuro no muy lejano con los dispositivos comunicándose unos con otros por medio de la nube.

1.2. Motivación

Como se mencionó, en la actualidad cada día se presentan nuevos avances tecnológicos donde los dispositivos inteligentes

el usuario tenga un control de funciones específicas.

1.3. Objetivos

Los objetivos para esta tesis se pueden dividir en dos partes, en un objetivo general y uno particular, así del siguiente modo:

Objetivos Generales:

Se debe demostrar que un sistema de comunicación electrónica mediante la tecnología IoT, puede ser realizado y desarrollado por medio de la implementación de un sistema de control.

Objetivo Particular:

Demostrar por medio del *microcontrolador Argon*, [16], la comunicación IoT (*Internet de las Cosas*) a través de un dispositivo inteligente que para este caso será una lámpara LED de potencia conectado con un microcontrolador con tecnología IoT, desarrollando el sistema de forma local y forma remota.

1.4. Alcances de la tesis

Es importante dejar claro el alcance de dicha tesis, en los siguientes puntos se observan las características del alcance o límite de la misma para con ello tener en cuenta hasta dónde llegará con respecto a lo que se desea obtener, aunque claro que se podría modificar para otros intereses deseados, los alcances de la tesis se muestran en forma de listado:

- Lograr la comunicación del microcontrolador IoT con un dispositivo en particular.
- Estudiar la hoja de datos del microcontrolador Argon (modelo 1.5.2) de la familia de microcontroladores Particle para poder conocer su funcionamiento.
- Entender las ventajas y desventajas así como las características del microcontrolador Argon de la hoja de datos estudiada.
- Realizar la programación correcta del microcontrolador Argon para llevar a cabo la comunicación con la lámpara LED o dispositivo de iluminación.

-
- Poder controlar el sistema de control conformado por el microcontrolador Argon con la lámpara LED a través de una computadora, un celular y control local.

Capítulo 2

Fundamentos de la arquitectura Particle

2.1. Fundamento del IoT (Internet of Things)

Se cree que para este año 2021 se encontrarán conectados entre 30 y 50 millones de dispositivos comunicándose unos con otros a través de Internet y haciendo uso de la tecnología IoT. Para entender de una manera más clara que es el Internet de las Cosas debemos explicar sus antecesores de este mismo, es decir, las anteriores revoluciones industriales que impactaron a la vida en la humanidad de manera significativa, las revoluciones industriales que han existido sin lugar a dudas abrieron camino a lo que hoy se conoce como la *Era digital*, y una de las principales

que encabezan la era digital en nuestros días es el Internet de las cosas, la Inteligencia Artificial y la Industria 4.0, [18]. Por ello es importante explicar brevemente cómo se desarrolló con ayuda de sus antecesores la era digital en la que se está en la actualidad y probablemente se mantenga un tiempo considerado en un futuro. Las revoluciones industriales que han existido hasta el momento a lo largo de la humanidad son las siguientes:

- Primera Revolución Industrial (Industria 1.0) y la máquina de vapor.
- Segunda Revolución Industrial (Industria 2.0), producción en masa, línea de montaje, energía eléctrica en la industria, así como la creación de la banda transportadora.
- Tercera Revolución Industrial (Industria 3.0) con la llegada de la Automatización, Informática y la Electrónica.
- Cuarta Revolución Industrial (Industria 4.0) con la llegada de los sistemas interconectados e Internet de las cosas (IoT).

En la siguiente figura se muestra la trascendencia a través

del tiempo de las revoluciones industriales creadas por el hombre para mejorar y satisfacer de una mejor manera la calidad de vida en la sociedad.



Figura 2.1: Revolución Industrial a través del tiempo.

Sin lugar a dudas cada revolución industrial ha hecho enormes cambios, cada una de forma importante y específica, a la vida en la sociedad produciendo beneficios en cuanto al mejoramiento en la calidad de productos para los consumidores, maquinaria y equipo más eficiente en la industria con avances tecnológicos impresionantes.

En la *primera revolución industrial* con la llegada de la máquina de vapor uno de sus principales sectores beneficiados fue la

industria del ferrocarril ya que con el poder del vapor se transportaba el producto de un lugar a otro haciendo crecer la economía de una región o país de manera significativa.

Después con la llegada de la *segunda revolución industrial*, la electricidad como uso doméstico e industrial mejoró de manera exponencial la calidad de vida de las personas así como también trajo con ello mayor producción en el sector industrial.

Con la *tercera revolución industrial* y la llegada de la automatización se lograron enormes avances industriales y tecnológicos ya que hizo enorme mejoría en la producción en masa de productos para su venta.

Finalmente con la llegada de la cuarta revolución industrial y la creciente *Era Digital*, marcaron aún más el mundo y siendo un parteaguas en la tecnología que actualmente se conoce, es un hecho que cada día se presentan grandes avances tecnológicos al rededor del mundo en distintas áreas del conocimiento humano.

2.1.1. Machine to Machine (M2M)

Básicamente el Internet de las Cosas es la conocida comunicación *Machine to Machine* (M2M), [15], que en realidad es una tecnología que permite la comunicación e intercambio de información entre dos dispositivos para el envío de datos con un fin en específico, como su nombre lo dice, es la comunicación máquina a máquina sin intervención alguna por el hombre una vez que se logra la comunicación entre las mismas, es por ello que esta tecnología es muy importante porque reduce tiempos significativos, como un ejemplo, para las industrias puede reducir gastos de operación. Por ello el concepto de M2M e Internet de las Cosas forman parte del mismo concepto, es decir, van de la mano, como se ha mencionado gracias al IoT existe una conectividad de máquina a máquina para intercambiar datos por medio de la nube y esta conectividad es a través de Bluetooth y Wi-Fi que es una conectividad de corto alcance, de 10 a 20 metros para el Bluetooth y 50 metros para el caso del Wi-Fi, aproximadamente. Sin embargo la conectividad de baja frecuencia maneja un alcance de hasta una longitud aproximadamente de 1000 kilómetros, también la red GSM o los satélites donde estos sistemas de

comunicación ya tienen un rango global de comunicación.

En realidad en la actualidad existen diferentes empresas que venden sus productos IoT para los usuarios como lo son microcontroladores, sensores, antenas, tarjetas de comunicación, entre otros, para este caso como ya se mencionó se usará la familia Particle, por ejemplo también se podría realizar con Arduino pero el detalle es que Arduino no cuenta con módulos Wi-Fi integrados como ya los tiene Particle; la diferencia radica en que todas las familias tienen sus ventajas y desventajas pero con un único fin, la comunicación remota entre dispositivos.

2.2. La arquitectura Particle

Los microcontroladores de la familia Particle trabajan con la tecnología del Internet de las Cosas y pueden ser utilizados en un sinnúmero de proyectos académicos, domésticos e industriales, como se mencionó, se trabajará con el microcontrolador *Argon*, en la siguiente figura se muestra el microcontrolador Particle que se usará:



Figura 2.2: Microcontrolador Argon.

Para esta tesis, se usaron en específico dos *Argon Kit (Wi-Fi - Mesh - Bluetooth)*, que constan de un kit básico de desarrollo para proyectos, en dichos kits se incluyen algunos componentes electrónicos, una antena para Wi-Fi, una protoboard, así como un cable USB para su alimentación, en la figura 2.3 se presenta el kit de Argon.

Todos los microcontroladores de la familia Particle tienen incluido el módulo para detectar la red Wi-Fi. Antes del uso del microcontrolador se tiene que realizar una breve configuración para acceder a un router (o fuente de Internet en un hogar) y poder conectarlo a Internet o a la nube. Este kit de Argon puede desarrollar algunas de las siguientes comunicaciones:

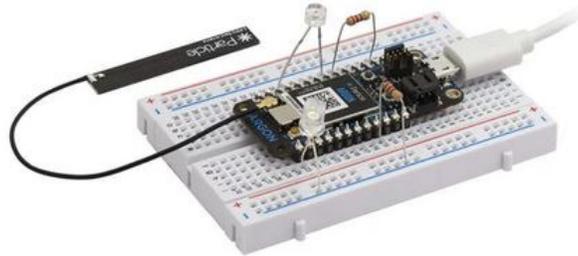


Figura 2.3: Kit de Argon.

- Wi-Fi
- Mesh (malla)
- Bluetooth

Para esta tesis se desarrollará por medio de comunicación Wi-Fi.

2.2.1. Características

Para la tesis se enfocará en el microcontrolador Argon, algunas de las características generales son las siguientes:

- Procesador de 32 bits ARM Cortex M4F con 64 MHz

-
- Utiliza un coprocesador Wi-Fi de 2.4 GHz
 - Micro USB full speed velocidad 12 Mbps
 - LED con estado RGB
 - Botones para Mode y Reset
 - Conector integrado para carga y batería Li-Po

2.2.2. Módulo de poder

Afortunadamente para este microcontrolador existen diferentes formas de poder alimentar la tarjeta principal, esto para dar más facilidad al usuario de poder manipular el dispositivo de una forma u otra, son las siguientes:

- Puerto USB
- El PIN VUSB
- Cargador LiPo
- Li + PIN
- El PIN de 3.3v

Para empezar se tiene como uso cotidiano el **Puerto USB** el cuál es el camino más fácil para alimentar al microcontrolador, básicamente se alimenta con la entrada de 5v de cualquier computadora para después regularse a 3.3v por medio del bloque regulador reductor XC9258A mostrado en la figura 2.3. Se debe tener en cuenta que la fuente de alimentación debe poder proporcionar al menos 500mA.

Esta conexión o PIN esta conectado de manera interna al **PIN VBUS** del puerto USB, el voltaje de salida es de entre 4.5 y 5 Vcc cuando está conectado y de cero cuando no lo está, este PIN se puede usar para alimentar periféricos que pueden funcionar a este voltaje dicho anteriormente y se recuerda no exceder de una corriente de 500mA. Este pin se encuentra protegido con un fusible de manera interna que tiene una capacidad de 1000mA.

Para proyectos inalámbricos se puede alimentar el microcontrolador con un **Cargador de LiPo** de 3.7v ya que la tarjeta tiene un cargador LiPo incorporado que cargará el dispositivo cuando la fuente USB este conectada para después si se quiere hacer uso del dispositivo se podrá utilizar sin la conexión del

USB directamente.

Este **Li + PIN** se encuentra conectada a la terminal positiva del conector LiPo, así que se puede conectar al microcontrolador alimentándolo, tomando en cuenta que el rango de voltaje de entrada es de 3.6 a 4.2 Vcc.

Este **PIN de 3.3v** también es importante y se puede usar para alimentar al microcontrolador (o en otro caso, puede alimentar a un LED simple) en alguna ausencia de la alimentación USB o LiPo recordando que al encenderlo se debe conectar el pin *enable* al conector *GND* para que el regulador integrado se encuentre desactivado.

2.2.3. Antena

Existen dos tipos de antenas de radio acopladas en el microcontrolador Argon, una es la antena nRF52840 (BLE) y una tipo Wi-Fi en la que se proporciona un conector u.FL para poder conectar la antena Wi-Fi para este tipo de conectividad en caso de querer usarla. Para la primera antena BLE, existen dos formas distintas de lograr la conexión por medio de ésta, la primera vie-

ne conectada a la placa de circuito impreso del microcontrolador en la que se tiene que seleccionar por medio del sistema operativo del dispositivo y en la segunda forma de conexión existe también un conector tipo u.FL, en caso que se desee conectar una antena externa utilizando los comandos apropiados para poder utilizarla.

2.2.4. El Conector de Depuración

Este microcontrolador goza de muchos beneficios, uno de los cuales es que posee un conector de depuración de 10 pines para poder hacer conexiones directas depurando el código escrito o reprogramando el arranque del Argon, a continuación se presenta en la figura 2.4 el conector de depuración con los 10 pines básicos.

2.2.5. Las partes del microcontrolador

A continuación se presentan los nombres de los elementos o partes que conforman al Argon, para su mejor visualización se mostrarán las imágenes superior e inferior del microcontrolador para su fácil identificación de los pines marcados.

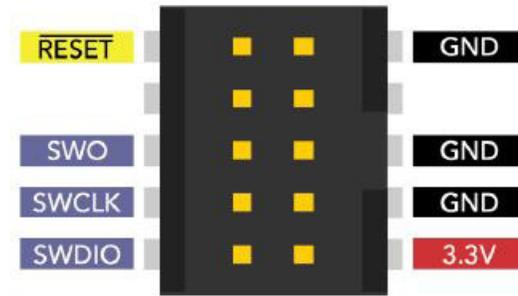


Figura 2.4: El conector de depuración.

1. USB
2. Botón de Mode
3. Botón de Reset
4. LED de estado RGB
5. LED de estado de carga
6. LED de usuario (Pin D7)
7. Conector de batería LiPo
8. Conector de depuración (SWD)
9. Conector u.FL para antena Wi-Fi (Requerido)
10. Conector u.FL para antena de malla (Opcional)



Figura 2.5: Vista superior del microcontrolador.

1. Conector u.FL para antena NFC
2. Antena incorporada en la PCB para malla
3. Escudo RF

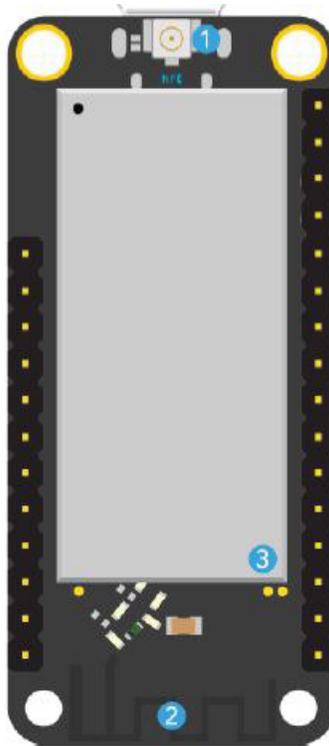


Figura 2.6: Vista inferior del microcontrolador.

2.2.6. Especificaciones de Radiocomunicación

El microcontrolador Argon cuenta con el módulo de radio: El Nordic Semiconductor nRF52840, el cual es un chip de alta tecnología (SoC) y se usa para un sinnúmero de aplicaciones inalámbricas para el cual se usa una baja potencia con una CPU ARM Cortex-M4F de 32 bits y con 1 MB de memoria flash y RAM de 256 kB en su chip y su transceptor integrado de 2.4 GHz el

CARACTERÍSTICA	DESCRIPCIÓN
Frecuencias de operación	2360 a 2500 MHz
Potencia de salida	Programable de -20dBm a +8dBm
Espaciado de canal PLL	1 MHz
Velocidad de los datos en el aire	De 125 a 2000 kbps

cual es compatible con un Bluetooth 5 de baja energía. Algunos de los muchos usos con los que se lleva a cabo este gran chip es con aplicaciones de realidad virtual, Internet de las Cosas aplicados a controladores y sensores para los hogares inteligentes o industriales, control de controles remotos para video-juegos.

2.3. Entorno de desarrollo de los microcontroladores Particle

A continuación se presentan las dos formas principales de programar por medio de los microcontroladores de la familia Particle (para este caso con Argon):

- Aplicación Oficial de Particle
- Particle Web IDE

Usar los microcontroladores Particle IoT es una experiencia muy retadora e interesante ya que requiere, como todos los microcontroladores, de un lenguaje de programación, para este caso se usará como base el lenguaje de programación en C en muchas de las instrucciones, de hecho la programación es muy similar también a Arduino aunque varía ligeramente con el uso de algunas librerías especiales que manejan los microcontroladores Particle IoT. Para su programación y funcionamiento de esta familia IoT se usa la **Aplicación Oficial de Particle**, como ya se mencionó y se puede descargar de Internet. Como muestra rápida y sencilla de una primera prueba con la aplicación se programa un pequeño LED, el LED D7 que viene incluido en la placa base del microcontrolador, o con cualquier LED común. También se puede configurar y programar la aplicación para proyectos de mayor complejidad.

Otra forma de programar es también haciendo uso de Internet, desde la página oficial de Particle, es decir, con **Particle Web IDE** escribiendo el código fuente, compilando y ejecutando (también depurando) el mismo en tiempo real, cabe señalar que antes de su uso primero se debe crear una cuenta y seleccio-

nar el microcontrolador que se usará, es decir, Argon, Electron, Photon, etc. Es importante que una vez que se tenga el programa correcto, se carga al microcontrolador y se queda en la memoria del dispositivo al menos que se sobre-escriba algún otro programa o se limpie la memoria.

Una vez que ya se carga el programa con su funcionamiento específico deseado, se puede colocar o instalar para la realización de alguna tarea continua o labor diaria que se busque ejecutar en el hogar, industria o algún sitio en específico, sin olvidar alimentar al dispositivo en todo momento, ya sea si la tarea la realiza desde la aplicación o con el programa cargado previamente.

2.3.1. Entorno de la Aplicación de Particle

La visualización de la aplicación se observa en la figura 2.7 y se aprecia que se deben introducir las credenciales de usuario y contraseña para dar acceso a la aplicación, también se usa la misma para poder reclamarlo o darlo de alta la primera vez que se utiliza un microcontrolador de la familia Particle IoT:

Una vez que se ha ingresado a la aplicación introduciendo



Figura 2.7: App de Particle.

en la cuenta el usuario y la contraseña, previamente habiendo configurado por primera vez y reclamado los dispositivos que se usarán, aparecerán los dispositivos Particle IoT que se encuentran dados de alta, como se observa en la figura 2.8.

2.3.2. Entorno de Particle Web IDE

En cuanto a la página de Internet de Particle Web IDE, se tiene un entorno de desarrollo parecido a los editores de programación en C como lo son CodeBlocks, Dev C++ ó Visual

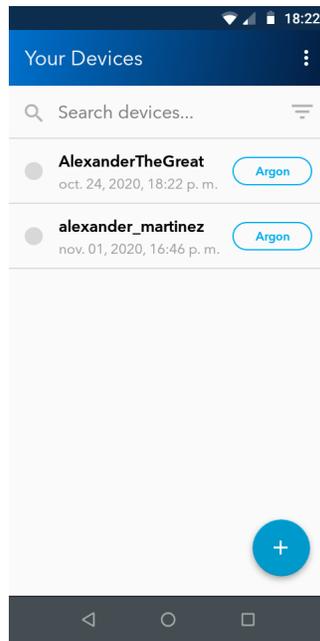


Figura 2.8: Entorno dentro de la aplicación.

Studio Code, la figura 2.9 presentada a continuación se visualiza una vez dentro de este entorno de programación, en el cual se pueden crear nuevas aplicaciones o códigos fuente así como visualizar las ventajas de estas diferentes herramientas o características de Particle Web IDE como se mostrará más adelante.

También se puede apreciar en la imagen anterior que en la pantalla derecha de fondo negro es donde se tiene que escribir y desarrollar el programa o código fuente para proyectos que se

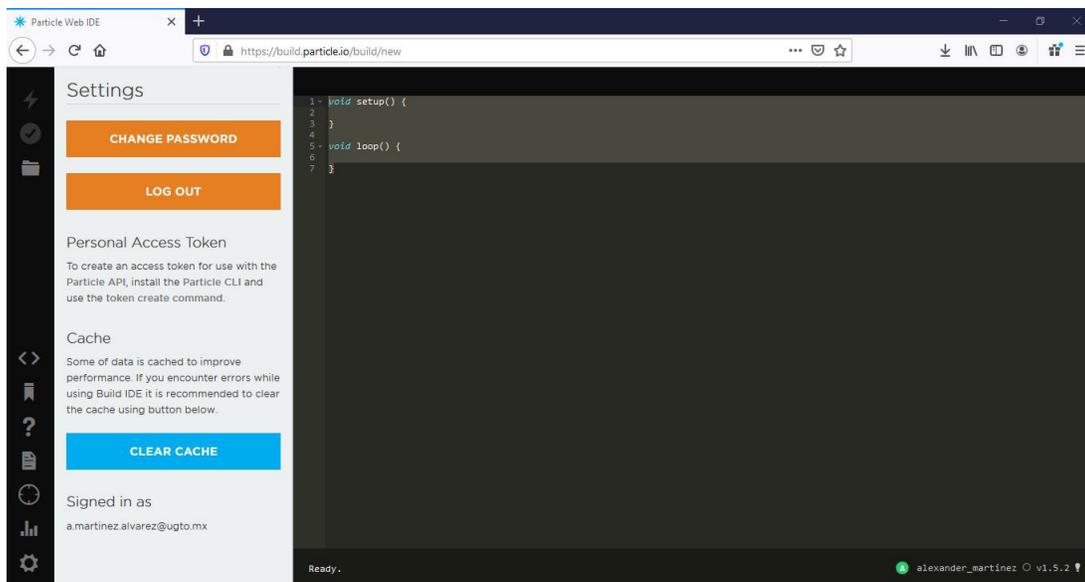


Figura 2.9: Particle Web IDE.

nombrarán y guardarán en la nube, al igual que hacer uso en cualquier momento y también cargar al dispositivo IoT dichos programas.

Es fácil crear una nueva aplicación o código fuente para un proyecto en particular en Particle, también poder eliminar proyectos que ya no se requieran en la nube, cabe mencionar que para usar el microcontrolador IoT por medio de Particle IDE, previamente como se mencionó en párrafos anteriores, se tuvo que haber reclamado el dispositivo, es decir, haberlo configurado la primera vez que se usa así como elegir un nombre al dispo-

sitivo y conectarlo a una red Wi-Fi, los pasos para reclamarlo son sencillos y se deja en la bibliografía un link donde se pueden seguir los pasos para poder dar de alta. Los dispositivos reconocidos por Particle Web IDE aparecen como se muestra en la imagen siguiente en forma de listado.

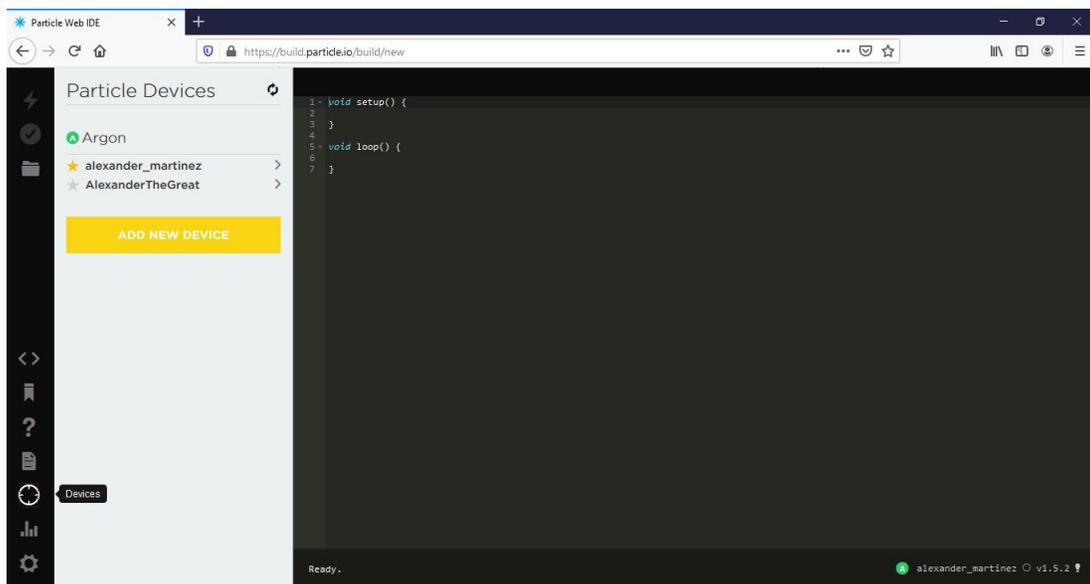


Figura 2.10: Listado de dispositivos reclamados.

2.4. Aplicación

Se describirá una aplicación sencilla con el microcontrolador Argon de inicio a fin, es decir, se describirá la programación en

código fuente y se compilará el programa para trasladarlo al microcontrolador por medio de Particle Web IDE. Será una programación sencilla y con la que se suele iniciar como programador cuando se escribe el primer programa en cualquier lenguaje de programación, el conocido **”Hola mundo”**. Para este caso será el encendido y apagado repetitivo (ciclo) de un LED, es decir el parpadeo del LED, para comenzar, se mostrarán a continuación los materiales y herramientas que se utilizarán:

- Microcontrolador Argon
- Dispositivo LED
- Protoboard
- Computadora

Este proyecto esta enfocado al desarrollo de un sistema de comunicación basado en el Internet de las Cosas, el cual mediante un microcontrolador de arquitectura Particle se comunicará a través de Internet con un dispositivo en el hogar que deseamos controlar, por ejemplo el encendido y apagado de un LED en ciertos horarios del día, básicamente de esto se trata el proyecto

y de hecho se puede expandir a diferentes proyectos con diferentes fines.

2.4.1. Descripción

Se realizará una comunicación mediante el encendido y apagado de un LED por medio de la página Particle Web IDE, es decir, mediante un programa escrito se darán instrucciones por medio del software para realizar el encendido y apagado de un LED ó parpadeo, estas instrucciones se enviarán a la nube. Ya que el microcontrolador posee un módulo Wi-Fi para conectarse a Internet que para este caso será un módem, se está ahorrando comprar y anexar módulos exteriores acoplados al sistema, entre otras ventajas. Una vez en la nube la información, se enviará al dispositivo inteligente (se carga el programa), para esta ocasión se mencionó que un LED prenderá y apagará dependiendo con que frecuencia sea el parpadeo de ciclo, realmente depende del programa creado por parte del usuario para qué tareas se defina que lleve a cabo el dispositivo IoT, en realidad hay un sinfín de posibilidades. Este microcontrolador debe mencionarse que

tiene cierto parecido en el software a los microcontroladores Arduino.

2.4.2. Programación

Técnicamente se puede programar en C para realizar algún tipo de instrucción en específico que desee y como ya se mencionó anteriormente, se puede realizar la programación parecida a la que se usa para el microcontrolador de la familia *Arduino*, para este caso realizaremos este tipo de programación ya que es muy didáctica, se debe tomar en cuenta la ventaja de esta programación, solo radica en modificar el software, es decir, en el código fuente, y así depurando o cambiando el código obtenemos diferentes resultados en comparación con software que no se pueda reciclar.

El código fuente es el siguiente:

Este código fuente funciona como sigue: se divide en dos partes medulares, en *void setup* y en *void loop*, la primera parte, al igual que Arduino sirve para declarar los puertos que se usarán

```
prueba-uno.ino
1
2 // Alexander Martínez Álvarez
3 // Ejemplo de aplicación
4
5
6 // Ejemplo de encendido del D2 del microcontrolador
7
8 int led1 = D2; // Se declara para su uso el pin D2
9 // en el microcontrolador
10
11
12
13 // Se declaran las entradas y salidas en el microcontrolador, tiene su estructura muy parecida
14 // a la forma de escribir código fuente en Arduino
15 // salvo algunas diferencias notables en tipos de funciones especiales de Particle, en este
16 // caso "void setup()" cumple la función de declarar
17 // las entradas y salidas que tendrá el programa, para este caso solo se maneja
18 // una salida que es el pin declarado en la variable anterior.
19
20
21 void setup() {
22
23     pinMode(led1, OUTPUT); // Se declara como salida la variable led1 que es el D2
24
25
26 // La función void loop(), al igual que en Arduino, sirve para ejecutar ciclos de programas
27 // de manera indefinida, es donde va la lógica de programación,
28 // por ejemplo, aquí podríamos estar comprobando constantemente la temperatura
29 // a través de un sensor, o bien comparando si se cumple una condición para prender o no un led.
30
31 void loop() {
32
33     digitalWrite(led1, HIGH); // Entrada digital en alto ó 1
34     delay(1000); // Tiempo de duración en encendido 1 seg
35
36     digitalWrite(led1, LOW); // Entrada digital en bajo ó 0
37     delay(1000); // Tiempo de duración apagado de 1 seg
38
39 }
```

Figura 2.11: Ejemplo de aplicación.

como entradas o como salidas, la segunda parte sirve para encender y apagar ciertos pines y mandarle cuanto tiempo estarán encendidos y cuanto apagados por medio de un ciclo infinito, para este caso se definió un *delay* de un segundo para mantenerlo encendido y un segundo mantenerlo apagado.

2.4.3. Pruebas y resultados

Las pruebas y resultados para este primer ejemplo de aplicación sencilla fueron logrados con éxito con base en el encendido y apagado de un LED controlado mediante un programa que envía instrucciones a través de Internet al microcontrolador, de este modo puede un usuario controlar, por ejemplo, ciertos horarios en un hogar en las cuales se pueden prender de manera controlada los LEDs que estén conectados al Argon con Internet, éste y otros ejemplos tienen un beneficio *tecnológico y social* debido a que puede ayudar a hacer ciertas actividades más fáciles a la vida en la sociedad como lo son algunas tareas diarias en el hogar.

En la figura 2.12 se observa como el microcontrolador debido a las instrucciones dadas en el programa mantiene el *encendido* del LED el tiempo establecido en el programa:

Ahora a continuación una vez que se mostró como se pueden mandar indicaciones al microcontrolador para encender de manera controlada el *encendido* del LED ahora se mostrará la manera en la que se estableció el control del apagado del LED el

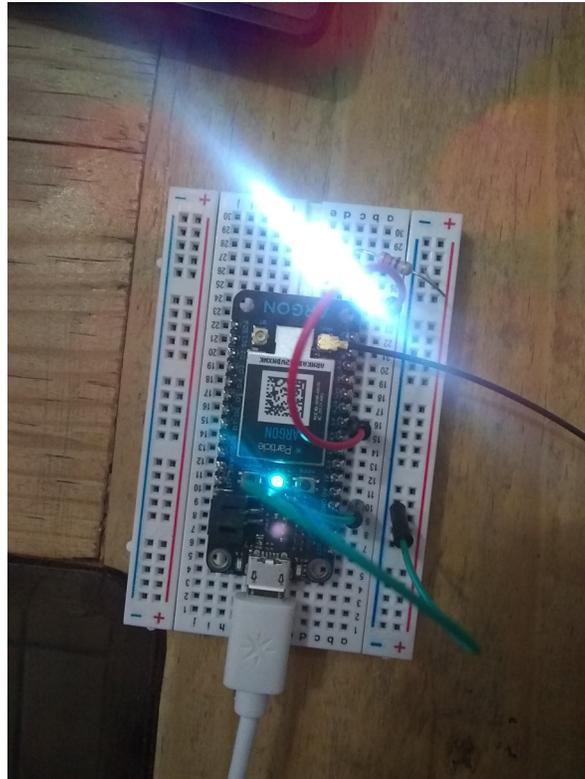


Figura 2.12: Encendido con loop controlado por Particle Web IDE.

tiempo especificado en el programa, así se muestra en la figura 2.13.

Como se puede observar si se obtuvo lo que se pedía para este caso particular, es decir, la comunicación del microcontrolador Argon de arquitectura Particle a través de la página de Particle Web IDE por medio de una comunicación sencilla así como tam-

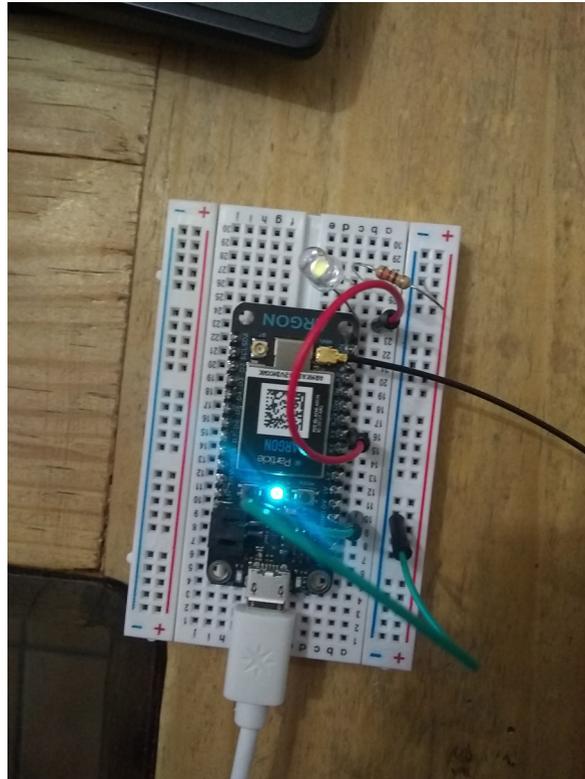


Figura 2.13: Apagado con loop controlado por Particle Web IDE.

bién observar y analizar que sí se obtuvo un impacto tecnológico y también social, ya que el IoT es una nueva tecnología que a futuro tendrá un mayor auge y será más productivo para la vida en la sociedad. Cabe señalar que este ejemplo aún no se indaga con respecto del IoT, ya que fue una comunicación sencilla para después habiendo entendido un poco el tipo de programación usado, se adentrará a la comunicación de IoT de forma remota

y local. Y aquí termine el ejemplo de aplicación práctico que se deseaba mostrar en esta sección.

Antes de terminar este capítulo se muestra por último el diagrama a bloques del microcontrolador y se presenta a continuación en la figura 2.14, en el diagrama se pueden observar los diferentes bloques de control para cada función específica que puede desarrollar el microcontrolador. Se observa que es sencillo de entender mediante esta figura la tarea que desarrolla cada bloque y como se conectan unas con otras para formar todo el sistema en conjunto.

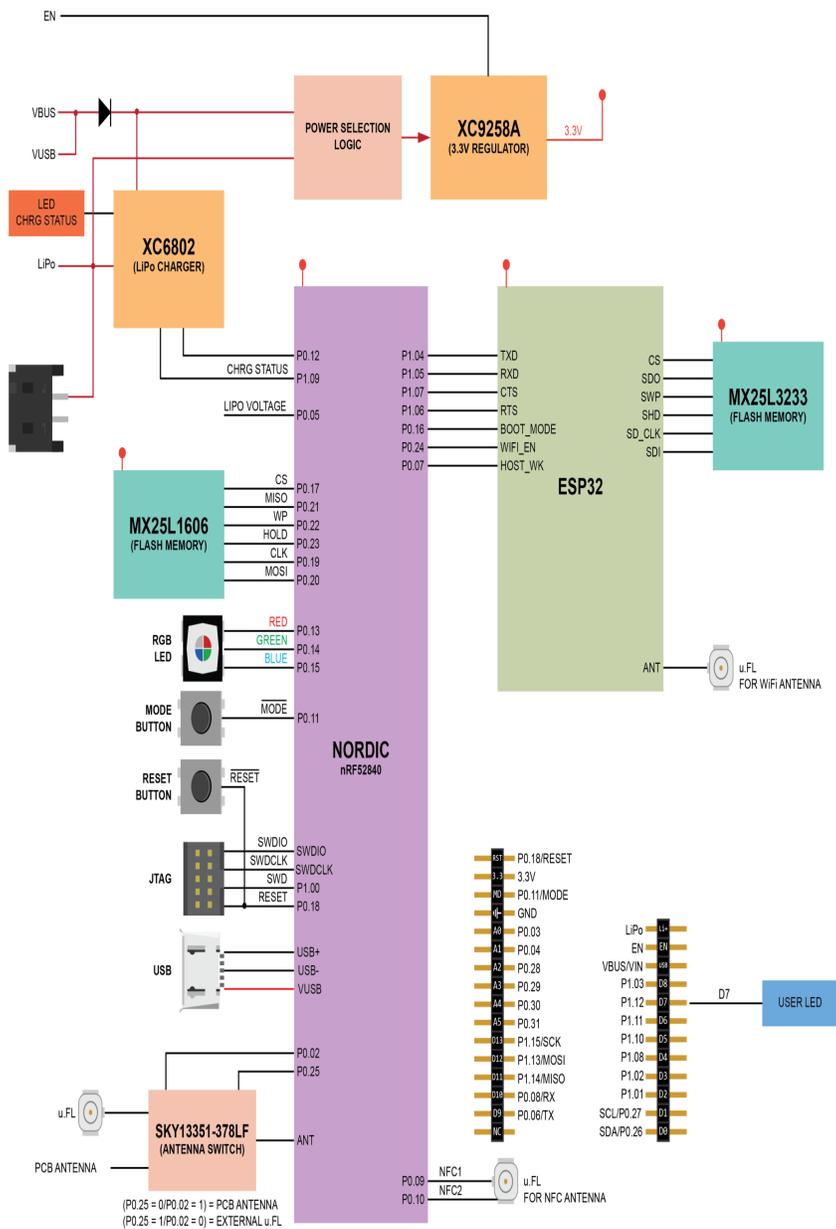


Figura 2.14: Diagrama a bloques del microcontrolador Argon.

Capítulo 3

Desarrollo de un sistema de control basado en IoT

3.1. Descripción general

Dos claros e importantes puntos que se abordarán en este capítulo son la descripción de:

- El software del sistema
- El hardware del sistema

Esta sección se dedicará a proponer y describir tanto el **Software** como el **Hardware** del sistema, los dos fundamentos importantes de los que consta cualquier microcontrolador cuando se realizan estos tipos de proyectos.

La descripción del sistema de control a desarrollar es crucial en esta sección, básicamente se desea manipular, como ya se ha comentado en anteriores ocasiones, por medio de un microcontrolador IoT una lámpara de potencia en la que más adelante se hará su descripción en hardware y en software, dicha lámpara de potencia estará conectada a un *led driver* para poder manipular de una mejor manera la corriente constante y evitar que se dañe así la lámpara, también se encontrará conectada directamente a un disipador de calor para evitar su sobrecalentamiento ante un uso prolongado del sistema IoT.

Es importante recalcar que se lograrán dos formas de controlar el sistema: de forma local y forma remota.

Cabe mencionar que la parte de programación, es la parte más importante, ya que el microcontrolador hará lo que esté escrito en las instrucciones del código fuente que se desarrolle, es decir, el programa hace lo que el usuario le propone siendo el microcontrolador el cerebro de todo el sistema desarrollado. En el ejemplo de aplicación del capítulo 2 se buscó, con éxito, establecer un primer contacto con la programación de los microcontroladores Particle con la finalidad de conocer al mi-

crocontrolador más a fondo, así como para ser un programador que se inicia en esta familia de microcontroladores IoT, es decir, para llevar a cabo el ejemplo sencillo de aplicación y así poder indagar más sobre el dispositivo IoT y conocer tanto el hardware como el software.

3.2. Planteamiento del sistema de control

Ahora para esta sección se propone el sistema de comunicación donde se visualizará su correcta ejecución, ya que básicamente el proyecto de tesis depende de la programación como su parte principal.

Básicamente una vez que se tiene el programa correcto en funcionamiento y depurado, se procede a cargarlo al dispositivo, la forma en la cual se usará el Argon se platicó brevemente (forma local y remota) y en el capítulo 2 se mostró un poco el *Entorno de Desarrollo* del microcontrolador utilizando una programación parecida a Arduino. En este capítulo y en el capítulo 4 se indagará mayormente a la utilización del hardware y software del dispositivo. Constan de lo siguiente:

Descripción de Particle Web IDE (forma local):

La descripción en forma local se puede resumir y exponer en las siguientes 3 sencillos pasos a seguir, siendo el de mayor importancia el último paso, ya que de ella consta el Push-Button acoplado que se oprime para que se encienda la lámpara LED de potencia con el programa ya cargado al microcontrolador mediante la nube:

1. Programación, compilación y ejecución en editor de la página Particle Web IDE.
2. Control por medio de consola Particle una vez compilado y ejecutado se enlaza el código con la consola para mostrar el encendido y apagado de un LED (aprender su utilización por medio de consola de Particle Web IDE).
3. Control en forma local de la lámpara LED de potencia con el programa cargado en memoria del microcontrolador mediante la nube (la forma final del encendido y apagado de la lámpara LED implementando físicamente un Push-Button).

Descripción por medio de la aplicación Particle (forma remota):

La descripción de la forma remota resulta mejor y más práctica para el usuario. En cierto sentido poder hacer uso de la forma remota, el usuario a distancia puede encender y apagar la lámpara LED así como también controlar la intensidad del LED por medio del PWM integrado en el microcontrolador usando IoT se pueden controlar distintas funciones deseadas, por ejemplo la intensidad de una lámpara. A continuación se muestran dos fases o pasos con las cuales se puede entender el desarrollo de la forma remota para el microcontrolador Argon.

1. Configurar y utilizar la app de Particle para su uso por medio de un móvil ó celular.
2. Aprender funcionamiento de la aplicación para su programación y ejecución por medio de un celular ó móvil logrando hacer una comunicación remota con el Internet de las Cosas.

3.3. Desarrollo del Hardware

El hardware principal que se requiere para llevar a cabo el desarrollo del proyecto de tesis, es una lámpara LED de potencia, dicha lámpara se describirá en esta sección, por ejemplo sus datos técnicos ó características, es decir, su hoja de datos en sí. Otro dispositivo importante conectado a la lámpara y anteriormente mencionado en las secciones anteriores es el led driver, se pueden describir sus características principales también de dicho dispositivo ya que se encontrará unido a la lámpara LED.

Muchas de las partes del hardware del microcontrolador se explicaron de manera general en el capítulo 2, por ejemplo las características de la *arquitectura del Argon*. En esta sección se trata sólo de los dispositivos exteriores que se conectarán con el microcontrolador, que en este caso será solamente la lámpara led junto con el led driver, es muy interesante conocer las características y límites de la lámpara LED ya que un mal o incorrecto uso puede llevar a estropear a dicho dispositivo o quemarlo así como no poder conectarlo de manera correcta al microcontrolador sin que pueda llevar a cabo una función alguna.

3.3.1. Lámpara LED de potencia

El LED a utilizar será de una potencia de 10W en el que se deberá alimentar con un voltaje de entrada de 12v como mínimo, también se le conoce como LED blanco frío de 10W y son componentes electrónicos semiconductores, es decir, diodos. Estos dispositivos al circular una corriente eléctrica pequeña a través de ellos, son capaces de emitir luz blanca para este caso. Las siglas de la palabra LED son *Light Emitting Diode* que significan *Diodo Emisor de Luz*.



Figura 3.1: Lámpara LED de 10W.

Cabe señalar que los diodos de este tipo, es decir, los que trabajan con una mayor potencia radican en que su diseño es

más completo en comparación de un LED común de 1.7 ó 3.3v, ya que se debe tener un mayor cuidado con el uso del control de su potencia para evitar un sobrecalentamiento o incluso llegar a quemarse el dispositivo.

¿Para que puede servir una lámpara LED (LED blanco frío de 10W)?

Se utilizan con muchos fines de proyectos tanto electrónicos como domésticos, por ejemplo pueden ser: iluminación en exteriores, iluminación en calles, iluminación en acuarios, iluminación en anuncios comerciales en las ciudades, etc., se pueden obtener en diferentes diseños de color y mayor potencia para una mejor y mayor iluminación. Otra variante de iluminación en cuanto a potencia en los otros valores de LED's que se pueden encontrar son en 1W, 3W, 10W, 50W y 100W.

Especificaciones y características:

- Potencia 10W
- Voltaje de entrada 12v
- Corriente 800mA
- Flujo luminoso de 900 a 1000 lúmenes

-
- Matriz de led 3x3
 - Tiempo de vida útil mayor a 50,000 horas
 - Dos terminales
 - Material shell aluminio
 - Pines:
 - Terminal negativa V(-) Cátodo
 - Terminal positiva V(+) Ánodo

3.3.2. LED driver

Como se mencionó anteriormente, la emisión de luz visible de un LED se debe a que se logran las condiciones eléctricas adecuadas para llevar a cabo dicha tarea, es decir, la iluminación. En muchas ocasiones se requiere, para lograr este objetivo, de algún mecanismo externo, ya sea por medio de un driver (controlador) o una fuente de alimentación que regule la energía recibida de la red para así de esta forma logre aportar al dispositivo semiconductor los parámetros correctos. Para este caso el LED driver que se usará es para solo un LED, aunque claro, se pueden realizar configuraciones para lograr conectar más de

un LED siguiendo las especificaciones eléctricas correctas en la hoja de datos.

Como ya se comentó, se usará un High Power 1A LED DC Driver, su nombre completo (modelo), es un elemento simplificado que se puede utilizar directamente para controlar a un LED sin necesidad de convertidores adicionales a este y por seguridad del dispositivo ó LED. Los componentes que conforman el led driver se encuentran soldados en la placa del mismo, este controlador maneja un empaque TO252 como convertidor DC a DC reductor. Es un chip de gran eficiencia donde su finalidad es poder entregar una corriente constante para así poder iluminar al LED de alta potencia y no varíe su iluminación, otras características importantes son:

- Las características del elemento MBI6651 integrado en el controlador hace que las aplicaciones sean mas seguras como el bloqueo por alto voltaje o protección contra la alta temperatura.
- Protección contra cortocircuito.
- Función de protección térmica (TP) para protección.

-
- Voltaje de entrada de 12v.

Algunas de las aplicación en las cuales se puede hacer uso este LED driver son:

- En señalizaciones y decoraciones para LED's
- Iluminación LED automotriz
- Iluminación de alta potencia
- Fuente de corriente constante

La siguiente tabla puede resultar útil para poder saber la función de los pines importantes del controlador:

Pin	Descripción
Vin	Terminal positiva para 12v
GND	Tierra
LED+	Conexion para anodo del LED
LED-	Conexion para catodo del LED
EN	Terminal para PWM

3.4. Desarrollo del Software

La programación para realizar lo que se desea es muy importante ya que con ello se desarrolla y describe el cerebro del sistema y por medio del software se lleva a cabo la tarea para que el hardware la ejecute sin problemas.

Se consideran estas 3 etapas de desarrollo y llevarán a cabo para este proyecto de tesis con el microcontrolador Argon donde se mostrarán sus pruebas y resultados para el siguiente capítulo, se enumeran como sigue:

1. Forma local inicial: Control del led on/off por medio de consola Particle (prueba 1).
2. Forma local final: Control por medio de push-button (prueba 2).

3. Forma remota: Por medio de la aplicación (prueba 3).

Se explicará una prueba previa al programa principal y se llevará a cabo para encender y apagar el LED cuando se desee pero por medio de comandos desde Internet a través de la consola de Particle IDE que se enlaza al programa de Particle IDE. Dicho programa sirvió mucho como previo antes de lograr el programa final de forma local mediante un push-button.

En el programa final desarrollado se trata de manipular el sistema de control de *forma local* mediante un *push-button* que se pueda encender y apagar el LED de potencia a través de una señal PWM del microcontrolador. Y en este caso resulta más completo que el anterior ya que se realizará una implementación física, es decir, en Hardware para poder realizarlo localmente.

Cuando desee hacerlo el usuario en el hogar, puede llevar a cabo esta tarea de forma local o si así lo desea de manera remota a través del celular.

El código fuente del programa previo al programa local se encuentra en las figuras 3.2 y 3.3 y en el que también se puede observar que primeramente se desarrolló dicho código en el editor de la página oficial del microcontrolador, es decir, en Particle

Web IDE, ahí se observarán en dichas imágenes las instrucciones comentadas explicando brevemente su función.

Como se pueden apreciar en las imágenes 3.2 y 3.3, donde aparece el código fuente para la prueba preliminar antes de la prueba principal local, ya no se utiliza en *void loop()* algún bucle en específico ya que como se controlará el encendido y apagado del LED a capricho del usuario y no de manera remitante, es por ello que no lleva ninguna instrucción en esa estructura del programa, aunque claro, hay excepciones en que sí se ocupa. Se puede observar también que la estructura principal del programa en C se lleva acabo mediante sentencias *if*, es decir de decisión.

Otras de algunas características principales es el uso de la función *Particle.function()*, esta función como tal es propia de Particle y no se encuentra en librerías de Arduino, cabe señalar que se debe tener especial atención en cada uno de los comentarios de las instrucciones y entender qué hace cada uno de ellos como se explica a detalle en las figuras 3.2 y 3.3.

```
web-connected-led.ino
1 // Programa desarrollado por Alexander Martínez Álvarez
2 // Proyecto de Tesis
3 // Licenciatura en Ing. en Comunicaciones y Electrónica
4 // Asesor de tesis: Óscar G. Ibarra Manzano
5
6
7 // -----
8 // Control del LED por internet por medio de consola
9 // -----
10
11 // Control del parpadeo de un LED
12 // el led1 es el pin D0, el led2 es el pin D7
13
14 int led1 = D0;
15 int led2 = D7;
16
17 // Esta vez se usará una nueva función, es decir, una función Particle
18
19 void setup()
20 {
21
22     // Se configuran los siguientes pines como salidas
23     pinMode(led1, OUTPUT);
24     pinMode(led2, OUTPUT);
25
26     // Se usará "Particle.function" con dicha función se encenderá y apagará el LED desde la nube
27     Particle.function("led",ledToggle);
28     // Cuando se pida a la nube la función "led", la nube empleará la función ledToggle() de esta aplicación
29     // Con las sig. instrucciones se asegura que estén apagados ambos LEDs en un inicio:
30     digitalWrite(led1, LOW);
31     digitalWrite(led2, LOW);
32
33 }
34
35
36 // En el ejemplo de aplicación del cap. 2 se encendía y apagaba continuamente el LED
37 // Ahora se esperara la instruccion a traves de la nube,
38 // Para void loop no se necesita escribir nada en el bucle, puesto que se espera
39 // una instruccion por parte del usuario
```

Figura 3.2: Código fuente por medio de consola, parte 1.

```
web-connected-led.ino
34
35
36 // En el ejemplo de aplicación del cap. 2 se encendía y apagaba continuamente el LED
37 // Ahora se esperara la instrucción a través de la nube,
38 // Para void loop no se necesita escribir nada en el bucle, puesto que se espera
39 // una instrucción por parte del usuario
40
41 void loop()
42 {
43     // Nada va aquí
44 }
45
46 // la función ledtoggle, es decir, la función creada se llama y se registra como
47 // función "led" en la Particle.function, es una función muy conveniente
48
49
50
51 int ledToggle(String command) {
52     /* Particle.function siempre toma una cadena como argumento y devuelve un num. entero.
53     Con dichas cadenas podemos ordenar al programa como debe usar dicha función.
54     En este caso, la función como cadena "on" encenderá el LED y "off" apagará el LED.
55     Después la función nos devuelve un valor para saber que sucedió.
56     En este caso, devolverá 1 para los LED que se encienden, 0 para los LED que se apagan
57     y un -1 si se recibe un comando totalmente falso que no hizo nada a los LED.*/
58
59     if (command=="on") {
60         digitalWrite(led1,HIGH);
61         digitalWrite(led2,HIGH);
62         return 1;
63     }
64     else if (command=="off") {
65         digitalWrite(led1,LOW);
66         digitalWrite(led2,LOW);
67         return 0;
68     }
69     else {
70         return -1;
71     }
72 }
73
```

Figura 3.3: Código fuente por medio de consola, parte 2.

Capítulo 4

Pruebas y resultados

4.1. Pruebas del sistema de control

4.1.1. Prueba 1

Una prueba sencilla que sirvió de introducción para la prueba número 2, fue descrita en la parte última de la sección anterior con el LED controlado por medio de la consola que se encuentra en la página oficial de Particle, es decir, el control del encendido y apagado del LED, el cual como se expresó, es un principio fundamental para poder controlar la forma local con un push-button acoplándolo físicamente en el microcontrolador, es por ello que fue indispensable mostrarlo.

4.1.2. Prueba 2

Para esta prueba se tiene básicamente el control físico del microcontrolador por medio de un push-button y siendo la sección final de la parte local donde el push-button se encuentra acoplado al sistema de comunicación para realizar dicha tarea, a continuación se muestra el código fuente desarrollado para esta prueba final, que cubre la forma local de la manipulación del microcontrolador.

```
1
2 // Programa desarrollado por Alexander Martínez Álvarez
3 // Proyecto de Tesis
4 // Licenciatura en Ing. en Comunicaciones y Electrónica
5 // Asesor de tesis: Oscar G. Ibarra Manzano
6
7
8 // -----
9 // Control de lámpara LED de forma local
10 // -----
11
12 int estado = 0; // Guarda la lectura del pin D8, inicia en 0
13
14 int estadoAnterior = 0; // Guarda el estado anterior del boton
15
16 int salida = 0; // si vale 0 = apagado, si vale 1 = encendido
17
18 // En void se declaran las entradas y salidas
19
20 void setup()
21
22   pinMode(D8, INPUT); // El pin D8 se declara como entrada
23   pinMode(D4, OUTPUT); // El pin D4 se declara como salida ya que es la lámpara LED
24
25
26 }
```

Figura 4.1: Código fuente para push-button, parte 1.

```
30 // En la función loop se declara la estructura principal del programa en C
31 void loop()
32 {
33
34     estado = digitalRead(D8); // Lee una entrada digital, para este caso es el pin D8
35
36     if((estado == HIGH) && (estadoAnterior == LOW)) { // Pregunta si esta en estado alto o bajo
37
38         salida = 1 - salida;
39
40         delay(10); // Para el rebote del boton, funcion delay detiene un lapso de tiempo el microcontrolador en ms
41     }
42
43     estadoAnterior = estado; // Guarda el valor actual
44
45
46     if(salida == 1){
47         digitalWrite(D4, HIGH); // Enciende el LED
48     }
49
50     else{
51         digitalWrite(D4, LOW); // Apaga el LED
52     }
53
54 }
55
56 }
```

Figura 4.2: Código fuente para push-button, parte 2.

4.1.3. Prueba 3

Para la realización de la prueba 3 de este proyecto, es decir, la prueba en forma remota, se aplicó el uso de PWM en el microcontrolador para lograr el control del encendido y apagado del LED así como controlar su intensidad de luz dependiendo de lo que decida el usuario y esto desde la aplicación de Particle, con esta etapa tres se termina el proyecto de control de la lámpara LED. Para controlar el microcontrolador se indagó sobre su uso y configuración de la aplicación y de esta forma poder hacer uso del Internet de las cosas. En el enlace anexado en los resultados de la forma local se aprecia tanto el encendido como el apagado del LED de potencia así como también la variación de la inten-

sidad de luz que se puede manipular gracias al PWM que está integrado en el microcontrolador. Fue necesario consultar el *datasheet* u hoja de datos del microcontrolador para observar los pines que son PWM, [16].

4.2. Resultados

4.2.1. Resultados forma local

Los resultados de la parte local para esta tesis se documentan en el enlace para el vídeo mostrado para la parte local, es decir, donde se agrega el Push-button.

Para entender los resultados se anexa el vínculo que llevará al resultado completo y satisfactorio obtenido para la lampara LED en forma local. Esta se refiere a la prueba número 2.

Link de resultado forma local:

<https://www.dropbox.com/s/mvj6s0inm4eybcl/prueba>

Adicional a ello, también se anexan adelante las figuras 4.3 y 4.4 donde se muestra el LED de potencia en su forma local

controlada, es decir, en estado de apagado y encendido.

4.2.2. Resultados forma remota

Los resultados de la forma remota, al igual que la forma local son importantes para entender en diferentes aspectos el potencial del microcontrolador IoT y abarca lo que es la prueba número 3.

Link de resultado forma remota:

<https://www.dropbox.com/s/sdgy8az7i3xn5tr/prueba>

Se muestra en la figura 4.5 la parte remota, es decir, la variación de intensidad de luz para la parte remota de la tesis controlada y configurada por medio de la aplicación de Particle con ayuda del PWM.

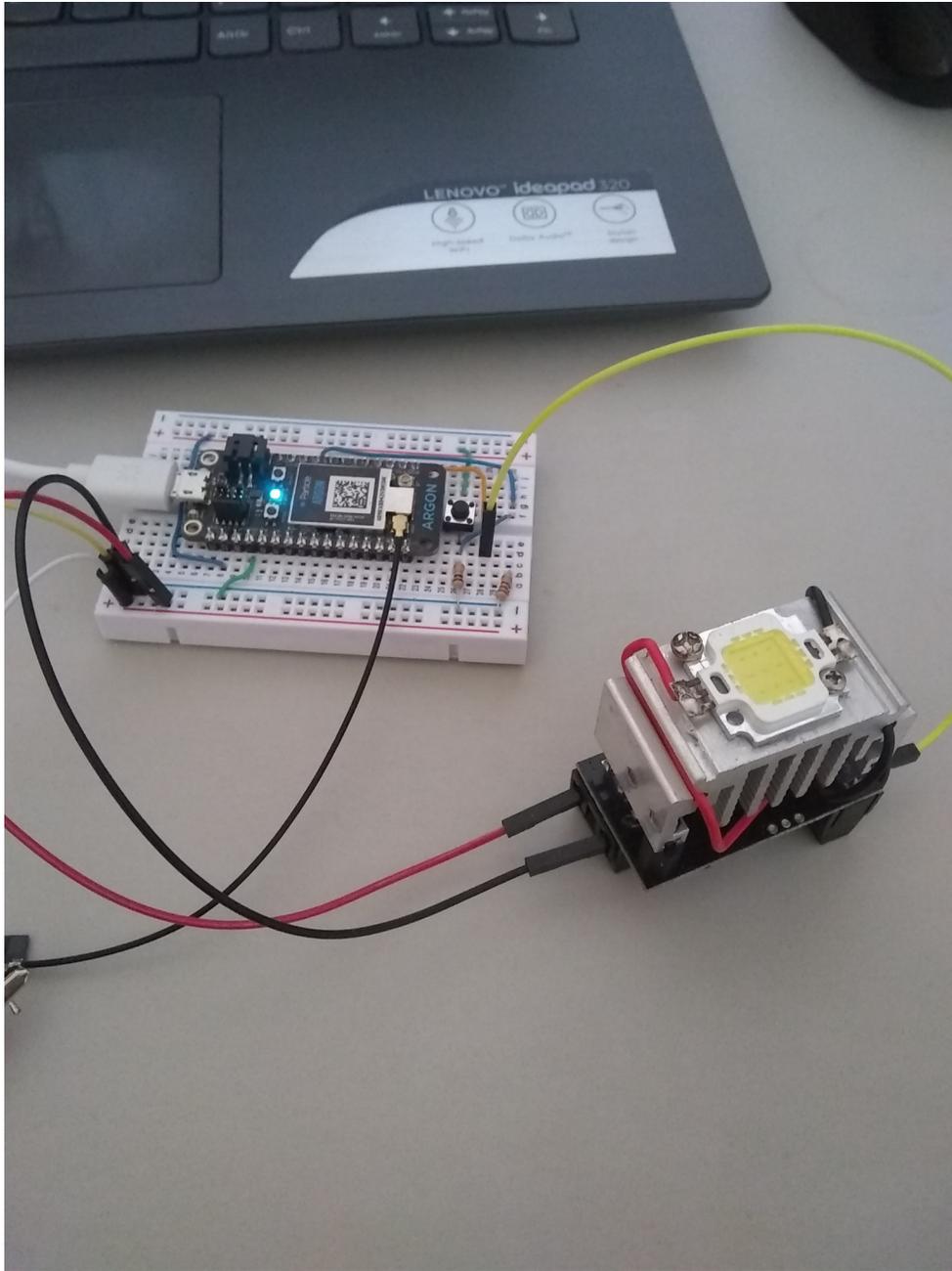


Figura 4.3: Lámpara de potencia apagada, forma local.

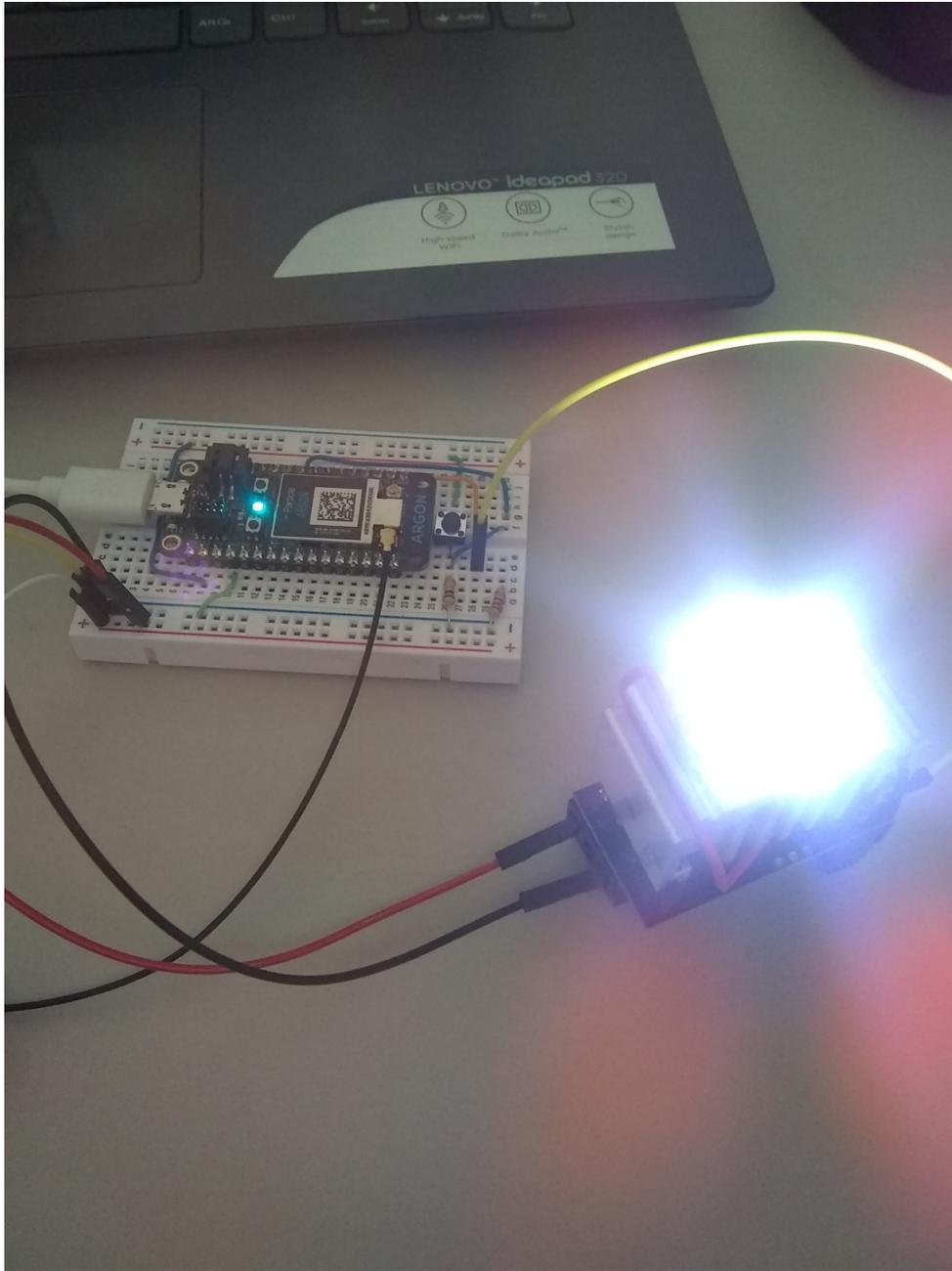
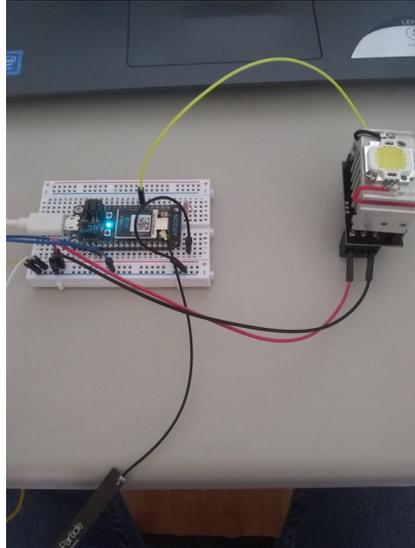
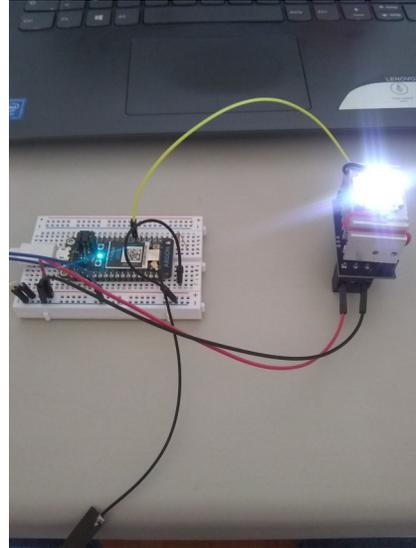


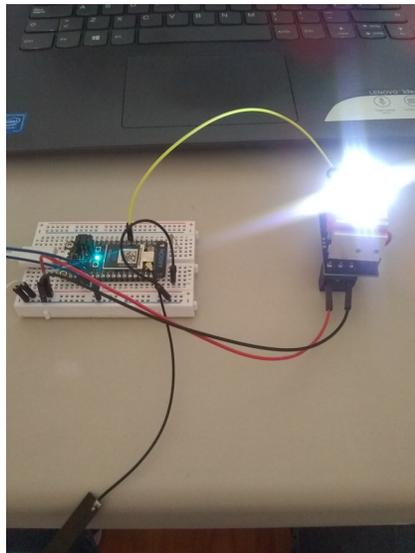
Figura 4.4: Lámpara de potencia encendida, forma local.



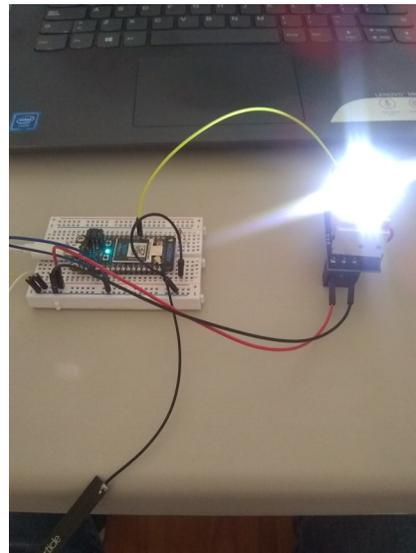
(a) LED apagado.



(b) LED encendido al 25 %.



(c) LED encendido al 75 %.



(d) LED encendido al 100 %.

Figura 4.5: Lámpara de potencia por medio de PWM, forma remota.

Capítulo 5

Conclusiones

5.1. Conclusiones generales de la tesis

La importancia de conocer la tecnología digital es crucial hoy en día para realizar mejores y más precisos sistemas electrónicos de comunicación, de potencia, de procesamiento, etc. En realidad a lo largo de este proyecto se entendió la importancia de poder reutilizar el software al sobre escribir modificando el programa o corrigiéndolo, ya no es como antes que se tenía que desechar tanto el hardware como el software sino que solo al modificar el software y cargando el programa, de nuevo se realizan nuevas ejecuciones con diferentes finalidades según sea el caso.

También se observó la importancia de aprender nuevas tec-

nologías como una formación de ingeniero así como mantenerse actualizado en diferentes ámbitos de avances científicos tecnológicos, proponiendo o mejorando como ingeniero algún sistema para su mejor uso también es una de las finalidades como ingeniero, es decir, darle solución a los problemas que se presenten en todo momento.

Así que se logró el objetivo, es decir la comunicación del sistema de Internet de las cosas (Internet of Things).

5.2. Generalidades del IoT

El IoT es una tecnología muy impresionante ya que con ella lo poco que se lleva desde su creación se han obtenido grandes avances en el mundo en diferentes áreas que no solo sea la ingeniería en sí, sino que se traslada dicha tecnología a la vida cotidiana para un mejor estilo de vida, o áreas en el deporte, en medicina con los aparatos médicos, etc. Sin lugar a dudas tiene mucho que dar esta nueva tecnología.

Bibliografía

- [1] DAVID ETTER. 2016. IOT(INTERNET OF THINGS) PROGRAMMING A SIMPLE AND FAST WAY OF LEARNING IOT.
- [2] CHRISTOPHER RUSH. 2016. PROGRAMMING THE PHOTON:GETTING STARTED WITH THE INTERNET OF THINGS. MCGRAW HILL.
- [3] SIMON MONK. 2015. GETTING STARTED WITH SPARK CORE AND PHOTON. UNITED STATES OF AMERICA, MAKER MEDIA.
- [4] AUTODESK. 2016, THE ESSENTIALS OF IOT FOR MODERN.
- [5] RYAN BETTS. 2016. ARCHITECTING FOR THE INTERNET OF THINGS. UNITED STATES OF AMERICA. O'REILLY MEDIA.

-
- [6] PERRY XIAO. 2018. DESIGNING EMBEDDED SYSTEMS AND THE IOT WITH THE ARM AND MBED. UNITED STATES OF AMERICA. WILEY.
- [7] RAJKUMAR BUYYA AND AMIR VAHID DASTJERDI. 2016, INTERNET OF THINGS: PRINCIPLES AND PARADIGMS. USA. ELSEVIER.
- [8] JOE BIRON AND JONATHAN FOLLETT. 2016. FOUNDATIONAL ELEMENTS OF AN IOT SOLUTION. UNITED STATES OF AMERICA. O'REILLY.
- [9] WHITE PAPER. 2018. IOT: TECHNOLOGY, ECONOMIC VIEW AND TECHNICAL STANDARDIZATION, ILNAS/ANEC.
- [10] CONSTANTINOS X. MAVROMOUSTAKIS, GEORGE MASTORAKIS AND JORDI MONGAY BATALLA. 2016. INTERNET OF THINGS (IoT) IN 5G MOBILE TECHNOLOGIES. SWITZERLAND. SPRINGER.
- [11] B. K. TRIPATHY AND J. ANURADHA. 2018. INTERNET OF THINGS (IoT): TECHNOLOGIES, APPLICATIONS, CHA-

ALLENGES AND SOLUTIONS. USA. TAYLOR AND FRANCIS GROUP.

- [12] DIMITRIOS SERPANOS AND MARILYN WOLF. 2018. INTERNET OF THINGS (IoT) SYSTEMS: ARCHITECTURES, ALGORITHMS AND METHODOLOGIES. SWITZELAND. SPRINGER.
- [13] ULRICH SENDLER. 2016. THE INTERNET OF THINGS: INDUSTRIE 4.0 UNLEASHED. GERMANY. SPRINGER.
- [14] LAURENCE MILLER. 2017. INTERNET OF THINGS FOR DUMMIES. UNITED STATES OF AMERICA. JOHN WILEY AND SONS INC.
- [15] [HTTPS://WWW.SCIENCEDIRECT.COM/TOPICS/ENGINEERING/MACHINE-TO-MACHINE-COMMUNICATION](https://www.sciencedirect.com/topics/engineering/machine-to-machine-communication)
- [16] [HTTPS://DOCS.PARTICLE.IO/DATASHEETS/WIFI/ARGON-DATASHEET/](https://docs.particle.io/datasheets/wifi/argon-datasheet/)
- [17] RENE DE JESUS ROMERO TRONCOSO. 2016. ELECTRONICA DIGITAL Y LOGICA PROGRAMABLE. GUANAJUATO, MEXICO. EDICIONES UG.

-
- [18] MOISES BARRIO ANDRES. 2018. INTERNET DE LAS COSAS. MADRID. EDITORIAL REUS.
- [19] ANA INES BASCO, GUSTAVO BELIZ, DIEGO COATZ, PAULA GARNERO. 2018. CIUDAD DE BUENOS AIRES. POR BANCO INTERAMERICANO DE DESARROLLO.
- [20] OLIVER GASSMANN, JONAS BOHM, MAXIMILIAN PALMIE. 2019. UNIVERSITY OF ST. GALLEN, SWITZERLAND. EMERALD PUBLISHING.
- [21] [HTTPS://WWW.ELECTRONICSHUB.ORG/INTERNET-OF-THINGS](https://www.electronicshub.org/internet-of-things)