



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO E IMPLEMENTACIÓN DE UNA INFRAESTRUCTURA DE
COMUNICACIÓN INALÁMBRICA DE BAJA POTENCIA Y LARGO ALCANCE**

Juan Antonio Valdez Méndez

Asesorado por la Inga. Ingrid Salomé Rodríguez de Loukota

Guatemala, noviembre de 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO E IMPLEMENTACIÓN DE UNA INFRAESTRUCURA DE
COMUNICACIÓN INALÁMBRICA DE BAJA POTENCIA Y LARGO ALCANCE**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

JUAN ANTONIO VALDEZ MÉNDEZ

ASESORADO POR LA INGA. INGRID SALOMÉ RODRÍGUEZ DE LOUKOTA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, NOVIEMBRE DE 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

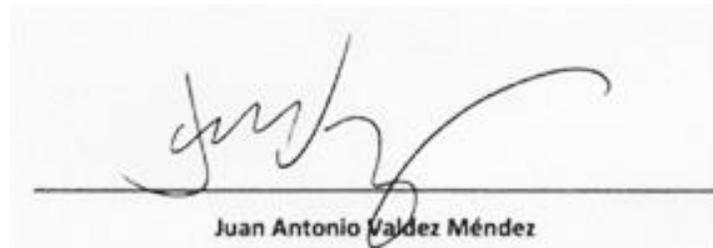
DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Guillermo Antonio Puente Romero
EXAMINADOR	Ing. Otto Fernando Andrino González
EXAMINADORA	Inga. María Magdalena Puente Romero
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO E IMPLEMENTACIÓN DE UNA INFRAESTRUCURA DE COMUNICACIÓN INALÁMBRICA DE BAJA POTENCIA Y LARGO ALCANCE

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 25 de septiembre de 2019.



Juan Antonio Valdez Méndez

Guatemala 2 de julio de 2020

Ingeniero
Julio César Solares Peñate
Coordinador del Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

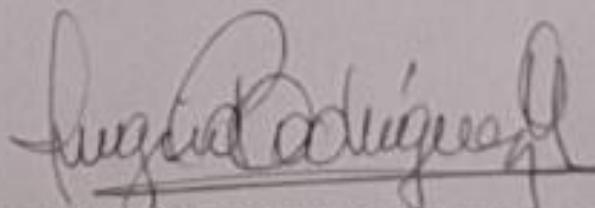
Apreciable Ingeniero Solares,

Me permito dar aprobación al trabajo de graduación titulado "**Diseño e implementación de una infraestructura de comunicación inalámbrica de baja potencia y largo alcance**", del señor **Juan Antonio Valdez Méndez**, por considerar que cumple con los requisitos establecidos.

Por tanto, el autor de este trabajo de graduación y, yo, como su asesora, nos hacemos responsables por el contenido y conclusiones del mismo.

Sin otro particular, me es grato saludarle.

Atentamente,



Inga. Ingrid Rodríguez de Loukota
Colegiada 5,356
Asesora

Ingrid Rodríguez de Loukota
Ingeniera en Electrónica
colegiada 5356

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

Guatemala, 7 de agosto de 2020

Señor Director
Armando Alonso Rivera Carrillo
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC

Estimado Señor Director:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado **DISEÑO E IMPLEMENTACIÓN DE UNA INFRAESTRUCTURA DE COMUNICACIÓN INALÁMBRICA DE BAJA POTENCIA Y LARGO ALCANCE**, desarrollado por el estudiante **Juan Antonio Valdez Méndez**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

ID Y ENSEÑAD A TODOS



Ing. Julio César Solares Peñate
Coordinador de Electrónica

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

REF. EIME 247.2020.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante Juan Antonio Valdez Méndez titulado: **DISEÑO E IMPLEMENTACIÓN DE UNA INFRAESTRUCTURA DE COMUNICACIÓN INALÁMBRICA DE BAJA POTENCIA Y LARGO ALCANCE**, procede a la autorización del mismo.



Ing. Armando Alonso Rivera Carrillo

Guatemala, 15 de Agosto de 2020.

DTG. 357.2020.

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Eléctrica, al Trabajo de Graduación titulado: **DISEÑO E IMPLEMENTACIÓN DE UNA INFRAESTRUCTURA DE COMUNICACIÓN INALÁMBRICA DE BAJA POTENCIA Y LARGO ALCANCE**, presentado por el estudiante universitario: **Juan Antonio Valdez Méndez**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Anabela Cordova Estrada
Decana

Guatemala, noviembre de 2020

AACE/asga

ACTO QUE DEDICO A:

- Dios** A quien agradezco todas las oportunidades brindadas a lo largo de mi vida para llegar a este punto.
- Mis padres** Juan Valdez y Sonia Méndez por sus sacrificios atención y cariño.
- Mi abuela** Por todo su apoyo y ser como una madre para mí.
- Mi familia** En especial a mi tía Almis y mi tío Luis, por ser un ejemplo y un gran apoyo a lo largo de mis estudios. A mi hermano y hermanas por todo su aprecio.
- Mis amigos** Por compartir su tiempo conmigo: por su apoyo estudiando o haciendo proyectos. Han sido, son y serán una parte importante de mi vida, por hacer de la universidad una experiencia grata e inolvidable.

Los ingenieros

Ingrid Rodríguez de Loukota, Armando Rivera, Iván Gonzales, Dr. Juan Carlos Córdova, Otto Andrino, Magdalena Puente y Guillermo puente. Por aportar a mi formación académica y guiarme con sus consejos.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por los servicios prestados a lo largo de mi carrera.
Facultad de Ingeniería	Por los conocimientos brindados, excelentes catedráticos y las puertas abiertas hacia nuevas oportunidades.
Escuela de mecánica eléctrica	Por darme la oportunidad para desempeñarme de manera profesional. Especialmente a los laboratorios de electrotecnia, por haber prestado las instalaciones donde crecí de manera académica y personal.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
LISTA DE SÍMBOLOS	XI
GLOSARIO	XIII
RESUMEN	XVII
OBJETIVOS.....	XIX
INTRODUCCIÓN.....	XXI
1. LA REPRESENTACIÓN Y TRANSMISIÓN DE INFORMACIÓN.....	1
1.1. Representación de la información	1
1.1.1. Señales físicas.....	1
1.1.2. Transductores.....	1
1.1.3. Variables analógicas.....	2
1.1.4. Variables digitales.....	3
1.2. Información digital.....	5
1.2.1. Digitalización.....	5
1.2.2. Comunicación digital.....	8
1.2.2.1. Comunicación digital paralela	8
1.2.2.2. Comunicación digital serial.....	9
2. TEORÍA DE LOS DISPOSITIVOS.....	11
2.1. Sistema en un chip ESP32	11
2.1.1. Especificaciones	12
2.1.2. Plataforma de desarrollo.....	15
2.1.3. Preparación del entorno de programación	16
2.1.3.1. Repositorios de utilidad	16

	2.1.3.2.	Instalación y reacondicionamiento	16	
2.2.		Módulo GPS.....	30	
	2.2.1.	Funcionamiento general	31	
		2.2.1.1. Antena	32	
		2.2.1.2. Sistema integrado	34	
		2.2.1.3. Exactitud.....	35	
		2.2.1.4. Velocidad de baudios	35	
		2.2.1.5. Canales	35	
		2.2.1.6. Chipset	36	
		2.2.1.7. GPS diferencial	36	
		2.2.1.8. Ganancia	36	
	2.2.2.	Módulo u-blox NEO-6M.....	37	
		2.2.2.1. Descripción del módulo.....	37	
		2.2.2.2. Límites máximos absolutos.....	38	
		2.2.2.3. Condiciones de operación.....	38	
		2.2.2.4. Circuitos propuestos.....	39	
		2.2.2.5. Tarjeta de pruebas	40	
			2.2.2.5.1. Diagramas.....	41
			2.2.2.5.2. Lectura de datos	46
		2.2.2.6. Monitoreo de posición	56	
			2.2.2.6.1. Gráficas de posición	56
2.3.		Módulo monitor de batería.....	57	
	2.3.1.	Terminología.....	59	
		2.3.1.1. Capacidad	59	
		2.3.1.2. Razón de descarga	59	
		2.3.1.3. Voltaje nominal de la celda	60	
		2.3.1.4. Recargable	60	
		2.3.1.5. Densidad de energía	60	
		2.3.1.6. Tarifa de descarga interna	61	

	2.3.1.7.	Seguridad	61
	2.3.1.8.	Polímeros de litio	62
	2.3.2.	Voltaje nominal	62
	2.3.3.	Carga y descarga	63
	2.3.4.	BQ27441-G1A	65
	2.3.4.1.	Descripción del circuito integrado	66
	2.3.4.2.	Límites máximos absolutos	68
	2.3.4.3.	Circuitos propuestos.....	68
	2.3.4.3.1.	Tarjeta de pruebas.....	68
	2.3.4.3.2.	Esquemático	70
	2.3.4.3.3.	Lectura de parámetros.....	71
	2.3.4.4.	Monitoreo de descarga en una celda...	73
	2.3.4.4.1.	Curvas de resultados	74
2.4.		Módulo LoRa.....	76
	2.4.1.	Descripción	77
	2.4.1.1.	Modulación	77
	2.4.2.	Transceptor SX1276.....	80
	2.4.2.1.	Descripción del circuito integrado	80
	2.4.2.2.	Límites máximos absolutos	82
	2.4.2.3.	Circuitos propuestos.....	83
	2.4.2.4.	Tarjeta de pruebas	84
	2.4.2.4.1.	Diagrama de conexión..	87
	2.4.2.4.2.	Lectura de parámetros.....	88
3.		NODO.....	93
	3.1.	Descripción y aplicaciones.....	93
	3.2.	Diagrama de bloques.....	94

3.3.	Esquemático	95
3.4.	Prototipo.....	97
3.4.1.	Circuito impreso.....	97
3.4.2.	Programación	100
3.5.	Pruebas de funcionamiento	105
4.	PUERTA DE ENLACE	107
4.1.	Descripción y aplicaciones	107
4.2.	Diagrama de bloques.....	109
4.3.	Esquemático	110
4.4.	Prototipo.....	111
4.4.1.	Circuito impreso.....	111
4.4.2.	Programación	111
4.5.	Pruebas de funcionamiento	114
5.	DISEÑO FINAL Y PRUEBAS DEL PROTOTIPO.....	117
5.1.	Diagrama de bloques.....	117
5.2.	Adquisición de datos.....	118
5.2.1.	Conexión a internet.....	118
5.2.1.1.	Monitoreo remoto de la información...	126
5.2.2.	Reporte de posición.....	127
5.2.3.	Reporte de consumo	131
5.2.4.	Resultados.....	133
	CONCLUSIONES.....	137
	RECOMENDACIONES	139
	BIBLIOGRAFÍA.....	141

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Transductores.....	2
2.	Onda senoidal.....	3
3.	Onda cuadrada.....	4
4.	Onda senoidal discreta.....	4
5.	Diagrama de bloques de un convertidor analógico a digital.....	6
6.	Señales de entrada y salida del ADC.....	8
7.	Comunicación paralela.....	9
8.	Comunicación serial.....	10
9.	Bloques de funciones simplificado del ESP32.....	12
10.	Patillaje del módulo WROOM32.....	13
11.	Weemos LOLIN32.....	15
12.	Contenidos del repositorio principal.....	17
13.	Carpeta del programa.....	18
14.	Ventana de inicio.....	19
15.	Arduino IDE.....	19
16.	Carpeta de documentos generales.....	20
17.	Ruta al administrador de tarjetas.....	21
18.	Administrador de tarjetas.....	22
19.	Selección de la tarjeta a utilizar.....	23
20.	Carpeta de librerías adicionales.....	24
21.	Lista de librerías y ruta de traslado.....	25
22.	Distintas tarjetas de desarrollo en torno al SoC ESP32.....	26
23.	Carpeta con definiciones de conexión.....	26

24.	Archivo de texto para la tarjeta Weemos LOLIN32.....	27
25.	Ubicación del ejemplo Blink.....	28
26.	Selección del puerto correspondiente	28
27.	Botones de: compilar, subir, nuevo, abrir y guardar un archivo.....	29
28.	Parpadeo del led instalado en la tarjeta	30
29.	Constelación GPS	31
30.	Triangulación	32
31.	Antena cerámica GPS	33
32.	Antena helicoidal GPS.....	33
33.	Módulo con conexión SMA.....	34
34.	Diagrama de bloques NEO-6.	37
35.	Circuito esquemático del módulo GY-NEO6MV2	39
36.	Tarjeta de pruebas o módulo GY-NEO6MV2.....	40
37.	Conector U.FL	41
38.	Conexión de la antena GPS.....	42
39.	Conexión a un puente de comunicación USB-UART.....	43
40.	Diagrama a bloque del CI CP2102.....	44
41.	Administrador de dispositivos.....	45
42.	Puertos virtuales	46
43.	Monitor serial de Arduino IDE.....	47
44.	Monitor serial	48
45.	u-center.....	52
46.	u-center en la ventana de inicio.....	53
47.	Ventana de inicial de configuraciones	54
48.	Puerto de comunicación en u-center.....	55
49.	Velocidad de transmisión en u-center	55
50.	Entorno gráfico u-center	56
51.	Descarga de una celda.....	58
52.	Baterías de polímeros de litio	62

53.	Curva típica de descarga normalizada	63
54.	Curva de carga para una batería LiPo	65
55.	Diagrama simplificado a bloques del circuito integrado BQ27441- G1A	66
56.	Gestor de baterías de litio.....	69
57.	Indicaciones del gestor de baterías.....	70
58.	Conexiones a la tarjeta weemos lolin 32.....	71
59.	Ejemplo básico para el gestor de baterías	72
60.	Asignación de la capacidad de la batería.....	72
61.	Parámetros de salida del gestor de baterías.....	73
62.	Carga resistiva	74
63.	Voltaje durante la descarga experimental de la celda.....	74
64.	Corriente durante la descarga experimental de la celda.....	75
65.	Voltaje durante la carga experimental de la celda	75
66.	Corriente durante la carga experimental de la celda.....	76
67.	Preámbulo de una señal LoRa	79
68.	Diagrama de bloques del integrado SX1276.....	81
69.	Módulos LoRa RFM96.....	83
70.	Identificación de conexiones del módulo RFM96.....	84
71.	Módulo LoRa de Adafruit	85
72.	Puerto de alimentación del módulo LoRa de Adafruit.....	85
73.	Puerto SPI del módulo LoRa de Adafruit.....	86
74.	Conexión del módulo RFM69	87
75.	Correcciones del ejemplo brindado en el paquete RadioHead.....	89
76.	Prueba de conexión hacia el módulo RFM95	90
77.	Prueba de enlace inalámbrico con el módulo RFM95.....	91
78.	Bloques fundamentales del nodo	94
79.	Conexiones al módulo ESP32	96
80.	Circuitos auxiliares de la tarjeta.....	96

81.	Capas de cobre	98
82.	Tarjeta del circuito impreso	99
83.	Circuito finalizado	99
84.	Ruta del archivo ejemplo	100
85.	Mensaje de éxito en la consola	101
86.	Librerías para utilizar en el nodo	102
87.	Variables y definiciones de conexión	103
88.	Puertos de comunicación UART	103
89.	Mensaje de alerta para el módulo de presión	104
90.	Lectura de posición satelital	105
91.	Salida del puerto serie del nodo	106
92.	Puerta de enlace.....	108
93.	Bloques fundamentales de la puerta de enlace	109
94.	Esquemático eléctrico para la puerta de enlace	110
95.	Circuito finalizado para la puerta de enlace	111
96.	Programación de la puerta de enlace	112
97.	Configuración de parámetros generales	112
98.	Dirección de la puerta de enlace	113
99.	Página generada por la puerta de enlace	114
100.	Botón de transmisión	115
101.	Mensajes recibidos en la puerta de enlace	115
102.	Montaje de campo	117
103.	The Things Network.....	118
104.	Usuario en The Things Network	119
105.	Registro del dispositivo.....	119
106.	Añadir aplicación	119
107.	Creación de la aplicación	120
108.	Registro del dispositivo.....	121
109.	Características extra.....	122

110.	Configuración completa	123
111.	Identificadores del nodo.....	123
112.	Decodificador estándar.....	124
113.	Formato LPP de Cayenne	124
114.	Integración de aplicaciones	125
115.	Aplicación myDevices.....	125
116.	Identificadores de la integración.....	126
117.	Visualización de texto	127
118.	Creación de usuario.....	128
119.	Usuario.....	128
120.	Tipos de dispositivos	129
121.	Formato de carga liviana	129
122.	Ajustes del dispositivo	130
123.	Tablero de información	131
124.	Consumo de corriente durante transmisión.....	132
125.	Estimador de tiempo de descarga.....	133
126.	Prueba de distancia 1	134
127.	Prueba de distancia 2	134
128.	Prueba de distancia 3	135
129.	Ejemplo de implementación en condición rural.....	135

TABLAS

I.	Límites de funcionamiento NEO-6M.....	38
II.	Condiciones de operación	39
III.	Condiciones de operación	45
IV.	Porción de la tabla del código ASCII	47
V.	Sentencia NEMA GPRMC	50
VI.	Sentencia NMEA GPGGA	50

VII.	Límites máximos BQ27441-G1A	68
VIII.	Detalle de conexiones	71
IX.	Límites máximos absolutos SX1276	82
X.	Parámetros de operación recomendados SX1276	82
XI.	Detalle de conexiones	87

LISTA DE SÍMBOLOS

A	Amperio
CA	Corriente Alterna
CC	Corriente Continua
CD	Corriente Directa
Hertz	Hertz
KΩ	Kilo Ohm
m	Metro
ms	Milisegundo
Ω	Ohm
s	Segundos
V	Voltios
W	Watts

GLOSARIO

ADC	Analog to Digital Converter. Dispositivo electrónico con la función de convertir una señal analógica en digital.
Álgebra booleana	Conjunto de reglas para manipular matemáticamente variables binarias.
Arduino	Plataforma de desarrollo conformada por hardware y software de código abierto para la elaboración de sistemas electrónicos.
Baudios	Cantidad de símbolos por segundo que se transmiten a través de un canal digital.
Bit	La unidad más pequeña de información.
Bluetooth	Protocolo de transmisión inalámbrica entre dispositivos electrónicos.
Bps	Unidad de medida utilizada para expresar velocidad de transmisión en bits por segundo.
Corriente	Flujo ordenado de electrones.
Corriente alterna	Corriente que cambia de sentido a través del tiempo.

Corriente continua	Corriente con valor constante a través del tiempo.
Corriente directa	Corriente que no cambia de signo a través del tiempo.
DAC	Digital to Analog Converter. dispositivo electrónico con la función de convertir una señal digital en analógica.
dBm	Unidad de medida para describir relación entre potencias, decibelio con referencia a la milésima parte de un watt.
Electrodos	Terminales metálicas con la función de establecer contacto con el medio de interés.
GND	Ground, Tierra eléctrica. Usualmente se utiliza como referencia en los esquemáticos de circuitos electrónicos.
GNSS	Global Navigation Satellite System, Sistema de satélites de posicionamiento global.
GPIO	General Port Input Output, puerto general de entrada o salida. Interfaz electrónica para transmitir o recibir estados digitales.
Hardware	Dispositivo electrónico físico, que interactúa con el entorno para adquirir o desplegar información.

IDE	Integrated Development Platform, Plataforma de Desarrollo Integrada. pieza de software con todas las herramientas para programar, compilar y cargar código.
I2C	Inter Integrated Circuit, Circuito integrado interconectado. Protocolo de comunicación digital serial.
IIS	Inter Integrated Circuit Sound, Circuito integrado interconectado para sonido. Protocolo de comunicación serial para transmitir sonido.
IP	Internet protocol. Estándar global definido para la transmisión y recepción de paquetes de información digitales a través de la red de internet.
LoRa	Long Range, largo alcance. Sistema de comunicación por radiofrecuencia con altos índices de propagación, diseñada específicamente para transmitir información a grandes distancias.
PCB	Printed Circuit Board, tarjeta de circuito impreso.
Protoboard	Tabla con conexiones eléctricas para el desarrollo de circuitos desmontables.
Repositorio	Archivo para el almacenamiento de información.

Router	Dispositivo que transmite paquetes de información a secciones de una red de computadoras de manera apropiada.
SPI	Serial Peripheral Interface, Interfaz periférica serial. Protocolo de comunicación digital serial.
Transceptor	Dispositivo capaz de convertir perturbaciones físicas en eléctricas y viceversa.
TTL	Transistor Transistor Logic, Lógica de transistor a transistor. En el desarrollo de electrónica digital se definió un estándar de 0 y 5 voltios como estados lógicos o binarios.
UART	Universal Asynchronous Receiver Transmitter. Protocolo de comunicación digital serial.
Voltaje	diferencia de potencial entre 2 puntos, capaz de realizar trabajo.
Wifi	Wireless Fidelity. Protocolo de transmisión inalámbrica para la comunicación de internet hacia dispositivos finales de red.

RESUMEN

Las ciencias afines al uso de tecnología son extremadamente sensibles al cambio. Los avances tecnológicos se dan gracias a los nuevos métodos de fabricación, algoritmos y dispositivos.

Esto representa un problema, una gran brecha entre las necesidades actuales y el conocimiento que las casas de estudio ofrecen. Nuevas ramas de la ingeniería utilizan datos a gran escala para ser procesados.

El estudiante obtendrá conocimiento teórico y práctico para la elaboración de un equipo de adquisición de datos y transmisión inalámbrica de baja potencia.

Se utilizará el estándar LoRa, por sus siglas en inglés: Largo Alcance, elegido por su versatilidad y manejo. Las características del dispositivo han sido orientadas al uso móvil con bajo costo de despliegue y mantenimiento.

El sistema propuesto puede generar los datos para un estudio estadístico y probabilístico estableciendo una base para estudios y soluciones frente a problemática nacional.

OBJETIVOS

General

Documentar, desarrollar e implementar un sistema de comunicación inalámbrica de baja potencia y largo alcance, para inculcar competencias de investigación y elaboración de equipo de uso final a futuros ingenieros.

Específicos

1. Brindar al estudiante la información para elaborar su propio equipo de conexión de bajo costo.
2. Brindar técnicas de medición y análisis para la prueba práctica de rendimiento de la alimentación.
3. Realizar y documentar pruebas de alcance en entorno urbano.
4. Realizar el primer nodo de la red y establecer un enlace exitoso.

INTRODUCCIÓN

La electrónica es un campo extenso y de fundamental estudio en la actualidad. Diversas aplicaciones dentro del campo integran soluciones a situaciones específicas que mezclan componentes de hardware y software para conformar un solo sistema. Es necesario experimentar con nuevas tecnologías, algoritmos y métodos para la resolución de la problemática actual.

La tecnología avanza a pasos agigantados, es imperativo buscar nuevos métodos para mejorar y acelerar el proceso de aprendizaje de los estudiantes. Es necesario mencionar las tecnologías de la información y comunicación. La red de telecomunicaciones es parte del ir día a día.

Cada año aumenta el número de usuarios. Como consecuencia los datos e información también incrementan. Aún más cuando se trata de tener un registro temporal multivariable.

Puede hacerse un estudio estadístico y probabilístico de variables con el fin de optimizar procesos. Existen casos donde la información requiere de sensores para ser medida.

Para practicar con los temas anteriormente descritos es necesario utilizar técnicas de instrumentación para adquirir datos de manera adecuada e informática para presentarla de manera legible. El desarrollo de una infraestructura en base a tecnologías actuales brindará al estudiante herramientas de entendimiento y prácticas apropiadas de diseño.

1. LA REPRESENTACIÓN Y TRANSMISIÓN DE INFORMACIÓN

1.1. Representación de la información

Información se define como el conocimiento que se adquiere a partir de la investigación o estudio de una variable. El observador le da a una variable el carácter de una señal. Una variable es un concepto más amplio y abstracto, mientras que la señal es una variable que representa el comportamiento físico de interés.

1.1.1. Señales físicas

Una señal física, representa el comportamiento de una variable de estudio: posición angular, temperatura, presión o luminosidad de un elemento.

Los sistemas de medición aprovechan estos fenómenos y su interacción con otras propiedades de la materia para indicar un cambio, por ejemplo: un termómetro de mercurio cambia de volumen en función de la temperatura, el mercurio es autosuficiente para medir la variable de interés y brindar una lectura denominada medición directa.

1.1.2. Transductores

Un transductor es un elemento que traduce, cualquier fenómeno causado por una perturbación física a una señal eléctrica y viceversa. Es importante que la salida sea de alguna forma función de la entrada, en otras palabras, el

comportamiento de salida no es aleatorio. Se busca una salida eléctrica ya sea un cambio en resistencia, corriente, voltaje o capacitancia porque es de interés realizar un circuito de adquisición de datos para obtener mediciones. Resulta conveniente ejecutar un estudio de los datos mencionados con sistemas computacionales para facilitar la caracterización, predicción y confiabilidad del comportamiento de la señal.

Figura 1. **Transductores**



Fuente: Blogspot. *Transductores eléctricos*.

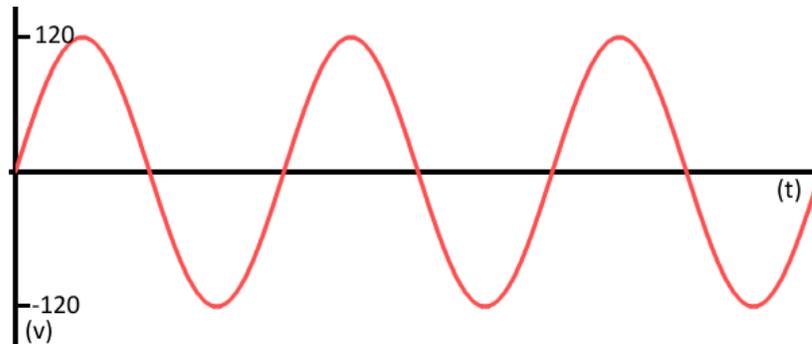
<http://transductoreselectricos2.blogspot.com/2015/06/transductores-un-transductor-es-un.html>.

Consulta: diciembre de 2019.

1.1.3. **Variables analógicas**

El ser humano vive en un mundo analógico. Existe una cantidad infinita de olores por percibir y de colores para crear una pintura. El tema en común entre ambas señales son sus infinitas posibilidades. Dicha variable generalmente cambia a través del tiempo, es útil representar en una gráfica donde el tiempo se traza en el eje horizontal y la magnitud de interés en el eje vertical, observar la gráfica es la forma más fácil de identificar si se trata de una señal continua o discreta. Una señal analógica es continua y suave. Ver figura 2.

Figura 2. **Onda senoidal**



Fuente: Sparkfun. *Onda senoidal*.

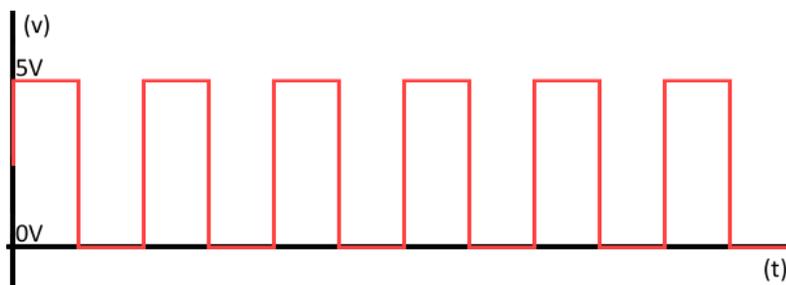
<https://cdn.sparkfun.com/assets/3/7/6/6/0/51c48875ce395f745a000000.png>. Consulta: diciembre de 2019.

Incluso si éstas señales tienen un rango y dominio limitado, que se establece entre su valor máximo y mínimo, siempre existe un infinito número de valores dentro del mismo.

1.1.4. Variables digitales

Las variables digitales que contienen información también suelen denominarse señales discretas o digitales, tienen un conjunto finito de valores, en contraposición de las señales analógicas que poseen un conjunto infinito. Las variables digitales generalmente adquieren uno de dos valores, por ejemplo, cero y cinco voltios. La gráfica temporal de esta variable suele apreciarse como onda cuadrada. Ver figura 3.

Figura 3. **Onda cuadrada**

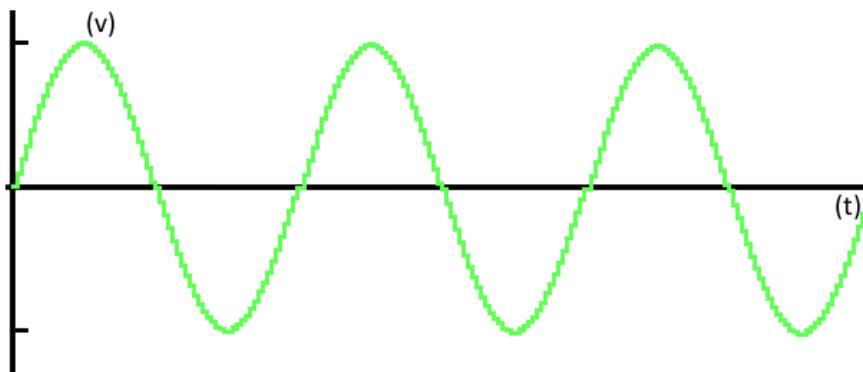


Fuente: Sparkfun. *Onda cuadrada*.

<https://cdn.sparkfun.com/assets/c/8/5/b/e/51c495ebce395f1b5a000000.png>. Consulta: diciembre de 2019.

Una señal también digital puede ser la representación discreta de una forma de onda analógica. Vista desde lejos, la onda puede parecer suave y continua, pero cuando se aprecia de cerca existen pequeños pasos que intentan aproximar la señal continua. Ver figura 4.

Figura 4. **Onda senoidal discreta**



Fuente: Sparkfun. *Onda discreta*.

<https://cdn.sparkfun.com/assets/0/2/8/4/6/51c85fbcce395fbc03000000.png>. Consulta: diciembre de 2019.

No todas las señales de audio son analógicas, estándares como HDMI para el video y el audio, MIDI, IIS son transmitidas digitalmente.

1.2. Información digital

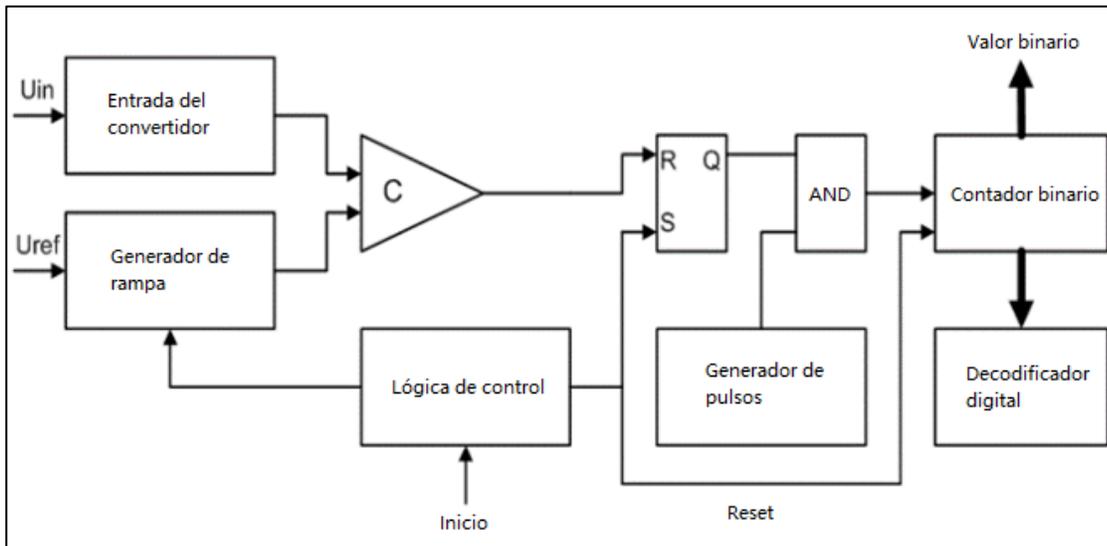
La información digital surgió a partir de un proceso de conversión desde una señal analógica a una discreta. Existen técnicas, circuitos y módulos con el fin de realizar la traducción descrita. La acción es necesaria para que la información pueda ser procesada mediante análisis computacional.

Digital o booleano, la lógica es un concepto fundamental en todos los sistemas computacionales modernos. Es el conjunto de reglas que permite realizar tareas complicadas basados en relativamente simples sí o no, uno y cero respectivamente. La principal ventaja es la velocidad para realizar operaciones matemáticas, imposibles de igualar por un ser humano.

1.2.1. Digitalización

Es la conversión de un valor analógico a discreto, existen distintas soluciones. Es de principal interés obtener una representación en el sistema de numeración binaria para su posterior uso en sistemas digitales. Tome a figura 5 para comprender mejor el proceso.

Figura 5. Diagrama de bloques de un convertidor analógico a digital



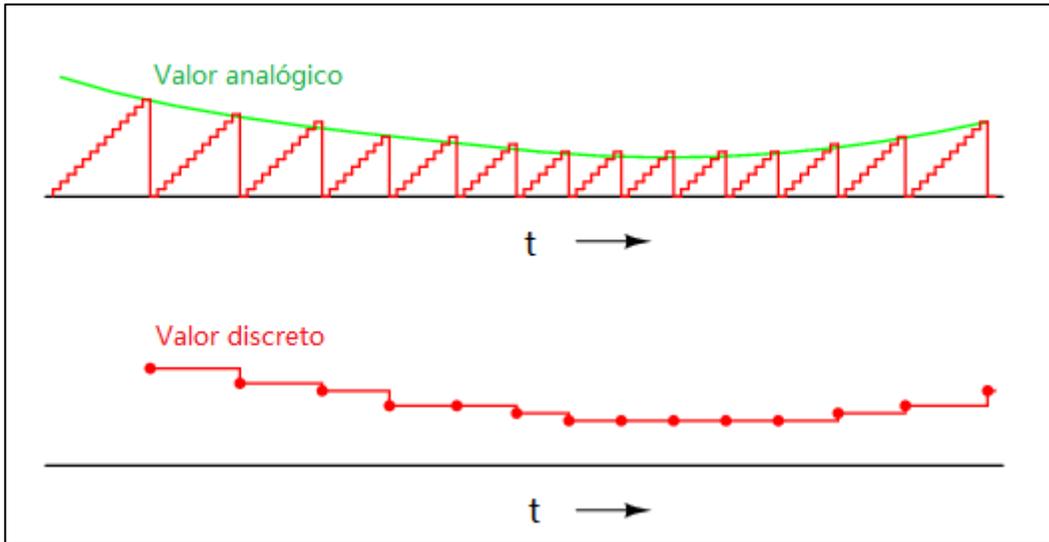
Fuente: elaboración propia, empleando Adobe Illustrator CC 2018.

- La entrada del convertidor es una etapa de reacondicionamiento de la señal analógica a ser procesada.
- El generador de rampa también llamado generador de voltaje incremental, es una señal de referencia.
- El comparador C , cambia su estado de salida cada vez que la señal de entrada iguala o supera la de referencia.
- Flip Flop SR es un circuito digital secuencial o unidad básica de memoria.
- Compuerta AND es un circuito digital combinatorio, su estado de salida es positivo si y solo si su entrada uno y su entrada dos son positivas.

- La unidad de control se encarga de sincronizar el inicio del proceso con los demás elementos.
- Generador de pulsos también llamado señal de reloj o sincronización, el muestreo y tiempo que dura el proceso es constante y periódico gracias al generador de pulsos.
- El contador digital es un módulo cuya entrada es una señal y la salida es un número binario que cuenta los pulsos.
- El display digital obtiene los valores binarios del contador digital y los despliega de forma decimal.

Como puede apreciarse en la figura 6. La entrada consta de dos señales: una de rampas o referencia y la segunda del valor analógico a digitalizar, la salida es el valor discretizado apreciable en el eje vertical. El proceso anteriormente descrito únicamente generó el valor digital, la información es procesable entonces por un sistema computacional. Es necesario transmitir los datos obtenidos a la unidad de procesamiento. Para llevarse a cabo se utilizan técnicas de comunicación digital.

Figura 6. **Señales de entrada y salida del ADC**



Fuente: elaboración propia, empleando Adobe Illustrator CC 2018.

1.2.2. **Comunicación digital**

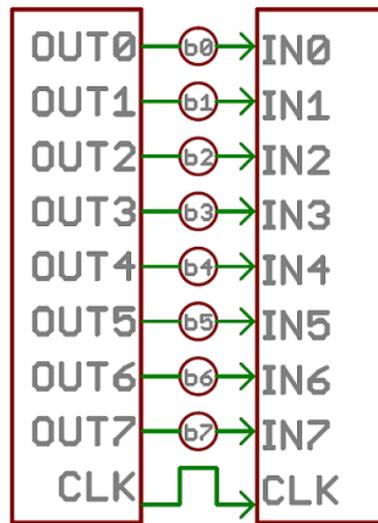
La electrónica embebida se crea a partir de interconectar circuitos, como procesadores y microcontroladores obteniendo un sistema simbiótico. Para que cada circuito intercambie la información ellos deben compartir un protocolo de comunicación. Se han definido cientos de distintos protocolos de comunicación para obtener intercambio de información. En general cada uno de ellos puede ser separado en dos grandes categorías: serial o paralela.

1.2.2.1. **Comunicación digital paralela**

Las interfaces paralelas, figura 7, transfieren múltiples unidades de información al mismo tiempo. Usualmente requieren de tantos buses como de datos transmitidos, es la forma más rápida de transmitir un paquete completo.

La desventaja se encuentra en el número de entradas y salidas necesarias para implementar el sistema. En la figura 6 se muestra un bus de 8 bits, controlado con un reloj transmite un byte cada ciclo de reloj. Se utilizan 9 cables.

Figura 7. **Comunicación paralela**



Fuente: Sparkfun. *Analog vs. Digital*. https://learn.sparkfun.com/tutorials/analog-vs-digital?_ga=2.41782754.1750709408.1575312359-80799570.1567000013. Consulta: diciembre de 2019.

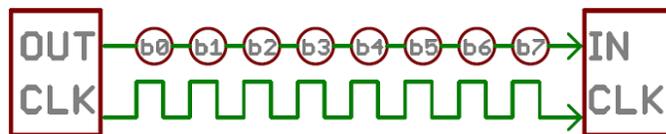
1.2.2.2. **Comunicación digital serial**

Las interfaces seriales transmiten sus datos un bit a la vez, es decir, la unidad básica de información binaria o memoria. A través de los años, docenas de protocolos seriales han sido diseñados para satisfacer necesidades particulares de sistemas embebidos. USB y Ethernet son algunos de los más conocidos. Otras interfaces seriales ampliamente utilizadas son SPI, IIC, UART éstas pueden ser separados en dos grandes grupos, síncronos y asíncronos.

Las primeras requieren de una señal de reloj o tren de pulsos para mantener sincronía, los posteriores no tienen señal física de reloj, pero mantienen sincronía al especificar la velocidad de transmisión tanto en el emisor como el receptor.

Se muestra en la figura 8 el ejemplo de una interfaz serial, se transmite un bit cada pulso de reloj. Únicamente se usan dos cables

Figura 8. **Comunicación serial**



Fuente: Sparkfun. *Analog vs. Digital*. https://learn.sparkfun.com/tutorials/analog-vs-digital?_ga=2.41782754.1750709408.1575312359-80799570.1567000013. Consulta: diciembre de 2019.

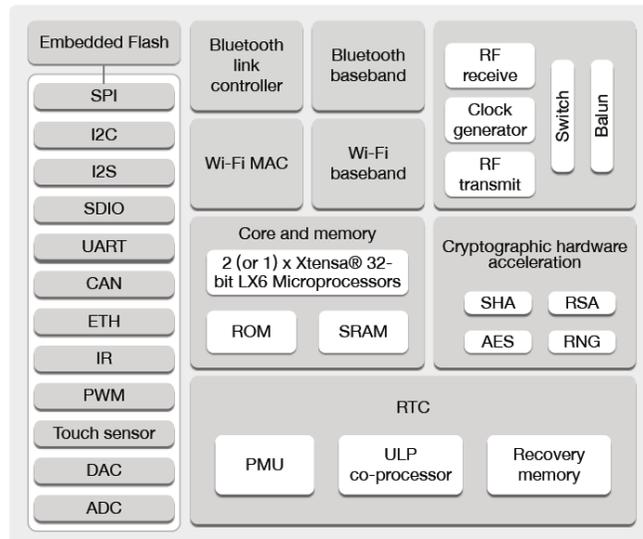
2. TEORÍA DE LOS DISPOSITIVOS

2.1. Sistema en un chip ESP32

El sistema ESP32 es un integrado que combina Bluetooth y Wifi de 2.4GHz. Está diseñado para obtener el mejor rendimiento en radio frecuencias, demostrando robustez, versatilidad y confiabilidad en gran variedad de aplicaciones y distintos escenarios de consumo de potencia. Elaborado para aplicaciones del internet de las cosas, electrónica de consumo y móviles, incluye la última tecnología característica de los chips de bajo consumo: múltiples opciones de consumo de potencia, dinámico y escalable, por ejemplo: fabricar un concentrador del internet de las cosas, la ESP se despierta periódica y únicamente cuando se detecta una condición en específico. Ciclos de baja latencia se utilizan para minimizar la cantidad de energía que el chip requiere. Entonces existe un gran equilibrio entre rango de comunicación, velocidad de transmisión y potencia de consumo.

El sistema ESP32 es una solución altamente integrada para aplicaciones Wifi y Bluetooth. Cuenta con aproximadamente 20 componentes, concentra en un solo empaque: un interruptor para la antena, balun de radio frecuencia, amplificador de potencia, amplificador de bajo ruido, filtros y módulos para la administración de poder. Además, la solución completa ocupa un área mínima del circuito impreso. La producción en masa para las soluciones basadas en el sistema ESP32 no requieren equipo de pruebas costoso y especializado para el análisis de Wifi. La figura 9 muestra un resumen de módulos y periféricos.

Figura 9. **Bloques de funciones simplificado del ESP32**



Fuente: Espressif Systems. *ESP32 series datasheet*.

https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Consulta: diciembre de 2019.

2.1.1. Especificaciones

EL módulo completo incluye una memoria SPI para almacenar las configuraciones, un cristal externo para generar los pulsos de reloj y una antena fabricada con el cobre del circuito impreso. Todo empaquetado dentro de una carcasa metálica contra interferencias electromagnéticas. Las terminales y diversas funciones se brindan en la figura 10.

Características Wifi principales.

- Compatible con el estándar 802.11 n (2.4 GHz), hasta 150 Mbps
- Desfragmentación
- Cuatro interfaces Wifi-virtuales
- Diversidad de antenas

Características Bluetooth principales.

- Transmisor clase 1, 2 y 3 sin amplificador de potencia externo
- Control de potencia mejorado
- Potencia de transmisión de hasta +12 dBm
- Conexión múltiple con Bluetooth clásico y Bluetooth de baja energía
- Escaneos y anuncios simultáneos

Microcontrolador y características avanzadas.

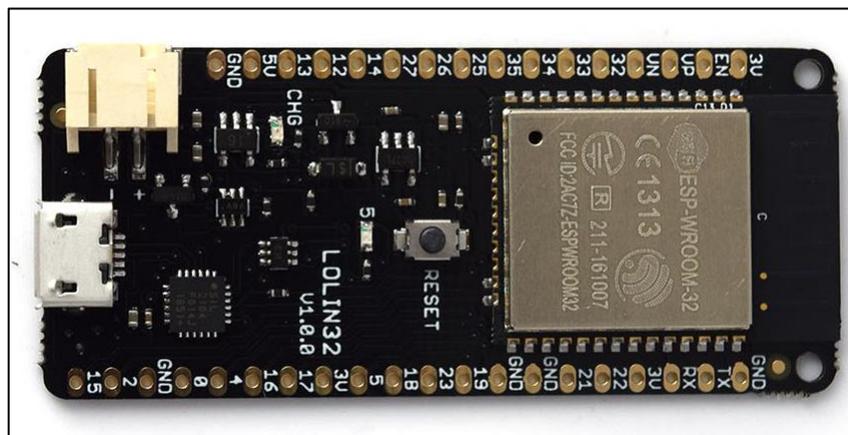
- Microprocesador de núcleo simple o doble marca Xtensa®
- 448 KB ROM
- 520 KB RAM
- Oscilador interno de 8MHz
- Temporizador de tiempo real
- 34 pines de propósito general
- Hasta 18 canales para un convertidor analógico a digital de 12 bits
- 2 convertidores digital a analógico de 8 bits
- 10 sensores táctiles
- 4 periféricos de interfaz serial
- 3 módulos de transmisión y recepción universal asíncrona

- Sensor de efecto Hall
- Hasta 16 canales para modulación de ancho de pulso

2.1.2. Plataforma de desarrollo

Para el uso del sistema ESP32 es necesario elegir una tarjeta de desarrollo. La tarjeta a utilizar llamada LOLIN32 figura 11. Integra un cargador de baterías de Litio y conexión USB para subir el código de funcionamiento de forma rápida y sencilla. Sin necesidad de adquirir un programador externo. Es recomendable elegir la tarjeta de desarrollo lo más completa posible. Algunas características importantes para elegir: número de pines accesibles, esquemáticos disponibles, conexión USB, dimensiones físicas y etiquetado legible de las terminales.

Figura 11. **Weemos LOLIN32**



Fuente: Diyprojects. *Wemos LoLin32 ESP32, wifi + bluetooth + lipo battery connector.*
<https://wiki.wemos.cc/products:lolin32:lolin32>. Consulta: diciembre de 2019.

2.1.3. Preparación del entorno de programación

Para desarrollar la programación y cargarlo a la tarjeta, se eligió software y hardware de código abierto. Permite entonces una solución de bajo coste y escalable, gracias a la constante actualización de la plataforma existen diversas versiones, dificultando el uso de un método generalizado para la correcta ejecución del prototipo por un tercero. Se preparó un repositorio con todos los archivos y enlaces necesarios para una implementación sencilla y rápida.

2.1.3.1. Repositorios de utilidad

Para acceder a los archivos únicamente es necesario descargarlos con el siguiente enlace: <https://github.com/valdezj95/ArchivosTesis>

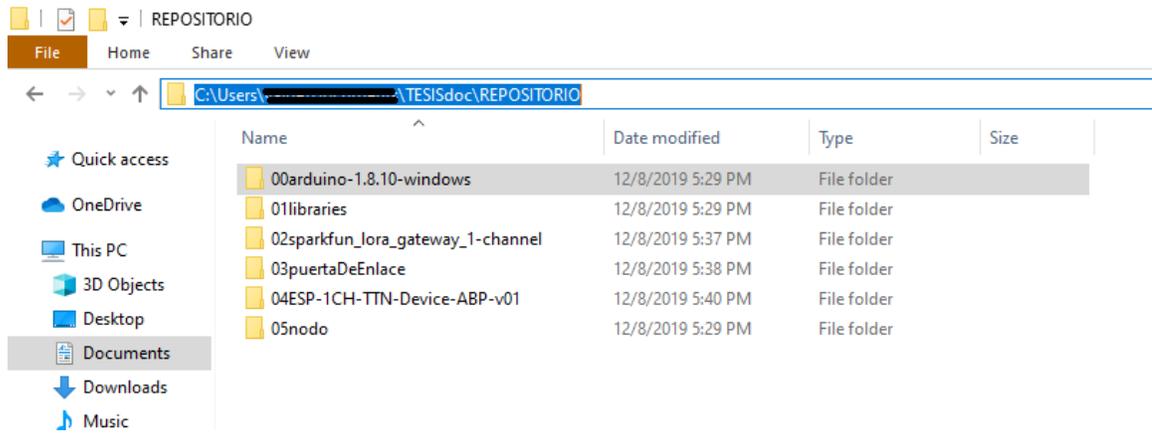
2.1.3.2. Instalación y reacondicionamiento

La computadora donde se desarrolló la implementación posee las siguientes características:

- Versión del Sistema operativo Windows 10 Home
- Procesador Intel® Core™ i3-4005U CPU @ 1.70GHz 1.7GHz
- Sistema operativo de 64 bits
- 500 GB de disco duro
- 6 GB de RAM

Cualquier computador personal con características similares o superiores serán suficientes para descargar y ejecutar los programas y carpetas sin problemas. Cabe destacar que la computadora debe tener al menos 10 GB de espacio libre.

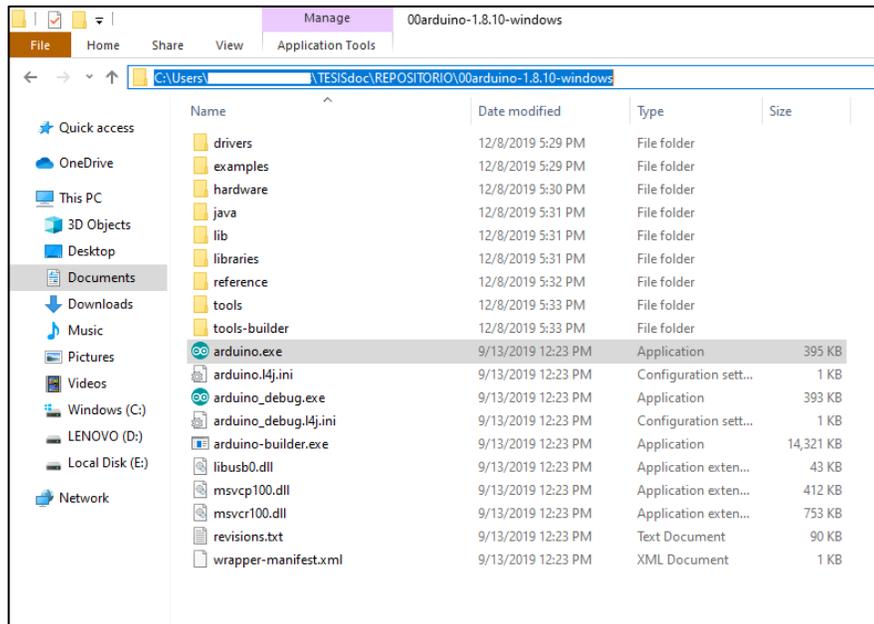
Figura 12. **Contenidos del repositorio principal**



Fuente: elaboración propia, empleando Arduino 1.8 2019.

Los archivos principales de trabajo, figura 12, contienen todo lo necesario para replicar un enlace de comunicación inalámbrica funcional. Sin embargo, en los pasos siguientes se indica dónde y cómo reemplazar las carpetas o archivos para adaptarlo a distintas tarjetas de desarrollo. En la figura 13 puede apreciarse la carpeta resaltada con el nombre: 00arduino-1.8.10-windows.

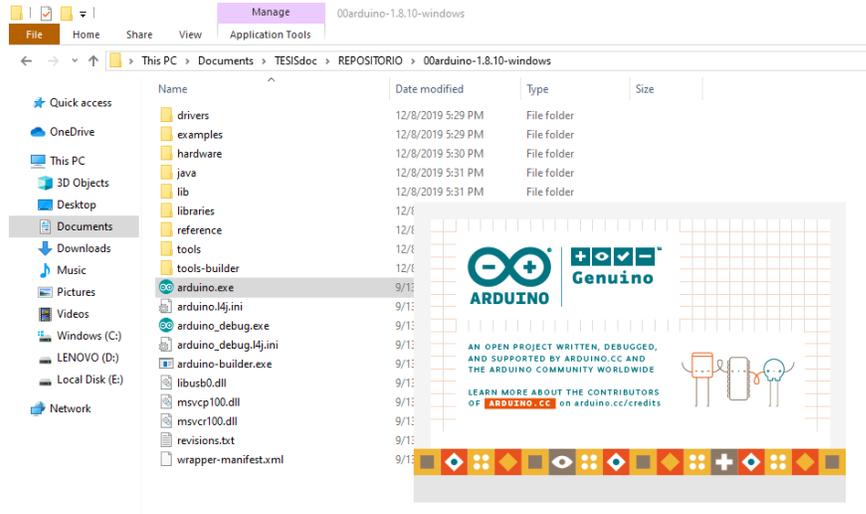
Figura 13. Carpeta del programa



Fuente: elaboración propia, empleando Arduino 1.8 2019.

Antes de realizar este paso es importante desinstalar versiones anteriores y borrar toda subcarpeta para evitar conflictos. Es posible que se requieran permisos de administrador para ejecutar el archivo indicado: arduino.exe Si se diera el caso la opción aparece al presionar el icono con el clic derecho. Al ejecutar el archivo aparecerá el mensaje de bienvenida de la figura 13.

Figura 14. Ventana de inicio



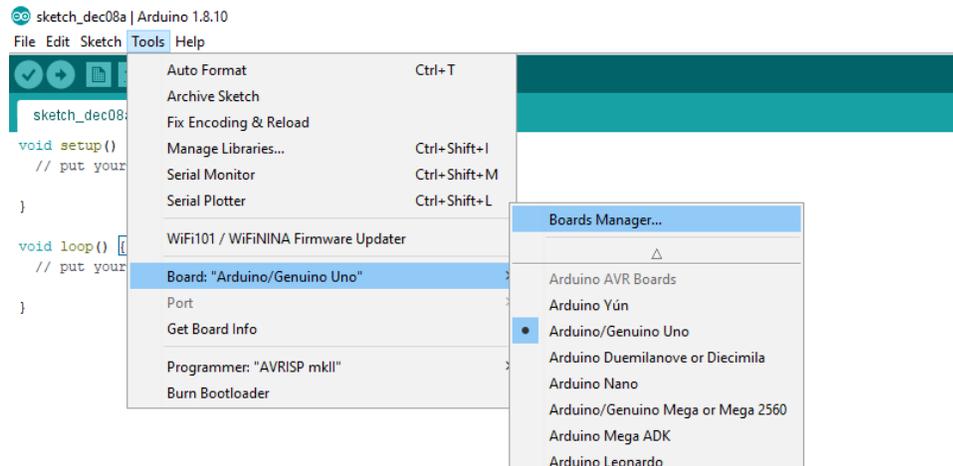
Fuente: elaboración propia, empleando Arduino 1.8 2019.

Figura 15. Arduino IDE



Fuente: elaboración propia, empleando Arduino 1.8 2019.

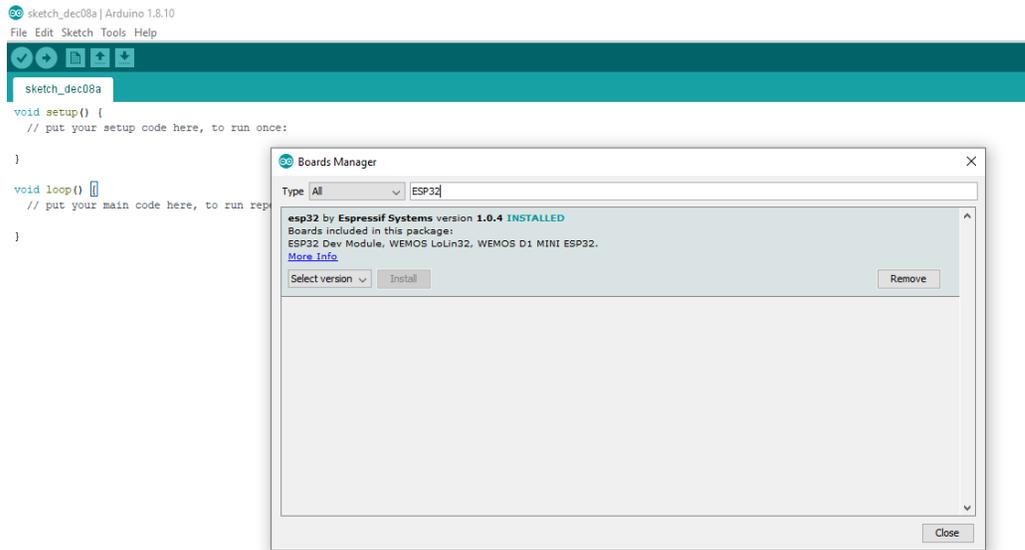
Figura 17. Ruta al administrador de tarjetas



Fuente: elaboración propia, empleando Arduino 1.8 2019.

Existen cientos de tarjetas compatibles con la plataforma haciendo extremadamente pesado el archivo de instalación. Se solucionó creando un administrador el cual se encarga de descargar el paquete de interés. Luego de seleccionar la ruta anterior se obtiene el cuadro de dialogo de la figura 18.

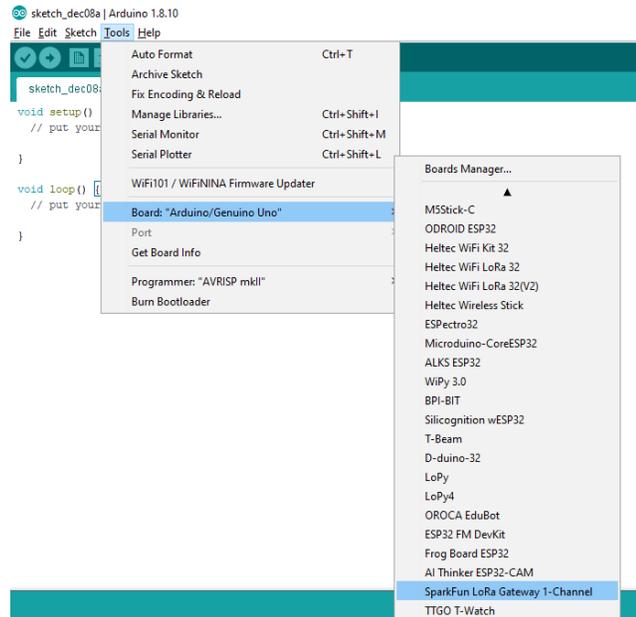
Figura 18. Administrador de tarjetas



Fuente: elaboración propia, empleando Arduino 1.8 2019.

Esperar a que la ventana de dialogo termine de cargar, proceder entonces a buscar ESP32. Seleccionar la versión 1.0.4 e instalarla, el paquete es de aproximadamente 500MB. Se requiere de una conexión a internet estable para descargarla apropiadamente.

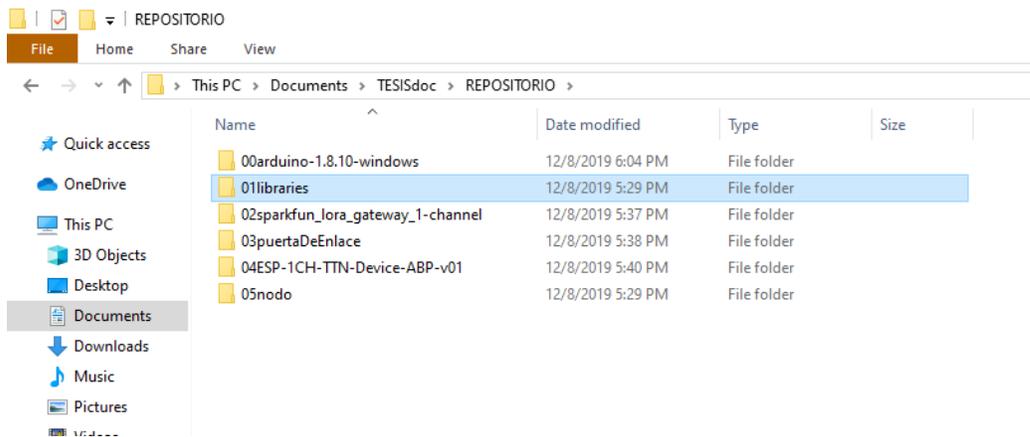
Figura 19. Selección de la tarjeta a utilizar



Fuente: elaboración propia, empleando Arduino 1.8 2019.

Una vez finalizado el proceso de descarga e instalación se recomienda cerrar el programa y abrirlo de nuevo. Posteriormente será necesario seleccionar la tarjeta en este caso: Sparkfun LoRa Gateway 1-Channel. Consultar la figura 19.

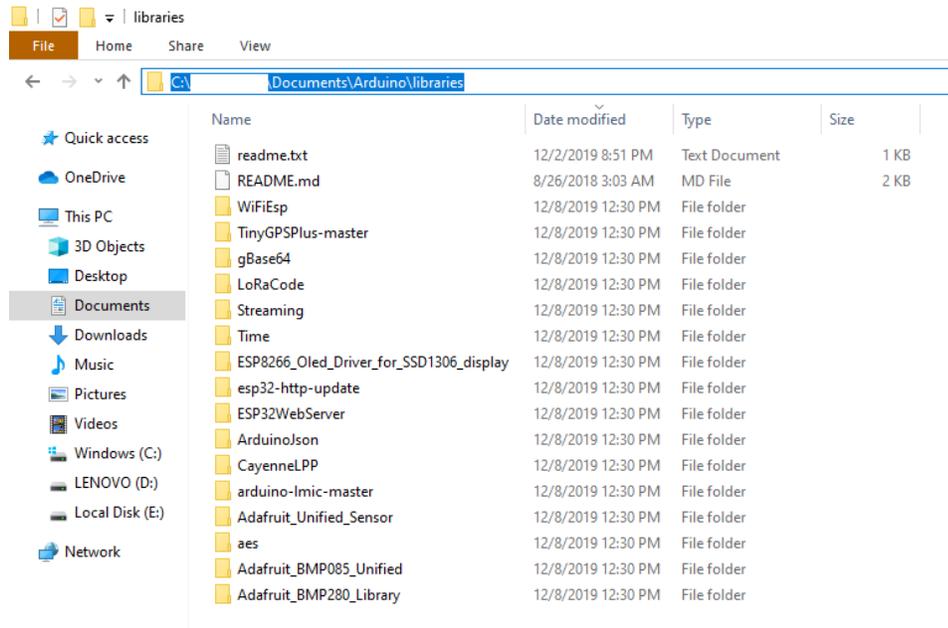
Figura 20. **Carpeta de librerías adicionales**



Fuente: elaboración propia, empleando Arduino 1.8 2019.

Al ubicar en los repositorios la segunda carpeta, figura 20, será necesario copiar sus contenidos y trasladarlos hacia la ruta: -> Documents/Arduino/libraries. Comprobar que los archivos coincidan con la figura 21.

Figura 21. Lista de librerías y ruta de traslado



Fuente: elaboración propia, empleando Arduino 1.8 2019.

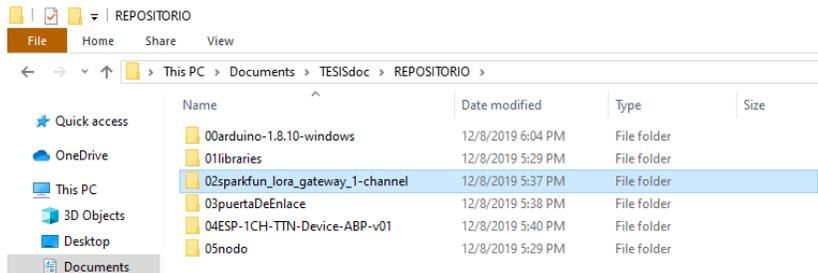
Aunque la tarjeta posea el mismo módulo, se desarrollaron distintas soluciones en las que el patillaje difiere entre modelos presentados en la figura 22. Es importante asegurarse de definir adecuadamente las funciones que cumplirán cuando se cargue el programa, esta herramienta permite al diseñador elegir otros recursos reduciendo el costo sin sacrificar funcionalidad o readaptación de otras plataformas para un uso particular.

Figura 22. Distintas tarjetas de desarrollo en torno al SoC ESP32



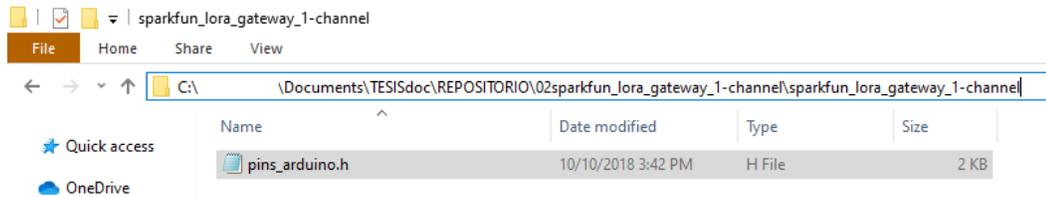
Fuente: Maker Advisor. *10 IoT development boards you need to get.*
<https://makeradvisor.com/10-iot-development-boards/>. Consulta: diciembre de 2019.

Figura 23. Carpeta con definiciones de conexión



Fuente: elaboración propia, empleando Arduino 1.8 2019.

Figura 24. **Archivo de texto para la tarjeta Weemos LOLIN32**



Fuente: elaboración propia, empleando Arduino 1.8 2019.

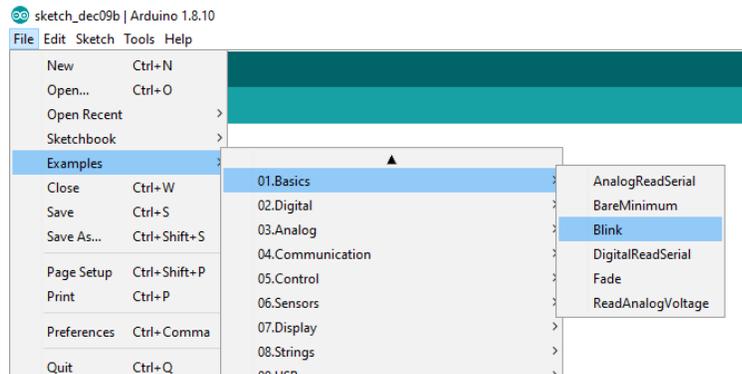
Seguir la ruta:

C:\Users\xxxxx\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4\variants\sparkfun_lora_gateway_1-channel.

Donde “xxxxx” indica el usuario en la computadora. Pegar aquí el archivo señalado en la figura 23. Es posible que alguna de las carpetas esté oculta, para hacerla visible: habilitar la opción de mostrar archivos y carpetas ocultas en la pestaña de vista. El archivo puede ser modificado para satisfacer necesidades específicas, por ejemplo, el uso de una tarjeta de desarrollo distinta a la sugerida.

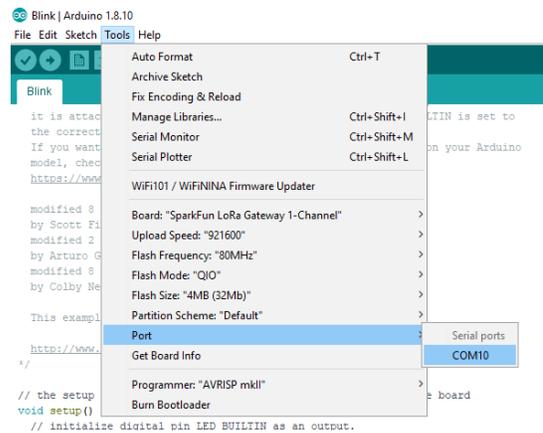
Reiniciar el programa para que se carguen los cambios, y abrir el ejemplo de Blink como se muestra en la figura.

Figura 25. Ubicación del ejemplo Blink



Fuente: elaboración propia, empleando Arduino 1.8 2019.

Figura 26. Selección del puerto correspondiente

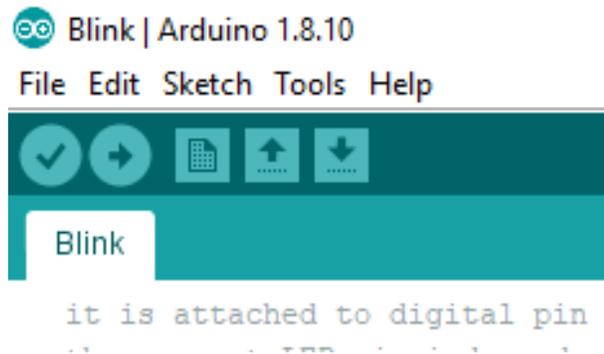


Fuente: elaboración propia, empleando Arduino 1.8 2019.

Al conectar cualquier tarjeta capaz de comunicarse a la computadora haciendo uso del puerto serial aparecerá entre las opciones disponibles como se muestra en la figura 26. El ejemplo se carga entonces con el segundo botón

de izquierda a derecha en la esquina superior izquierda, como se muestra en la figura 27.

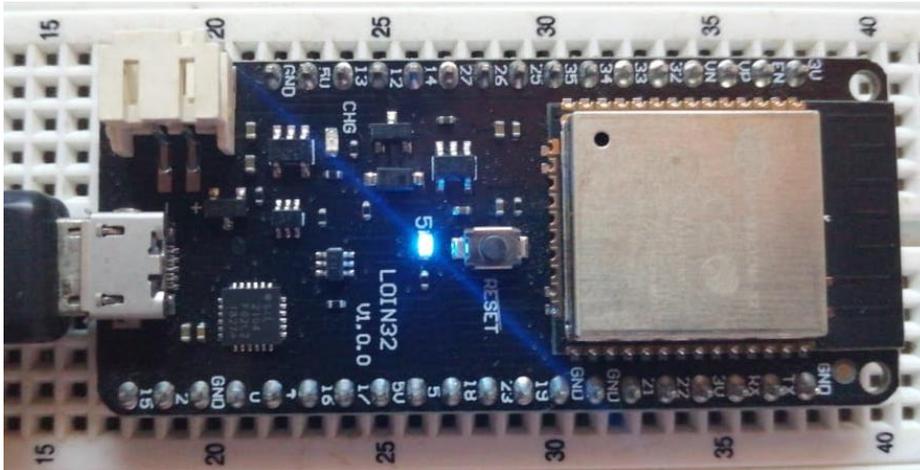
Figura 27. **Botones de: compilar, subir, nuevo, abrir y guardar un archivo**



Fuente: elaboración propia, empleando Arduino 1.8 2019.

Esperar un momento a que el compilador verifique por errores y cargue el código, el tiempo varía dependiendo del tamaño de la programación. La cantidad de tiempo requerido es directamente proporcional al tamaño del código.

Figura 28. **Parpadeo del led instalado en la tarjeta**



Fuente: elaboración propia.

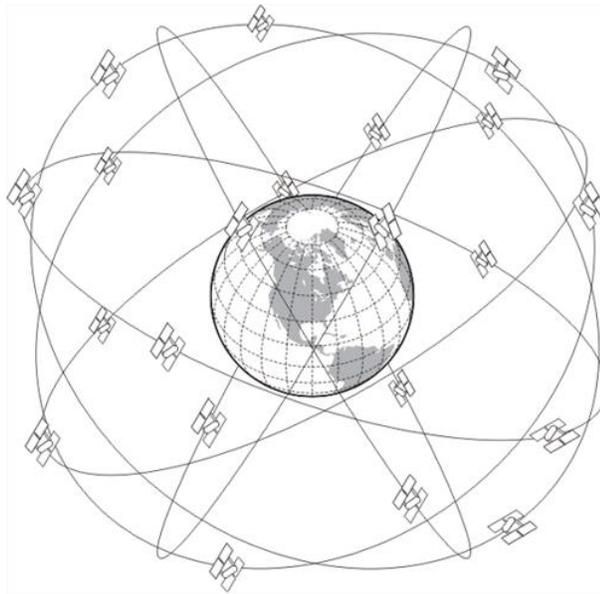
Se verifica entonces el correcto funcionamiento del entorno de trabajo en conjunto con la plataforma de desarrollo, si se aprecia el parpadeo del led aproximadamente una vez por segundo. En este punto se dispone del procesador ESP32 para el progreso de las pruebas de cada dispositivo presentado en los puntos posteriores.

2.2. Módulo GPS

El sistema de posicionamiento Geológico es una red conformada por aproximadamente 30 satélites orbitando la tierra a 20 000 Km de altura. El sistema fue diseñado originalmente por el ejército de los Estados Unidos, ahora cualquier dispositivo móvil con unidad GPS puede recibir las señales que los satélites transmiten. Los cuales orbitan la Tierra completando una vuelta cada 12 horas. Son uniformemente distribuidos para que al menos 4 satélites

siempre sean accesibles por una línea de vista desde cualquier punto del planeta.

Figura 29. **Constelación GPS**



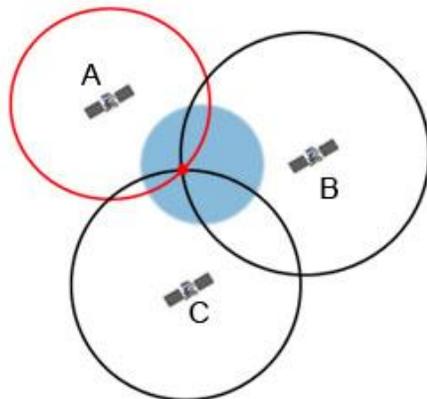
Fuente: PennState. *The space segment*. <https://www.e-education.psu.edu/geog862/node/1770>.

Consulta: diciembre de 2019.

2.2.1. Funcionamiento general

Cada satélite difunde un mensaje que contiene su posición, órbita y tiempo exacto. Un receptor GPS recibe la combinación de los mensajes difundidos por varios satélites para calcular su posición exacta utilizando un proceso llamado triangulación, figura 30. Un mínimo de 3 satélites se utiliza para obtener la posición, obtener una conexión con 4 satélites es ideal puesto que provee una mayor exactitud.

Figura 30. **Triangulación**



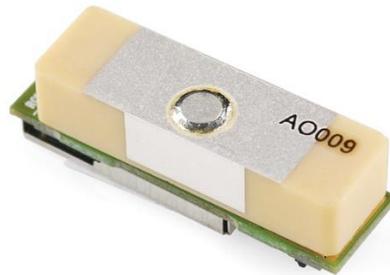
Fuente: Institute of physics. *Triangulación de satélites*. <http://www.physics.org/article-questions.asp?id=55>. Consulta: diciembre de 2019.

Para que un dispositivo GPS funcione correctamente primero tiene que establecer una conexión entre el requerido número de satélites. Dicho proceso puede demorar desde algunos segundos hasta pocos minutos dependiendo de la sensibilidad del receptor y factores ambientales. La antena es el dispositivo que cumple la función de recepción

2.2.1.1. Antena

Un módulo GPS recibe las señales de satélites a miles de kilómetros de distancia, en cualquier punto del cielo. Para un mejor desempeño se desea un camino claro entre la antena y el satélite. El clima, las nubes, y las tormentas no deberían afectar la señal, pero los árboles, edificios y montañas, crearán interferencias que afectan la precisión del GPS. Existen muchas opciones para elegir una antena, se muestra una típica antena en la figura 31.

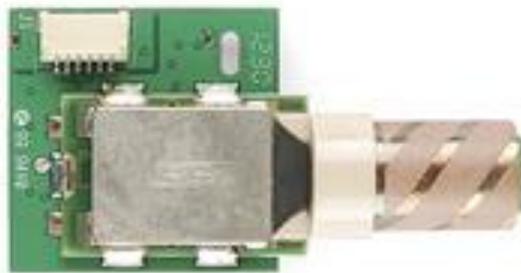
Figura 31. **Antena cerámica GPS**



Fuente: Sparkfun. *GPS Receiver - EM-506 (48 Channel)*.
<https://www.sparkfun.com/products/12751>. Consulta: diciembre de 2019.

La antena de la figura 31 es de bajo perfil, de bajo costo y compacta. Pero tiene baja recepción comparada con otros tipos de antena. Este tipo de antena necesita ser instalada con la placa de cobre hacia el suelo y poseer una vista clara al cielo lograr una buena señal de recepción. Por ejemplo, la ganancia de la antena es máxima cuando se instala viendo hacia arriba.

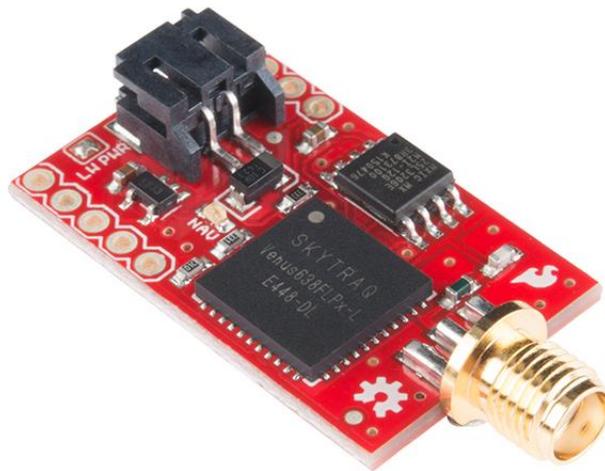
Figura 32. **Antena helicoidal GPS**



Fuente: Sparkfun. *Antena helicoidal GPS*.
<https://cdn.sparkfun.com/assets/6/3/7/5/f/50f8899f395f1b3f000000.jpg>. Consulta: diciembre de 2019.

Este tipo de antena, figura 32, suele ocupar más espacio que una antena cerámica, pero la construcción permite una mejor recepción en cualquier orientación.

Figura 33. **Módulo con conexión SMA**



Fuente: Sparkfun. *Módulo con conexión SMA.*

<https://cdn.sparkfun.com/assets/parts/6/0/5/3/10920-01a.jpg>. Consulta: diciembre de 2019.

Un acoplamiento por conector SMA proporciona la habilidad de montar la antena en un lugar distinto al del circuito principal. Este conjunto puede ser beneficioso si el sistema principal no tiene acceso directo al cielo, por ejemplo, dentro de un edificio o un carro.

2.2.1.2. Sistema integrado

Es importante identificar los componentes, conceptos y capacidades del receptor GPS, estas incluyen la antena, el procesador, canales, velocidad, formato y el software correspondiente para decodificar el mensaje. Los puntos

posteriores, hacen referencia a algunas de las características más importantes a tomar en cuenta para elegir el sistema GPS.

2.2.1.3. Exactitud

La Exactitud de un GPS puede variar. Normalmente es posible ubicarse en cualquier punto del planeta en 30 segundos, con un error de +/- 5 metros. El error existe porque la posición puede variar entre módulos, la hora, claridad en la recepción, entre otros. La mayoría de los módulos pueden obtener hasta +/- 3 metros de exactitud con funciones especiales habilitadas. Existen sistemas de corrección en tiempo real que pueden conseguir un error en el orden de los centímetros, haciéndolos sumamente costosos. En conjunto para obtener la mejor exactitud, se necesita una vista clara al cielo.

2.2.1.4. Velocidad de baudios

Algunos receptores GPS utilizan comunicación serial asíncrona para transmitir los datos a una velocidad en específico. Para los receptores de 1Hz la más común es de 9600 bps, pero 57600 bps se ha vuelto muy popular entre los nuevos dispositivos. Siempre se debe de verificar la información en la correspondiente hoja de datos.

2.2.1.5. Canales

El número de canales que el módulo GPS es capaz de ejecutar afectará el primer tiempo fijo también denominado TTFF. Puesto que el receptor no conoce cual satélite establecerá comunicación, mientras más canales pueda monitorear al mismo tiempo, más rápido encontrará un punto fijo. Algunas plataformas apagan los demás canales al estar anclados esto quiere decir que hay al menos

4 satélites disponibles para obtener información, reduciendo la potencia de consumo.

2.2.1.6. Chipset

Es el responsable de realizar todo proceso electrónico, desde ejecutar cálculos hasta proveer la circuitería analógica para la antena, desde la etapa de potencia hasta la interfaz para transmitir la información. El Chipset es independiente al tipo de antena, por lo tanto, existe un amplio rango de opciones para elegir. Algunos como: ublox, SiRF, y SkyTraq contienen procesadores poderosos que permiten una rápida adquisición y gran confiabilidad. Las diferencias entre los Chipsets usualmente se obtienen con el balance de la potencia de consumo con tiempos de adquisición, y accesibilidad de hardware.

2.2.1.7. GPS diferencial

También denominado DGPS, es un tipo específico de receptor GPS. Contienen una antena adicional para obtener señales de estaciones terrestres, además de las satelitales. Por ese motivo los módulos GPS diferenciales requieren de 2 antenas, solución más voluminosa y costosa, pero consiguiendo posición con centímetros de tolerancia.

2.2.1.8. Ganancia

Se denomina ganancia a la eficiencia de la antena dada una orientación en específico, aplicando tanto a las antenas de recepción como de transmisión.

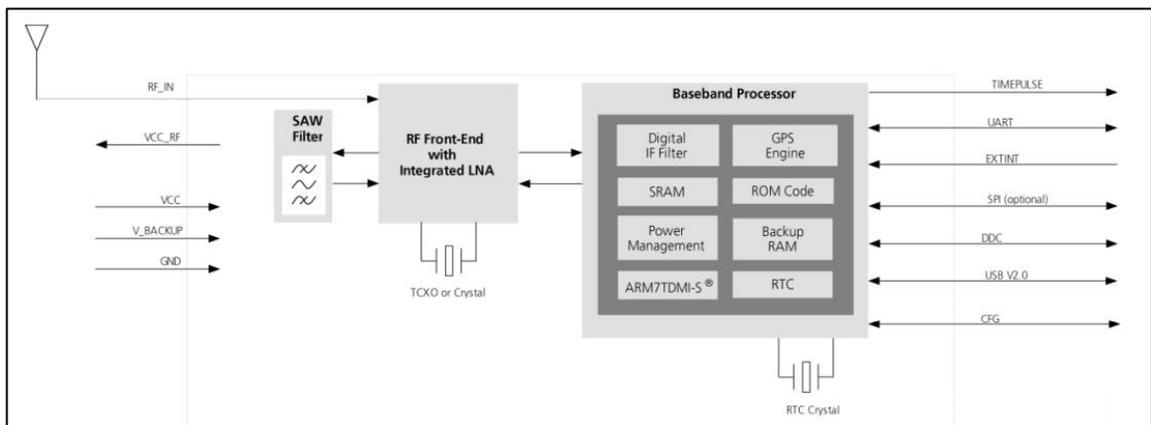
2.2.2. Módulo u-blox NEO-6M

El dispositivo forma parte de una familia de receptores GPS totalmente integrados proporcionando alto rendimiento. Estos flexibles y económicos receptores ofrecen numerosas opciones de conectividad en un tamaño compacto. La arquitectura, potencia de consumo y opciones de memoria hacen estos módulos la opción ideal para aparatos móviles operados con batería que además poseen restricciones tanto en espacio como en costo.

2.2.2.1. Descripción del módulo

Es importante reunir las características generales de funcionamiento para incorporar adecuadamente la lógica y no dañar el dispositivo.

Figura 34. Diagrama de bloques NEO-6



Fuente: U-blox. *NEO – 6*. [https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_\(GPS.G6-HW-09005\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf). Consulta: diciembre de 2019.

El motor de posicionamiento de 50 canales ofrece un anclaje del dispositivo en menos de 1 segundo. El motor de adquisición es capaz de realizar búsquedas masivas tanto en tiempo como en frecuencia, habilitándolo para encontrar satélites instantáneamente. Tecnología y diseño innovador suprime y mitiga fuentes de ruido, dándole al módulo de recepción GPS NEO-6 excelente desempeño en navegación incluso en ambientes complicados.

2.2.2.2. Límites máximos absolutos

Sobrepasar los límites máximos absolutos puede ocasionar daño permanente al dispositivo.

Tabla I. **Límites de funcionamiento NEO-6M**

Parámetro	Símbolo	Mín.	Máx.	Unidades
Voltaje de alimentación	VCC	-0,5	3,6	V
Voltaje de la batería de respaldo	V_BCKP	-0,5	3,6	V
Voltaje USB	VDDUSB	-0,5	3,6	V
Voltaje de entrada de las terminales	Vin	-0,5	3,6	V
Corriente a través de las terminales	Ipin		10,0	mA

Fuente: elaboración propia.

El módulo no cuenta con protecciones contra sobre voltajes o polaridad invertida. En caso de ser necesario, picos de voltaje que excedan los valores dados en la tabla I tendrán que ser limitados dentro de las fronteras establecidas utilizando diodos de protección apropiados.

2.2.2.3. Condiciones de operación

Se establecen los parámetros ideales de funcionamiento en las cuales el dispositivo puede realizar las tareas asignadas.

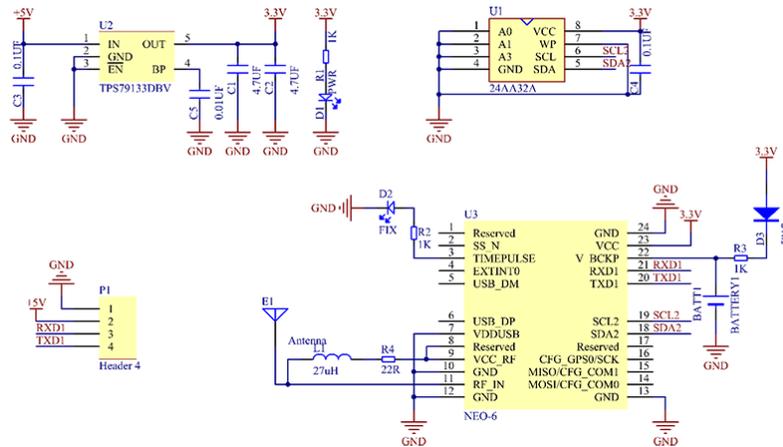
Tabla II. Condiciones de operación

Parámetro	Símbolo	Mín.	Típ.	Máx.	Unidades
Voltaje de alimentación	VCC	2,7	3,0	3,6	V
Voltaje de la batería de respaldo	V_BCKP	1,4		3,6	V
Voltaje USB	VDDUS				
Voltaje USB	B	3,0	3,3	3,6	V
Voltaje de entrada de las terminales	Vin	0		VCC	V
Voltaje de entrada, 0 lógico	Vil	0		0,2*VCC	V
Voltaje de entrada, 1 lógico	Vih	0,7*VCC		VCC	V
Voltaje de salida, 0 lógico	Vol			0,4	V
Voltaje de salida, 1 lógico	Voh	VCC-0,4			V
Temperatura de trabajo	Topr	-40		85	°C

Fuente: elaboración propia.

2.2.2.4. Circuitos propuestos

Figura 35. Circuito esquemático del módulo GY-NEO6MV2



Fuente: Sunfounder. *Ublox NEO-6M GPS module*.

http://wiki.sunfounder.cc/index.php?title=Ublox_NEO-6M_GPS_Module. Consulta: diciembre de 2019.

El lado superior izquierdo de la figura 35 corresponde al bloque para la regulación de voltaje, garantizando el rango de alimentación VCC sugerido en la tabla II. Además, se incluye una memoria externa para guardar configuraciones de respaldo. Las terminales de comunicación RXD1 y TXD1 hacen referencia al protocolo de comunicación UART.

2.2.2.5. Tarjeta de pruebas

El mercado ofrece de manera conveniente las tarjetas de prueba e identificación de patillaje.

Figura 36. **Tarjeta de pruebas o módulo GY-NEO6MV2**



Fuente: Eprolabs. Archivo *WL-00144.jpg*. <https://wiki.eprolabs.com/index.php?title=File:WL-00144.jpg>. Consulta: diciembre de 2019.

Uno de los módulos más populares y económicos del mercado presentado en la figura 36, adquirido individualmente o en juego. Puesto que el módulo es inutilizable sin antena, usualmente se venden juntos, Se puede apreciar en la

esquina superior derecha un conector U.FL macho: es una terminal diseñada especialmente para adjuntar antenas de alta frecuencia de manera estándar.

El conector U.FL de la figura 37 no está diseñado para conexiones y desconexiones de manera constante o repetitiva, se recomienda evitar el movimiento del conector para reducir desgaste.

Figura 37. **Conector U.FL**



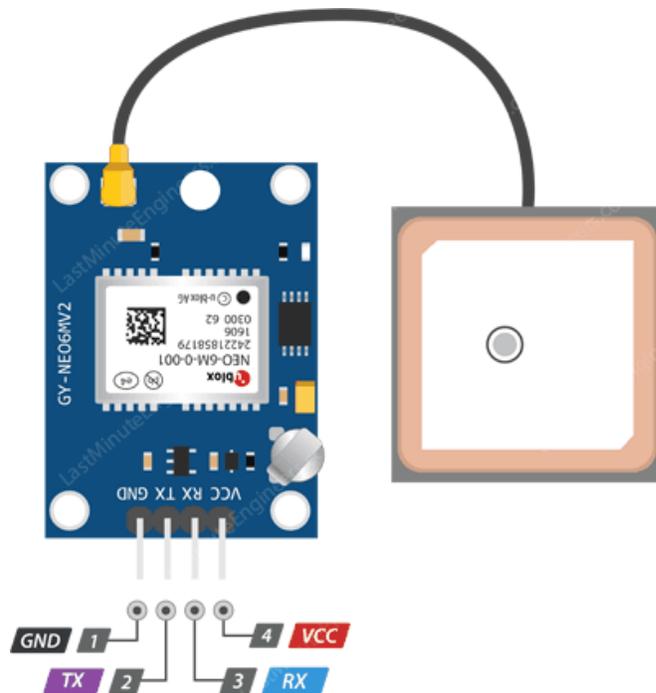
Fuente. Adafruit. *uFL SMT antenna connector*. <https://www.adafruit.com/product/1661>. Consulta: diciembre de 2019.

2.2.2.5.1. Diagramas

Existe una gran variedad de módulos, antenas, chipsets, precios y métodos de comunicación por elegir. Abriendo el camino a un sinnúmero de combinaciones funcionales. Para garantizar la adecuada coordinación del sistema completo es indispensable conocer los parámetros eléctricos de todos los dispositivos. Información que se brindará en el documento para el desarrollo de la solución trabajada. Los datos fueron extraídos de la hoja de datos del circuito integrado en particular, en general el lector puede elegir el juego de piezas a su criterio conservando funcionalidad si respeta compatibilidad de

protocolos de comunicación, conexiones físicas y parámetros eléctricos entre dispositivos.

Figura 38. **Conexión de la antena GPS**



Fuente: Last Minute Engineers. *Ublox-NEO-6M-GPS module pinout.*

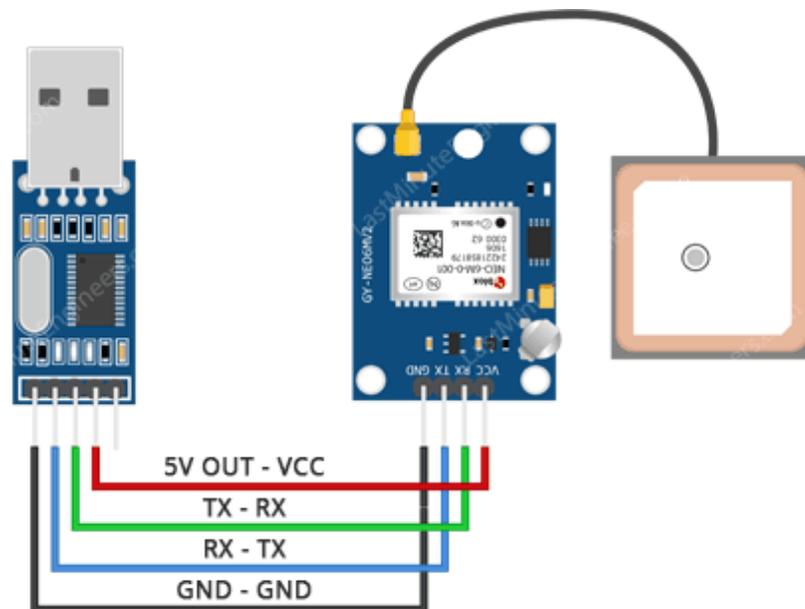
[https://lastminuteengineers.com/wp-content/uploads/2018/11/Ublox-NEO-6M-GPS-Module-](https://lastminuteengineers.com/wp-content/uploads/2018/11/Ublox-NEO-6M-GPS-Module-Pinout.png)

[Pinout.png.](https://lastminuteengineers.com/wp-content/uploads/2018/11/Ublox-NEO-6M-GPS-Module-Pinout.png) Consulta: diciembre de 2019.

- La terminal 1 de la figura 38. corresponde a la referencia de la alimentación GND.
- La terminal 2 de la figura 38. transmite los datos del módulo en formato digital serial.

- La terminal 3 de la figura 38. recibe los datos del controlador en formato digital serial.
- La terminal 4 de la figura 38. corresponde al voltaje de alimentación VCC.

Figura 39. **Conexión a un puente de comunicación USB-UART**



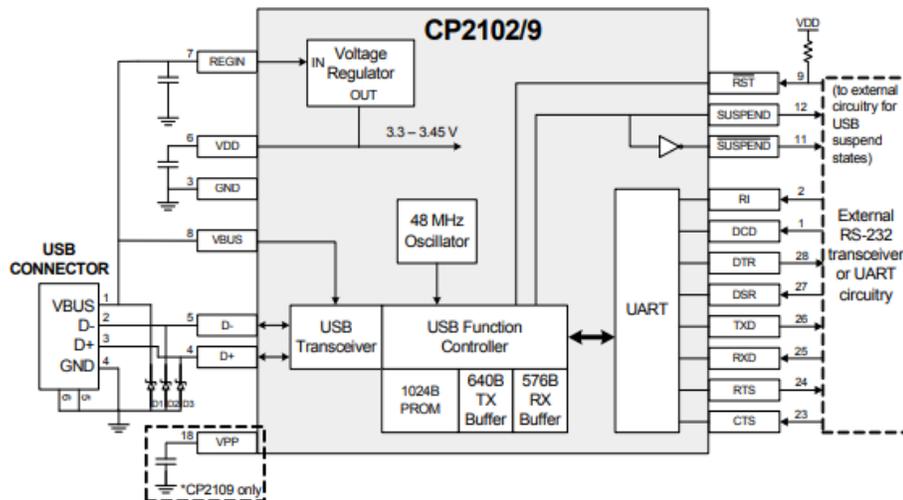
Fuente: Last Minute Engineers. *Direct computer wiring NEO 6M GPS module with PL2303 USB to TTL converter.* <https://lastminuteengineers.com/wp-content/uploads/2018/11/Direct-Computer-Wiring-NEO-6M-GPS-Module-with-PL2303-USB-to-TTL-Converter.png>. Consulta: diciembre de 2019.

Un puente de comunicación, figura 39, se encarga de establecer la traducción entre 2 protocolos distintos, existen puentes más sofisticados que pueden interactuar con diversos formatos, elevando su precio y tamaño. Gracias a que los sensores y módulos poco a poco han adquirido la capacidad de comunicarse digitalmente con protocolos estándar, vuelven al puente de

comunicación una herramienta que no puede faltar en la mesa de trabajo. De hecho, la tarjeta de desarrollo Weemos Iolín32 utiliza este tipo de sistema para establecer comunicación con la computadora.

El puente de comunicación USB-UART, también conocido como USB-TTL o USB-SERIAL. Realizará entonces las traducciones correspondientes sin necesidad de conocer intensivamente ambos protocolos. De cualquier forma, es importante verificar la compatibilidad eléctrica entre ambos dispositivos, esta dependerá del puente disponible. En este caso se muestra la información del Circuito integrado CP2102.

Figura 40. Diagrama a bloque del CI CP2102



Fuente: Silicon labs. *Single chip usb to uart bridge datasheet*.

<https://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf>. Consulta: diciembre de 2019.

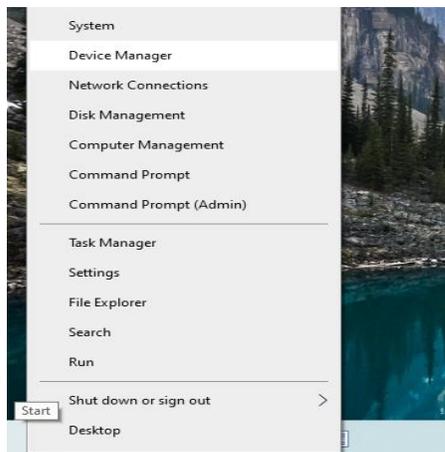
Otras opciones populares son FTDI232 y CH340 que corresponden a marcas distintas ofreciendo distintos enfoques en el uso de software. Es posible que los puentes de comunicación USB no sean reconocidos por la computadora en un primer uso, puesto que en su mayoría necesitan un controlador instalado en la computadora para funcionar adecuadamente. Las últimas versiones de Windows traen varios paquetes por defecto, solucionando el inconveniente.

Tabla III. **Condiciones de operación**

Parámetro	Símbolo	Mín.	Típ.	Máx.	Unidades
Voltaje de alimentación	VDD	3,0	3,3	3,6	V
Rango de voltaje de entrada	VREGIN	4,0		5,25	V
Voltaje de entrada, 0 lógico	Vil	0		0,8	V
Voltaje de entrada, 1 lógico	Vih	2,0			V
Voltaje de salida, 0 lógico	Vol		1		V
Voltaje de salida, 1 lógico	Voh		VDD-0,8		V

Fuente: elaboración propia.

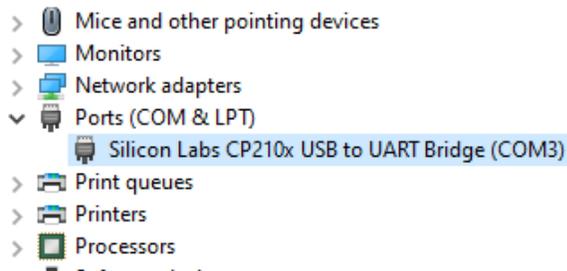
Figura 41. **Administrador de dispositivos**



Fuente: elaboración propia, empleando Printscreen de Windows.

Para desplegar las opciones como se muestra en la figura 41. Presionar con clic derecho el botón de inicio.

Figura 42. **Puertos virtuales**



Fuente: elaboración propia, empleando Printscreen de Windows.

Como se aprecia en la figura 42. Existe un puerto COM3, el número de puerto será útil para que el software a utilizar establezca conexión con ese canal. Además, se tiene un número asignado para no mezclar la información que provenga de otros puertos virtuales utilizados por la computadora.

2.2.2.5.2. Lectura de datos

El puente comunicación tiene un controlador para instalar en la computadora. Para tener una lectura de datos es necesario configurar algunas opciones del monitor serial. Un monitor serial es una consola que despliega y envía información del puerto serial utilizando el código ASCII. ASCII significa, código estándar americano para intercambio de información, es un código de 7 u 8 bits estándar o extendido respectivamente. Un carácter corresponde a una única combinación de bits similar a un diccionario o una tabla, es útil el uso de sistemas de numeración binaria, decimal y hexadecimal para abreviar los

equivalentes. Es más común trabajar una tabla de símbolos ASCII en su correspondiente equivalente hexadecimal como se muestra en la tabla IV.

Tabla IV. **Porción de la tabla del código ASCII**

DEC	HEX	SIMBOLO
40	28	(
41	29)
42	2A	*
43	2B	+
44	2C	,
45	2D	-

Fuente: elaboración propia.

El número decimal 40 corresponde al símbolo o carácter “(“el monitor serial conoce todas y cada una de las traducciones para desplegar y ejecutar en pantalla. Existen caracteres de control por ejemplo ENTER y caracteres imprimibles como las letras. Es importante entonces diferenciar entre un número decimal 100 y una cadena de 3 caracteres conformada por 3 números binarios correspondientes.

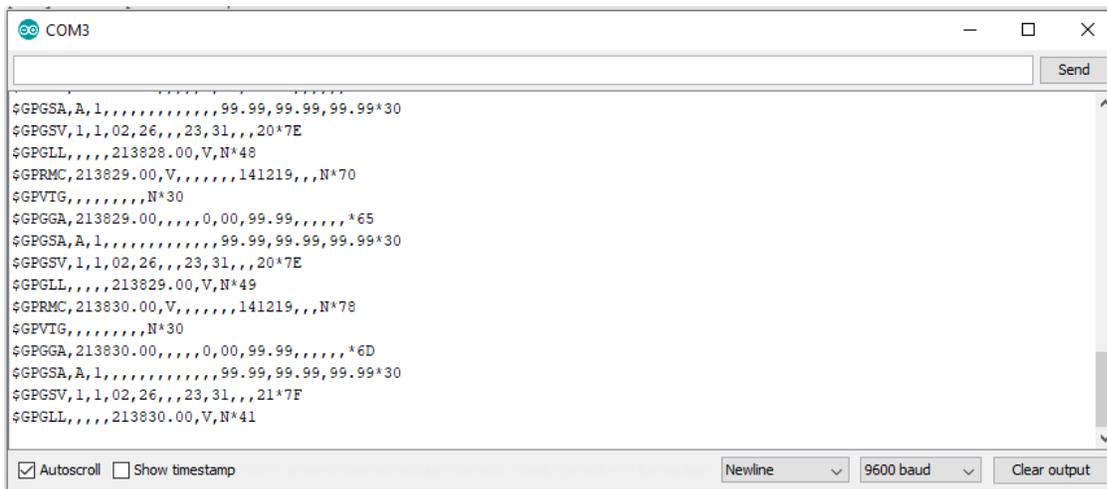
Figura 43. **Monitor serial de Arduino IDE**



Fuente: elaboración propia, empleando Printscreen de Windows.

Al ejecutar el archivo Arduino.exe brindado en el repositorio principal como se muestra en la figura 12. Y repetir el proceso de la figura 26. Para verificar el puerto en uso. Buscar en la esquina superior derecha el símbolo de la figura 43. Y presionar el clic derecho.

Figura 44. **Monitor serial**



Fuente: elaboración propia, empleando Printscreen de Windows.

Se muestra en la figura 44 el monitor serial de Arduino IDE, posee opciones de configuración sumamente básicas, pero necesarias. Si se desea enviar un carácter desde la computadora hacia el dispositivo, carácter de finalización y velocidad de transmisión. Recordar que el protocolo de comunicación digital en cuestión es UART también conocida como transmisión y recepción universal asíncrona. Ya que se carece de una línea de reloj para la sincronización, tanto el dispositivo que transmite como el que recibe poseen la misma velocidad de comunicación, llamada baudio.

Además, pueden apreciarse algunas cadenas de caracteres en la ventana de recepción. Las cadenas de caracteres en conjunto forman oraciones. La información obtenida en la interfaz serial son oraciones NMEA. NMEA es un acrónimo traducido de: Asociación Electrónica Nacional Marítima. Esto es un formato de mensaje estandarizado, prácticamente para todos los receptores GPS, que separan la información utilizando comas y facilitar su aislamiento mediante computadoras y microcontroladores. Las oraciones NMEA se refrescan a intervalos regulares también conocido como frecuencia de actualización.

Por defecto, el módulo NEO-6M actualizan su información 1 vez por segundo o 1Hz de frecuencia. Para analizar las oraciones NMEA es necesario conocer sus contenidos, existen muchos tipos de oraciones, pero las más comunes son:

\$GPRMC: contienen la información mínima de coordenadas recomendadas para el posicionamiento global: tiempo, fecha, latitud, longitud, altitud y velocidad estimada.

\$GPGGA: provee la información esencial para generar el anclaje, esta se refiere a la localización 3D y exactitud.

Por ejemplo:
\$GPRMC,223542.00,A,1439.95,N,09029.07,W,0.117,,141219,,,A*6F
y su significado dado en la tabla V.

Tabla V. **Sentencia NEMA GPRMC**

\$	Todas las oraciones NMEA inician con este carácter.
GPRMC	Coordenadas mínimas recomendadas de posicionamiento global.
223542.00	Tiempo en UTC - 22:35:42
A	Estado A= activo V= inválido
1439.95,N	Latitud 14 grados 39.95' N
9029.07,W	Latitud 90 grados 29.08' O
0.117	Velocidad en nudos
141219	Fecha 14/12/19
6F	Dato de comprobación, siempre inicia con *

Fuente: elaboración propia.

Otro formato disponible:

\$GPGGA,224117.00,1439.96304,N,09029.06806,W,1,06,2.30,1474.9,M,-4.8,M,,*69 y su significado dado en la tabla VI.

Tabla VI. **Sentencia NMEA GPGGA**

\$	Inicio de la sentencia NMEA
GPGGA	Datos de anclaje para el sistema de posicionamiento global.
224117.00	Tiempo actual en UTC-22:41:17
1439.96,N	Latitud 14 grados 39,95' N
9029.06,W	Latitud 90 grados 29,08' O
1	Anclaje GPS
06	Número de satélites rastreados.
2.30	Dilución horizontal de posición.
1474.9,M	Altura promedio sobre el nivel de mar.
-4.8,M	Altura del geoide promedio al nivel del mar.
*69	Dato de comprobación, siempre inicia con *

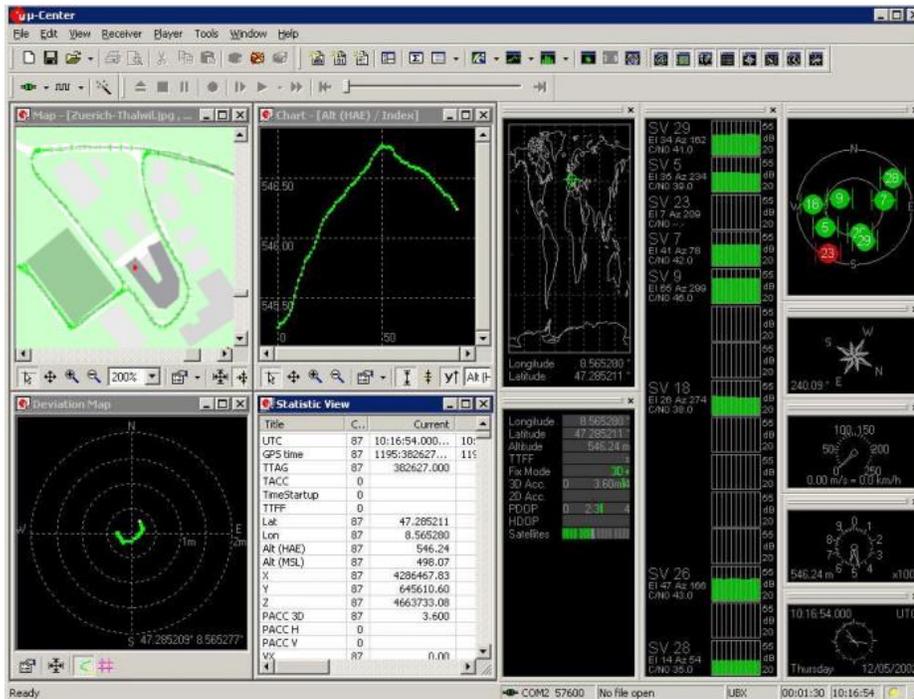
Fuente. elaboración propia.

- u-center

Existen más de 2 tipos de sentencias NMEA, no es necesario consultar una tabla y darle una traducción a la posición que le corresponde a cada sección, volvería el proceso muy lento si se desea adquirir toda la información proveniente del módulo. U-blox el fabricante del módulo ofrece una solución integrada para visualizar todos los datos de forma cómoda: u-center, consultar figura 45, es un software de evaluación GNSS para Windows. Algunas de sus características más importantes:

- Altamente interactivo y fácil de utilizar
- Soporte completo para todos los módulos u-blox receptores GNSS.
- Extensas configuraciones y control de características.
- Visualización en tiempo real de un receptor GNSS vía RS232 o USB.

Figura 45. u-center

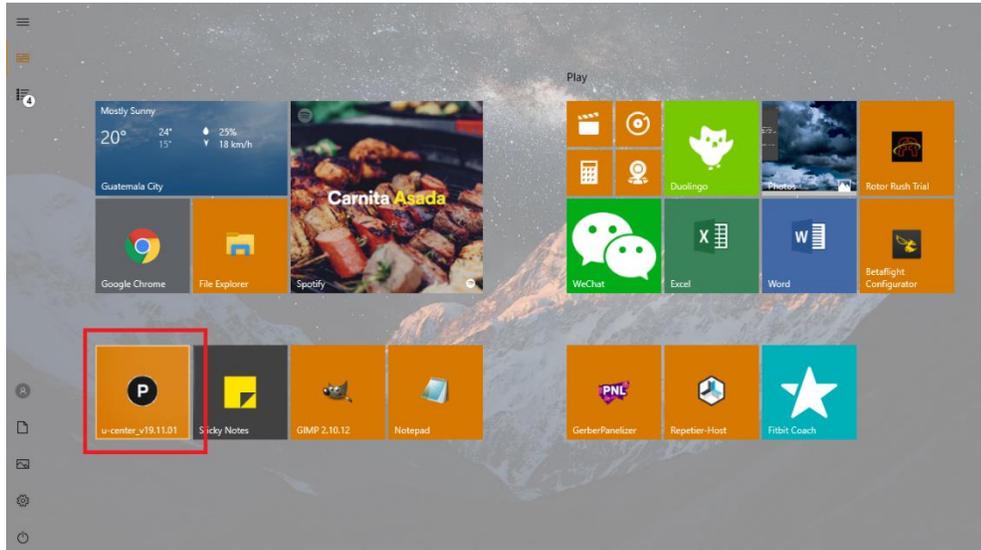


Fuente: U blox. *u-center*. <https://www.u-blox.com/en/product/u-center>. Consulta: diciembre de 2019.

- Instalación del programa u-center

El programa de instalación sirve de guía para realizar los pasos necesarios y efectuar una instalación exitosa del programa. Luego de la instalación, se puede iniciar u-center desde el menú de inicio como se muestra en la figura 46.

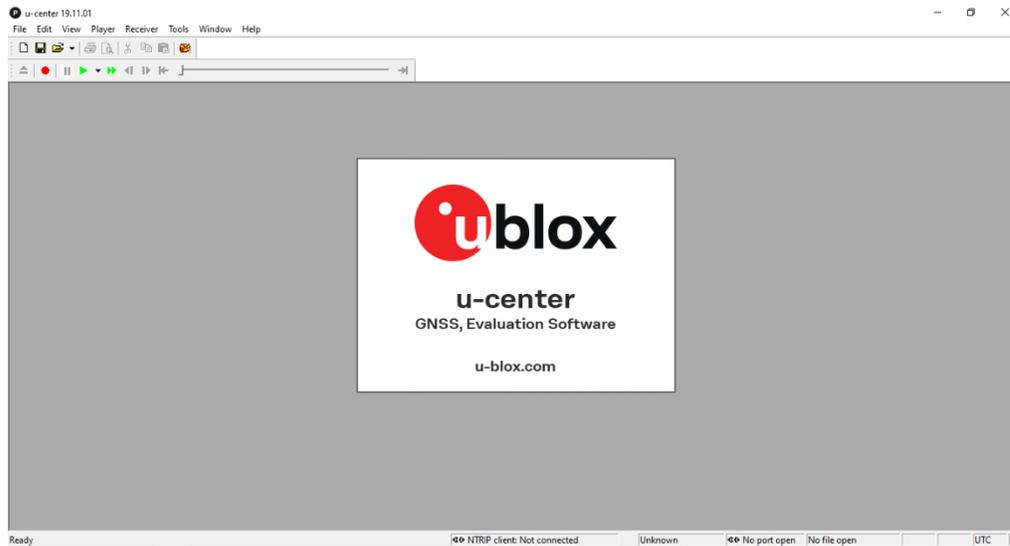
Figura 46. **u-center en la ventana de inicio**



Fuente: elaboración propia, empleando u-center 2020.

Luego de iniciar el programa aparecerá la ventana predeterminada de configuraciones como se muestra en la figura 47, el módulo se encuentra conectado a la computadora indicado en la figura 42, pero aún no se ha configurado el puerto que dirige la información al software.

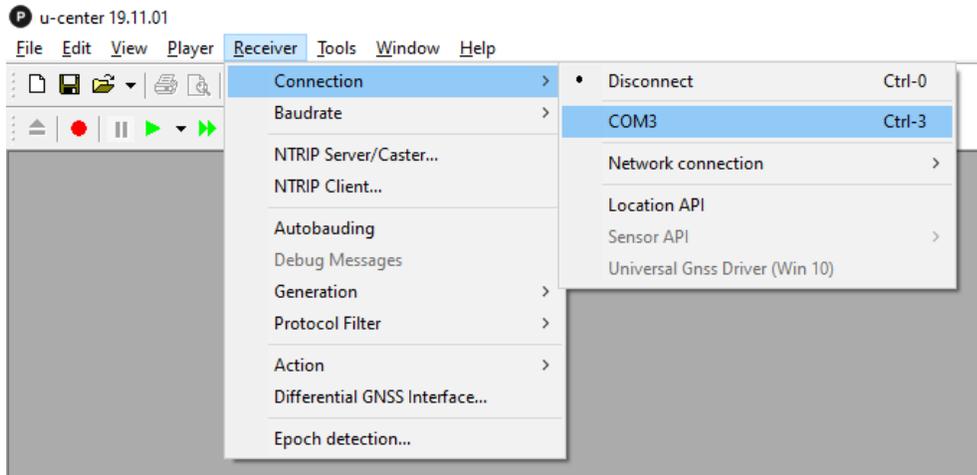
Figura 47. Ventana de inicial de configuraciones



Fuente: elaboración propia, empleando u-center 2020.

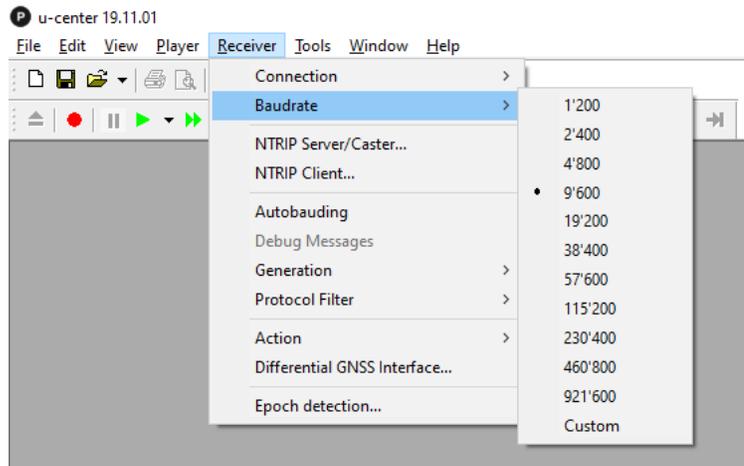
Para establecer conexión: localizar la barra del receptor como se visualiza en la figura 48, en este punto se podrá apreciar el puerto, COM3. Luego elegir la velocidad de comunicación como se muestra en la figura 49, usualmente 9600 baudios.

Figura 48. Puerto de comunicación en u-center



Fuente: elaboración propia, empleando u-center 2020.

Figura 49. Velocidad de transmisión en u-center



Fuente: elaboración propia, empleando u-center 2020.

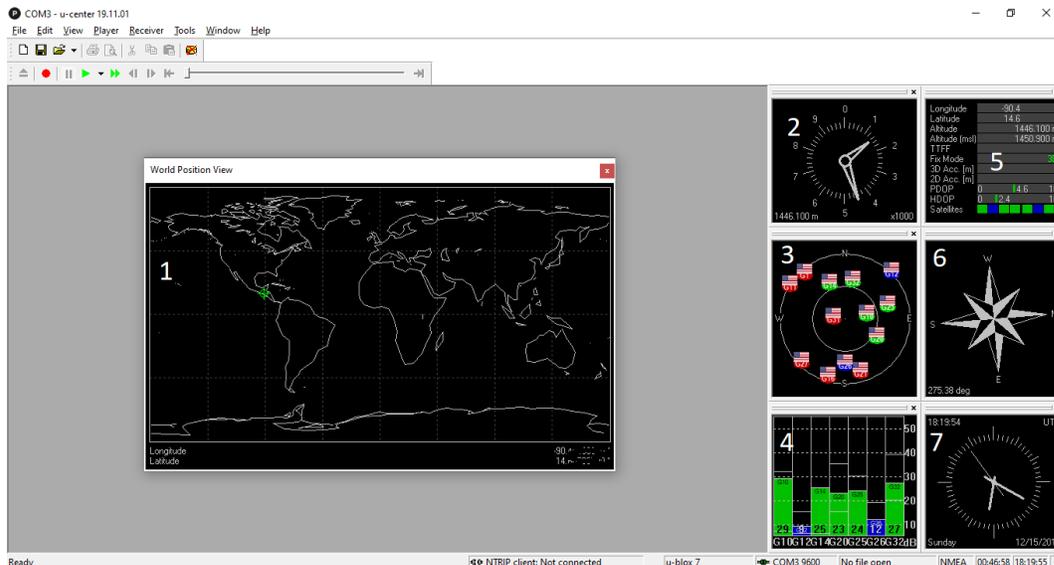
2.2.2.6. Monitoreo de posición

Al configurar todos los pasos anteriores el programa será capaz de desplegar de una manera gráfica la información previamente descrita de las sentencias NMEA. Facilitando la interpretación de los datos. Como se presenta en la figura 46.

2.2.2.6.1. Gráficas de posición

El software para comprobar el funcionamiento del dispositivo y desplegar información de manera amigable con el usuario.

Figura 50. Entorno gráfico u-center



Fuente: elaboración propia, empleando u-center 2020.

La ventana de posición global utiliza la longitud y latitud para ubicar el sistema en un mapa bidimensional. En la figura 50.1 pueden apreciarse las

coordenadas de Guatemala. Un medidor de altura: figura 50.2 proporciona un método similar a los medidores analógicos presentes en los aviones y sistemas de navegación antiguos, únicamente para desplegar altura de manera gráfica. Todos los satélites se encuentran en constante movimiento, el mapa presentado en la figura 50.3 Indica la posición de los satélites que han enviado sus datos. Cabe resaltar que únicamente los satélites al alcance del módulo son desplegados. En la figura 50.4 Puede apreciarse la identificación del satélite y la intensidad de la señal recibida en decibelios, en telecomunicaciones se le conoce como RSSI y la tabla de datos generales: figura 50.5, muestra la información extraída de la sentencia NMEA.

El posicionamiento 3D además de incluir: longitud, latitud y altitud. Proporciona orientación respecto al norte terrestre, El dato no es generado propiamente desde el módulo GPS, particularmente el modelo utilizado carece del sistema requerido. La información de orientación es un estimado realizado a partir de cálculos.

Las observaciones del clima alrededor del mundo siempre se toman respecto a un tiempo estándar. Por convención, antes del año 1972, se denominaba el promedio de tiempo de Greenwich GMT. Ahora es referido como tiempo universal coordinado UTC. Las zonas horarias corresponden a la cantidad de horas que se añaden o sustraen al tiempo universal coordinado, Por ejemplo, la zona horaria en Guatemala es UTC-6.

2.3. Módulo monitor de batería

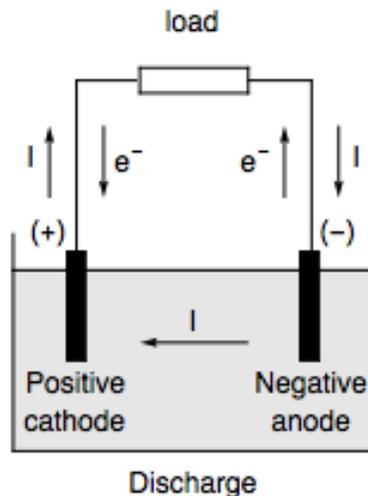
Las baterías son colecciones de una o varias celdas, que a base de reacciones químicas crean el flujo de electrones presente en un circuito. Todas las baterías poseen 3 componentes básicos: ánodo, cátodo y algún tipo de

electrolito, una sustancia que reacciona con los dos primeros. Esta reacción produce el flujo de los electrones en el circuito que a su vez regresan a la terminal negativa. Cuando el material en el cátodo o el ánodo se consumen o ya no puede usarse en la reacción, la batería es incapaz de producir electricidad. En ese punto la batería se encontrará vacía o muerta.

Las baterías que tienen que ser tiradas después de usadas se denominan baterías primarias. Aquellas que pueden ser recargadas son conocidas como baterías secundarias. Ambos, el ánodo y el cátodo son tipos de electrodos los cuales conducen la electricidad que entra o sale en los componentes de un circuito.

La corriente de electrones fluye desde aquí, eso quiere decir que la corriente convencional entra al ánodo. Como se muestra en la figura 51.

Figura 51. **Descarga de una celda**



Fuente: Wikimedia Commons. *File: fig bat discharge charge3.*

<https://commons.wikimedia.org/w/index.php?curid=79179542>. Consulta: diciembre de 2019.

El electrolito es comúnmente una sustancia líquida o gelatinosa capaz de transportar iones entre las reacciones químicas que ocurren con los electrodos. También inhibe el flujo de electrones desde el cátodo hacia el ánodo directamente. Para que la corriente viaje con mayor facilidad en el circuito externo, también conocida como carga. El objetivo de que la corriente viaje a través de la carga: Realizar un trabajo útil.

2.3.1. Terminología

Una batería es un dispositivo que convierte energía química en energía eléctrica y viceversa. Para clasificarlas y elegir a opción correcta para alimentar un circuito es indispensable conocer los conceptos básicos que permitan el adecuado dimensionamiento dada una aplicación en específico.

2.3.1.1. Capacidad

Las baterías poseen distintas categorías para expresar la cantidad de energía que pueden almacenar. Cuando una batería está completamente cargada, la capacidad es la cantidad de potencia que posee. Generalmente se clasifican con la cantidad de corriente que pueden otorgar durante un lapso determinado. Por ejemplo, existen baterías de 1000mAH (miliamperios Hora), y 2000mAH.

2.3.1.2. Razón de descarga

También conocida como tarifa de descarga relativa, al describir una batería la corriente de descarga usualmente se expresa en radio-C y de esta manera normalizar contra distintas capacidades. Es una medida de la razón en la que la batería se descarga relativo a su máxima capacidad. Por ejemplo: 1C

significa que la corriente de consumo descarga la batería en 1 hora, Otro caso sería de 2C para representar el agotamiento de la batería en 30 minutos. Visto desde otra forma la carga se aumenta de manera proporcional al número de la razón de descarga, Aumentando la corriente y disminuyendo el tiempo.

2.3.1.3. Voltaje nominal de la celda

El voltaje promedio de una celda al estar cargada. El voltaje nominal de una batería depende de las reacciones químicas que ocurren dentro de ellas. Una batería con ácido de plomo usualmente se encuentra de 12 voltios y las baterías de botón de litio a 3 voltios, la palabra clave es nominal. En la realidad el voltaje medido en la batería decrece a medida que se descarga. Una batería de polímeros de Litio producirá alrededor de 4,23 voltios, mientras que descargada poseerá cerca de 2,7 voltios.

2.3.1.4. Recargable

Para que una batería sea recargada se requiere de otra fuente de alimentación, pero estas pueden ser cargadas y descargadas varias veces durante su vida útil. En general las baterías primarias o desechables tienen una menor tasa de descarga lo cual les permite durar más. Pero a largo plazo llegan a ser menos económicas que una batería recargable o secundaria.

2.3.1.5. Densidad de energía

Al combinar la capacidad con la forma y el tamaño de la batería se puede calcular la densidad de energía. Se refiere a la cantidad de potencia que la batería puede otorgar respecto a su volumen. Distintas tecnologías permiten

diversas densidades. Por ejemplo: en general una batería de Polímeros de Litio posee más energía que una batería alcalina del mismo tamaño.

2.3.1.6. Tarifa de descarga interna

Las baterías también se descargan al estar almacenadas sin uso alguno. La tarifa de descarga interna es la velocidad de autoconsumo dadas condiciones de fuga dentro del electrolito.

2.3.1.7. Seguridad

Básicamente las baterías son pequeños explosivos, para prevenir daños las baterías están diseñadas para ser lo más seguras posible. La mayoría de las baterías se fabrican para que se descarguen de manera paulatina y segura en caso de no utilizarse. Si una batería alcalina se conecta de manera errónea es posible sufrir sobrecalentamientos o fuga de electrolito. Las baterías de Litio poseen circuitos integrados embebidos o añadidos a la celda, como se muestra en la figura 52, para prevenir eventos inseguros o accidentales.

Figura 52. **Baterías de polímeros de litio**



Fuente: elaboración propia.

2.3.1.8. Polímeros de litio

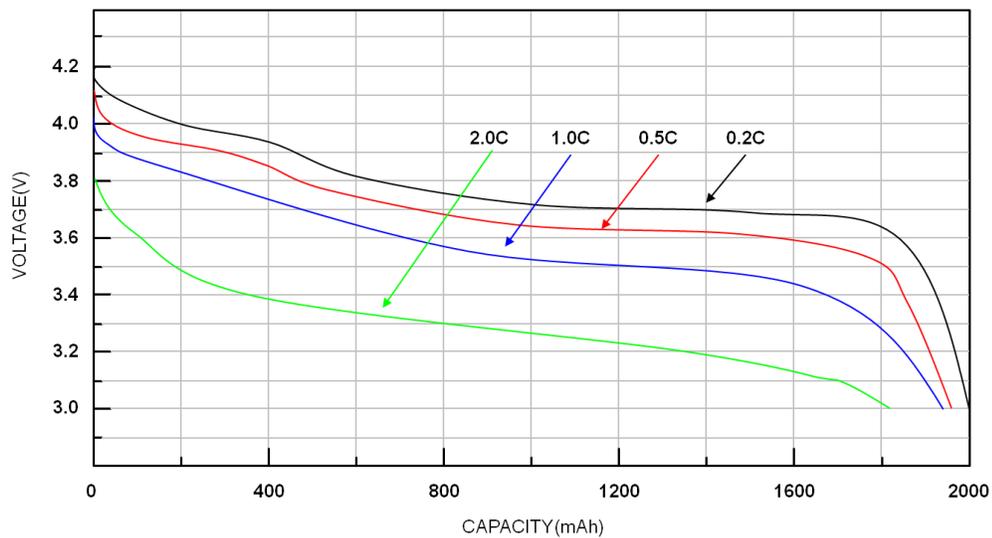
La tecnología basada en polímeros de Litio (usualmente abreviada como LiPo). es muy útil para la electrónica móvil. Ofrece la densidad de energía más alta en el mercado actual a precios competitivos. Convirtiéndolas en un nuevo estándar para teléfonos celulares. Sin embargo, necesitan ser recargadas de una manera especial. La elección de la capacidad de la batería dependerá del tiempo estimado de uso, restricciones de tamaño y otros factores.

2.3.2. Voltaje nominal

Una celda de batería de Litio posee un voltaje nominal de 3,7 V. Si se carga totalmente es posible hallarla cerca de los 4,3 V sin embargo decaerá

rápidamente a 3,7 V bajo uso normal. Cuando se agotan, la celda rodeara los 3 V. Una curva típica para distintos perfiles de descarga relativa se muestra en la figura 53.

Figura 53. **Curva típica de descarga normalizada**



Fuente: Richtek. *Li-ion battery and gauge introduction*.

<https://www.richtek.com/Design%20Support/Technical%20Document/AN024>. Consulta: diciembre de 2019.

2.3.3. Carga y descarga

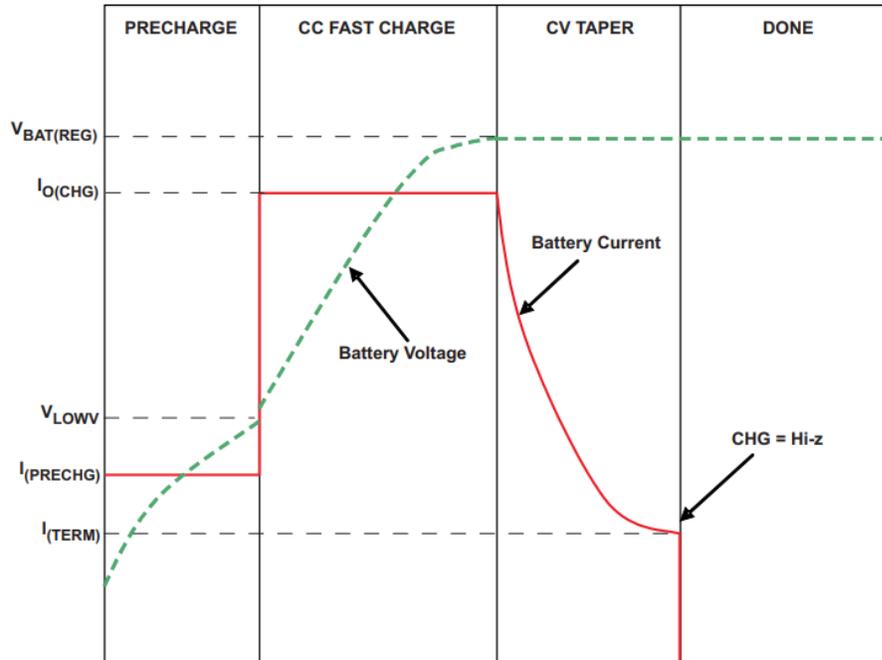
El ciclo de descarga de una batería LiPo depende principalmente del consumo y temperatura de la celda, otros factores como la edad de la batería también afectan el desempeño. En general se puede tener un consumo constante o dinámico, no existe un estándar o regla para descargar una batería, pues esto depende de la aplicación, es importante remarcar que las gráficas como la presentada en la figura 53 se realizó con una carga a corriente constante, independientemente del voltaje en la batería. Imposible de lograr

solamente con componentes pasivos, existe equipo especialmente diseñado para realizar este tipo de pruebas. El fabricante recomienda o añade circuitos de protección para que la batería no se agote hasta un punto muerto.

El proceso de carga para una batería LiPo está definida con un algoritmo de corriente y voltaje constante. Combinado ambos casos para recargar la celda de manera segura y garantizar la vida útil de la misma. El proceso es delicado y difícil de hacer manualmente, ingenieros desarrollaron soluciones integradas que operan todo el proceso de manera automática. Un algoritmo ejemplo se muestra en la figura 54. Es común que el circuito se encargue de un reacondicionamiento o precarga para luego recurrir a una inyección de corriente constante, al llegar a un nivel de voltaje de carga completo, la corriente disminuye paulatinamente. Todo el proceso esta optimizado para no lastimar o dañar la celda durante el ciclo de carga. Es común encontrarse con cargadores de para baterías de litio de 1 o 2 celdas.

Se propuso desarrollar una aplicación de baja potencia, por lo cual se ha elegido una batería de 1 celda. Las baterías de polímeros de litio son tan populares que muchas tarjetas de desarrollo poseen un cargador embebido en la placa para este tipo de acumuladores, como es el caso con la tarjeta Weemos Lolin 32 sugerida en este texto.

Figura 54. **Curva de carta para una batería LiPo**



Fuente: Texas Instrument. *BQ2407x Standalone 1-Cell 1.5-A linear battery charger with powerpath*. <http://www.ti.com/lit/ds/symlink/bq24075.pdf>. Consulta: diciembre de 2019.

2.3.4. **BQ27441-G1A**

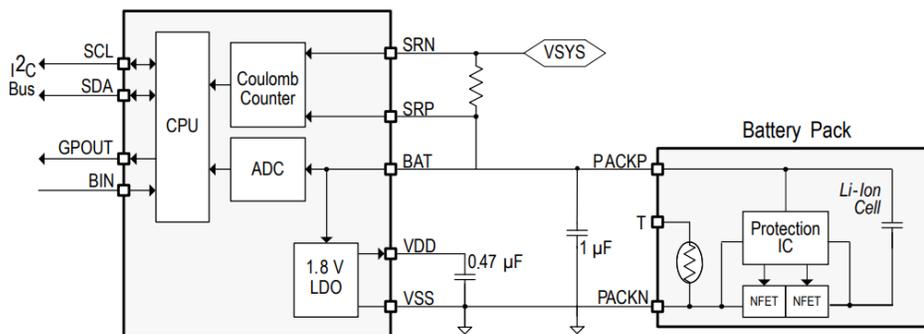
El circuito integrado BQ27441-G1A de Texas Instruments es un microcontrolador cuyos periféricos proveen medición del combustible para baterías de una sola celda con iones de Litio. El dispositivo requiere una configuración mínima por parte del usuario. Utiliza un algoritmo patentado de Rastreo de Impedancia™ para la medición de combustible y provee información como la cantidad disponible de carga (mAh), estado de carga (%), y voltaje de la batería (mV). Se adjunta el sistema de funcionamiento interno en la figura 55.

2.3.4.1. Descripción del circuito integrado

Medidor de combustible para baterías con iones de litio.

- Existe embebido en el sistema
- Soporta baterías removibles o embebidas
- Admite una Resistencia de sensado ultra baja (10 mili Ohm)
- Reporta la capacidad restante y el estado de carga (SOC) con filtro suavizante.
- Automáticamente se ajusta al envejecimiento de la batería, auto descarga y Temperatura.
- Interfaz serial I2C.
- Salida digital de advertencia para nivel bajo de batería.
- Reporte de temperatura con sensor interno o externo.

Figura 55. Diagrama simplificado a bloques del circuito integrado BQ27441-G1A



Fuente: Texas Instrument. *bq27441-G1 system-side impedance track™ fuel gauge.*

<https://www.ti.com/lit/ds/symlink/bq27441->

[g1.pdf?ts=1599801929584&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252F](https://www.ti.com/lit/ds/symlink/bq27441-g1.pdf?ts=1599801929584&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252F)

BQ27441-G1. Consulta: diciembre de 2019.

El conteo de Colombios se resume en una estimación del estado de carga (SOC), basado en corriente. Para determinar la capacidad restante, el conteo de colombios se fundamenta en la integración de la corriente absorbida y otorgada a la batería a través del tiempo, $\tau \in [t_0, t]$ dada la siguiente ecuación:

$$SOC(t) = SOC(t_0) - \int_{t_0}^t \frac{\eta(\tau)I(\tau)}{C_b^0} d\tau$$

Donde:

η = eficiencia durante el proceso de carga y descarga

I = corriente y el denominador es la capacidad nominal de la batería

El método posee ventajas en términos de simplicidad estructural y facilidad de implementación. Sin embargo, existen problemas que dificultan obtener el estado de carga de manera precisa. Porque la capacidad de la batería y eficiencia se distorsionan en base a la temperatura de operación y dirección de la corriente, estos factores deben considerarse para corregir la estimación del estado de carga. La precisión del sensor también juega un papel importante porque los errores acumulados afectarán desplazando el valor a calcular. La descripción del proceso entero excede el alcance del texto, además el circuito integrado se encargará de realizar todos los cálculos, correcciones y mediciones necesarias para proporcionar la información requerida.

2.3.4.2. Límites máximos absolutos

Sobrepasar los límites máximos absolutos puede ocasionar daño permanente al dispositivo.

Tabla VII. Límites máximos BQ27441-G1A

Parámetro	Símbolo	Mín.	Máx.	Unidades
Voltaje de alimentación	VCC	0,3	2	V
Voltaje de batería	VBAT	-0,3	6	V
Voltaje de entrada de las terminales	VIOD	-0,3	6	V
Temperatura ambiente de operación	TA	-65	150	°C

Fuente: elaboración propia.

2.3.4.3. Circuitos propuestos

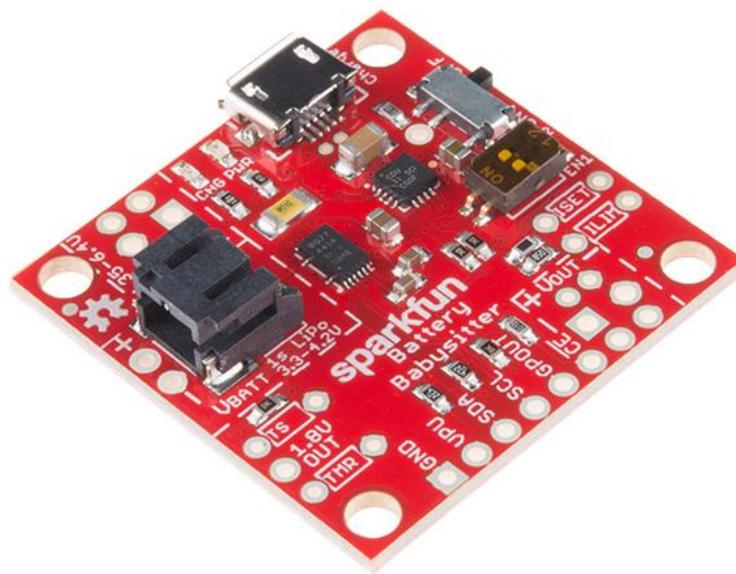
El circuito propuesto para la batería tiene la funcionalidad de monitorear y cargar la batería, contiene todo lo necesario para mantener los proyectos alimentados de forma segura y extensa. Ya que se está trabajando con una aplicación móvil, el método de alimentación seleccionado es el actual estándar en el mercado, baterías de Litio.

2.3.4.3.1. Tarjeta de pruebas

La tarjeta propuesta, figura 56, posee una pareja de circuitos de la marca Texas Instruments: El cargador de baterías BQ24075 y el medidor de combustible BQ27441-G1A, el cargador soporta corrientes hasta 1,5 A y el estándar USB de 500 mA o 100 mA. También presenta un gestor de alimentación, el cual garantiza alimentar la aplicación incluso si la batería se descarga, es decir redirecciona la alimentación en caso de fallas, el algoritmo

descrito requiere de otra fuente de energía, distinta a la batería. El circuito integrado BQ27441-G1A mide el voltaje de la batería para estimar el porcentaje de carga y la capacidad restante. Se detalla una breve descripción de las terminales en la figura 57.

Figura 56. **Gestor de baterías de litio**



Fuente: Sparkfun. *Gestor de baterías de litio.*

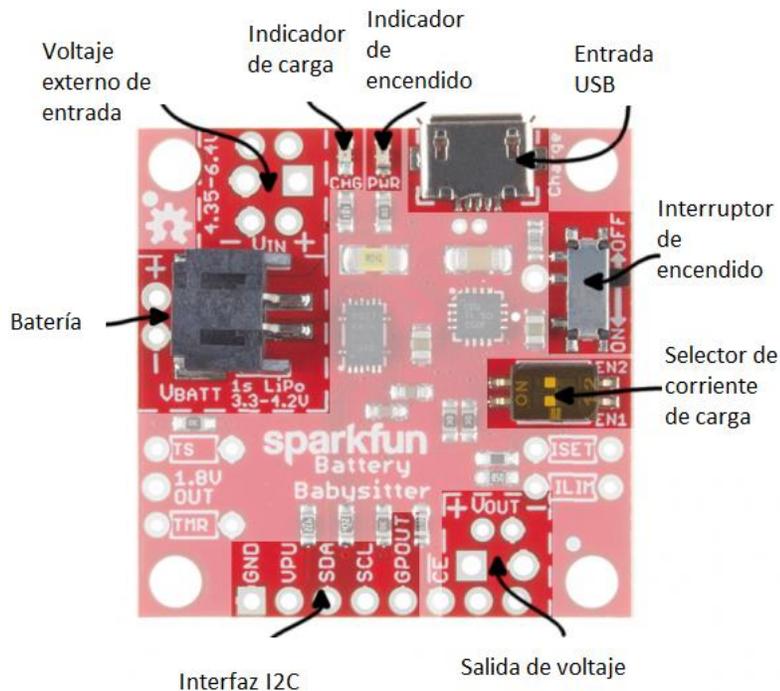
<https://cdn.sparkfun.com/assets/parts/1/1/3/3/1/13777-01.jpg>. Consulta: diciembre de 2019.

Características:

- Cargador y monitor de baterías LiPo combinados en una tarjeta
- Tarifas de carga seleccionables 100 mA, 500 mA o 1,5 A
- Puerto microUSB para carga u opciones externas
- Gestor dinámico de direccionamiento de alimentación
- Interfaz I2C con tolerancia a 5 V

- Medición de voltaje, porcentaje de carga y corriente

Figura 57. **Indicaciones del gestor de baterías**

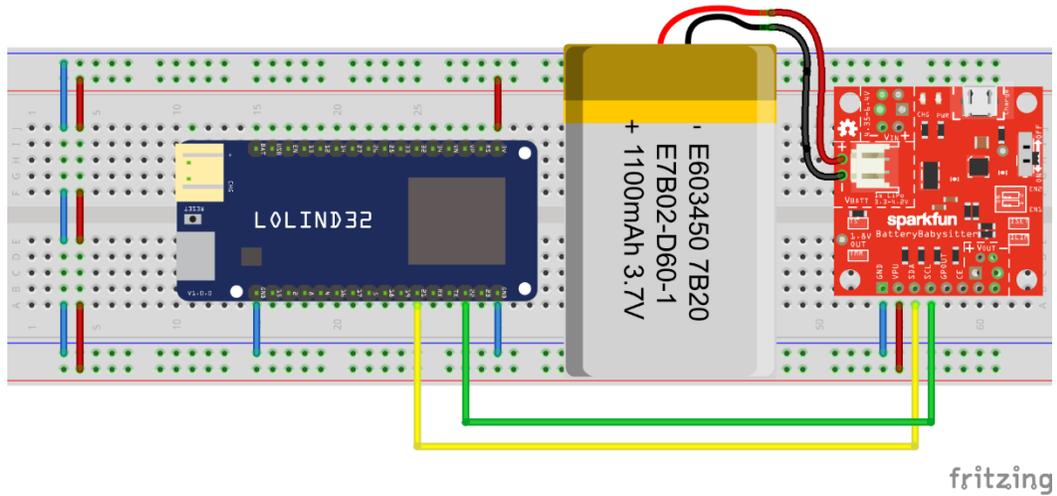


Fuente: Sparkfun. *Indicaciones del gestor de baterías*. https://cdn.sparkfun.com/r/600-600/assets/learn_tutorials/5/3/0/top-annotated-01.png. Consulta: diciembre de 2019.

2.3.4.3.2. Esquemático

La interfaz I2C proporciona los datos de manera digital, es necesario utilizar un controlador compatible. Dentro del repositorio proporcionado se encuentran ejemplos listos para cargar a la tarjeta de desarrollo weemos lolin 32. Las conexiones se presentan en la figura 58.

Figura 58. **Conexiones a la tarjeta weemos lolin 32**



Fuente: elaboración propia, empleando Fritzing 2019.

Tabla VIII. **Detalle de conexiones**

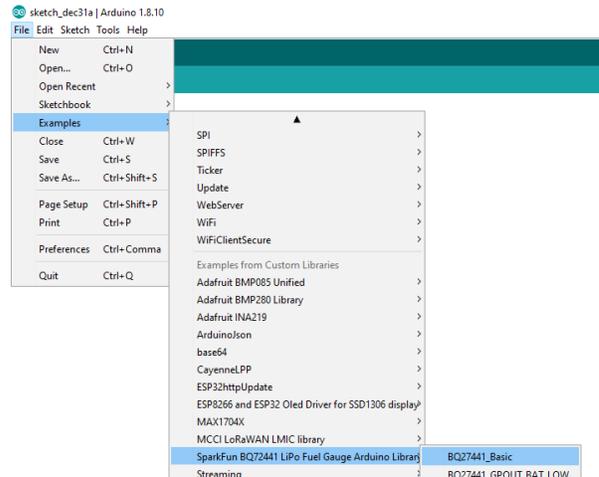
PINES	
WEEMOS	SPARKFUN
3,3V	VPU
GND	GND
SDA	SDA
SCL	SCL

Fuente: elaboración propia.

2.3.4.3.3. **Lectura de parámetros**

Para hallar el código correspondiente, abrir el IDE de Arduino como se presentó en la figura 12. Ubicar el ejemplo descrito en la figura 59. Se considera una buena práctica revisar conexiones y módulos individualmente antes de hacer el prototipo final.

Figura 59. **Ejemplo básico para el gestor de baterías**



Fuente: elaboración propia, empleando Arduino 1.8 2019.

La capacidad de la batería se mide en mAh, se necesita el dato para configurar el programa a cargar en la tarjeta, en este caso la batería posee una capacidad de 850 mAh, para asignar el valor, buscar la línea de código denotada en la figura 60 y actualizar el dato dependiendo de la batería en uso.

Figura 60. **Asignación de la capacidad de la batería**

```
// Set BATTERY_CAPACITY to the design capacity of your battery. |  
const unsigned int BATTERY_CAPACITY = 850; // e.g. 850mAh battery
```

Fuente: elaboración propia, empleando Arduino 1.8 2019.

Para finalizar hay que corroborar que la tarjeta esté conectada a la computadora y cargar el código repitiendo los pasos vistos en la figura 26 y 27. Una vez el programa entre en funcionamiento utilizar la herramienta del monitor serial y comparar las lecturas con la figura 61.

Figura 61. Parámetros de salida del gestor de baterías



```
ets Jun 8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:9720
ho 0 tail 12 room 4
load:0x40080400,len:6352
entry 0x400806b8
Connected to BQ27441!
90% | 4046 mV | -15 mA | 680 / 757 mAh | -61 mW | 91%
```

Fuente: elaboración propia, empleando Arduino 1.8 2019.

Dependiendo de la tarjeta de desarrollo o microcontrolador las lecturas iniciales de la figura 61 pueden variar, el procesador ESP32 posee información de fábrica que envía por el puerto serial cada vez que se enciende o sufre un reinicio.

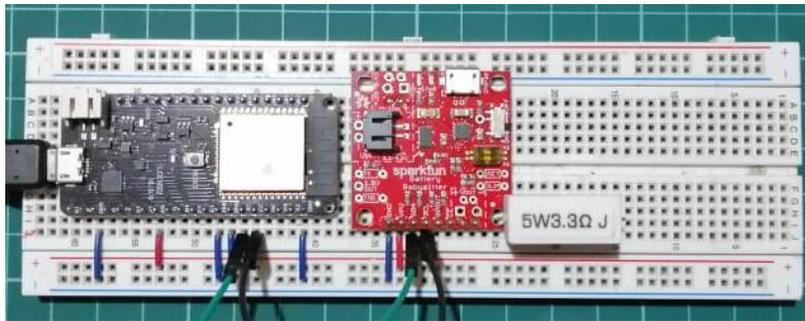
2.3.4.4. Monitoreo de descarga en una celda

Los datos extraídos de la batería se despliegan luego de que el programa corrobora el estado de comunicación. La última línea muestra la primera lectura de las propiedades de la batería, de izquierda a derecha en a figura 61: estado de carga, voltaje, corriente promedio, relación de la capacidad actual con la inicial, potencia y estado de salud. Se puede apreciar en la corriente y potencia un signo negativo, únicamente indica que la batería está siendo descargada. Y la potencia consumida. Un signo positivo denota la corriente otorgada a la batería durante el proceso de carga.

2.3.4.4.1. Curvas de resultados

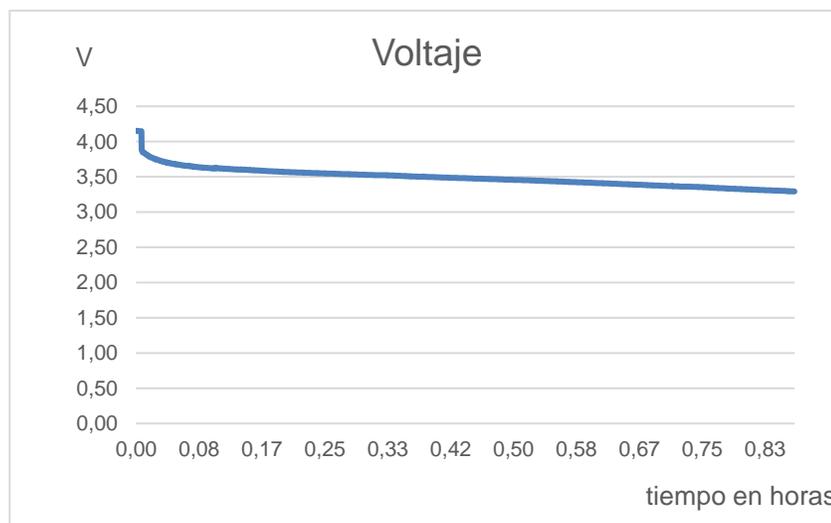
Se utilizará una carga resistiva para graficar la descarga de la batería de Litio conectado como se muestra en la figura 62.

Figura 62. Carga resistiva



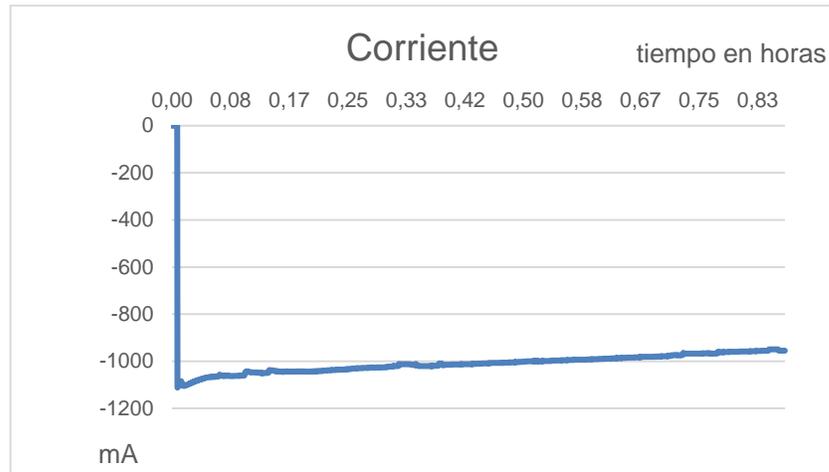
Fuente: elaboración propia.

Figura 63. Voltaje durante la descarga experimental de la celda



Fuente: elaboración propia.

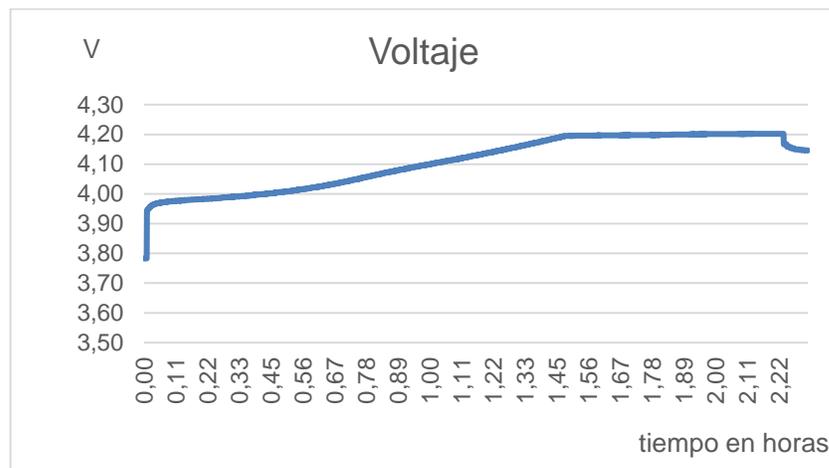
Figura 64. **Corriente durante la descarga experimental de la celda**



Fuente: elaboración propia.

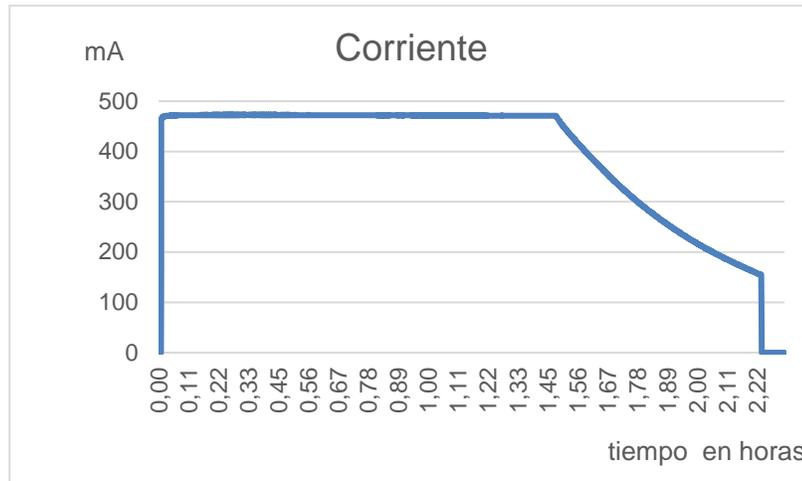
Además, se registra el proceso de carga para realizar una comparación teórica de la figura 54 y prácticas figuras 65 y 66.

Figura 65. **Voltaje durante la carga experimental de la celda**



Fuente: elaboración propia.

Figura 66. **Corriente durante la carga experimental de la celda**



Fuente: elaboración propia.

2.4. Módulo LoRa

LoRa es una técnica de modulación de desplazamiento de espectro desarrollada por una compañía llamada Semtech. Los dispositivos LoRa poseen tecnología inalámbrica de largo alcance y baja potencia. Está diseñado específicamente para redes de comunicación de máquina a máquina y el internet de las cosas. Abriendo camino al desarrollo de aplicaciones como: gestión de energía, control de contaminación, eficiencia estructural, prevención de desastres y un sinfín de necesidades próximas y futuras. La seguridad de transmisión es un factor importante puesto que puede utilizarse en redes públicas, privadas o mixtas. Ofrece mejor cobertura que las tecnologías celulares. La tecnología LoRa se integra fácilmente con la infraestructura existente.

2.4.1. Descripción

La interfaz de radio frecuencia LoRa define la señal que está siendo transmitida por un módulo o dispositivo correspondiente. Las propiedades para definir incluyen la modulación de la forma de onda, niveles de potencia admisibles y las bandas que pueden ser utilizadas en sus respectivos continentes, a la mano del protocolo de radio frecuencia y la interfaz de uso. El sistema de comunicación inalámbrico LoRa utiliza frecuencias libres de licencia disponibles alrededor de mundo, por ejemplo:

- Europa: 868 MHz
- Norteamérica: 915 MHz
- Asia: 433 MHz

Utilizar frecuencias o bandas menores a 2,4 GHz de uso industrial, científico y médico, permite brindar una mejor cobertura por parte de los dispositivos, especialmente si los nodos se encuentran dentro de los edificios.

2.4.1.1. Modulación

La modulación LoRa se basa en el desplazamiento de chirridos en frecuencia. Lo cual le otorga un buen desempeño en ambientes ruidosos o multicaminos, incluso en bajos niveles de potencia. El esquema controla pulsos lineales en frecuencia con banda ancha. El nivel de la frecuencia disminuye o aumenta a través del tiempo para codificar los datos a transmitir. Brindando una apariencia de transmisión pseudoaleatoria, otras tecnologías de recepción la detectan como ruido, dificultando la obtención de la información con métodos convencionales, aumentando la seguridad del enlace. Otra ventaja de la solución es su gran tolerancia a compensaciones de frecuencia, puesto que los

datos se agrupan en incrementos y decrementos, es permitido un ligero cambio en frecuencias centrales. Evitando incorporar los costosos osciladores de precisión estabilizados en amplios rangos de temperatura.

La comunicación entre diferentes dispositivos de uso final y puertas de enlace utilizan varios canales de frecuencias y velocidad de transmisión. La elección de la tasa de velocidad es un balance entre el rango de enlace y duración del mensaje. Este tipo de modulación permite que distintos enlaces de comunicación con diversas velocidades de enlace no interfieran entre sí. Un conjunto de vínculos virtuales puede ser creados para incrementar la capacidad de la puerta de enlace.

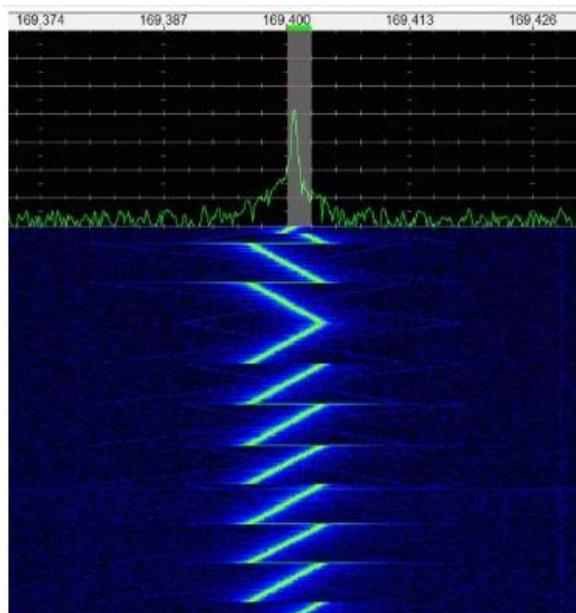
El hecho de que únicamente se usen bajas velocidades de enlace y en general poca información de transferencia, significa que los anchos de banda requeridos también son pequeños. Los principales son: 125 kHz, 250 kHz y 500 kHz. Además de los distintos anchos de banda, LoRa trabaja con 6 distintos factores de propagación designados desde SF7 hasta SF12 para adaptarse a las tasas de transmisión y alcance, con el fin de encontrar un equilibrio.

Factores de propagación elevados permiten más cobertura a expensas de menores velocidades de transmisión y viceversa. La combinación del ancho de banda con los factores de propagación puede elegirse de acuerdo con las condiciones del enlace y la cantidad de información a ser transmitida. Un factor de propagación elevado mejora el rendimiento para un ancho de banda dado, pero también aumenta el tiempo transmisión, variando desde los 18 bps hasta los 40 kbps

Cuando se procesa un mensaje, se obtiene ganancia de procesamiento gracias a la habilidad del modem para filtrar la rampa constante de la señal. Así es como se obtiene alta sensibilidad. En orden de acondicionar al módulo para recibir la señal, el transmisor emite el preámbulo mostrado en la figura 65.

Ambas gráficas representan la magnitud en función de la frecuencia, la gráfica superior es instantánea mientras que la inferior es un registro temporal, consultar la figura 67. Donde la magnitud se representa con una escala de colores, mayores magnitudes se simbolizan con tonos cálidos. El tiempo en el eje vertical y la frecuencia sobre el eje horizontal también conocidas como gráficas de cascada.

Figura 67. **Preámbulo de una señal LoRa**



Fuente: Link labs. *What is LoRa? a technical breakdown.* <https://www.link-labs.com/blog/what-is-lora>. Consulta: enero de 2020.

No existe selectividad de parte del receptor. Si hay una señal constante a la tasa y frecuencia correcta, el demodulador LoRa lo escuchará, este caso se define como enganche, sea dirigido al receptor o no. Luego, se finaliza el preámbulo con la misma señal, pero invertida. Convenientemente todos los detalles técnicos y teóricos han sido empaquetados en circuitos integrados de manera eficiente por la misma empresa que desarrollo la tecnología. Es tal la densidad implementada que el módulo completo entra en la categoría de sistema en un chip al igual que la ESP32.

2.4.2. Transceptor SX1276

Es un circuito integrado UHF de baja potencia capaz de establecer comunicación bidireccional, es posible configurarlo como emisor o receptor. Es una solución altamente integrada capaz de operar sobre un gran rango de frecuencias: 433 MHz, 868 MHz y 915 MHz. Arquitectura tan compacta minimiza la necesidad de incorporar elementos externos conservando flexibilidad. Los parámetros principales en radiofrecuencia son programables y la mayoría de ellos pueden ser manipulados dinámicamente. El transceptor SX1276 ofrece la ventaja de poseer ancho banda programable, habilitando intercambios entre modos de comunicación sin necesidad de cambiar componentes externos.

2.4.2.1. Descripción del circuito integrado

Aplicaciones:

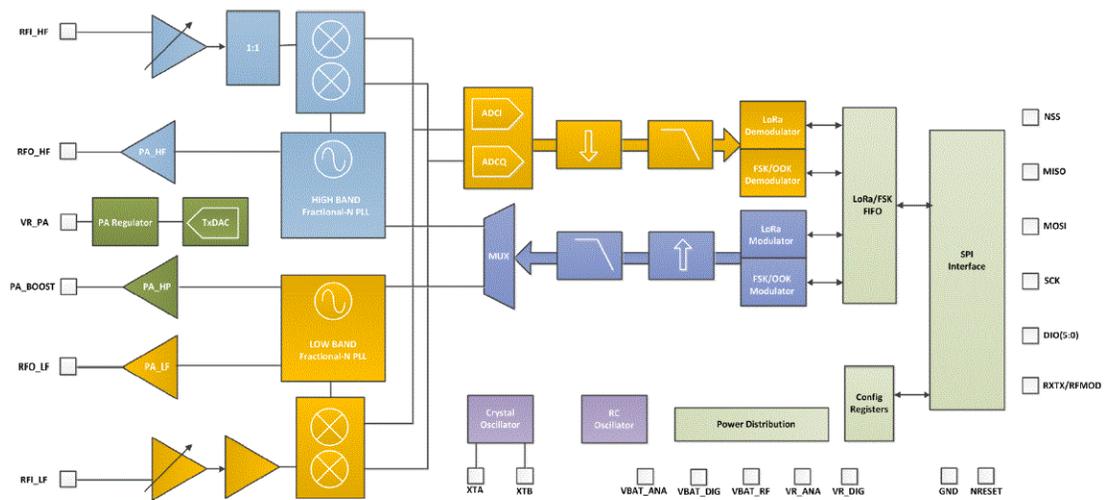
- Lectura automática de medidores
- Red de sensores inalámbricos
- Automatización domiciliar y de construcciones

- Alarmas y sistemas de seguridad inalámbrica
- Control y monitoreo industrial

Características:

- Alta sensibilidad: -120 dBm como mínimo a a 1,2 kbps
- Baja corriente: Rx 16 mA, 100 nA en retención de registro
- Potencia programable de salida, desde -18 dBm hasta +17 dBm en pasos d 1 dBm.
- Sintetizador de frecuencia totalmente integrado con resolución de 61 Hz.
- Sensor de temperatura y alarma de batería baja incorporado.
- Interfaz serial paralela.

Figura 68. Diagrama de bloques del integrado SX1276



Fuente: Semtech. *Diagrama de bloques del integrado SX1276.*

<https://www.semtech.com/uploads/products/sx1276-bd.gif>. Consulta: enero de 2020.

2.4.2.2. Límites máximos absolutos

Sobrepasar los límites máximos absolutos puede ocasionar daño permanente al dispositivo.

Tabla IX. Límites máximos absolutos SX1276

Parámetro	Símbolo	Mín.	Máx.	Unidades
Voltaje de alimentación	VDDmr	-0,5	3,9	V
Temperatura	Tj	-55	115	° C
Nivel de entrada RF	Pmr		6	dBm

Fuente: elaboración propia.

Exponer el chip a valores superiores o inferiores a los máximos y mínimos respectivamente, pueden dañarlo de forma permanente. Existen parámetros de operación recomendados para asegurar el correcto funcionamiento del sistema:

Tabla X. Parámetros de operación recomendados SX1276

Parámetro	Símbolo	Mín.	Máx.	Unidades
Voltaje de alimentación	VDDop	1,8	3,6	V
Temperatura	Top	-40	85	° C
nivel de entrada RF	ML		0	dBm
voltaje de entrada, 1 logico	VIH	0,8		VDD
voltaje de entrada, 0 logico	VIL		0,2	VDD
voltaje de salida, 1 logico	VOH	0,9		VDD
voltaje de salida, 0 logico	VOL		0,1	VDD

Fuente: elaboración propia.

2.4.2.3. Circuitos propuestos

Para hacer uso del circuito integrado es necesario acceder a los pines y realizar las conexiones mínimas de funcionamiento. La empresa HOPERF, un aliado de Semtech, se dedicó a la elaboración tarjetas con el chip incorporando circuitería extra para una implementación simplificada tanto en programación como fabricación. El módulo RFM95/96/97/98. En la figura 69 se muestran las tarjetas y su respectiva banda de frecuencia de trabajo, puede apreciarse que el chip de mayor tamaño es esencialmente el mismo, son los componentes extras que se arreglaron como filtro para limitar el rango de frecuencias.

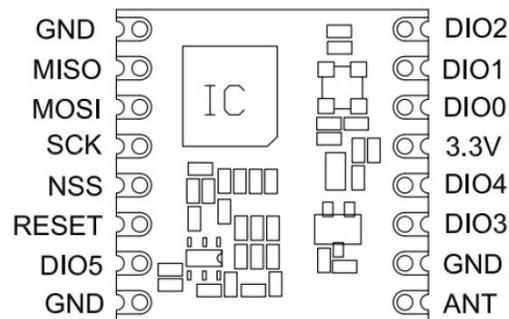
Figura 69. Módulos LoRa RFM96



Fuente: The things network. *Fitxer: Rfm95 vs rfm96.*

https://thethingsnetwork.cat/index.php/Fitxer:Rfm95_vs_rfm96.jpg. Consulta: enero de 2020.

Figura 70. **Identificación de conexiones del módulo RFM96**



Fuente: Microchip. *Low power long range transceiver module.*

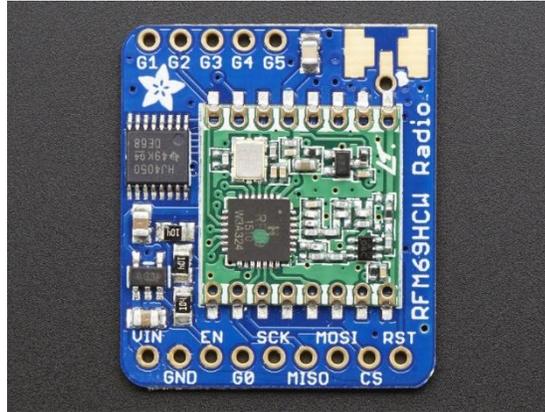
<http://ww1.microchip.com/downloads/en/DeviceDoc/RN2483-Low-Power-Long-Range-LoRa-Technology-Transceiver-Module-Data-Sheet-DS50002346D.pdf>. Consulta: enero de 2020.

La tarjeta modular RFM96 permite el acceso a funciones como alimentación, comunicación y antena. El protocolo de comunicación es SPI. Sin embargo, el tamaño es muy reducido y carece de protecciones para trabajarlo en una tarjeta de prototipos o protoboard. Es común hallar estos módulos embebidos en tarjetas que acomodan, señalan y protegen el patillaje del sistema, empresas como Sparkfun, Adafruit y Seeedstudio, La tarjeta de pruebas es una manera aún más cómoda para realizar pruebas de manera segura.

2.4.2.4. Tarjeta de pruebas

El mercado ofrece de manera conveniente las tarjetas de prueba e identificación de patillaje.

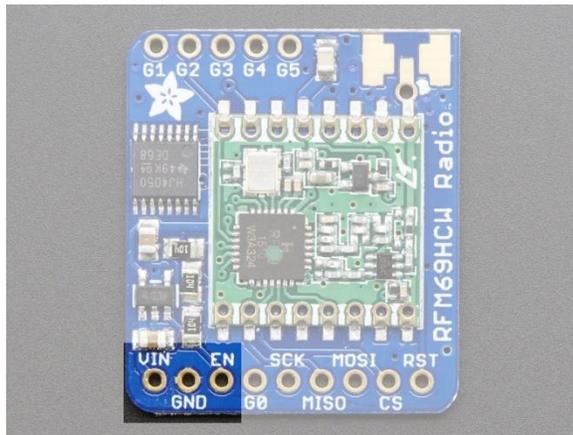
Figura 71. **Módulo LoRa de Adafruit**



Fuente: Adafruit. *Adafruit RFM69HCW and RFM9X LoRa*.
<https://learn.adafruit.com/assets/31722>. Consulta: enero de 2020.

El punto verde sobre el integrado en la figura 71 significa que la frecuencia central de trabajo es de 915 MHz.

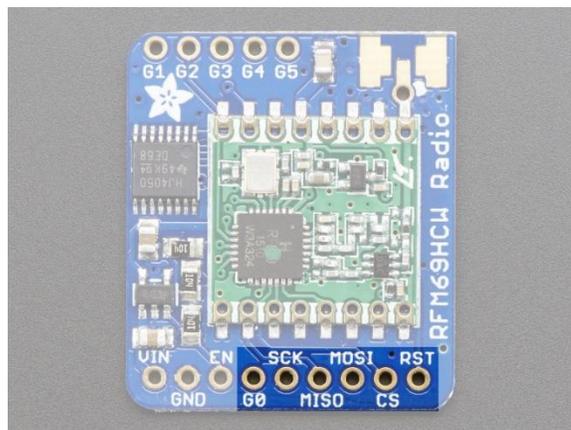
Figura 72. **Puerto de alimentación del módulo LoRa de Adafruit**



Fuente: Adafruit. *Adafruit RFM69HCW and RFM9X LoRa*.
<https://learn.adafruit.com/assets/31722>. Consulta: enero de 2020.

Los puertos de alimentación sirven para administrar la fuente, admite desde 3,3 hasta 6 VDC. Como mínimo se espera una fuente capaz de otorgar 150 mA, puesto a que los picos de corriente pueden llegar a existir durante el proceso de transmisión. El rango de entrada de voltaje es compatible para sistemas de 3,3 y 5VDC. Los más populares y estándar en el mercado. Los sistemas de 3,3VDC han ganado auge con el desarrollo de la tecnología. Referirse a la figura 72.

Figura 73. **Puerto SPI del módulo LoRa de Adafruit**



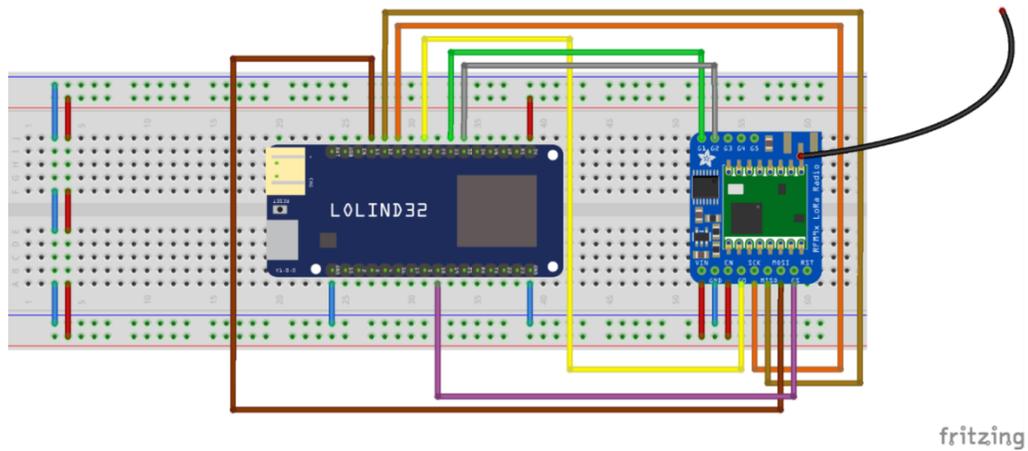
Fuente: Adafruit. *Adafruit RFM69HCW and RFM9X LoRa*.
<https://learn.adafruit.com/assets/31722>. Consulta: enero de 2020.

El protocolo de comunicación se encargará de transmitir la información del módulo hacia el microcontrolador sugerido, El fabricante propone librerías compatibles para el sistema, y se brindaron dentro del repositorio de trabajo.

2.4.2.4.1. Diagrama de conexión

Otra manera de presentar las conexiones entre módulos es utilizar una captura de galleta de prototipos.

Figura 74. **Conexión del módulo RFM69**



Fuente: elaboración propia, empleando Fritzing 2019.

Tabla XI. **Detalle de conexiones**

PINES	
WEEMO S	LOR A
26	G0
33	G1
32	G2
14	SCK
12	MISO
13	MOSI
18	CS

Fuente: elaboración propia.

2.4.2.4.2. Lectura de parámetros

Para comprobar la comunicación digital del controlador hacia el módulo RFM96, basta con armar el circuito de la figura 74. Para verificar la conexión inalámbrica es necesario realizar 2 circuitos iguales. Un programa será configurado como emisor y el otro como receptor. Dentro del repositorio existen 2 ejemplos simples, `rf95_serverTESIS.ino` y `rf95_clientTESIS.ino` basados en los ejemplos originales de la librería RadioHead, un paquete conformado por diversos radios para microprocesadores embebidos. Provee librerías orientadas a objetos para enviar y recibir mensajes empaquetados. La versión utilizada en este documento es la 1.98. versiones más actuales pueden ser referidas a la página <http://www.airspayce.com/mikem/arduino/RadioHead/>.

Es necesario readaptar algunos parámetros para que coincidan con la tarjeta de desarrollo. En el ejemplo `rf95_clientTESIS.ino` hay 3 objetos de radio como se muestra en la figura 75 en las líneas 15, 14 y 16. El primero se utiliza en controladores genéricos como el Arduino UNO, se modificó la línea 14 donde el 18 denota el pin de selección del chip y el 26 el pin de interrupción 0 también llamado G0. Además, se elevó la velocidad de transmisión hacia la computadora a 115200 en la línea 29. Tomarlo en cuenta para elegir correctamente los baudios en la terminal serial.

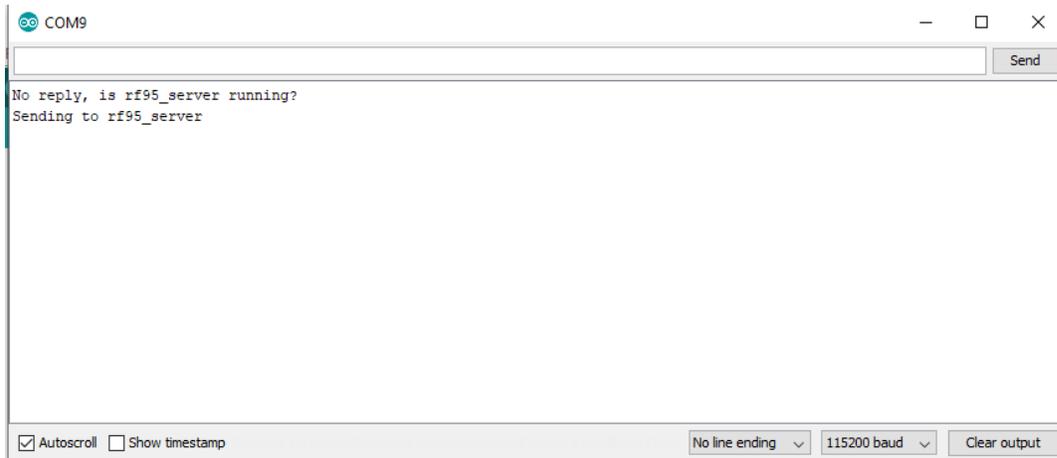
Figura 75. Correcciones del ejemplo brindado en el paquete RadioHead

```
14 // Singleton instance of the radio driver
15 //RH_RF95 rf95;
16 RH_RF95 rf95(18, 26); // Rocket Scream Mini Ultra Pro with the RFM95W
17 //RH_RF95 rf95(8, 3); // Adafruit Feather M0 with RFM95
18
19 // Need this on Arduino Zero with SerialUSB port (eg RocketScream Mini Ultra Pro)
20 // #define Serial SerialUSB
21
22 void setup()
23 {
24   // Rocket Scream Mini Ultra Pro with the RFM95W only:
25   // Ensure serial flash is not interfering with radio communication on SPI bus
26   // pinMode(4, OUTPUT);
27   // digitalWrite(4, HIGH);
28
29   Serial.begin(115200);
30   while (!Serial) ; // Wait for serial port to be available
```

Fuente: elaboración propia, empleando módulo RFM95.

Al cargar el código en la tarjeta, esperar un momento para que el procesador ESP32 reconozca el módulo RFM95, como puede observarse en la figura 75, de lo contrario revisar que las conexiones estén correctamente elaboradas. El código completo envía una pequeña cadena de caracteres y espera respuesta de manera inalámbrica. Los datos extraídos del enlace son retransmitidos hacia el puerto UART, repitiendo el proceso de manera cíclica. En contraste se tiene el código rf95_serverTESIS.ino sumamente similar, la diferencia radica en el orden de la lógica. Empieza por esperar un mensaje para luego enviar respuesta. Para que el enlace funcione la tarjeta 1 posee el código de servidor y la tarjeta 2 el código de cliente, de lo contrario no existe coordinación entre algoritmos.

Figura 76. Prueba de conexión hacia el módulo RFM95



Fuente: elaboración propia, empleando módulo RFM95.

Luego de cargar el código correspondiente a la otra tarjeta, comparar la figura 76 con la figura 77, se aprecia que, en la segunda, el enlace se realizó con éxito, es importante corroborar cada módulo por separado para identificar de manera puntual un posible fallo. Armar la solución completa sin corroborar de manera secuencial es una mala práctica.

Figura 77. Prueba de enlace inalámbrico con el módulo RFM95



Fuente: elaboración propia, empleando módulo RFM95.

Al momento, el texto se ha desarrollado en torno a la presentación básica, verificación de compatibilidad y funcionamiento modular, la flexibilidad que se presenta es de interés para implementar a diversos panoramas, entornos u aplicaciones.

3. NODO

3.1. Descripción y aplicaciones

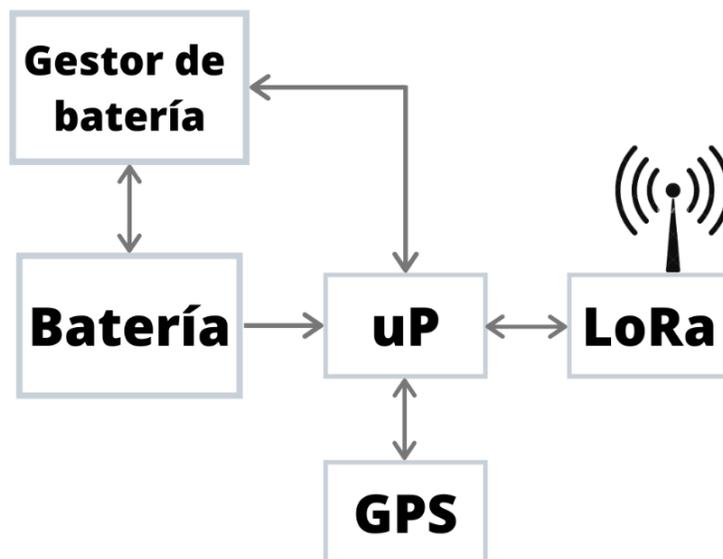
El concepto de nodo existe en diversas áreas, se adopta primero una definición más familiar: como el nodo que puede encontrarse en un diagrama eléctrico, un punto de conexión entre 2 o más componentes. Un esquemático brinda información gráfica general o específica del funcionamiento de este. En telecomunicaciones: la red que une las computadoras, routers, servidores y demás, también es representada gráficamente con una topología de red. Un nodo es definido dependiendo del ambiente de trabajo, comúnmente conocido como capa. Puesto a que se está elaborando un dispositivo se puede concluir que la definición recae en la capa física.

El nodo es un punto final, un dispositivo capaz de albergar una dirección única que lo diferencie dentro de una red. El cual puede generar información de utilidad para transferir a través de algún canal presente en la red. El nodo en cuestión tendrá la funcionalidad de obtener su ubicación geográfica utilizando tecnología satelital y brindarla en un formato conveniente para transmitirla por el sistema inalámbrico LoRa. Datos de interés para conclusiones posteriores.

3.2. Diagrama de bloques

Para describir la tarea del dispositivo a grandes rasgos y describir el flujo de funcionamiento.

Figura 78. Bloques fundamentales del nodo



Fuente: elaboración propia, empleando Canva.

Comúnmente un nodo carece de comunicación Wifi, pues representa una carga muy pesada para una batería. La información es relativamente sencilla de codificar, no se trata de una imagen o video, además el canal LoRa demoraría mucho tiempo en transmitir tantos datos. Los nodos están diseñados para utilizar la menor cantidad de energía posible.

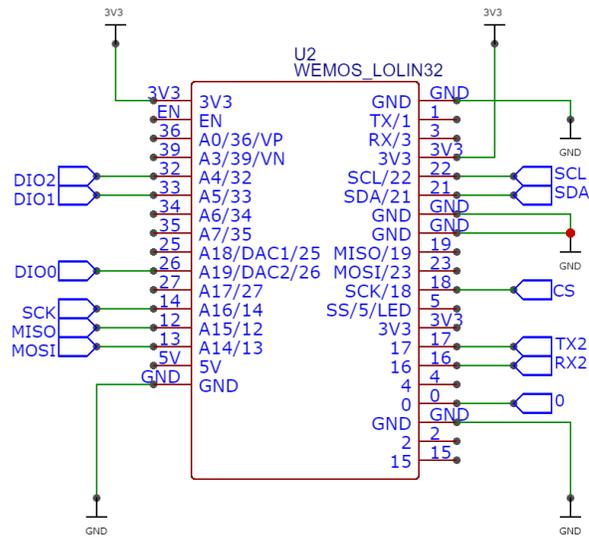
Como se muestra en la figura 78, la información a transmitir es obtenida mediante el módulo GPS. La corriente de consumo se adquiere con el gestor de

baterías para realizar los cálculos de potencia de transmisión el cual monitorea el consumo y estado de la aplicación en tiempo real, la batería alimentará el circuito, el módulo LoRa proporciona el enlace inalámbrico. El procesador se encarga de coordinar las tareas de cada dispositivo de manera secuencial y ordenada. El protocolo de comunicación dependerá de la naturaleza del módulo.

3.3. Esquemático

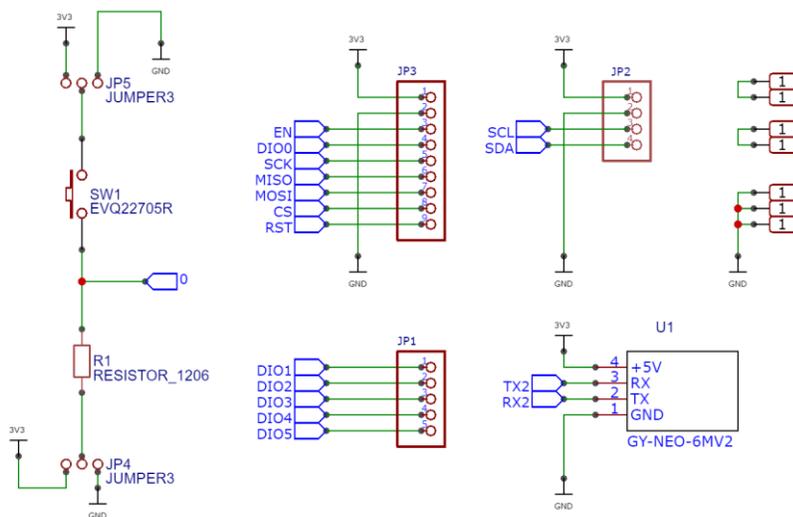
Al realizar un prototipo formal es común emplear las tarjetas de circuitos impresos, puesto a que reducen la probabilidad de falsos contactos dada la sujeción que se presenta en cada componente. El software para realizar el esquemático y circuito impreso también es gratuito, con la posibilidad de ser utilizado en línea. En resumen, la tarjeta será para acomodar los módulos y garantizar el funcionamiento en campo. Se puede apreciar en la figura 79 todos los periféricos de comunicación seriales utilizados; por ejemplo, SCK, MISO y MOSI, pertenecen al protocolo SPI, las terminales SCL y SDA al protocolo IIC, por último: las terminales RX2 y TX2 al protocolo UART.

Figura 79. **Conexiones al módulo ESP32**



Fuente: elaboración propia, empleando easyEDA.

Figura 80. **Circuitos auxiliares de la tarjeta**



Fuente: elaboración propia, empleando easyEDA.

En la izquierda de la figura 80 puede observarse una entrada digital. El pulsador puede ser utilizado con lógica positiva o negativa, la primera le asigna un número 1 a la presencia de voltaje y un 0 a la ausencia de este. La lógica negativa contradice lo anterior asignándole un número 0 a la presencia de voltaje y 1 a su ausencia. Se adoptó el circuito con fines experimentales, sin embargo, una de las opciones más populares al trabajar con entradas digitales es la lógica negativa. Para configurarlo adecuadamente, el jumper 3 se asigna a tierra y el jumper 4 se conecta al voltaje de alimentación.

3.4. Prototipo

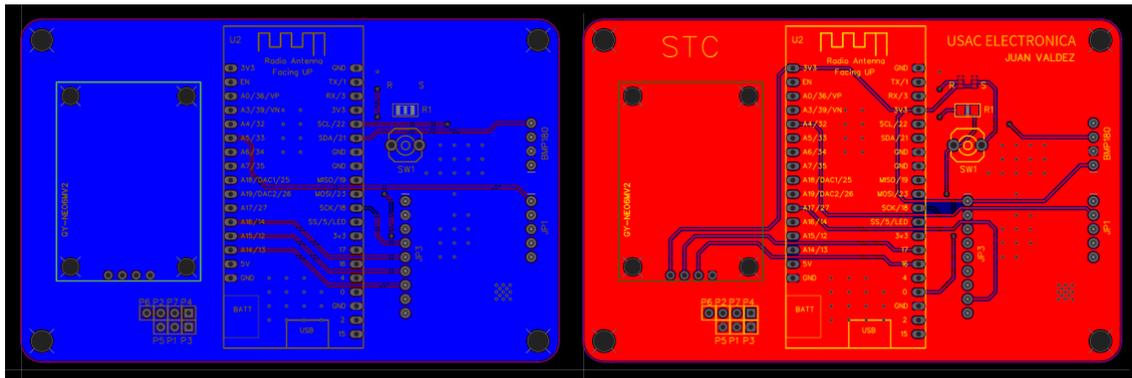
Desarrollar tarjetas y hacerlas en una casa de fabricación tiene ventajas respecto a hacerlas artesanalmente. La inversión inicial de los materiales y habilidad manual pueden ser un cuello de botella, puesto que todo el proceso tiene etapas que requieren cuidado. Existe riesgo del mal manejo de herramienta y dependiendo del método de revelado; los químicos y gases pueden ser delicados. El tiempo que se dedica a trabajo manual puede ser destinado a desarrollo y depuración de errores.

3.4.1. Circuito impreso

Otra ventaja de hacer circuitos impresos es la resolución y capacidad de hacer tarjetas de doble cara, que ahorra espacio y facilita el proceso de ruteo. En la figura 80, se presenta de lado derecho la capa superior y de lado izquierdo la capa inferior. Por defecto se realizan los diseños vistos desde enfrente. Lo último puede representar un error común, la impresión de la capa inferior en modo espejo tener esto en cuenta si se desea desarrollar la tarjeta de forma manual.

Se presenta en la figura 81 el circuito listo para soldar los componentes, observar a lado izquierdo de la misma y compararla con la figura 81, se nota de manera invertida.

Figura 81. Capas de cobre



Fuente: elaboración propia, empleando easyEDA.

Por conveniencia y presentación compacta, se eligió importar la tarjeta, con un tiempo de espera de 3 semanas e inversión de aproximadamente 250 Quetzales por 10 piezas. Es importante aclarar que comprar la placa no es un proceso indispensable pues el estudiante es capaz de utilizar otras técnicas para su manufactura. La figura 83 muestra el nodo finalizado, listo para programar.

Figura 82. Tarjeta del circuito impreso



Fuente: elaboración propia, empleando easyEDA.

Figura 83. Circuito finalizado



Fuente: elaboración propia, empleando easyEDA.

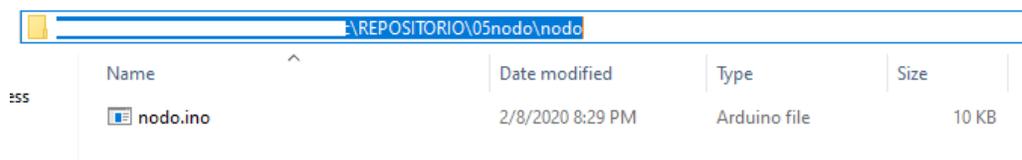
3.4.2. Programación

Anteriormente se hicieron pruebas de funcionamiento por módulo: procesador, GPS y LoRa. De ser posible, es una buena práctica realizar el circuito completo en una galleta de prototipos, al terminar de ensamblar una tarjeta se recomienda realizar pruebas de continuidad; especialmente en las terminales de alimentación.

El código utiliza numerosas librerías auxiliares, existen posiciones específicas dentro del programa donde es posible ingresar código por el usuario sin interferir con los procesos propios de las mismas. El repositorio de trabajo posee por defecto todas las configuraciones utilizadas durante el desarrollo del dispositivo.

Para hallar el código correspondiente, abrir el IDE de Arduino como se presentó en la figura 12. Ubicar el ejemplo descrito en la figura 84.

Figura 84. Ruta del archivo ejemplo



The image shows a file explorer window with the address bar containing the path `\\REPOSITORIO\05nodo\nodo`. Below the address bar is a table with the following columns: Name, Date modified, Type, and Size. The table contains one entry: `nodo.ino`, which is an Arduino file of 10 KB, last modified on 2/8/2020 at 8:29 PM.

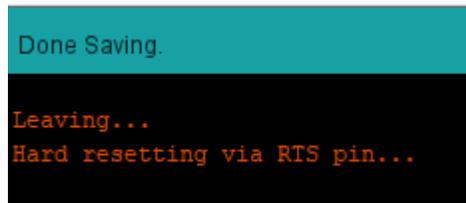
Name	Date modified	Type	Size
nodo.ino	2/8/2020 8:29 PM	Arduino file	10 KB

Fuente: elaboración propia, empleando Arduino 1.8 2019.

Todas las configuraciones y periféricos están configurados para la tarjeta de trabajo indicada en la figura 19. En resumen, verificar las conexiones, compilar el código, revisar el puerto serial al que se está conectados y cargar el

código a la tarjeta. Si se siguieron las indicaciones notará el mensaje de la figura 85.

Figura 85. **Mensaje de éxito en la consola**



```
Done Saving.  
Leaving...  
Hard resetting via RTS pin...
```

Fuente: elaboración propia, empleando Arduino 1.8 2019.

Ahora bien, al verificar que el código funciona se examinan los segmentos que fueron modificados. Cuando el desarrollador utiliza librerías de terceros, no acostumbra a indagar a detalle la estructura y composición de este a menos que sea necesario optimizarlo, además, ya se hicieron pruebas individuales de los módulos. Se debe tener especial cuidado al trabajar con varios protocolos de comunicación puesto que; si no se conoce la estructura de los periféricos existe el riesgo de mezclarlos.

Si el compilador falla o existe algún mensaje inesperado, este es el punto adecuado para verificar paso a paso lo previamente expuesto. En la figura 86 se presentan las librerías a utilizar. Lmic.h y hal.h se encuentran fuera del alcance de este texto, pero serán de utilidad para visualizar de una manera más cómoda.

SPI.h se utiliza para comunicarse con el módulo RFM96. CayenneLPP.h optimiza y le da formato a una cadena de caracteres para reducir los símbolos transmitidos por el canal LoRa. Recuerde que a mayores distancias de

transmisión el tiempo de emisión aumenta, por ese motivo es preferible disminuir lo más posible el uso de caracteres en la transmisión. Wire.h, Adafruit_Sensor.h y Adafruit_BMP085_U.h son opcionales, en este caso también se utiliza un sensor de presión esto servirá de ejemplo porque: la idea de desarrollar el dispositivo es brindarle al estudiante una herramienta para desplegar su propia red. Es altamente probable que los requerimientos cambien de acuerdo con sus necesidades. Si el estudiante carece del dispositivo solo hará falta comentar las secciones que utilizan el sensor. TinyGPS++.h será de gran utilidad para obtener las cadenas provenientes del módulo GPS y obtener las coordenadas de geolocalización de una manera más cómoda.

Figura 86. **Librerías para utilizar en el nodo**

```
29 |
30 | // Include the arduino-lmic library:
31 | #include <lmic.h>
32 | #include <hal/hal.h>
33 | #include <SPI.h>
34 | #include <CayenneLPP.h>
35 |
36 | #include <Wire.h>
37 | #include <Adafruit_Sensor.h>
38 | #include <Adafruit_BMP085_U.h>
39 |
40 | #include <TinyGPS++.h>
41 |
42 |
```

Fuente: elaboración propia, empleando Arduino 1.8 2019.

Las variables extra utilizadas por los sensores junto al patillaje hacia el módulo LoRa se presentan en la figura 87. El patillaje tiene que coincidir con las conexiones físicas, estas pueden ser consultadas a detalle en la tabla XI.

Figura 87. **Variables y definiciones de conexión**

```
67 float temperature;
68 float presion;
69 float latitud;
70 float longitud;
71 float altura;
72
73 static osjob_t sendjob;
74
75 // Pin mapping for the SparkX ESP32 LoRa 1-CH Gateway
76 const lmic_pinmap lmic_pins = {
77   .nss = 18,
78   .rxtx = LMIC_UNUSED_PIN,
79   .rst = LMIC_UNUSED_PIN,
80   .dio = {26, 33, 32},
81 };
^^
```

Fuente: elaboración propia, empleando Arduino 1.8 2019.

El módulo GPS requiere de un puerto UART dedicado, la línea 202 de la figura 88 indica el uso del puerto 2 a una velocidad de 9600 baudios y sus respectivas terminales de comunicación. El primer puerto serial se encuentra reservado por defecto para la comunicación hacia la computadora.

Figura 88. **Puertos de comunicación UART**

```
200 void setup() {
201   Serial.begin(115200);
202   Serial2.begin(9600, SERIAL_8N1, 16, 17);
```

Fuente: elaboración propia, empleando Arduino 1.8 2019.

Muchas veces, es de utilidad enviar mensajes de alerta y evitar que la aplicación entre en funcionamiento hasta solventar el inconveniente. La figura 89 presenta dicho ejemplo.

Figura 89. **Mensaje de alerta para el módulo de presión**

```
256 Serial.println("Pressure Sensor Test"); Serial.println("");
257
258 /* Initialise the sensor */
259 if(!bmp.begin())
260 {
261   /* There was a problem detecting the BMP085 ... check your connections */
262   Serial.print("Ooops, no BMP085 detected ... Check your wiring or I2C ADDR!");
263   while(1);
264 }
265
266 /* Display some basic information on this sensor */
267 displaySensorDetails();
268
269 }
270
```

Fuente: elaboración propia, empleando Arduino 1.8 2019.

A continuación, se obtiene la información del GPS, figura 90, sin necesidad de descifrar las cadenas NMEA para redireccionar la información por el canal LoRa. Cuando el desarrollador elige una librería, el principal criterio que se necesita tomar en cuenta es la documentación de esta.

Figura 90. **Lectura de posición satelital**

```
299     smartDelay(1000);
300     if (millis() > 5000 && gps.charsProcessed() < 10)
301         Serial.println(F("No GPS data received: check wiring"));
302
303     latitud = gps.location.lat();
304     longitud = gps.location.lng();
305     altura = gps.altitude.meters();
306
307     Serial.println(latitud,5);
308     Serial.println(longitud,5);
309     Serial.println(altura);
310
311     lpp.reset();
312     lpp.addGPS(1, latitud, longitud, altura);
313     lpp.addTemperature(2, temperature);
314     lpp.addBarometricPressure(3,presion);
315
316     do_send(&sendjob);
317 }
```

Fuente: elaboración propia, empleando Arduino 1.8 2019.

3.5. Pruebas de funcionamiento

Podrá apreciar entonces, que el progreso del dispositivo se basa en la constante depuración de cada etapa. La consola del puerto UART o monitor serial presentado en la figura 43 contiene ahora el texto de la figura 91 al reiniciar o encender el dispositivo. Cabe aclarar, no es común que una aplicación inalámbrica posea comunicación directa hacia la computadora (cable USB). A menos que se requiera una constante actualización de firmware. La tecnología LoRa no es un canal adecuado para realizar este tipo de intercambios.

Figura 91. Salida del puerto serie del nodo

```
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:9720
ho 0 tail 12 room 4
load:0x40080400,len:6352
entry 0x400806b8
RXMODE_RSSI
3284: engineUpdate, opmode=0x808
3309: Unknown event
3313: engineUpdate, opmode=0x888
3370: TXMODE, freq=903900000, len=13, SF=7, BW=125, CR=4/5, IH=0
Packet queued
Pressure Sensor Test

-----
Sensor:      BMP085
Driver Ver:  1
Unique ID:   10085
Max Value:   300.00 hPa
Min Value:   1100.00 hPa
Resolution:  0.01 hPa
-----

98463: setupRx1 txrxFlags 00 --> 01
start single rx: now-rxtime: 3
98594: RXMODE_SINGLE, freq=923300000, SF=7, BW=500, CR=4/5, IH=0
rxtimeout: entry: 98678 rxtime: 98588 entry-rxtime: 90 now-entry: 4 rxtime-txend: 62524
161323: setupRx2 txrxFlags 0x1 --> 02
start single rx: now-rxtime: 3
161454: RXMODE_SINGLE, freq=923300000, SF=9, BW=125, CR=4/5, IH=0
rxtimeout: entry: 162751 rxtime: 161448 entry-rxtime: 1303 now-entry: 3 rxtime-txend: 125384
162767: processRx2DnData txrxFlags 0x2 --> 00
216597: processDnData txrxFlags 00 --> 20
216600: EV_TXCOMPLETE (includes waiting for RX windows)
```

Fuente: elaboración propia, empleando Arduino 1.8 2019.

4. PUERTA DE ENLACE

4.1. Descripción y aplicaciones

Las puertas de enlace son una parada clave para los datos que se dirigen hacia otras redes, a diferencia de un nodo, estas pueden direccionar la información de acuerdo con especificaciones del usuario. El entorno de instalación es más estático y demanda mayor potencia, es posible que el equipo disponga de varios canales de comunicación. Como mínimo una puerta de enlace debe tener 2 distintos módulos de transmisión inalámbrica y al menos una cableada. La cableada puede ser conexión ethernet o USB y en este caso particular las inalámbricas: Wifi y LoRa.

Las puertas de enlace son un puente entre los dispositivos que se comunican utilizando LoRa y otra red, privada o pública (internet). Generalmente la información se guía hacia una base de datos para poder acceder a ella desde internet. Por esta razón la puerta de enlace también posee conexión Wifi o Ethernet.

Figura 92. Puerta de enlace



Fuente: The things network. *The things gateway*.

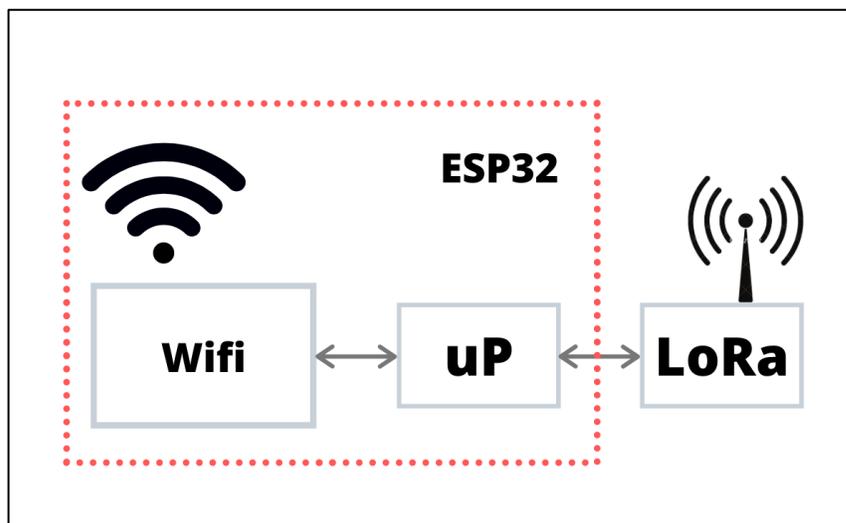
<https://www.thethingsnetwork.org/docs/gateways/gateway/>. Consulta: febrero de 2020.

Por supuesto existen puertas de enlace listas para instalar, en la figura 92 se presenta una de las opciones más comunes, diseñada para interiores y con un precio de aproximadamente Q 3 000,00. Este capítulo se enfoca en los requerimientos físicos mínimos que requiere uno de estos dispositivos para explotar las capacidades del equipo que se tiene a disposición.

4.2. Diagrama de bloques

Para describir la tarea del dispositivo a grandes rasgos y describir el flujo de funcionamiento.

Figura 93. **Bloques fundamentales de la puerta de enlace**



Fuente: elaboración propia, empleando Canva.

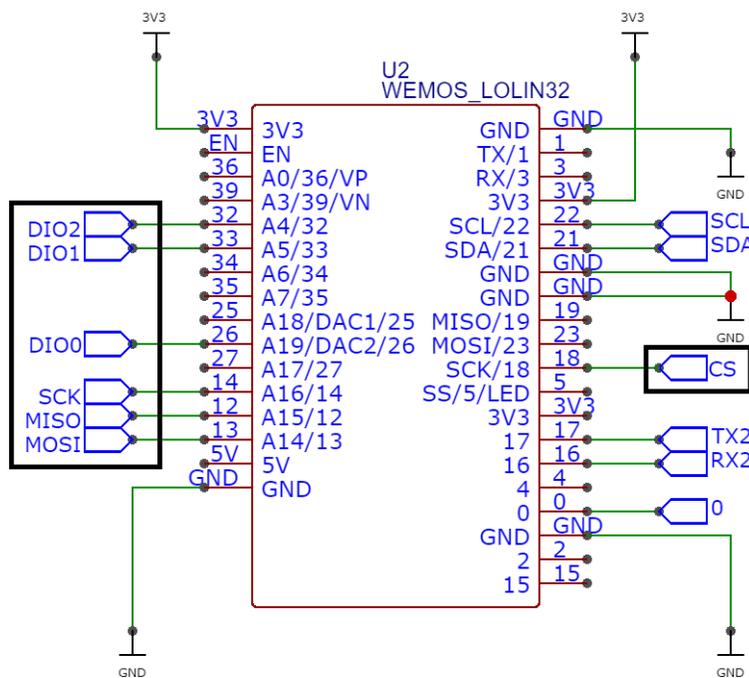
En la figura 93 se pueden apreciar un número menor de bloques, aunque sea de una naturaleza simple, es un dispositivo esencial en la aplicación. La línea punteada reúne al microprocesador y el módulo Wifi, en este caso se eligió la tarjeta de desarrollo Weemos lolin 32, que posee el SoC ESP32. Una batería de respaldo es opcional pero no fundamental para su construcción. El uso de una batería como fuente principal se descarta por los siguientes motivos: la comunicación Wifi requiere de potencia relativamente alta para transmitir y procesar los paquetes. El dispositivo debe estar abierto a la escucha de cualquier mensaje, es decir, que se mantenga “despierto”. Se hace esta

referencia puesto que el modo de descanso es una técnica ampliamente utilizada para aumentar la vida de la batería.

4.3. Esquemático

El lector puede apreciar la similitud física entre el modo y la puerta de enlace. En función de los requerimientos actuales: un prototipo funcional y técnicas de desarrollo se añade la optimización de tiempo de ejecución y capital requerido. En este caso la tarjeta del nodo también puede ser utilizada para instalar la puerta de enlace. En la figura 94 se encierran las conexiones necesarias para comunicarse hacia el módulo LoRa.

Figura 94. Esquemático eléctrico para la puerta de enlace



Fuente: elaboración propia, empleando easyEDA.

4.4. Prototipo

El dispositivo más cercano a la aplicación final es el prototipo, un paso clave para verificar el sistema de alimentación y líneas de comunicación. Existen factores que no pueden ser simulados en una computadora. Por lo cual una prueba en campo es sumamente útil.

4.4.1. Circuito impreso

El circuito impreso posee las conexiones eléctricas y mantiene mecánicamente fijas las piezas con pequeñas soldaduras de estaño.

Figura 95. **Circuito finalizado para la puerta de enlace**

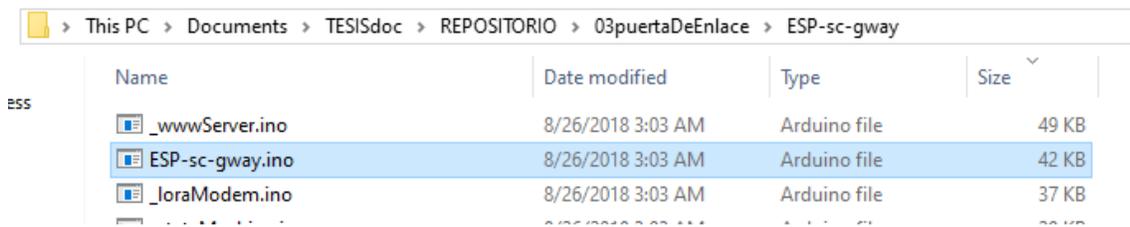


Fuente: elaboración propia.

4.4.2. Programación

Para hallar el código correspondiente, abrir el IDE de Arduino como se presentó en la figura 12. Ubicar el ejemplo ESP-sc-gway.ino descrito en la figura 96. Una vez abierto, el lector puede darse cuenta de que el archivo automáticamente abre más ficheros. No es necesario modificar nada si el desarrollador posee los módulos y conexiones definidas durante el texto.

Figura 96. **Programación de la puerta de enlace**



Fuente: elaboración propia, empleando Arduino 1.8 2019.

Figura 97. **Configuración de parámetros generales**



Fuente: elaboración propia, empleando Arduino 1.8 2019.

La figura 97 presenta el fichero donde se encuentran las opciones a tomar en cuenta. El desarrollador tiene que estar consiente de:

- `_LFREQ`: frecuencia central de transmisión para elegir la antena adecuadamente, el ejemplo se encuentra configurado a 915 MHz.
- `_PINOUT`: son las conexiones entre el SoC y el módulo RFM96.

- `_EMAIL`: La puerta de enlace tendrá enlace hacia internet y será necesaria una cuenta de correo electrónico para configurarla.
- `wpa wpa[]`: El usuario proporciona el nombre de la red y contraseña correspondiente para acceder a internet.

Al terminar de familiarizarse con los parámetros mencionados cargar el programa hacia la tarjeta, después el monitor serial desplegará la dirección donde se encuentra la puerta de enlace. La dirección que se muestra en la figura 98 es válida dentro de una red, sólo si la computadora está conectada a la misma. Al ingresar la dirección en un buscador web el desarrollador podrá apreciar una página como la indicada en la figura 99.

Figura 98. **Dirección de la puerta de enlace**

```
Host esp32-fc9ae8 WiFi Connected to labTeo on IP=192.168.1.168
Gateway ID: 3C71BFFFFFFC9AE8, Listening at SF7 on 903.90 Mhz.
setupOta:: Started
Ready
IP address: 192.168.1.168
Time: Saturday 20:24:29
```

Fuente: elaboración propia, empleando Arduino 1.8 2019.

Figura 99. **Página generada por la puerta de enlace**

The screenshot shows a web browser window with the address bar displaying '192.168.1.168'. The page title is 'ESP Gateway Config'. Below the title, there is a version string 'Version: V1.3.3.H.180824', system information 'ESP-8266 module: Saturday, 22-2-2020 20:24:29, Uptime: 0:00:07.06', and current time 'Current time: Saturday, 22-2-2020 20:20:47'. There are three navigation links: 'Documentation', 'Basic Mode', and 'Log Files'. The main content area is titled 'Package Statistics' and contains a table with the following data:

Counter	C 0	C 1	C 2	Pkgs	Pkgs/hr
Packages Downlink				0	
Packages Uplink Total				0	0
Packages Uplink OK				0	
SF7 rcvd	0	0	0	0	0 %
SF8 rcvd	0	0	0	0	0 %
SF9 rcvd	0	0	0	0	0 %
SF10 rcvd	0	0	0	0	0 %
SF11 rcvd	0	0	0	0	0 %
SF12 rcvd	0	0	0	0	0 %

Fuente: elaboración propia, empleando Arduino 1.8 2019.

4.5. Pruebas de funcionamiento

La puerta de enlace posee estadísticas y niveles de depurado seleccionable desde su página, el nodo está configurado para enviar mensajes cuando el desarrollador lo decida. Hay que recordar que el módulo GPS debe anclar su posición, éste parpadea 1 vez por segundo para indicar que ya conoce su posición satelital, una vez anclado puede presionar el botón indicado en la figura 100 para iniciar el enlace.

Figura 100. **Botón de transmisión**



Fuente: elaboración propia.

No hace falta configurar nada en la página de la puerta de enlace, para fines prácticos esta se utiliza para confirmar que el módulo este transmitiendo el paquete en el canal y a la frecuencia elegida. Si el desarrollador ha ejecutado exitosamente cada etapa el radioenlace ha quedado establecido.

Figura 101. **Mensajes recibidos en la puerta de enlace**

Package Statistics

Counter	C 0	C 1	C 2	Pkgs	Pkgs/hr
Packages Downlink				0	
Packages Uplink Total				3	29
Packages Uplink OK				3	
SF7 rcvd	3	0	0	3	100 %
SF8 rcvd	0	0	0	0	0 %
SF9 rcvd	0	0	0	0	0 %
SF10 rcvd	0	0	0	0	0 %
SF11 rcvd	0	0	0	0	0 %
SF12 rcvd	0	0	0	0	0 %

Message History

Time	Node	C	Freq	SF	pRSSI
Tuesday 23-2-2020 18:48:45	26 02 11 b4	0	903900000	7	-35
Tuesday 23-2-2020 18:44:06	26 02 11 b4	0	903900000	7	-37
Tuesday 23-2-2020 18:43:37	26 02 11 b4	0	903900000	7	-36

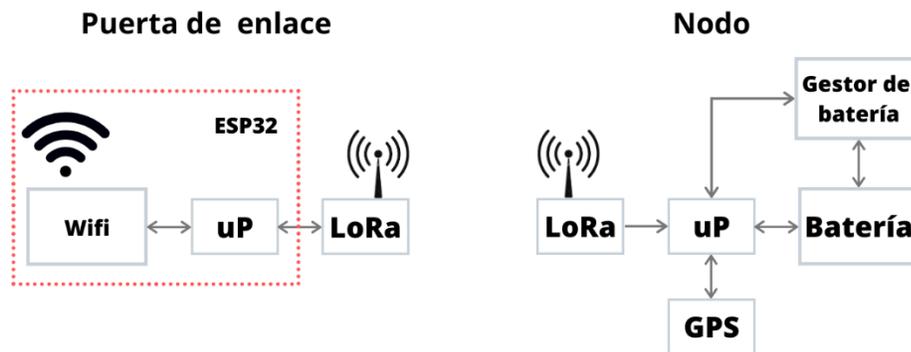
Fuente: elaboración propia. Arduino 1.8 2019.

5. DISEÑO FINAL Y PRUEBAS DEL PROTOTIPO

5.1. Diagrama de bloques

Para describir la tarea del dispositivo a grandes rasgos y describir el flujo de funcionamiento.

Figura 102. Montaje de campo



Fuente: elaboración propia, empleando Canva.

Al disponer de la puerta de enlace y nodo se procede a una descripción más detallada de algunos parámetros de programación dedicados a una presentación más conveniente de la información. Similar al algoritmo de seccionamiento que se debe aplicar a una sentencia NMEA para extraer un punto de ubicación terrestre. En general, el mensaje que se obtiene desde el nodo llega a la puerta de enlace para ser presentado en texto plano. Para

acceder al texto que contiene la información. Se puede apreciar el equipo completo en la figura 102.

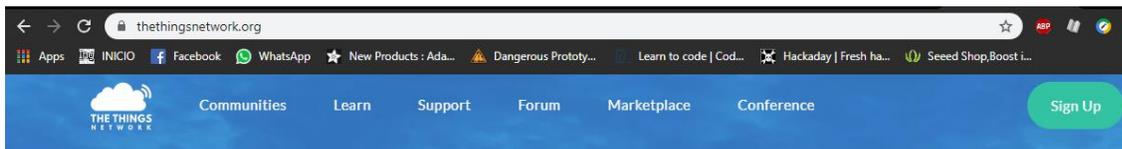
5.2. Adquisición de datos

Un tema extenso en el área de instrumentación, en la presente sección se refiere al direccionamiento de la información a la página de presentación al usuario.

5.2.1. Conexión a internet

Se utilizará una página para gestionar la información, The Things Network: consultar figura 103. Esta contiene un servidor para crear aplicaciones de manera segura y encriptada, de esta manera solamente el usuario con los derechos correspondientes podrá acceder a los datos. Para registrar un dispositivo entrar a la página thethingsnetwork.org y crear un usuario.

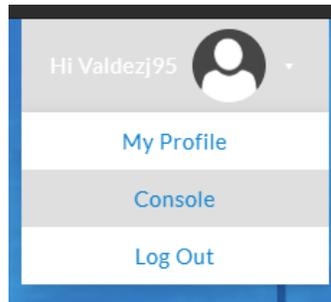
Figura 103. The Things Network



Fuente: elaboración propia, empleando Printscreen de Windows.

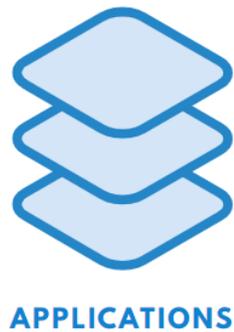
Una vez finalizado el usuario, ingresar con las credenciales y elegir la opción: consola como se muestra en la figura 104. Para crear la aplicación elegir la opción correspondiente y crear o añadir un dispositivo como se muestra en la figura 106.

Figura 104. **Usuario en The Things Network**



Fuente: elaboración propia, empleando Printscreen de Windows.

Figura 105. **Registro del dispositivo**



Fuente: elaboración propia, empleando Printscreen de Windows.

Figura 106. **Añadir aplicación**



Fuente: elaboración propia, empleando Printscreen de Windows.

La figura 107 muestra configuraciones mínimas u obligatorias para crear una aplicación: Nombre y Descripción. El EUI Identificador Único Extendido y gestor de registro se otorgan por defecto. La parte física correspondiente es la puerta de enlace.

A continuación, será necesario configurar el nodo. Existen 2 parámetros de especial interés durante esta etapa, el identificador único extendido del dispositivo y la llave de aplicación para la sesión, consultar en la figura 108. El radioenlace existe como mínimo entre 2 dispositivos. Será de utilidad entonces, la capacidad de identificar y transmitir comandos a un nodo en específico.

Una vez obtenidas, configurar el modo de activación y el contador de cuadros como se indica en la figura 109.

Figura 107. **Creación de la aplicación**

ADD APPLICATION

Application ID
The unique identifier of your application on the network
esp32-1ch-test-01

Description
A human readable description of your new app
Test application for ESP32 1-CH gateway

Application EUI
An application EUI will be issued for The Things Network block for convenience; you can add your own in the application settings page.
EUI issued by The Things Network

Handler registration
Select the handler you want to register this application to
ttn-handler-us-west

Fuente. elaboración propia, empleando Printscreen de Windows.

Figura 108. Registro del dispositivo

Overview Devices

REGISTER DEVICE

Device ID
This is the unique identifier for the device in this app. The device ID will be immutable.

esp32-1ch-gw-dev

Device EUI
The device EUI is the unique identifier for this device on the network. You can change the EUI later.

App Key
The App Key will be used to secure the communication between you device and the network.

App EUI

Fuente: elaboración propia, empleando Printscreen de Windows.

Figura 109. Características extra

The screenshot shows a configuration window with the following sections:

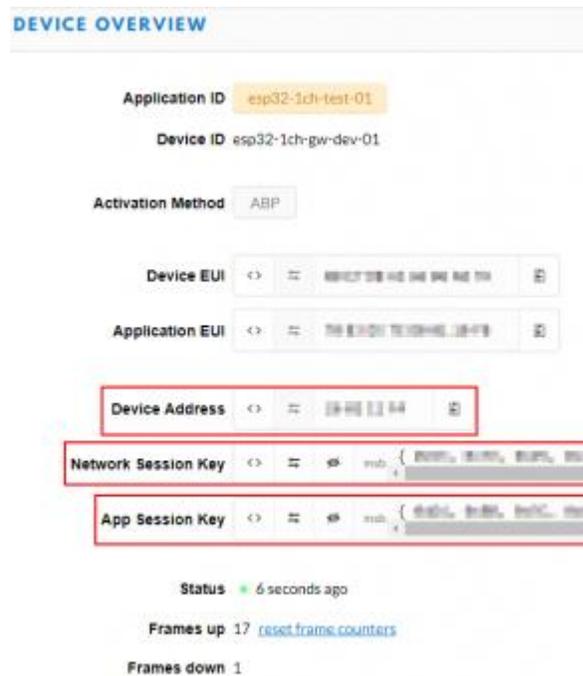
- Activation Method:** Two radio buttons, 'OTAA' (selected) and 'ABP'.
- Device Address:** A text field containing '26 02 11 F4' and a green indicator '4 bytes'.
- Network Session Key:** A text field containing a hexadecimal string and a green indicator '16 bytes'.
- App Session Key:** A text field containing a hexadecimal string and a green indicator '16 bytes'.
- Frame Counter Width:** Two radio buttons, '16 bit' (selected) and '32 bit'.
- Frame Counter Checks:** A checkbox that is unchecked, with a warning message: 'Disabling frame counter checks drastically reduces security and should only be used for development purposes'.

At the bottom of the window, there are three buttons: 'Delete Device' (in red), 'Cancel', and 'Save'.

Fuente: elaboración propia, empleando Printscreen de Windows.

Para finalizar de configurar la pareja de elementos es necesario que el nodo contenga la identificación desde el código fuente. Cuya ruta se encuentra en la figura 87. El texto generado en: Network sesión key, App Session Key y Device address corresponden a NWKSKEY[16], APPSKEY[16] y DEVADDR respectivamente. Migrar los datos de la figura 110 al código de la figura111.

Figura 110. Configuración completa



Fuente: elaboración propia, empleando Printscreen de Windows.

Figura 111. Identificadores del nodo

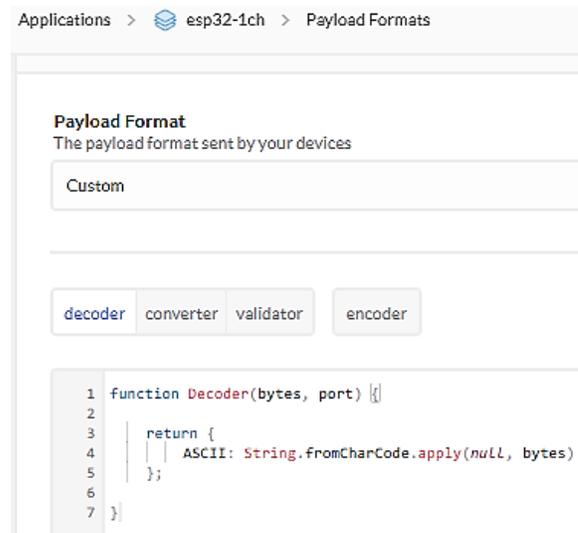
```
50 // LoRaWAN NwsKEY, network session key
51 static const PROGMEM u1_t NWKSKEY[16] = { 0xFF,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5};
52
53 // LoRaWAN AppSKey, application session key
54 static const u1_t PROGMEM APPSKEY[16] = { 0xFF,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5,0xA5};
55
56 // LoRaWAN end-device address (DevAddr)
57 static const u4_t DEVADDR = 0x01234567;
```

Fuente: elaboración propia, empleando Arduino 1.8.

Al finalizar de cargar el código, se puede acceder a la información, ésta se encuentra codificada, puede elegirse el formato e incluso realizar un decodificador a conveniencia, un método común de acondicionamiento se

muestra en la figura 112. En este caso se eligió un formato de carga liviano desarrollado por Cayenne figura 113.

Figura 112. **Decodificador estándar**



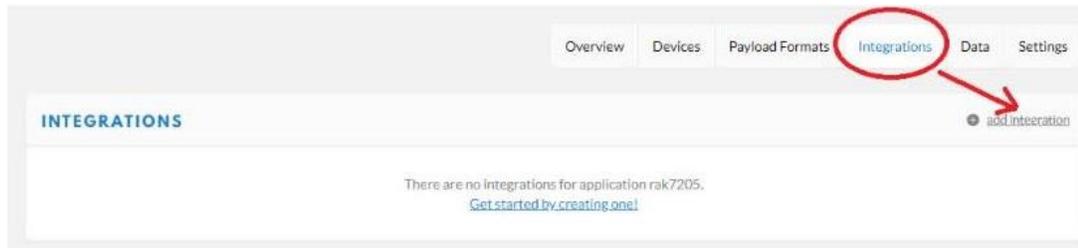
Fuente: elaboración propia, empleando Arduino 1.8 2019.

Figura 113. **Formato LPP de Cayenne**



Fuente: elaboración propia. empleando Arduino 1.8 2019.

Figura 114. Integración de aplicaciones



Fuente: elaboración propia. empleando Arduino 1.8 2019.

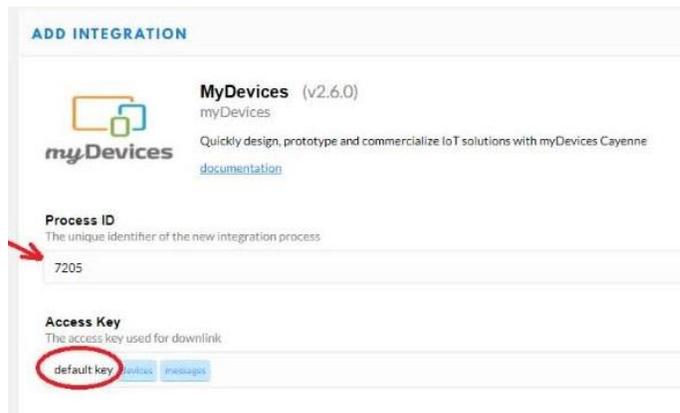
Para un posterior enlace de aplicación: registrar una integración con myDevices de Cayenne, explicados en las Figuras: 114, 115 y 116.

Figura 115. Aplicación myDevices



Fuente: elaboración propia. empleando Arduino 1.8 2019.

Figura 116. **Identificadores de la integración**

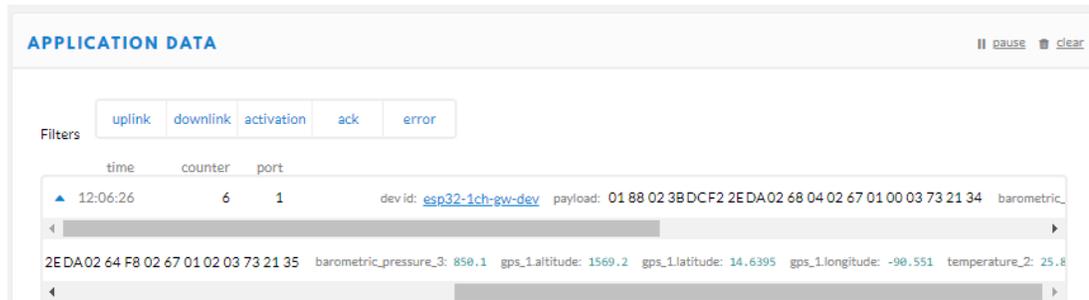


Fuente: elaboración propia. empleando Arduino 1.8 2019.

5.2.1.1. **Monitoreo remoto de la información**

La información del nodo se muestra en texto plano además se puede apreciar la información decodificada en la figura 117. En este punto el desarrollador puede hacer uso de su información: almacenarla, realizar acciones basadas en mediciones y estudios de interés. Dentro del contexto, un tablero de información presenta las variables de manera gráfica y cómoda al usuario. Se eligió la plataforma de Cayenne para elaborar el tablero.

Figura 117. **Visualización de texto**

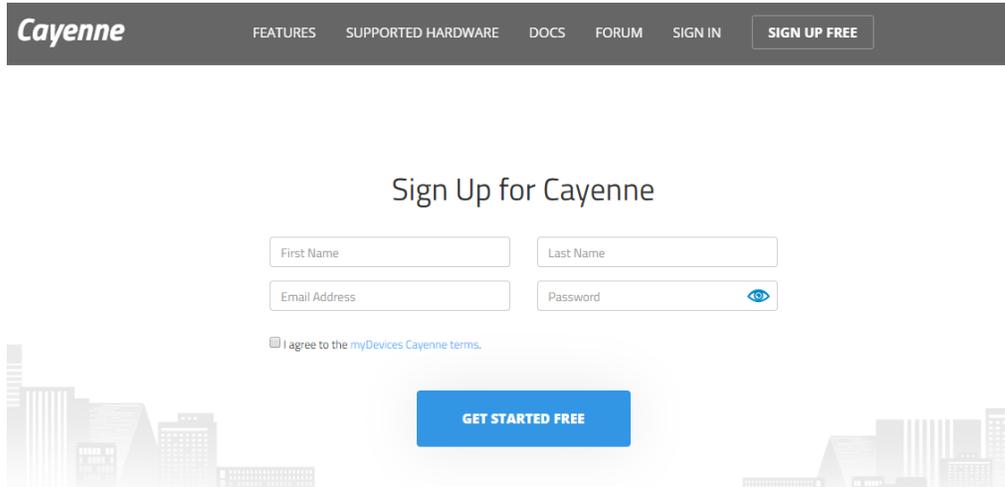


Fuente: elaboración propia. empleando Arduino 1.8 2019.

5.2.2. **Reporte de posición**

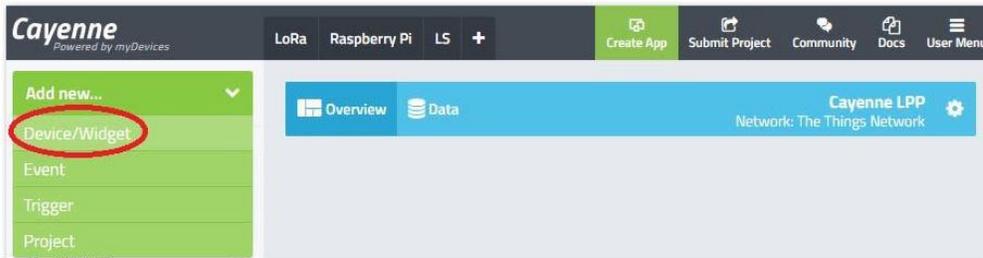
Para realizar generar el tablero visual generar un usuario ingresando a <https://developers.mydevices.com/cayenne/signup/> seguir los pasos correspondientes de las figuras 118 y 119 para añadir un nuevo dispositivo. Al aparecer una nueva pestaña elegir la categoría LoRa y subcategoría The Things Network como se aprecia en la figura 120. Elegir la opción de la figura 121.

Figura 118. Creación de usuario



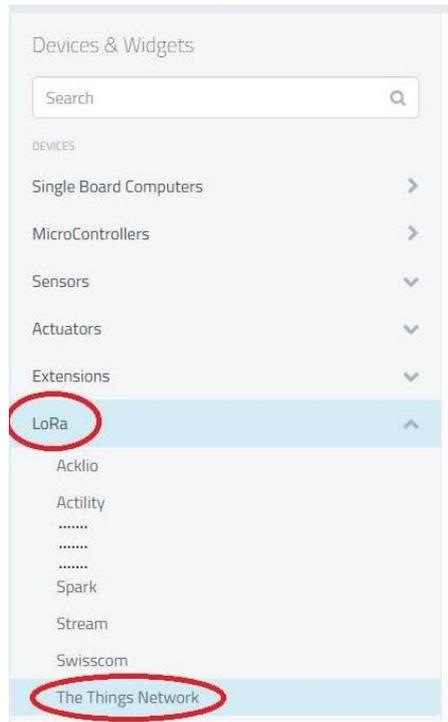
Fuente: elaboración propia, empleando Cayenne.

Figura 119. Usuario



Fuente: elaboración propia, empleando Cayenne.

Figura 120. **Tipos de dispositivos**



Fuente: elaboración propia, empleando Cayenne.

Figura 121. **Formato de carga liviana**



Fuente: elaboración propia, empleando Cayenne.

Los ajustes correspondientes a la aplicación utilizada se muestran en la figura 122. La identificación del dispositivo devEUI se toma de la figura 110. El

tablero de información es personalizable y es posible elegir la posición, tamaño y datos a desplegar. Consultar figura 123.

Figura 122. **Ajustes del dispositivo**

Enter Settings

 Cayenne Cayenne LPP
Cayenne Low Power Payload

This device uses Cayenne LPP

Name
Cayenne LPP

DevEUI

Activation Mode
Already Registered ▼

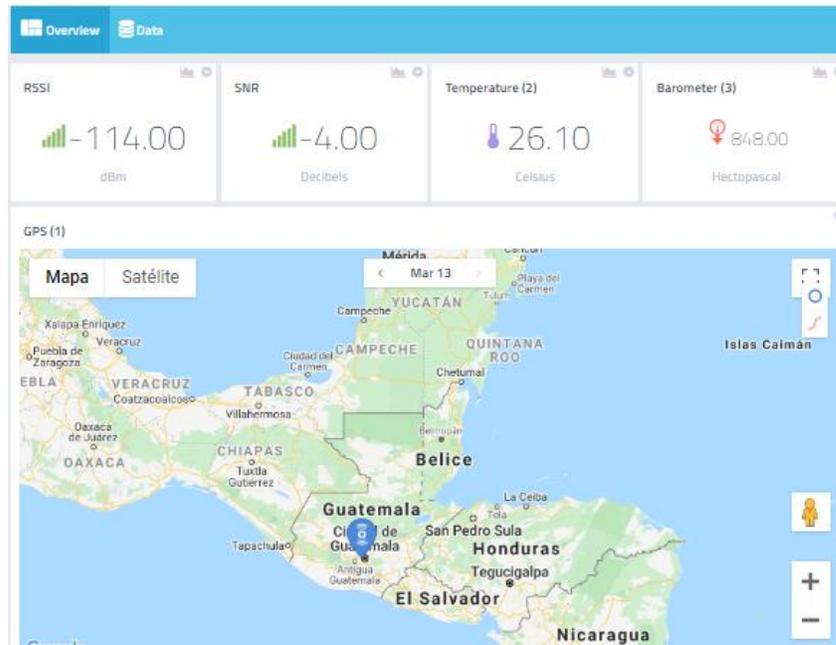
Tracking

Location
This device moves ▼

Add device

Fuente: elaboración propia, empleando Cayenne.

Figura 123. **Tablero de información**

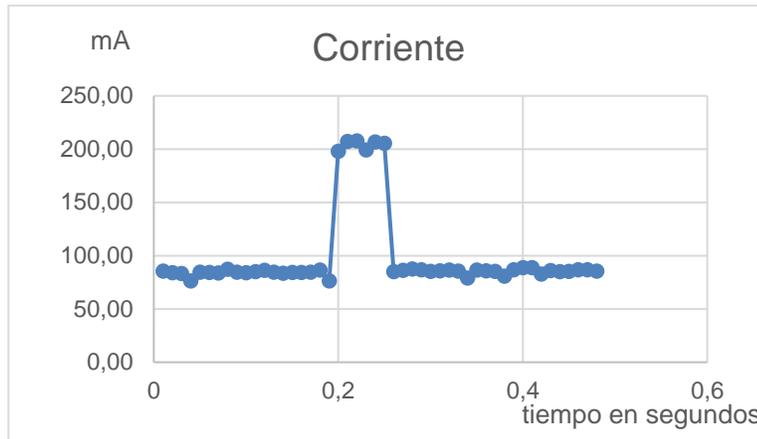


Fuente: elaboración propia, empleando Cayenne.

5.2.3. **Reporte de consumo**

La figura 124 registra el funcionamiento normal y de transmisión, con un promedio de 85 mA y un pico de corriente de aproximadamente 205 mA con una duración de 65 milisegundos por mensaje. La batería utilizada para alimentar el circuito posee una capacidad de 850mAh, en el peor de los casos la aplicación consumiría 205 mA constantemente hasta agotar la batería, dando como resultado una vida útil de aproximadamente 4 horas. En el mejor de los casos consumirá 85 mA sin enviar información, resultando en una vida útil de aproximadamente 10 horas por carga.

Figura 124. **Consumo de corriente durante transmisión**



Fuente: elaboración propia.

Si bien una aplicación que consume alrededor de 1 W por mensaje entra en la categoría de bajo consumo, un dispositivo inalámbrico con ciclos de funcionamiento de 4 a 10 horas no es eficiente. Existen técnicas para agotar energía óptimamente: consiste en desconectar periféricos y módulos tanto internos como externos mientras un circuito auxiliar de ultra bajo consumo mantiene un registro temporal, cuando termina el tiempo de descanso toda la circuitería entra en funcionamiento, se ejecuta el código principal para que al terminar el nodo hiberne y repita el ciclo.

Para obtener el estimado de vida útil, se puede utilizar el enlace <http://www.of-things.de/battery-life-calculator.php> y los valores que se muestran en la figura 125. La duración de ejecución de transmisión experimental fue de 0,065 segundos, se eligió un tiempo de hibernado de 10 minutos, consumo experimental de 205 mA y un estimado de corriente de hibernación de 200 uA. Los resultados representan una vida útil de 136 días, aproximadamente 4

meses. La implementación, el modo de ultra bajo consumo queda fuera del alcance del texto.

Figura 125. **Estimador de tiempo de descarga**

Your results:
Your device will probably run for **3267 hours** or around **136 days and 3 hours**
Its estimated, average power consumption per hour **0.23 mAh**

Software

duration of code execution: **sec** sleep time: **sec**

Hardware

consumption during code execution: **mA** consumption in sleep mode*: **µA**

Battery

power of battery: **mAh** discharge safety: **%**

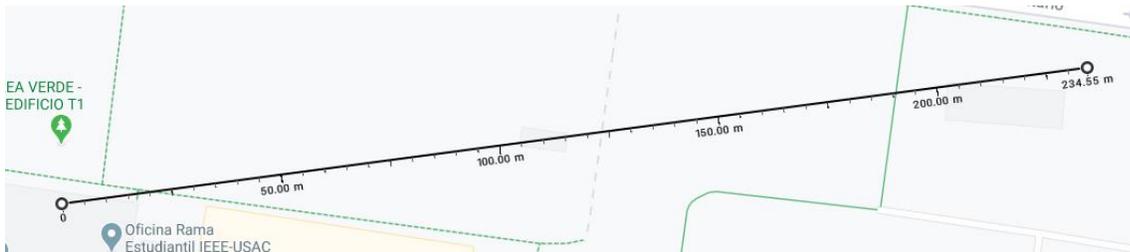
Fuente: Toolbox for developing IOT. *Simple calculator for estimating a (LiPo) battery's life.*
<http://www.of-things.de/battery-life-calculator.php>. Consulta: marzo de 2020.

5.2.4. **Resultados**

El rango del enlace depende fuertemente del entorno y tipo de antena. Una buena línea de vista entre los módulos es importante. Las pruebas de las figuras 126, 127 y 128 se realizaron en un entorno urbano con obstáculos como paredes, árboles, estructuras metálicas e interferencia. Aproximadamente con una altura de 1 m sobre el nivel del suelo. Con antenas 915 MHz, de 50 Ohmios, omnidireccionales y ganancia de 3 dBi. La figura muestra una prueba en terreno rural y línea de vista, es de útil referencia para dimensionar las capacidades mínimas y máximas del enlace.

Los dispositivos fueron fabricados con herramientas de código abierto y materiales disponibles en el mercado actual, es posible replicar el enlace con un presupuesto de Q 1 000,00, dependiendo de los proveedores. Cada mensaje consume menos de 1 W en fracciones de segundo con registros de distancia experimental de 7 Km bajo condiciones óptimas. Tomar como referencia la figura 129.

Figura 126. **Prueba de distancia 1**



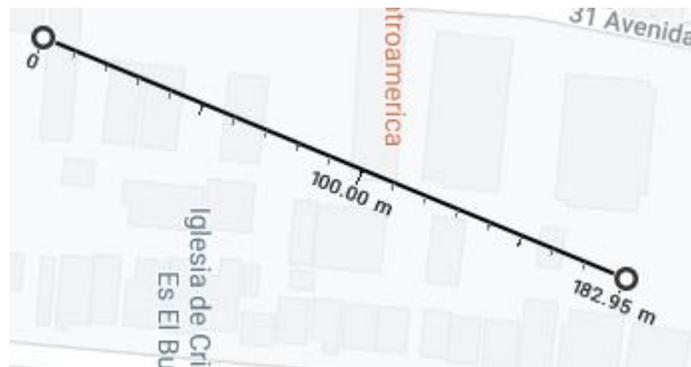
Fuente: elaboración propia, empleando Cayenne.

Figura 127. **Prueba de distancia 2**



Fuente: elaboración propia, empleando Cayenne.

Figura 128. **Prueba de distancia 3**



Fuente: elaboración propia, empleando Cayenne.

Figura 129. **Ejemplo de implementación en condición rural**



Fuente: YouTube. *Testing long range wireless 915Mhz RF LoRa transceiver module REYAX RYLR890 RYLR896 Arduino part 3.* <https://www.youtube.com/watch?v=BuOtvYOykjU>. Consulta: marzo de 2020.

CONCLUSIONES

1. Se otorga una guía de elaboración práctica de un sistema de comunicación de baja potencia y largo alcance con tecnologías actuales utilizadas a nivel mundial.
2. Se presenta el proceso de prueba y depuración modular como principio básico de desarrollo.
3. Se utiliza equipo de bajo costo como punto de entrada efectivo frente a nuevas tecnologías.
4. El dispositivo inalámbrico es capaz de funcionar con baterías de bajo voltaje, con un consumo corriente de aproximadamente 200mA durante el proceso de transmisión.
5. Existen pruebas que documentan enlaces de hasta 7Km bajo condiciones óptimas.

RECOMENDACIONES

1. Debido al surgimiento de nuevas tecnologías afines a la rama de la electrónica, es necesario que el estudiante e ingeniero posea habilidades de desarrollo a lo largo de su vida competitiva. Una serie de principios adquiridos de forma práctica y teórica son esenciales en el proceso de auto aprendizaje. Las destrezas mencionadas tienen que formar parte de sus herramientas profesionales. Puesto que a consecuencia de un cambio; una solución, se vuelve necesaria.
2. El desarrollo de una solución altamente integrada generalmente es diseñada, elaborada, gestionada y aplicada dentro de un grupo multidisciplinario, el trabajo y coordinación de un equipo se nutre a base de experiencia profesional, intercambio de ideas y diversidad de criterios. Es importante identificar cuando se requiere de asistencia para el progreso adecuado del objetivo de interés.
3. El estudiante de ingeniería electrónica de la Universidad de San Carlos de Guatemala adquiere una gran parte de sus destrezas durante su curso de laboratorio. A medida que la carrera avanza los proyectos se vuelven más complejos en su materia de estudio, sin embargo, existe poca relación entre ellos. Se sugiere combinar al menos 2 proyectos para su elaboración durante 2 semestres.

4. Los reportes de progreso son documentos hechos por los alumnos que demuestran a grandes rasgos detalles importantes de la elaboración de su proyecto de laboratorio, dichos documentos quedan archivados como propiedad exclusiva de quien diseñó el proyecto. Es información valiosa que vale la pena documentar y utilizar como base para futuros participantes. Los detalles que surgen a partir de la experiencia de previos alumnos pueden utilizarse para proponer y elaborar nuevos temas de investigación.

BIBLIOGRAFÍA

1. Advantech. *Advantech's LoRa solutions*. [en línea]. <<http://select.advantech.com/lora/en-us/>>. [Consulta: 16 de enero de 2020].
2. Avnet. *Sensor Node Lora documentation*. [en línea]. <<https://loranode.readthedocs.io/en/latest/>>. [Consulta: 21 de enero de 2020].
3. Bitbucket. *Whisper Node - AVR LoRa*. [en línea]. <<https://bitbucket.org/talk2/whisper-node-avr-lora/src/master/>>. [Consulta: 24 de enero de 2020].
4. IBM. *Nodos de red*. [en línea]. <https://www.ibm.com/support/knowledgecenter/es/SSXK2N_1.4.0/com.ibm.powervc.standard.help.doc/powervc_network_nodes.html>. [Consulta: 29 de diciembre de 2019].
5. Sparkfun. *Hardware y software de código abierto*. [en línea]. <<https://learn.sparkfun.com/>>. [Consulta: 20 de diciembre de 2019].
6. Technopedia. *Node*. [en línea]. <<https://www.techopedia.com/definition/5307/node>>. [Consulta: 7 de enero de 2020].

7. Wikipedia: la enciclopedia libre. *Electrónica analógica*. [en línea]. <https://es.wikipedia.org/wiki/Electrónica_analógica>. [Consulta: 15 de diciembre de 2019].

8. _____. *Electrónica digital*. [en línea]. <https://es.wikipedia.org/wiki/Electrónica_digital>. [Consulta: 15 de diciembre de 2019].