

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“SERVICIO DE DRONES APLICADO AL MONITOREO
DE LA AGRICULTURA EN EL ECUADOR CON
TECNOLOGÍA LORA”**

EXAMEN DE GRADO

Previa a la obtención del Título de:

MAGISTER EN TELECOMUNICACIONES

GISELL ESTHER LITARDO SANDOVAL

GUAYAQUIL - ECUADOR

AÑO: 2020

AGRADECIMIENTOS

En primer lugar, agradezco a Dios por haberme acompañado y guiado en cada etapa de estudio, por darme la fortaleza en momentos de debilidad, y por darme la sabiduría necesaria para enfrentar cada desafío presentado.

Le doy gracias a mi mamá, María de Lourdes, por apoyarme en todo momento siendo un pilar fundamental en mi vida, por sus palabras de aliento, por su ejemplo de fortaleza, y por impulsarme a seguir creciendo en el ámbito profesional.

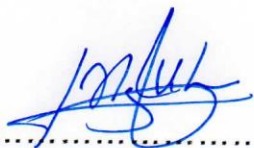
A mi hermana Gisella, por ser un ejemplo de desarrollo profesional a seguir, a mis sobrinos Jimena y Emilio Isaac, por siempre preocuparse por mis estudios y por saber cómo me iba en la universidad.

A Andrés, por ser alguien especial en mi vida, quien estuvo en cada momento de alegría y de tristeza durante mis estudios, por siempre apoyarme, darme ánimo, y sobre todo por su paciencia y amor incondicional.

DEDICATORIA

Dedico este trabajo, a mi mamá María de Lourdes, mi hermana Gisella, sobrinos Jimena y Emilio Isaac, y a mi enamorado Andrés, quienes creyeron en mí, me dieron aliento y fuerza necesaria para seguir adelante y llevar a cabo este desafío.

TRIBUNAL DE EVALUACIÓN



.....
MSc. Alfredo Nuñez
PROFESOR EVALUADOR

.....
PhD. María Antonieta Álvarez
PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me corresponde exclusivamente; y doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Gisell Esther Litardo Sandoval

RESUMEN

El presente proyecto consiste en el diseño de un sistema de monitoreo para el sector agrícola del Ecuador, por medio de Drones usando la tecnología LoRa. LoRa es una de las tecnologías de redes de bajo consumo y área extensa conocida por su nombre en inglés Low-Power Wide Area Network (LPWAN), que cuenta con grandes ventajas para proyectos de Internet de las Cosas (IoT).

En los drones se incorporará una placa de desarrollo, pycom lopy4, con una placa de expansión pycom pysense, la cual contiene sensores de temperatura, humedad, presión, altitud y el voltaje de la misma; que a su vez transmitirán dicha data por medio de la tecnología LoRa hacia el gateway LoRaWAN.

La plataforma que permitirá la integración de la placa de desarrollo con el gateway es The Things Network (TTN), ya que está desarrollada con el objetivo de que los dispositivos que utilicen LoRaWAN logren una enlazarse a Internet. En la plataforma The Things Network, se realizará el registro tanto del gateway LoraWAN como de la placa de desarrollo, pycom lopy4, para poder realizar la transmisión de la data que ha sido sensada. Así mismo, se realizará una integración en TTN hacia thingspeak, que es la plataforma de IoT la cual permite recoger, almacenar y visualizar la data de forma gráfica y en tiempo real, permitiendo que los especialistas analicen los resultados de la data sensada.

ÍNDICE GENERAL

AGRADECIMIENTOS	I
DEDICATORIA	II
TRIBUNAL DE EVALUACIÓN	III
DECLARACIÓN EXPRESA	IV
RESUMEN	V
ÍNDICE GENERAL	VI
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	
XI CAPÍTULO 1	1
1. INTRODUCCIÓN	1
1.1. DESCRIPCIÓN DEL PROBLEMA	1
1.2. JUSTIFICACIÓN	1
1.3. OBJETIVOS	2
1.3.1. Objetivo General	2
1.3.2. Objetivos Específicos:	2
1.4. MARCO TEÓRICO	3
1.4.1. LPWAN	3
1.4.2. LoRa	5
1.4.3. LoRaWAN	6

CAPÍTULO 2	8
2. DISEÑO DEL SISTEMA DE MONITOREO CON TECNOLOGÍA LORA	8
2.1. DISPOSITIVOS UTILIZADOS EN EL SISTEMA DE MONITOREO	8
2.1.1. Pycom LoPy4	8
2.1.2. Pysense	12
2.1.3. Gateway LoRaWAN.....	13
2.1.4. Drone	14
2.2. DIAGRAMA DE LA SOLUCIÓN	15
2.3. DIAGRAMA DE FLUJO DEL SOFTWARE	22
2.3.1. Importación de librerías	24
2.3.2. Configuración Inicial	24
2.3.3. Instanciación de variables	25
2.3.4. Conexión al Servidor LoRaWAN.....	26
2.3.5. Configuración del socket LoRaWAN.....	28
2.3.6. Eventos LoRa	28
2.3.7. Adquisición de datos y transmisión	29
2.3.8. Decodificación	30
CAPÍTULO 3	32
3. ANÁLISIS DE RESULTADOS	32
CONCLUSIONES Y RECOMENDACIONES	37
CONCLUSIONES	37
RECOMENDACIONES	37
BIBLIOGRAFÍA	38

ÍNDICE DE FIGURAS

Figura 1.1. Espacio regido por tecnologías inalámbricas en términos de ancho de banda y alcance físico.	4
Figura 1.2. Tecnologías LPWAN: Características principales.	4
Figura 1.3. Arquitectura para LoRaWAN	6
Figura 1.4. Estructura de red LoraWAN	7
Figura 1.5. LoRaWAN: Casos de Uso	5
Figura 2.1. Lopy4 Board	8
Figura 2.2. Versión inicial del Lopy4: v1.9.4-94bb382 on 2019-08-22	8
Figura 2.3. Versión final del Lopy4: v1.11-db33be7 on 2020-07-01	9
Figura 2.4. Configuración de los parámetros del firmware	10
Figura 2.5. Resultados del software empleado para actualizar el firmware ...	10
Figura 2.6. Configuración de los parámetros del firmware	11
Figura 2.7. Resultados del software empleado para actualizar el firmware ...	11
Figura 2.8. Pysense	12
Figura 2.9. Pinout del Pysense	13
Figura 2.10. Gateway LoRaWAN.....	14
Figura 2.11. Drone.....	15
Figura 2.12. Prueba de los sensores	16
Figura 2.13. Integración dron con Lopy4 y pysense	17

Figura 2.14. Parámetros de Registro de la Aplicación en TTN.....	17
Figura 2.15. Registro del Gateway en TTN	17
Figura 2.16. Flujo de mensajes recibidos por el gateway	18
Figura 2.17. Flujo de datos en la aplicación del dispositivo	18
Figura 2.18. Método de Registro ABP en TTN	19
Figura 2.19. Contador de paquetes de subida del dispositivo en TTN	19
Figura 2.20. Reset Frame Counters	19
Figura 2.21. Intefración de Thingspeak	20
Figura 2.22. Parámetros de Thingspeak.....	20
Figura 2.23. Configuraciones del Canal.....	21
Figura 2.24. Diagrama de Bloques de la Solución.....	22
Figura 2.25. Diagrama de la Solución.....	22
Figura 2.26. Diagrama de Flujo	23
Figura 2.27. Librerías.....	24
Figura 2.28. Configuración Inicial	25
Figura 2.29. Terminal de Visual Studio Code: Configuración Inicial	25
Figura 2.30. Configuración Inicial: Led Blanco encendido	25
Figura 2.31. Instanciación de variables	26
Figura 2.32. Terminal de Visual Studio Code: Instancias creadas.....	26
Figura 2.33. Conexión al Servidor	27
Figura 2.34. Terminal de Visual Studio Code: Conexión a TTN	28
Figura 2.35. Configuración del socket LoRaWAN.....	28
Figura 2.36. Eventos de LoRa	28

Figura 2.38. Terminal de Visual Studio Code: Mensaje de Envío del paquete.	29
Figura 2.39. Adquisición de datos y transmisión.....	29
Figura 2.41. Adquisición de datos: Led Azul encendido	30
Figura 2.42. Transmisión de datos: Led Verde encendido	30
Figura 2.43. Decodificación	31
Figura 3.1. Visual Studio Code: Packets sent.....	33
Figura 3.2. Payload Formats	33
Figura 3.3. Applications Data	33
Figura 3.4. Gateway Traffic	34
Figura 3.5. Monitoreo de la Temperatura, Humedad, Altitud, Presión y Voltaje	35
Figura 3.6. Exportar de data reciente	36
Figura 3.7. Exportar de data del Canal	36
Figura 3.8. Data Exportada .CSV	36

ÍNDICE DE TABLAS

Tabla 1. Significado de los colores del LED.	23
---	----

ABREVIATURAS

LPWAN	Low-Power Wide Area Network
IoT	Internet of Things
TTN	The Things Network
CR	Coding Rate

CAPÍTULO 1

1. INTRODUCCIÓN

1.1. DESCRIPCIÓN DEL PROBLEMA

En los últimos 6 años se ha podido identificar que uno de los principales problemas en el sector agrícola es la zonificación, ya que en algunos casos las unidades productivas se encuentran en zonas agroecológicas, y no deberían estar donde la tierra no es óptima, el clima no es aparente, son lugares bajos y cuando llegan las lluvias hay problemas. Por tal motivo una de las formas de realizar un frente ante esta realidad, es por medio de programas de incremento de productividad en las zonas de producción visitando [1], levantando información, monitoreando las condiciones de la tierra para analizar y verificar si es rentable el cultivo en la zona. Este monitoreo de las condiciones de la tierra tiende a ser agotador y demorado ya que el personal de campo debe realizar recorridos por el área de cultivo (hectáreas de cultivo) con el fin de recopilar información de la tierra para detectar problemas en la misma.

La información que suelen recoger sobre su área de cultivo (humedad, temperatura, intensidad del viento, entre otros) la extraen de forma manual, obteniéndola a través de una muestra de un área o algunas áreas, las cuales son anotadas para luego ser llevadas y analizadas por los expertos, quienes son los que toman las decisiones en base a la información obtenida, las mismas que pueden llegar a ser generalizadas para todo el cultivo y probablemente erróneas, que poco a poco traen consigo un impacto en su cultivo como en su economía.

1.2. JUSTIFICACIÓN

La agricultura es uno de los sectores relevantes en donde se desarrolla la economía del país, ya sea en el sector económico como en la seguridad alimentaria. De acuerdo a los informes de Productividad Agrícola del Ecuador se verifica que este sector aporta un promedio de 8.5% al PIB, obteniendo el sexto lugar de entre los sectores que aportan a la producción del país [2]; por

tal motivo la agricultura desempeña un rol fundamental ya que no sólo se trata de proporcionar alimentos sino también oportunidades de empleos, sin embargo tanto el empresario como agricultor necesita tierras productivas aún en una etapa difícil para el campo de producción, ya que cada vez las condiciones meteorológicas son más adversas y el agricultor debe procurarse en hacerle frente.

El uso de drones en la Agricultura como un servicio puede ser una oportunidad de innovación tecnológica, así como también un desafío en intervenir en la acción contra el cambio del sector agrícola, por lo que el sistema de monitoreo por medio de los drones con tecnología LoRa, ayudará en los múltiples procesos de la agricultura captando información constante y relevante para los especialistas, quienes evaluarán las condiciones del área monitoreada.

Realizar este monitoreo con los sensores integrados permitirá brindar información en tiempo real, determinando de forma temprana y eficiente el estado del cultivo permitiendo detectar las condiciones en las que se encuentra el área para prevenir plagas, maleza, y posibles efectos de daños climáticos.

Este monitoreo, será eficiente y de gran utilidad para el analista, productor y agricultor ya que tendrá un impacto tanto en el aspecto ambiental como económico, permitiendo tomar decisiones de manera oportuna y con mayor acierto.

1.3. OBJETIVOS

1.3.1. Objetivo General

- Diseñar un sistema de monitoreo agrícola en el Ecuador por medio de drones con tecnología LoRa.

1.3.2. Objetivos Específicos

- Implementar la transmisión de los datos de los sensores por medio de la tecnología LoRa.

- Incorporar sensores humedad, temperatura, precisión y altitud al sistema del dron para monitoreo del área de cultivo.
- Incorporar WebServices para visualización y monitoreo en tiempo real de la Data.
- Generar reportes de la data obtenida para uso de los expertos designados en el monitoreo del cultivo.

1.4. MARCO TEÓRICO

1.4.1. LPWAN

LPWAN es un protocolo de transporte inalámbrico el cual se lo utiliza en proyectos de IoT, siendo sus características más relevantes para estas implementaciones las siguientes:

- Su alcance ya que está diseñado para la transmisión de data inalámbrica-mente entre equipos, mismos que se encuentran distanciados en el rango de kilómetros.
- La capacidad de información transmitida, cuando se quiere regular el transporte no constante de cantidades pequeñas de datos.
- El bajo gasto eléctrico, con el fin de que las baterías de los equipos mantengan una duración de varios años en vez de semanas o meses.

Siendo estas las más relevantes, ya que permiten distinguir a LPWAN de otras tecnologías como se detalla a continuación: Wi-Fi, Bluetooth, 3GPP y Zigbee [3]. Figura 1.1.

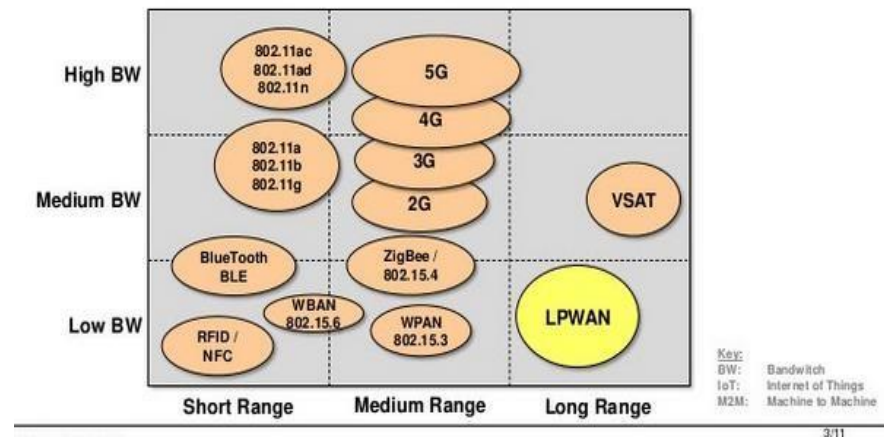


Figura 1.1. Espacio regido por tecnologías inalámbricas en términos de ancho de banda y alcance físico.

Existen diversas implementaciones en donde se usa protocolo LPWAN, de los cuales se puede mencionar a: Sigfox, NB-IoT, Weightless, RPMA y LoRaWAN, entre otras; pudiendo identificar variedades entre ellas como lo es su modulación, alcance, capacidad de información transmitida, encriptación y autenticación [3]. Figura 1.2.

	Weightless -N	Weightless -P	Sigfox	LoRaWAN	Ingenu RPMA
Frequency band	EU: 868 MHz US: 900 MHz	Sub-GHz ISM	EU: 868 MHz US: 902 MHz	EU: 433/868 MHz US: 915 MHz	2.4 GHz
Range (estimated)	3 km (urban)	2 km (urban)	3– 8 km (urban) 30–50 km (rural)	3–8 km (urban) 30–50 km (rural)	1–3 km (urban) 5–10 km (rural)
Link budget⁷			156 dB	156 dB	172 dB
Data throughput	10 bps – 10 kbps	200 bps – 100 kbps	EU: 100 bps US: 600 bps up to 140 messages per day	0.3 – 50 kbps adaptive	Uplink: 100 kbps Downlink: 600 kbps
Power consumption			Tx: < 50 mA Rx: 10-40 mA Sleep: <0.01 mA	Tx: < 50 mA Rx: 10–40 mA Sleep: <0.01 mA	Tx: 750 mA Rx: 300 mA Sleep: 0.072 mA
Packet size	Up to 20 bytes	Minimum of 10 bytes	12 bytes	Defined by user	6 bytes to 10 kB
Devices per access point	Unlimited	Unlimited	1 million	1 million	< 500,000
Security	AES-128 encryption	AES-128/256 encryption	No network encryption	AES CCM (128-bit) for encryption and authentication	128/256 bit encryption
Status	Early deployments	Under development	Being deployed	Being deployed	Being deployed

Figura 1.2. Tecnologías LPWAN: Características principales.

1.4.2. LoRa

LoRa es un tipo de modulación en radiofrecuencia patentado por Semtech y administrada por la “LoRa Alliance” destacando sus principales características como las siguientes:

- Alta tolerancia a las interferencias
- Alta sensibilidad para recibir datos (-168dB)
- Basado en modulación chirp
- Bajo Consumo (hasta 10 años con una batería*)
- Largo alcance 10 a 20km
- Baja transferencia de datos (hasta 255 bytes)
- Conexión punto a punto
- Frecuencias de trabajo: 915Mhz América, 868 Europa, 433 Asia

Características que logra realizar conexiones a grandes distancias, siendo de utilidad en las redes de IoT ya que se puedan utilizar en ciudades inteligentes, lugares que tengan poca cobertura celular o redes privadas de sensores o actuadores [4]. Figura 1.5.

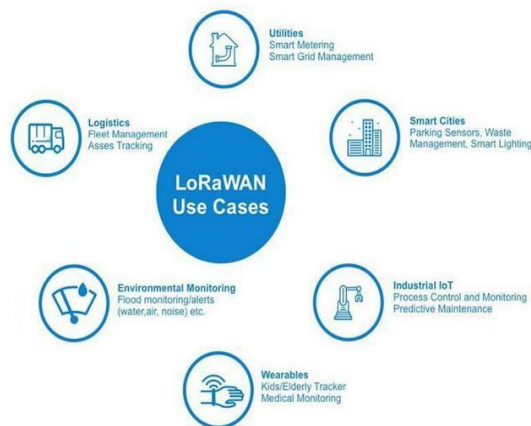


Figura 1.3. LoRaWAN: Casos de Uso

1.4.3. LoRaWAN

LoRaWAN es un protocolo de red que utiliza la tecnología LoRa el cual permite administrar y comunicar dispositivos LoRa, compuesto por: gateways y nodos; siendo los Gateway los encargados de enviar y recibir la información a los nodos, mientras que estos últimos son los encargados de enviar y recibir la información hacia el gateway.

Es una incorporación del protocolo LPWAN que maneja la tecnología LoRa (Figura 1.3), creador por LoRa Alliance en conjunto con Cisco, IBM y Orange, mismo que se la conceptúa como protocolo y arquitectura de red en la cual van a relacionarse directamente las aplicaciones [5].

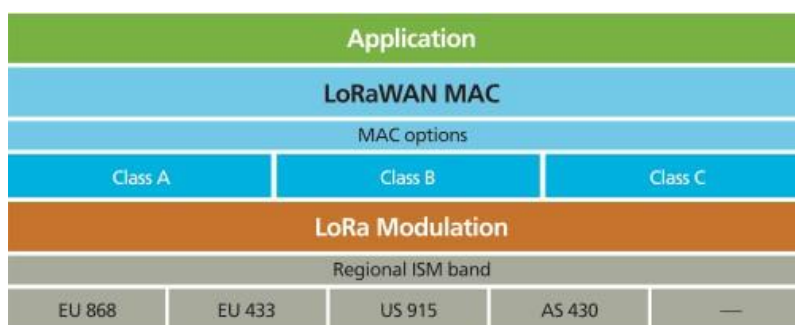


Figura 1.4. Arquitectura para LoRaWAN

Las características más importantes de LoRaWAN son:

- Topología estrella
- Alcance de 10 a 15km en línea de vista
- Encriptación AES 128
- Soporte para 3 clases de nodos
- Administración de dispositivos
- Redes públicas y privadas
- Bajo consumo y largo alcance
- Baja transferencia de datos (hasta 242 bytes)

LoRaWAN mantiene dos estructuras de red:

- Malla: en donde los dispositivos finales transmitan directamente sin requerir de un dispositivo en el medio, permitiendo una comunicación básica entre elementos.
- Estrella: los nodos finales envían sus mensajes a los Gateway, en la cual dicho mensaje puede ser captado por más de un Gateway. La red en donde intervienen dispositivos LoRa se comunicará de forma inalámbrica. Figura 1.4

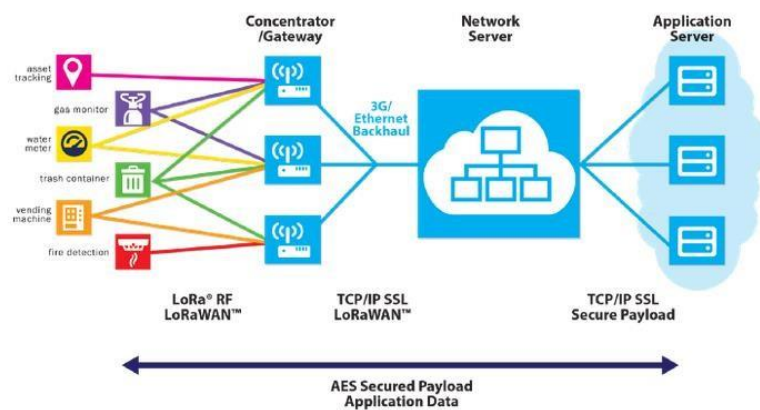


Figura 1.5. Estructura de red LoraWAN

CAPÍTULO 2

2. DISEÑO DEL SISTEMA DE MONITOREO CON TECNOLOGÍA LORA

2.1. DISPOSITIVOS UTILIZADOS EN EL SISTEMA DE MONITOREO

2.1.1. Pycom LoPy4

El LoPy4 es una placa programable por Micropython que funciona con LoRa, Sigfox, WiFi y Bluetooth. Figura 2.1.

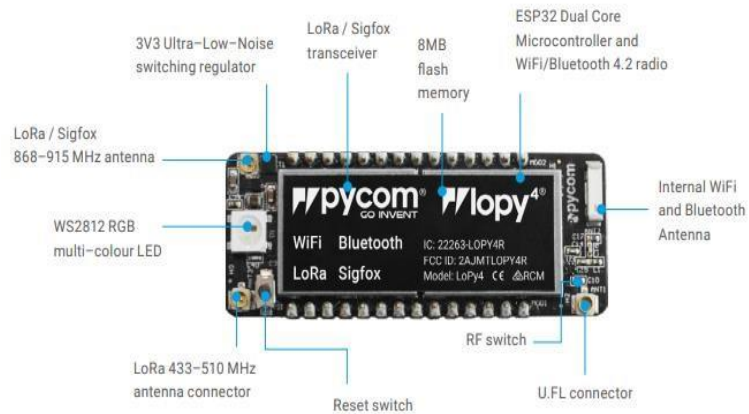


Figura 2.1. Lopy4 Board

Para empezar a hacer uso de la placa, se debe realizar la actualización del firmware a la última versión con la configuración de la banda de frecuencias para AU915. La versión inicial se observa en la Figura 2.2, la versión actual se muestra en la Figura 2.3.

```

Connecting to /dev/ttyACM0...
>>> ets Jun  8 2016 00:22:57

rst:0x1 (POWERON RESET),boot:0x16 (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:0x3fff0020,len:8
load:0x3fff0030,len:2156
ho 0 tail 12 room 4
load:0x4009fa00,len:19208
entry 0x400a05f4
Initializing filesystem as FatFS!
Pycom MicroPython 1.20.0.rc13 [v1.9.4-94bb382] on 2019-08-22; LoPy4 with ESP32
Type "help()" for more information.
>>> import os
>>> os.uname()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'on' is not defined
>>> os.uname()
(sysname='LoPy4', nodename='LoPy4', release='1.20.0.rc13', version='v1.9.4-94bb382 on 2019-08-22', machine='LoPy4 with ESP32', lorawan='1.0.2', sigfox='1.0.1')
>>>

```

Figura 2.2. Versión inicial del Lopy4: v1.9.4-94bb382 on 2019-0822

```

AutoConnect enabled, ignoring 'address' setting (see Global Settings)
Searching for PyCom boards on serial...
No PyCom boards found on USB
Connecting to /dev/ttyACM0...
Smart Provisioning started in the background
See https://docs.pycom.io/smart for details
Pycom MicroPython 1.20.3.b0 [v1.11-db33be7] on 2020-07-01; LoPy4 with ESP32
Pybytes Version: 1.5.0
Type "help()" for more information.
>>> import os
>>> os.uname()
(sysname='LoPy4', nodename='LoPy4', release='1.20.3.b0', version='v1.11-db33be7 on 2020-07-01', machine='LoPy4 with ESP32', lorawan='1.0.2', sigfox='1.0.1', pybytes='1.5.0')
>>>

```

Figura 2.3. Versión final del Lopy4: v1.11-db33be7 on 2020-07-01

Esta actualización es importante debido a que varias de las funciones necesarias en la programación no trabajaban de forma correcta con la versión inicial con la que contaba el Lopy4. Los comandos para obtener la versión del firmware son los siguientes [6].

```
>>> import os  
  
>>> os.uname()
```

El resultado de ejecutar los comandos anteriores se muestra a continuación:

```
(sysname='LoPy4', nodename='LoPy4', release='1.20.0.rc13',  
version='v1.9.4-94bb382 on 2019-08-22', machine='LoPy4 with  
ESP32', lorawan='1.0.2', sigfox='1.0.1')
```

Los resultados del software empleado para realizar la actualización del firmware se muestran en la Figura 2.4 y 2.5 a continuación.

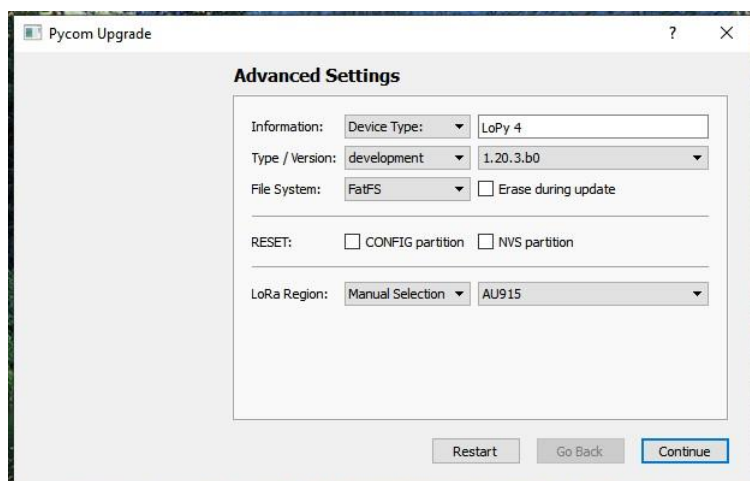
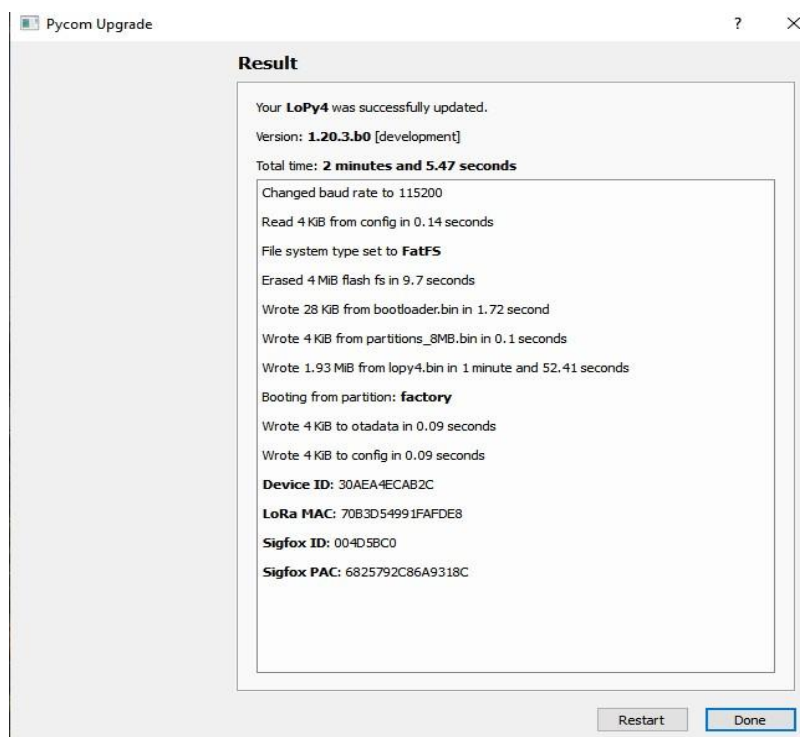


Figura 2.4. Configuración de los parámetros del firmware.**Figura 2.5. Resultados del software empleado para actualizar el firmware.**

Luego de varias pruebas usando la configuración anterior, se notaron problemas de pérdida constantes de paquetes, después de investigar se llegó a la conclusión de que la configuración para el firmware en la banda AU915 contiene errores. Por lo tanto, se tuvo que volver a grabar el firmware con la configuración US915 [7]. Esto se muestra en las Figuras 2.6 y 2.7.

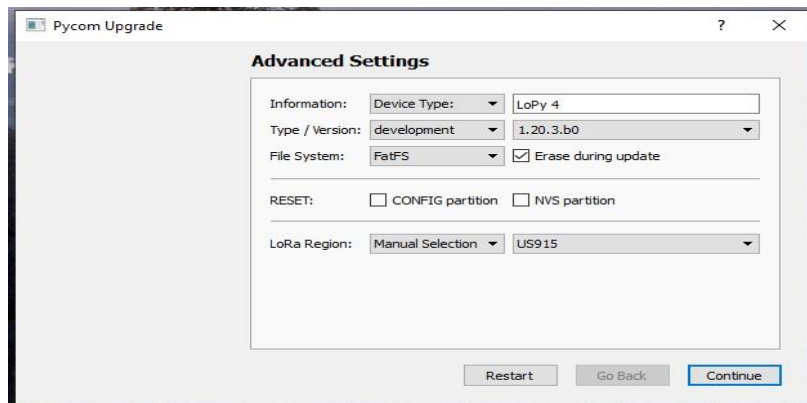


Figura 2.6. Configuración de los parámetros del firmware.

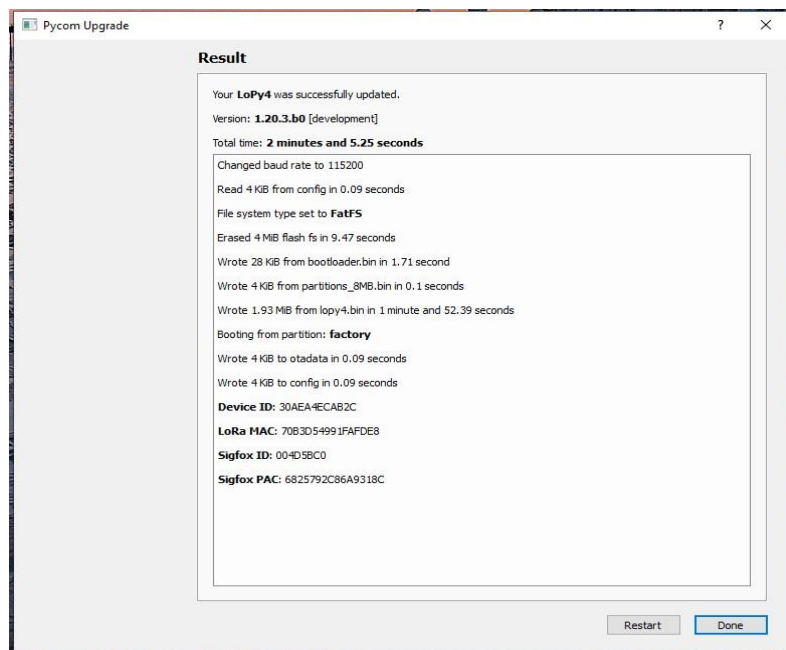


Figura 2.7. Resultados del software empleado para actualizar el firmware.

2.1.2. Pysense

Shield de sensores que se puede usar con cualquiera de sus módulos de red múltiple de Pycom y es compatible con LoPy4 [8]. Figura 2.8.

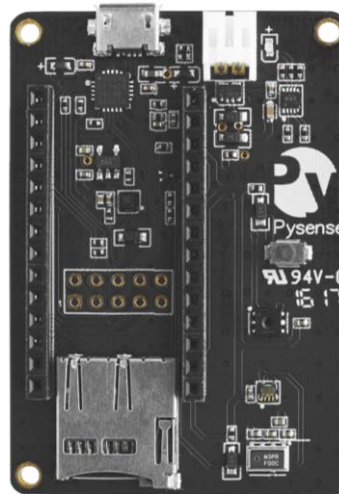


Figura 2.8. Pysense

Las características del shield pysense se detallan a continuación:

- Sensor de luz ambiental
- Sensor de presión barométrica
- Sensor de humedad
- Acelerómetro de 3 ejes y 12 bits
- Sensor de temperatura
- Puerto USB con acceso serial
- Cargador de batería LiPo
- Compatibilidad con tarjetas MicroSD
- Dimensiones: 55x35x10mm
- Peso: 11g

A continuación, se muestra el Pinout del shield pysense. Figura 2.9.

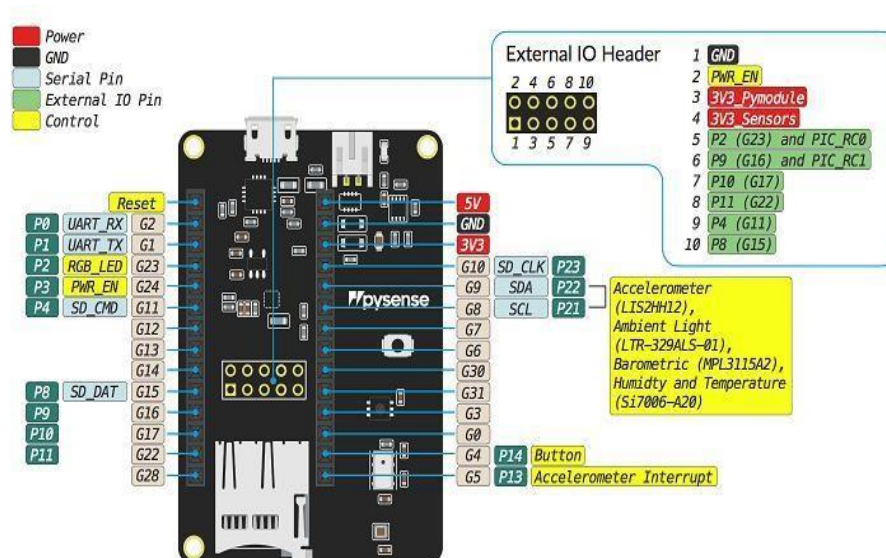


Figura 2.9. Pinout del Pysense

2.1.3. Gateway LoRaWAN

Un gateway es un dispositivo que recibe los paquetes de datos enviados por los nodos sensores a través de tecnología LoRa y los transmite vía internet a un servidor [9]. Figura 2.10.

Entre las especificaciones que se encuentran sobre el equipo Gateway son:

- Rango de hasta 10 km.
- Permite conexión vía Ethernet o Wifi.
- Permitir la conexión Bluetooth 4.2 orientado a nodos IoT
- Seguridad por medio de https y mantiene protocolo LoRaWAN.
- Hardware y software Open Source.
- Permite la integración hasta a 10.000 nodos.



Figura 2.10. Gateway LoRaWAN

2.1.4. Drone

La RAE lo define como “aeronave no tripulada”, es decir funciona por control remoto. También puede aparecer escrito como RPA (Remotely Piloted Aircraft). Figura 2.11.

Dentro del sector agrícola, el drone desempeña un gran número de tareas como:

- Inspección y monitoreo de instalaciones y obras de infraestructura.
- Investigaciones atmosféricas.
- Topografía y cartografía temática.
- Geología y prospección petrolífera y gasífera.
- Gestión de riesgos y desastres naturales (incendios, inundaciones, etc.).
- Exploración de lugares de difícil acceso, salvamento y rescate.
- Cinematografía y fotografía comercial, artística y/o deportiva.
- Control medioambiental.
- Limnología y oceanografía.

- Investigaciones sobre conservación de la biodiversidad.
- Medios de comunicación y entretenimiento.
- Movilidad, tráfico y logística en general.
- Actividades agrícolas y pecuaria.
- Aplicación de productos fitosanitarios.



Figura 2.11. Drone.

2.2. DIAGRAMA DE LA SOLUCIÓN

El Diseño de la solución consta de la integración de una placa de desarrollo, pycom lopy4, con un shield de expansión pycom pysense, la cual contiene sensores de temperatura, humedad, presión, altitud y el voltaje de la misma. Se probaron los sensores del módulo Pysense, cuyos resultados se muestran en la Figura 2.12 a continuación.

```
rst:0x1 (POWERON_RESET),boot:0x16 (SPI_FAST_FLASH_BOOT)
config: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:0x3fff0028,len:8
load:0x3fff0030,len:1992
load:0x40078000,len:12104
load:0x40080400,len:5032
entry 0x4008060c
Smart Provisioning started in the background
See https://docs.pycom.io/smart for details
MPL3115A2 temperature: 31.6875
Altitude: -24.6875
Pressure: 101622.7
Temperature: 32.82675 deg C and Relative Humidity: 51.59811 %RH
Dew point: 21.52392 deg C
Humidity Ambient for 24.4 deg C is 83.88988%RH
Light (channel Blue lux, channel Red lux): (0, 0)
Acceleration: (0.0, 0.0, 0.0)
Roll: 2.730628
Pitch: -2.950377
Battery voltage: 4.711792
Previous board is not available anymore
> Failed to connect (Error: Port is not open). Click here to try again.
```

Figura 2.12. Prueba de los sensores.

El LoPy4 con el shield expansor pysense, se integrará de forma física al dron para que este pueda tomar datos en diferentes altitudes y ubicaciones. Figura 2.13.

El dron que se puede emplear para esta aplicación es el AeroDron MR4 (Figura 2.11), con 4 motores, el cual puede transportar 0.27 Kg durante 20 minutos.

El aeroDron MR4 posee las siguientes características:

- Tiempo de vuelo teórico: hasta 23.8 min. ·
Provee comunicaciones confiables hasta 5 Km · Es open hardware.
- Fácil de añadir elementos adicionales. · Usa Ardupilot como software de vuelo.
- Controlador de vuelo Pixhawk
- Retención de posición dinámica.
- Retención de altitud dinámica.
- Despegue y aterrizaje automático.
- Navegación de waypoints.

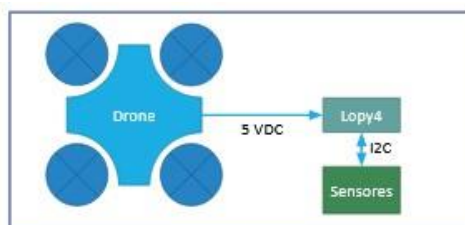


Figura 2.13. Integración dron con Lopy4 y pysense

El LoPy4 es una placa programable por Micropython que funciona con LoRa, que al integrarse con el shield de expansión pysense, se programa para que pueda registrarse a la plataforma de The Things Network como una aplicación en la misma indicando su Device EUI, Application EUI, App Session Key y Network Session Key [10]. Figura 2.14.

Device EUI	<>	70 B3 D5 49 91 FA FD E8	🗑️
Application EUI	<>	70 B3 D5 7E D0 03 34 74	🗑️
Device Address	<>	26 02 16 28	🗑️
Network Session Key	<>	🗑️
App Session Key	<>	🗑️

Figura 2.14. Parámetros de Registro de la Aplicación en TTN

Por otro lado, el Gateway LoRaWAN, se registra como Gateway en la plataforma TTN con frecuencia US915. Figura 2.15.

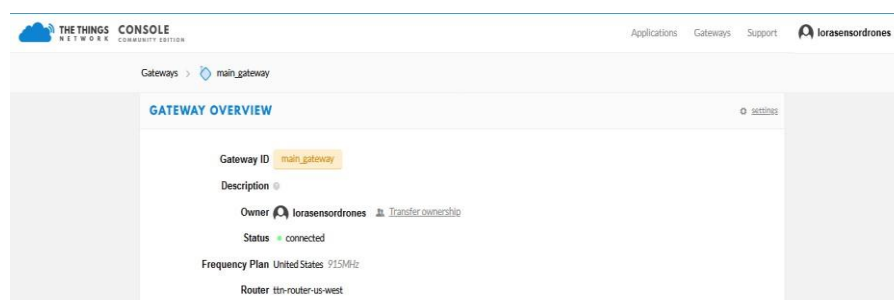


Figura 2.15. Registro del Gateway en TTN

Durante el registro de la aplicación en la red TTN se probaron dos métodos: OTAA y ABP. Al iniciar la prueba con OTAA, el dispositivo envía los joinrequest y estos son recibidos por el servidor en la red de TTN, el cual inmediatamente genera el join-accept; sin embargo, el dispositivo no es capaz de recibir este mensaje por lo cual no se completa el proceso de registro en la red. Esto se muestra en las Figuras 2.16 y 2.17.

time	frequency	mod.	CR	data rate	airtime (ms)	cnt	
▲ 19:57:33	917.6	lorawan	4/5	SF 12 BW 125	1318.9	2814	dev addr: 26 02 25 D5 payload size: 19 bytes
★ 19:52:16	923.3	lorawan	4/5	SF 12 BW 500	288.8		
★ 19:52:11	917.2	lorawan	4/5	SF 10 BW 125	370.7		app eui: 70 B3D57ED003 34 74 dev eui: 70 B3D5 49 91 FAFDE8
★ 19:49:10	923.3	lorawan	4/5	SF 12 BW 500	288.8		
★ 19:49:05	917.2	lorawan	4/5	SF 10 BW 125	370.7		app eui: 70 B3D57ED003 34 74 dev eui: 70 B3D5 49 91 FAFDE8
★ 19:47:41	923.3	lorawan	4/5	SF 12 BW 500	288.8		
★ 19:47:37	917.8	lorawan	4/5	SF 10 BW 125	370.7		app eui: 70 B3D57ED003 34 74 dev eui: 70 B3D5 49 91 FAFDE8
★ 19:47:36	917.4	lorawan	4/5	SF 10 BW 125	370.7		app eui: 70 B3D57ED003 34 74 dev eui: 70 B3D5 49 91 FAFDE8
▲ 19:39:32	918	lorawan	4/5	SF 12 BW 125	1318.9	2796	dev addr: 26 02 25 D5 payload size: 19 bytes
▲ 19:32:31	917.4	lorawan	4/5	SF 12 BW 125	1318.9	2789	dev addr: 26 02 25 D5 payload size: 19 bytes

Figura 2.16. Flujo de mensajes recibidos por el gateway.

En el costado izquierdo de la Figura 2.16 se puede observar los rayos de color amarillo y verde, donde el color amarillo indica que llegó un joinrequest y el color verde indica que la red envió un join-accept al dispositivo. Los triángulos azules son mensajes de uplink de un dispositivo usado para verificar que todo esté funcionando correctamente.

time	counter	port	
★ 19:52:11			dev addr: 26 02 2E 49 app eui: 70 B3D57ED003 34 74 dev eui: 70 B3D5 49 91 FAFDE8
★ 19:49:05			dev addr: 26 02 23 A4 app eui: 70 B3D57ED003 34 74 dev eui: 70 B3D5 49 91 FAFDE8
★ 19:47:36			dev addr: 26 02 2A 6B app eui: 70 B3D57ED003 34 74 dev eui: 70 B3D5 49 91 FAFDE8

Figura 2.17. Flujo de datos en la aplicación del dispositivo.

Después de realizar una investigación se logró determinar que el problema con el método de registro OTAA se debe a un problema con el firmware del dispositivo. Por lo tanto, esto no va a estar disponible hasta que el desarrollador arregle este fallo. [7]

Se procedió a probar con el método ABP, siendo este el actual método de registro con el que se encuentra funcionando el dispositivo. Figura 2.18.

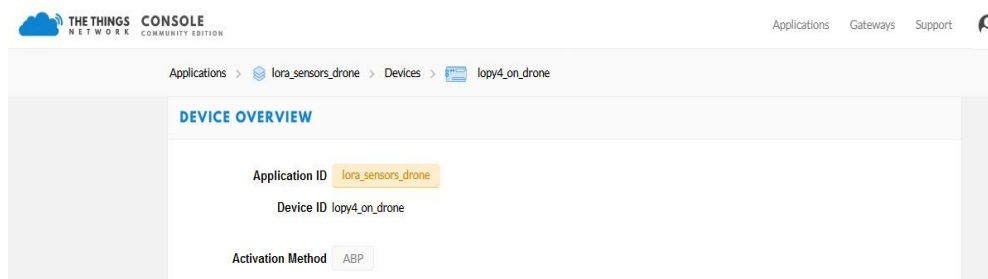


Figura 2.18. Método de Registro ABP en TTN

A pesar de que el registro es exitoso existe un problema asociado al firmware del dispositivo ya que cada vez que el dispositivo es desconectado de la fuente de energía, el contador de paquetes de subida se reinicia a 0, debido a esto, se dejan de recibir mensajes del dispositivo en el servidor de aplicación de TTN. Esto se debe a que TTN también lleva la cuenta de paquetes y al ser diferentes los números, todos los paquetes inferiores al conteo que se encuentra en la plataforma se descartan [11]. Esto se puede observar en la Figura 2.19.

Frames up 40 [reset frame counters](#)

Figura 2.19. Contador de paquetes de subida del dispositivo en TTN.

Para solucionar este problema antes de encender el dispositivo se debe dar clic en “*reset frame counters*” y aceptar lo que aparece en el cuadro de diálogo. Una vez hecho lo anterior se procede a conectar el dispositivo a la fuente de energía, como se muestra en la Figura 2.20.



Figura 2.20. Reset Frame Counters.

Una vez registrada la placa como el gateway, se realiza la programación en VisualStudio para que que la data sensada por el pysense sea enviada al

Gateway LoRaWAN, la cual irá codificada en bytes, como se muestra en la Figura 2.39.

En la plataforma TTN, se recibe la información codificada, por lo que se requiere decodificarla para obtener los datos sensados (Figura 2.43). La información decodificada es enviada a la plataforma de ThingSpeak mediante una integración que se configuró en la plataforma de TTN indicando su Channel ID. Figura 2.21 y 2.22.

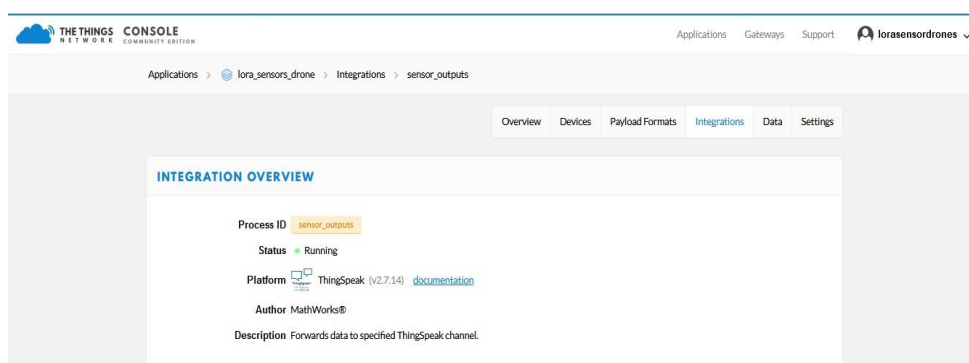


Figura 2.21. Intefración de Thingspeak



Figura 2.22. Parámetros de Thingspeak

En Thingspeak, se configura el canal por el cual será mostrada la data, indicando sus campos de monitoreo llamados Field 1 – 5, y estos a su vez son los que se podrás monitorear para uso del especialista. Figura 2.23.

Channel Settings

Percentage complete	50%	
Channel ID	1116740	
Name	<input type="text" value="Parámetros ambientales"/>	
Description	<input type="text" value="En esta aplicación se muestran las mediciones de parámetros ambientales registrados por sensores ubicados en drones."/>	
Field 1	<input type="text" value="Temperatura [°C]"/>	<input checked="" type="checkbox"/>
Field 2	<input type="text" value="Humedad [%]"/>	<input checked="" type="checkbox"/>
Field 3	<input type="text" value="Altitud [m]"/>	<input checked="" type="checkbox"/>
Field 4	<input type="text" value="Presión [Pa]"/>	<input checked="" type="checkbox"/>
Field 5	<input type="text" value="Voltaje de batería [V]"/>	<input checked="" type="checkbox"/>

Figura 2.23. Configuraciones del Canal.

A continuación, se muestran los Diagramas del Diseño de la solución del proyecto, el cual servirá para monitoreo de la agricultura en el Ecuador. En la Figura 2.24, se detalla por diagramas de bloques la solución, en donde los sensores estarán conectados al board lopy4 y a su vez adjunto a un dispositivo móvil (drone), mismo que utilizarán la tecnología LoRa con frecuencia US915 para conectarse al Gateway LoRaWAN. Una vez recolectado los datos en el Gateway, estos son enviados al server TTN que contiene la integración con Thingspeak donde se tendrá almacenada la data para poder ser visualizada y monitoreada por el usuario Final. En la Figura 2.25, se muestra las imágenes de cada dispositivo de la solución.

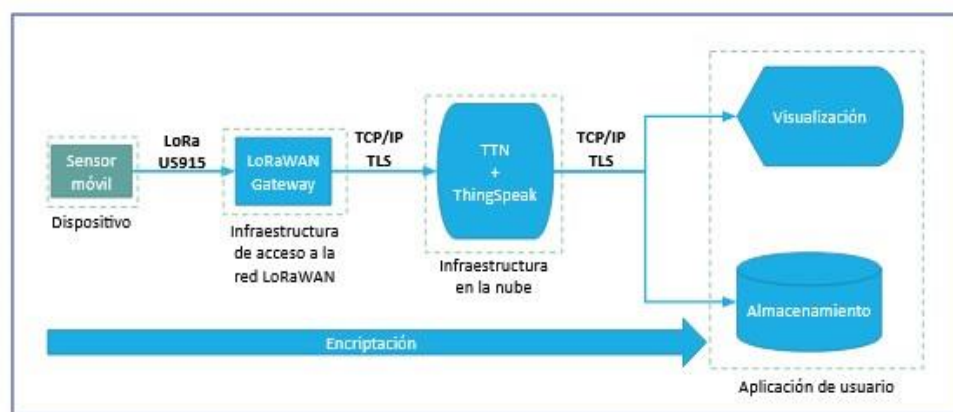


Figura 2.24. Diagrama de Bloques de la Solución

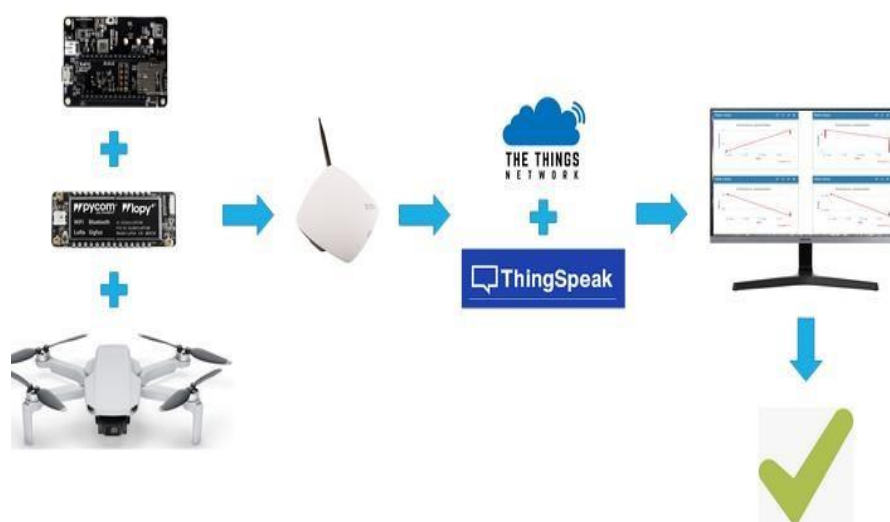


Figura 2.25. Diagrama de la Solución

2.3. DIAGRAMA DE FLUJO DEL SOFTWARE

El software se encuentra dividido en siete secciones que se encargan del funcionamiento del sistema. En la Figura 2.26 se muestra el diagrama de flujo en el cual se muestra cada sección programada en el microcontrolador Lopy4: inicio, importación de librerías, configuración inicial, su instanciación de variables, su forma de registro del lopy4 al Server TTN, configuración del socket, adquisición de datos, Codificación y Transmisión.

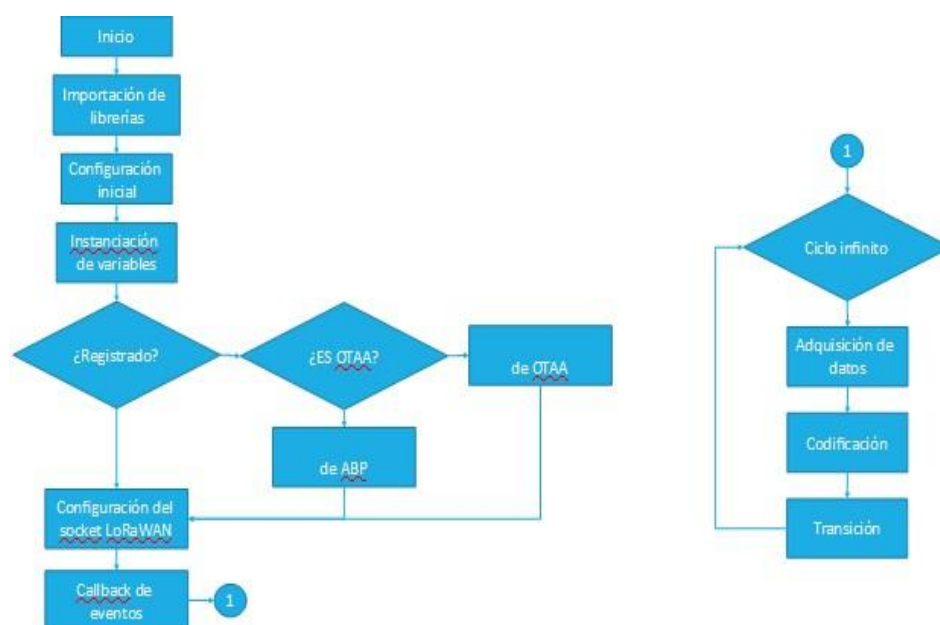


Figura 2.26. Diagrama de Flujo

En la **Tabla 1** se muestra el significado de los colores del LED que se ha utilizada para cada sección:

Color del LED	Descripción
Blanco	Inicialización del sistema
Amarillo	Se está realizando el proceso de JOIN
Verde brillante	Configuración inicial completa.
Naranja	Creación del socket
Azul	Inicio de la transmisión

Verde	Transmisión finalizada
-------	------------------------

Tabla 1. Significado de los colores del LED.

2.3.1. Importación de librerías

Se realiza la importación de las librerías necesarias para el desarrollo de las funciones utilizadas en la programación del proyecto. Figura 2.27.

```

#-----
#                               Librerías
#-----
import time
import pycom
from pysense import Pysense
from network import LoRa
import machine
import math
import socket
import ubinascii
import binascii
import struct

from LIS2HH12 import LIS2HH12
from SI7006A20 import SI7006A20
from LTR329ALS01 import LTR329ALS01
from MPL3115A2 import MPL3115A2, ALTITUDE, PRESSURE

```

Figura 2.27. Librerías

2.3.2. Configuración Inicial

En esta sección se realiza la configuración inicial del Lopy4, se desactivan elementos que no son necesarios para la aplicación. Figuras 2.28 y 2.29.

```

#-----
#                               Configuración inicial
#-----
pycom.wifi_on_boot(False) # deshabilita el wifi
pycom.heartbeat(False) # deshabilita el heratbeat
pycom.rgbled(0x0A0A08) # LED color blanco
print("LoRa sensor drone v1.00")

```

Figura 2.28. Configuración Inicial

Se visualiza en la terminal de Visual Studio la programación que corresponde a la configuración Inicial (Figura 2.29), y a su vez se observa el led blanco encendido en el Lopy4 (Figura 2.30).

```
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd
mode:DIO, clock div:1
load:0x3fff0028,len:8
load:0x3fff0030,len:1992
load:0x40080400,len:5032
entry 0x4008060c
LoRa sensor drone v1.00
```

Figura 2.29. Terminal de Visual Studio Code: Configuración Inicial

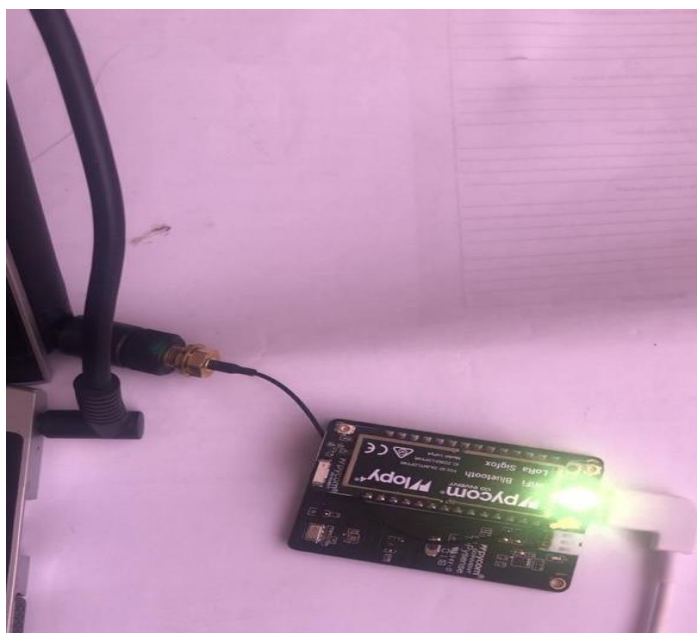


Figura 2.30. Configuración Inicial: Led Blanco encendido

2.3.3. Instanciación de variables

En esta sección se crean e instancian variables empleadas a lo largo del desarrollo del software, como la frecuencia, separación de frecuencia, Device address, network address, app address, App EUI, App Key.

```

print("Creando módulos necesarios...")
py = Pysense() # modulo pysense

start_frequency = 903900000 # Canal de frecuencia con el que inicia la banda
step_frequency = 200000 # Se paración entre canales

# DevEUI 70B3D54991FAFDE8
# Parámetros para autenticación ABP
dev_addr = struct.unpack(">1", ubinascii.unhexlify('26021628'))[0] # Dirección del dispositivo
nwk_swkey = ubinascii.unhexlify('D190F751A69C9FCD04746E3D88FEC206D') # Clave de sesión de la red
app_swkey = ubinascii.unhexlify('4F728F076630D6GAD3B7E079AC535CDB') # Clave de aplicación de la red
ABP = True

# Parametros para autenticación OTAA
app_eui = ubinascii.unhexlify('70B3D57ED0083474') # EUI de la aplicación
app_key = ubinascii.unhexlify('AE5FDE58FB9D3C884EDB14754C393CE0') # Clave de la aplicación
OTAA = False

LORA_NODE_DR = 3 # Data rate
lora = LoRa(mode=LoRa.LORAWAN, public=True, adr=False, region=LoRa.US915, tx_retries=4, power_mode=LoRa.ALWAYS_ON, device_class=LoRa.CLASS_A)
lora.nvram_restore() # Carga de la memoria la configuración lora previa

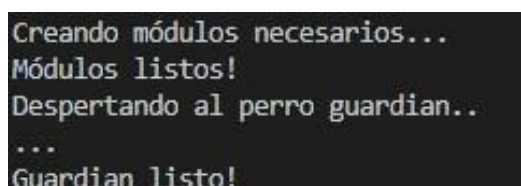
print("Módulos listos!")

print("Despertando al perro guardian..")
print("...")
# Start WDT
wdt = machine.WDT(timeout=300000) # watchdog
print("Guardian listo!")

```

Figura 2.31. Instanciación de variables

Se visualiza en la terminal de Visual Studio el mensaje de Guardián Listo luego de haber instanciado las variables. Figura 2.32.



```

Creando módulos necesarios...
Módulos listos!
Despertando al perro guardian..
...
Guardian listo!

```

Figura 2.32. Terminal de Visual Studio Code: Instancias creadas.

2.3.4. Conexión al Servidor LoRaWAN

En esta sección se realiza el proceso de registro en la red LoRaWAN, el método de activación por defecto es ABP. Para cambiar el tipo de autenticación se debe intercambiar el contenido de las variables “ABP” y “OTAA” de la sección 3, por “False” y “True” correspondientemente. Figura 2.33.


```

if lora.has_joined(): # Verifica si el dispositivo ya se encuentra registrado en la red
    print("Dispositivo ya se encuentra registrado en una red.")
    pycom.rgbled(0x00FF00) # LED color verde
else:
    pycom.rgbled(0xFFFF00) # LED color amarillo mientras no ha realizado join
    if OTAA:
        print("Iniciando en proceso de registro mediante activación OTAA...")
        lora.nvram_erase() # Borra de memoria las configuraciones LoRa previas
        # Elimina todos los canales por defecto
        for channel in range(0, 72):
            lora.remove_channel(channel)
        # Configura la banda de canales de frecuencia para US 915 en TTN USA
        for channel in range(8, 16):
            if channel <= 15:
                channel_frequency = start_frequency + (channel - 8)*step_frequency
                channel_dr_min = 0
                channel_dr_max = 3
            else:
                channel_frequency = 904600000
                channel_dr_min = 12
                channel_dr_max = 12
        lora.add_channel(channel, frequency=channel_frequency, dr_min=channel_dr_min, dr_max=channel_dr_max)
        #print("Canal añadido: ", channel, channel_frequency, channel_dr_min, channel_dr_max)
        lora.join(activation=LoRa.OTAA, auth=(app_eui, app_key), timeout=0, dr=0) # Se registra a la red usando OT
        pycom.rgbled(0xFFFF00) # LED color amarillo
        while not lora.has_joined(): # Espera hasta que el dispositivo se haya registrado en la red
            time.sleep(2.5)
            wdt.feed() # Alimenta al WDT para evitar que el dispositivo se reinicie.
            pass
        loraStatus = str(lora.has_joined()) # Confirmación del estado
        print("Dispositivo registrado satisfactoriamente: " + loraStatus)
        lora.nvram_save() # Guarda las configuraciones LoRa en la memoria
        wdt.feed() # Alimenta al WDT para evitar que el dispositivo se reinicie.
        pycom.rgbled(0x00FF00) # LED color verde
    else:
        print("Iniciando en proceso de registro mediante activación ABP...")
        lora.nvram_erase() # Borra de memoria las configuraciones LoRa previas
        # Elimina todos los canales por defecto
        for channel in range(0, 72):
            lora.remove_channel(channel)
        # Configura la banda de canales de frecuencia para US 915 en TTN USA
        for channel in range(8, 16):
            if channel <= 15:
                channel_frequency = start_frequency + (channel - 8)*step_frequency
                channel_dr_min = 0
                channel_dr_max = 3
            else:
                channel_frequency = 904600000
                channel_dr_min = 12
                channel_dr_max = 12
        lora.add_channel(channel, frequency=channel_frequency, dr_min=channel_dr_min, dr_max=channel_dr_max)
        #print("Canal añadido: ", channel, channel_frequency, channel_dr_min, channel_dr_max)
        lora.join(activation=LoRa.ABP, auth=(dev_addr, nwk_swkey, app_swkey)) # Se registra a la red usando ABP (Activ:

```

Figura 2.33. Conexión al Servidor

Se muestra el mensaje de conexión a la red LoRaWAN en la terminal Consola de Visual Studio Code. Figura 2.34.

```

Conectandose a la red LoRaWAN...
Iniciando en proceso de registro mediante activación ABP...
Lora joined: True
Dispositivo registrado satisfactoriamente: True

```

Figura 2.34. Terminal de Visual Studio Code: Conexión a TTN.

2.3.5. Configuración del socket LoRaWAN

En esta sección se configura el socket para hacer uso de las comunicaciones LoRaWAN. Figura 2.35.

```

#-----
#                               Configuración del socket LoRaWAN
#-----
pycom.rgbled(0xEC7C26) # LED color naranja
time.sleep(1) # Se espera 1 segundo
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW) # Crea el socket LoRa
s.setsockopt(socket.SOL_LORA, socket.SO_DR, LORA_NODE_DR) # Configura las opciones del socket
s.bind(1) # Número del puerto de subida
s.setblocking(False) # Hace el socket non-blocking
pycom.rgbled(0x000000) # LED apagado

```

Figura 2.35. Configuración del socket LoRaWAN

2.3.6. Eventos LoRa

En esta sección se registran los eventos de transmisión y recepción de datos. Es decir, en cada ocasión que ocurra uno de estos eventos, se mostrará un mensaje por la consola. Figura 2.36.

```

#-----
#                               Eventos LoRa
#-----
def lora_cb(lora):
    events = lora.events()
    if events & LoRa.RX_PACKET_EVENT:
        print('Lora packet received')
        wdt.feed() # Alimenta al WDT para evitar que el dispositivo se reinicie.
    if events & LoRa.TX_PACKET_EVENT:
        print('Lora packet sent')
        wdt.feed() # Alimenta al WDT para evitar que el dispositivo se reinicie.

lora.callback(trigger=(LoRa.RX_PACKET_EVENT | LoRa.TX_PACKET_EVENT), handler=lora_cb) #

```

Figura 2.36. Eventos de LoRa

En la Figura 2.38, se observa el mensaje de “LoRa packet sent”, indicando que los paquetes se están enviando.

```

Lora packet sent
Lora packet sent
Lora packet sent
Lora packet sent

```

Figura 2.38. Terminal de Visual Studio Code: Mensaje de Envío del paquete.

2.3.7. Adquisición de datos y transmisión

En esta sección se realiza la adquisición de datos de los sensores y se realiza su posterior transmisión al servidor de red LoRaWAN TTN. Se realizan dos transmisiones debido a la capacidad del payload, por el puerto 1 se envían los datos de temperatura, humedad, altitud y presión. Por puerto 2 se envía el voltaje de la batería. Figura 2.39.

```
# Adquisición de datos y transmisión
#-----
while True: # Ciclo infinito de sensado y transmisión.
    mp = MPL3115A2(py,mode=ALTITUDE) # Retorna la altura en metros.
    mpp = MPL3115A2(py,mode=PRESSURE) # Retorna la altura en Pa.
    si = SI7006A20(py) # Retorna la temperatura en grados C y la humedad relativa en %.
    #lt = LTR329ALS01(py) # Retorna la intensidad de luz en los canales rojo y azul.
    ### Inicio del proceso de codificación
    # Procesamiento de datos, se convierte a enteros los valores decimales obtenidos.
    temperature = int(round(si.temperature(), 2) * 100)
    humidity = int(round(si.humidity(), 2) * 100)
    altitude = int(round(mp.altitude(), 2) * 100)
    pressure = int(round(mpp.pressure(), 2) * 100)
    battery = int(round(py.read_battery_voltage(), 2) * 100) # Se obtiene el voltaje de la fuente de energía.
    # Se convierte a hexadecimal los valores enteros
    temperatureHex = hex(temperature)
    humidityHex = hex(humidity)
    altitudeHex = hex(altitude)
    pressureHex = hex(pressure)
    batteryHex = hex(battery)
    # Se convierten a bytes los valores obtenidos
    temperatureHex = temperatureHex.to_bytes(2, 'big')
    humidityHEX = humidityHex.to_bytes(2, 'big')
    altitudeHEX = altitudeHex.to_bytes(2, 'big')
    pressureHEX = pressureHex.to_bytes(4, 'big')
    batteryHEX = batteryHex.to_bytes(2, 'big')
    ### Fin del proceso de codificación
    pycom.rgbled(0x0100F0) # LED color azul.
    s.setblocking(True) # Hace el socket blocking.
    s.bind(1) # Se envia al puerto 1 los datos de temperatura, humedad, altitud y presión.
    s.send(bytes(temperatureHex+humidityHEX+altitudeHEX+pressureHEX)) # Primer mensaje codificado.
    time.sleep(3) # Se esperan 3 segundos antes de realizar la siguiente transmisión.
    s.bind(2) # Se envia al puerto 2 el voltaje de la batería.
    s.send(bytes(batteryHEX)) # Segundo mensaje codificado
    s.setblocking(False) # Hace el socket non-blocking.
    pycom.rgbled(0x00FF00) # LED color verde.
    lora.nvram_save() # Se guarda el estado actual de los parametros LoRaWAN en la memoria.
    time.sleep(10) # Se esperan 10 segundos antes de repetir el ciclo.
    wdt.feed() # Alimenta al WDT para evitar que el dispositivo se reinicie.
```

Figura 2.39. Adquisición de datos y transmisión

En la Figura 2.40, se visualiza el led azul encendido, que de acuerdo a la programación corresponde a la adquisición de datos, mientras que en la Figura 2.41 se observa el led verde correspondiendo a la transmisión de datos.



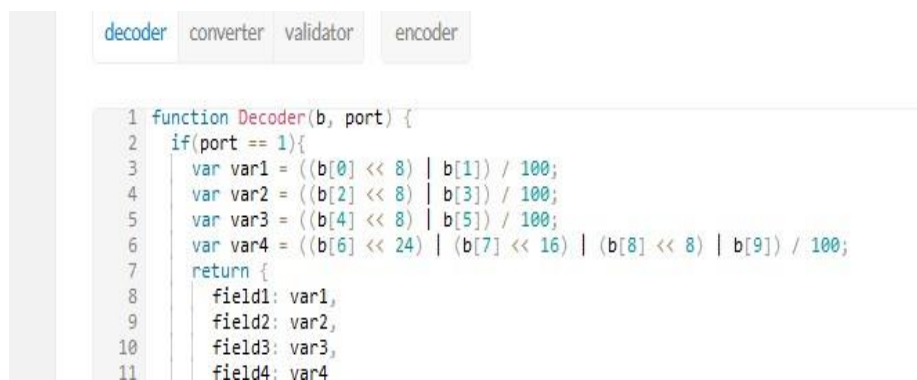
Figura 2.41. Adquisición de datos: Led Azul encendido



Figura 2.42. Transmisión de datos: Led Verde encendido

2.3.8. Decodificación

La función decodificadora se encarga de transformar los bytes recibidos en la plataforma de TTN a información útil. Esta función recibe el payload y el puerto. En base al puerto por el cual llegó el payload se realiza un tratamiento diferente de los bytes para así poder obtener la información. La programación empleada se encuentra a continuación. Figura 2.43.



```
1 function Decoder(b, port) {
2   if(port == 1){
3     var var1 = ((b[0] << 8) | b[1]) / 100;
4     var var2 = ((b[2] << 8) | b[3]) / 100;
5     var var3 = ((b[4] << 8) | b[5]) / 100;
6     var var4 = ((b[6] << 24) | (b[7] << 16) | (b[8] << 8) | b[9]) / 100;
7     return {
8       field1: var1,
9       field2: var2,
10      field3: var3,
11      field4: var4
12    };
13  }
14 }
```

Figura 2.43. Decodificación

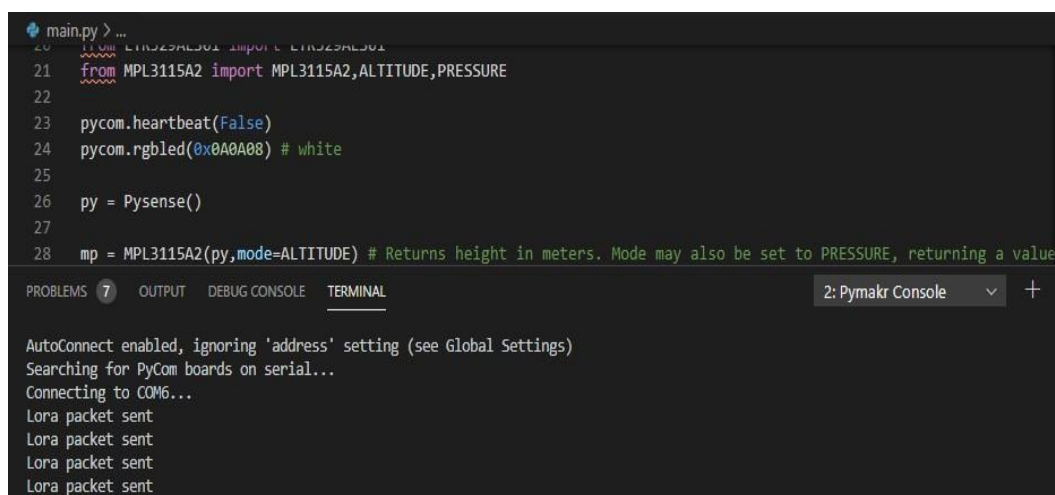
CAPÍTULO 3

3. ANÁLISIS DE RESULTADOS

En la plataforma TTN, dentro de la sección de Applications se puede visualizar el payload, que es la data codificada, misma que fue enviada por la placa lopy4 (Figura 3.1) utilizando la tecnología LoRa. Este payload luego es decodificado en la sección Payload Formats (Figura 3.2), permitiendo obtener los datos reales de cada sensor en variables definidas como field 1 – 4. Figura 3.3.

Junto con el payload recibido se muestran los siguientes parámetros: hora, contador de la información recibida, como también el puerto por donde recibe dicha información.

En el puerto 1, se recibe la información de los sensores de: Temperatura, Humeda, Altura y Presión; mientras que en el puerto 2, se recibe la información del Voltaje de la Placa.



```
main.py > ...
20 from LRU29AL301 import LRU29AL301
21 from MPL3115A2 import MPL3115A2, ALTITUDE, PRESSURE
22
23 pycom.heartbeat(False)
24 pycom.rgblcd(0x0A0A08) # white
25
26 py = Pysense()
27
28 mp = MPL3115A2(py, mode=ALTITUDE) # Returns height in meters. Mode may also be set to PRESSURE, returning a value

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL 2: Pymakr Console +
AutoConnect enabled, ignoring 'address' setting (see Global Settings)
Searching for PyCom boards on serial...
Connecting to COM6...
Lora packet sent
Lora packet sent
Lora packet sent
Lora packet sent
```

Figura 3.1. Visual Studio Code: Packets sent.

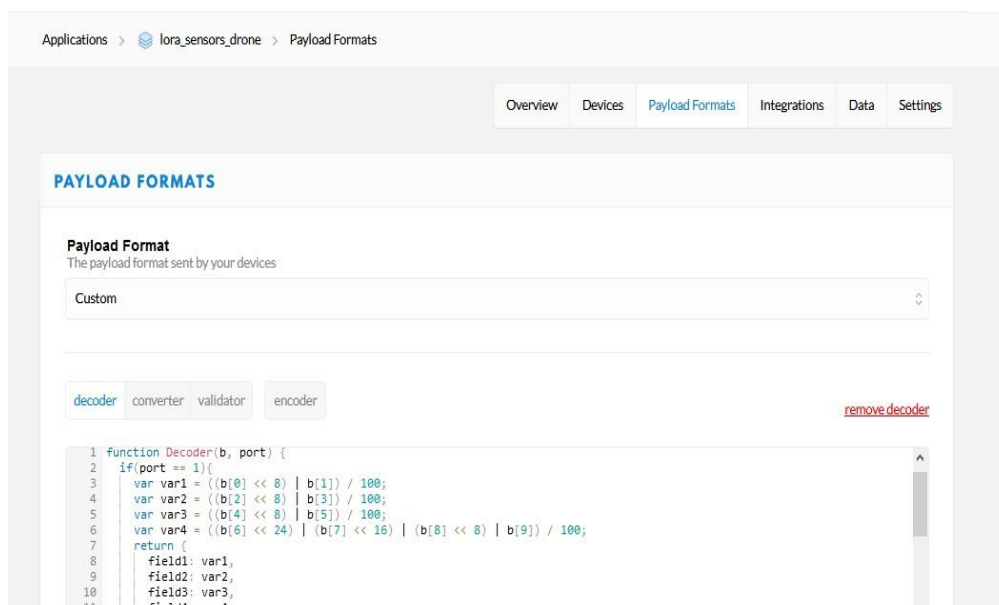


Figura 3.2. Payload Formats

time	counter	port	payload	fields
20:17:32	8	2	payload: 01D5	field5: 4.69
20:17:27	7	1	payload: OCC414B6B2DC009ACB71	field1: 32.68 field2: 53.82 field3: 457.88 field4: 181446.25
20:17:14	6	2	payload: 01D5	field5: 4.69
20:17:08	5	1	payload: OCC214BDB2B0009ACAC2	field1: 32.66 field2: 53.89 field3: 457.44 field4: 181444.5
20:16:54	4	2	payload: 01D5	field5: 4.69
20:16:49	3	1	payload: OCC014E7B2CA009ACB26	field1: 32.64 field2: 53.51 field3: 457.7 field4: 181445.5
20:16:35	2	2	payload: 01D7	field5: 4.71
20:16:30	1	1	payload: OCC21545B2B0009ACAC2	field1: 32.66 field2: 54.45 field3: 457.44 field4: 181444.5

Figura 3.3. Applications Data

El Gateway, que es el que forma un puente entre el dispositivo (lopy4) y The Things Network utiliza redes de baja potencia como LoRaWAN para conectarse a la puerta de enlace, y a su vez transmitir la data utilizando redes de gran ancho de banda como WiFi para conectarse a The Things Network. Figura 3.4.

Se puede observar que los parámetros de la información obtenida en la sección Traffic son:

- Hora: en el momento que el gateway recibe la data,
- Frecuencia: es la banda de frecuencia con la que registró, en este caso US915,
- Modulación: Lora,
- CR: Coding Rate 4/5
- Data rate: se muestran 2 valores SF, que es el factor de expansión y el BW, que depende del ancho de banda utilizado, en este caso 125 kHz.
- Airtime: Tiempo Aire
- Counter: contador de la data recibida,
- Device address: como su nombre lo dice la dirección del dispositivo del cual recibe la data que es el lopy4,
- Payload size: tamaño en bytes de la data recibida.

The screenshot shows the 'GATEWAY TRAFFIC' page in The Things Console. The page title is 'GATEWAY TRAFFIC' with a 'beta' badge. The breadcrumb navigation is 'Gateways > main_gateway > Traffic'. The page has a search bar with '0 bytes' and a 'clear' button. The table below shows traffic data for a gateway named 'main_gateway'.

time	frequency	mod.	CR	data rate	airtime (ms)	cnt	dev addr	payload size
20:18:11	904.5	lora	4/5	SF 7 BW 125	46.3	12	26 02 16 28	15 bytes
20:18:05	904.7	lora	4/5	SF 7 BW 125	61.7	11	26 02 16 28	23 bytes
20:17:52	904.9	lora	4/5	SF 7 BW 125	46.3	10	26 02 16 28	15 bytes
20:17:46	905.1	lora	4/5	SF 7 BW 125	61.7	9	26 02 16 28	23 bytes
20:17:33	905.3	lora	4/5	SF 7 BW 125	46.3	8	26 02 16 28	15 bytes
20:17:27	904.7	lora	4/5	SF 7 BW 125	61.7	7	26 02 16 28	23 bytes
20:17:14	904.1	lora	4/5	SF 7 BW 125	46.3	6	26 02 16 28	15 bytes
20:17:08	904.5	lora	4/5	SF 7 BW 125	61.7	5	26 02 16 28	23 bytes
20:16:55	905.1	lora	4/5	SF 7 BW 125	46.3	4	26 02 16 28	15 bytes
20:16:49	903.9	lora	4/5	SF 7 BW 125	61.7	3	26 02 16 28	23 bytes

Figura 3.4. Gateway Traffic

Luego de llegar al gateway, se realiza la conexión hacia the thingspeak, que es la plataforma en la cual se realiza el monitoreo en tiempo real de la data sensada, como se muestra en la Figura 3.5.

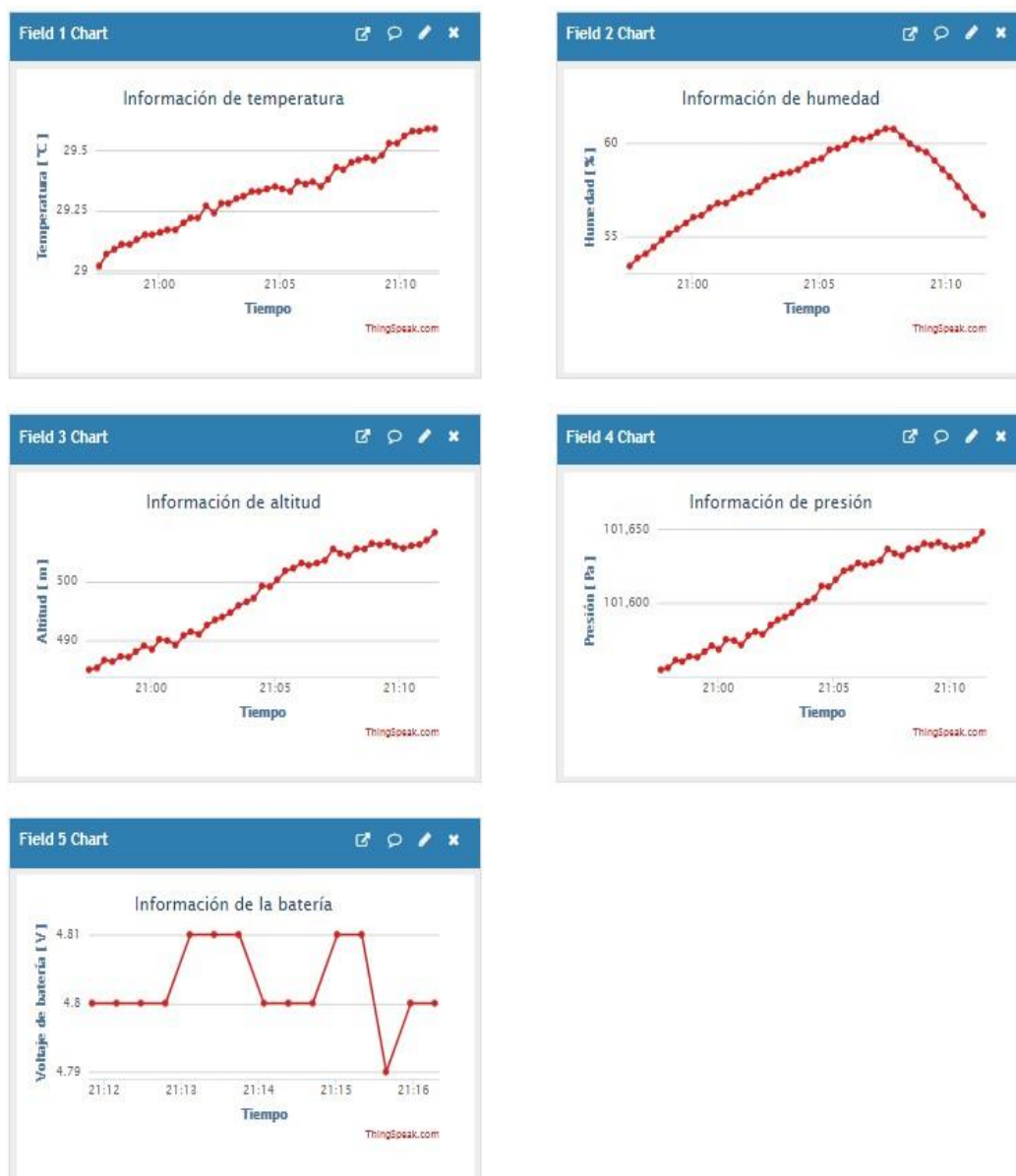


Figura 3.5. Monitoreo de la Temperatura, Humedad, Altitud, Presión y Voltaje

Finalmente, en la Plataforma de Thingspeak, se puede realizar la exportación en formato .CSV de toda la data del Canal que ha sido sensado, como también la data recientemente senada. Figuras 3.6, 3.7, y 3.8.



Figura 3.6. Exportar de data reciente



Figura 3.7. Exportar de data del Canal

created_at	entry_id	field1	field2	field3	field4
2020-08-15 22:29:54 -05	1	30.27	59	512.8	101665.75
2020-08-15 22:30:13 -05	2	30.22	58.95	512.6	101665.25
2020-08-15 22:30:31 -05	3	30.3	58.87	513.2	101667.5
2020-08-15 22:30:51 -05	4	30.35	58.69	512.9	101666.25
2020-08-15 22:31:10 -05	5	30.38	58.53	512.4	101664.5
2020-08-15 22:31:28 -05	6	30.41	58.42	512.8	101666
2020-08-15 22:31:47 -05	7	30.41	58.38	513.4	101668.25
2020-08-15 22:32:07 -05	8	30.38	58.43	512.9	101666.5
2020-08-15 22:32:26 -05	9	30.36	58.46	513.9	101670.25
2020-08-15 22:32:44 -05	10	30.33	58.52	513.4	101668.5
2020-08-15 22:33:04 -05	11	30.32	58.53	513.1	101667.25
2020-08-15 22:33:23 -05	12	30.31	58.51	514	101670.75
2020-08-15 22:33:42 -05	13	30.34	58.46	513.9	101670.5
2020-08-15 22:34:01 -05	14	30.37	58.38	514.8	101673.75
2020-08-15 22:34:19 -05	15	30.34	58.37	514.8	101674
2020-08-15 22:34:39 -05	16	30.41	58.45	514.7	101673.5
2020-08-15 22:34:58 -05	17	30.42	58.33	514.9	101674.25
2020-08-15 22:35:17 -05	18	30.45	58.25	515.8	101678
2020-08-15 22:35:36 -05	19	30.38	58.32	515.1	101675
2020-08-15 22:35:55 -05	20	30.41	58.37	515.1	101675
2020-08-15 22:36:14 -05	21	30.37	58.4	515.4	101676.25
2020-08-15 22:36:33 -05	22	30.4	58.37	516.2	101679.5
2020-08-15 22:36:52 -05	23	30.41	58.27	515.5	101676.75
2020-08-15 22:37:11 -05	24	30.44	58.18	516.5	101680.75
2020-08-15 22:37:30 -05	25	30.42	58.31	515.8	101678.25

Figura 3.8. Data Exportada .CSV

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

El sistema de monitoreo en el que se incorpora la tecnología LoRa con el microcontrolador Lopy4 y su Gateway LoraWAN, posibilita realizar la transmisión de data a largas distancias, permitiendo al usuario final tener de forma accesible y en

tiempo real la información del estado del área que se monitorea por medio de la plataforma Thingspeak.

Los sensores empleados en este sistema sirven como propuesta de la viabilidad de realizar un proyecto de esta naturaleza, ya que se puede incluir más parámetros como es la ubicación GPS, acelerómetro con el fin de obtener más información área donde se encuentra el cultivo monitoreado.

En la plataforma de IoT, Thingspeak, se encuentra disponible la información monitoreada, adicionalmente, se puede realizar la exportación de los campos sensados en un archivo .CSV que sirve de reportería

RECOMENDACIONES

La tecnología LoRaWAN tiene mucho potencial en aplicaciones que requieran el sensado remoto de parámetros de diferentes índoles, en donde el acceso recurrente para obtener esta información sea complejo puesto que un dispositivo con esta tecnología permite una duración de la batería de entre 5 y 10 años.

Se puede analizar otras opciones de microcontrolador que incorpore tecnología LoRa con el fin validar que al ser configurado con OTTA, la comunicación con el server TTN sea estable, ya que dicho método es el recomendado por el proveedor TTN.

BIBLIOGRAFÍA

- [1] «El Telégrafo,» 21 Abril 2014. [En línea]. Available: <https://www.eltelegrafo.com.ec/noticias/economia/8/el-primer-problema-de-la-agricultura-es-la-zonificacion>. [Último acceso: 02 Septiembre 2020].
- [2] nandrade, «Universidad Técnica del Norte,» [En línea]. Available: <https://www.utn.edu.ec/ficaya/carreras/agropecuaria/?p=1091>. [Último acceso: 05 Septiembre 2020]. [Último acceso: 20 Agosto 2020].
- [3] PANDORAFMS, «PANDORA FMS,» 20 Septiembre 2018. En línea]. Available: <https://pandorafms.com/blog/es/que-es-lpwan/>
- [4] M. Á. C. González, «AlfaIoT,» [En línea]. Available: <https://alfaiot.com/blog/ultimas-noticias-2/post/que-es-lora-2>. [Último acceso: 20 08 2020].
- [5] «Medium,» 25 Septiembre 2017. [En línea]. Available: <https://medium.com/beelan/haciendo-iot-con-lora-cap%C3%ADtulo-1qu%C3%A9-es-lora-y-lorawan-8c08d44208e8>. [Último acceso: Agosto 21 2020].
- [6] «Docs.python.org,» [En línea]. Available:

- <https://docs.python.org/3/library/os.html>. [Último acceso: 21 Agosto 2020].
- [7] G. Started, «Forum,» [En línea]. Available: <https://forum.pycom.io/topic/3025/lopy4-lorawan-nano-gateway-and-node-abpand-otaa-issues/7>. [Último acceso: 23 Agosto 2020].
- [8] Pycom, «Pycom Go Invent,» [En línea]. Available: <https://alepycom.gitbooks.io/pycomdocumentation/content/chapter/datasheets/downloads/pysense-specsheet.pdf>. [Último acceso: 02 Septiembre 2020].
- [9] «IoT Consulting,» [En línea]. Available: <https://iotconsulting.tech/el-gatewaylorade-ttn-para-su-comunidad/>. [Último acceso: 22 08 2020].
- [10] T. T. Network, «The Things Network,» [En línea]. Available: <https://www.thethingsnetwork.org/docs/lorawan/security.html>. [Último acceso: 22 Agosto 2020].
- [11] T. T. Network, «The Things Network,» [En línea]. Available: <https://www.thethingsnetwork.org/docs/network/troubleshooting.html>. [Último acceso: 22 Agosto 2020].