

DISEÑO DEL SISTEMA DE SENSORES, CÁMARAS Y TELECOMUNICACIONES
DEL GLOBO SONDA ESTRATOSFÉRICO DE LA MISIÓN E3TRATOS

David Mantilla Jaramillo

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Bucaramanga
2023

DISEÑO DEL SISTEMA DE SENSORES, CÁMARAS Y TELECOMUNICACIONES
DEL GLOBO SONDA ESTRATOSFÉRICO DE LA MISIÓN E3TRATOS

David Mantilla Jaramillo

Trabajo de Grado para optar al título de Ingeniero Electrónico

Director

Julián Gustavo Rodríguez Ferreira
Ph.D Astrofísica

Codirector

Jaime Guillermo Barrero Perez
Mg. potencia eléctrica

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Bucaramanga
2023

Agradecimientos

Agradezco a aquellas personas que influyeron en mi desarrollo como profesional durante todos estos años.

A mis familiares, quienes me brindaron su apoyo incondicional para continuar con mis estudios.

Así mismo, agradecer a mis profesores quienes, con esfuerzo y dedicación durante mi trayecto, me enseñaron las herramientas para salir a desempeñarme como profesional.

Y finalmente, pero no menos importante, a mis amigos, ya que me impulsaron a seguir cuando los días se tornaban oscuros.

Dedicatoria

A mis padres, Edinson Mantilla Cagua y Marielly Jaramillo Camacho quienes, aunque no pudieron terminar sus estudios y ser profesionales, lograron que yo pudiera tener la oportunidad de desarrollarme como profesional y alcanzar todas mis metas gracias a su esfuerzo, a ellos les debo la vida y les dedico esto de corazón.

A mis amigos, quienes, sabiendo que no tenía suficiente para alcanzar la finalización de mi carrera, cada uno aportó su granito de arena para estar donde estoy hoy día, a ellos les deseo lo mejor.

A mi novia, Paola Andrea Contreras Torres, quien me ayudó a encontrar el camino que había perdido hace mucho tiempo y me alentó para continuar por más dificultades que surgieran, a ella le agradezco con el alma.

Índice general

Introducción	1
1. Objetivos	3
1.1. Objetivo general	3
1.2. Objetivos específicos	3
2. Marco de Referencia	4
2.1. Estado del arte	4
2.2. Misión E3Tratos	5
2.3. Variables atmosféricas	6
2.3.1. Gases de efecto invernadero	7
2.4. Raspberry Pi 3B+	7
2.4.1. Raspbian	9
2.5. Radio enlace	9
2.5.1. Presupuesto de potencia	9
2.5.2. RSSI	11
3. Arquitectura y Diseño	12
3.1. Arquitectura	12
3.1.1. Gondola	12
3.1.2. Estación terrena	13
3.2. Diseño	13
3.2.1. Variables medibles	13
3.2.2. Cerebro de la misión	14
3.2.3. Sensores usados	15
3.2.3.1. PT100	15
3.2.3.2. BMP180	16
3.2.3.3. GUVA-S12SD	17
3.2.3.4. IMU MPU9250	18
3.2.4. Captación de imagen	19
3.2.4.1. Cámara de Raspberry Pi	19
3.2.5. Radios de comunicación	20
3.2.5.1. RFD900+	20
3.2.5.2. Yagi-Uda	22
3.2.5.3. Monopolar 1/4 de onda	22
3.2.6. Alimentación	23
3.2.6.1. VTA 10000[mAh]	24
3.2.7. PCB	25
3.2.8. Código de la gondola	27
3.2.8.1. Diagrama de flujo de la gondola	28
3.2.9. Código de la base terrena	31

3.2.9.1.	Conexión de la antena con la base terrena	31
3.2.9.2.	Diagrama de flujo de la base terrena	32
4.	Resultados	33
4.1.	Pruebas de sensores	33
4.2.	Pruebas a la cámara	35
4.3.	Potencia de consumo de la Raspberry Pi	35
4.4.	Configuración de comunicación	37
4.5.	Almacenamiento de datos	39
4.5.1.	Trama de datos	39
4.5.2.	Imágenes	40
4.6.	Montaje final	42
4.7.	Interfaz de la Base terrena	44
4.8.	Ubicación y resultados de las pruebas de comunicación	45
4.9.	Valores de RSSI según la distancia	46
4.10.	Pruebas de presupuesto de potencia	48
4.11.	Pruebas en Radio Mobile para comunicación	49
4.11.1.	Facultad de Ingeniería Fisicomecánicas - Aeropuerto Palonegro	50
4.11.2.	Aeropuerto Palonegro - Peaje Picacho	52
5.	Trabajo futuro	56
5.1.	Sensores	56
5.2.	Cámaras	57
5.2.1.	GoPro Max360	57
5.2.2.	Camarray Hat Arducam	58
5.3.	Alimentación	59
5.3.1.	INIU 20000[mAh]	59
6.	Recomendaciones	62
7.	Conclusiones	64
8.	Anexos	65
8.1.	Repositorios	65
8.2.	Coeficiente de reflexión de cada antena usada	65
8.3.	Potencia de los datos de transmisión a corta distancia (10[cm])	67
8.4.	Distancias de las pruebas mostrado en Google Earth	68
8.4.1.	Estadio 1°de Marzo	68
8.4.2.	Teatrino - Administración III	69
8.4.3.	Teatrino - Camilo (UIS-75)	69
8.4.4.	Teatrino - Edificio Santa Isabella	70
8.5.	Gráficas de RSSI para cada prueba	70
8.5.1.	Estadio 1°de Marzo	70
8.5.2.	Teatrino - Administración III	71
8.5.3.	Teatrino - Camilo (UIS-75)	71

8.5.4.	Teatrino - Edificio Santa Isabella	72
8.6.	Gráficas de RSSI con LNA para las últimas dos pruebas	72
8.6.1.	Teatrino - Camilo (UIS-75)	72
8.6.2.	Teatrino - Edificio Santa Isabella	73
8.7.	Fotos tomadas	73
8.7.1.	Estadio 1°de Marzo	73
8.7.2.	Teatrino - Administración III	74
8.7.3.	Teatrino - Camilo (UIS-75)	75
8.7.4.	Teatrino - Edificio Santa Isabella	76

Índice de figuras

2.1. Balance de radiación en la atmósfera	5
2.2. Estructura de la Góndola	6
2.3. Visión general de gases de efecto invernadero	7
2.4. Raspberry Pi 3B+	8
3.1. Arquitectura del sistema	12
3.2. Esquema ampliado de pines de Raspberry Pi 3B+	15
3.3. Conexión de PT100 con MAX31865	16
3.4. Módulo BMP180	16
3.5. Modulo sensor UV	18
3.6. IMU MPU-9250	18
3.7. Modulo de cámara Pi	19
3.8. Módem RFD900+	20
3.9. PCB RFD900+ pinout	21
3.10. Antenas direccionales para la recepción de datos	22
3.11. Antena monopolar de 1/4 de onda	22
3.12. Banco de baterías de polímero de litio VTA 10000[mAh]	24
3.13. PCB de extensión de puertos	25
3.14. Diagrama de conexiones de dispositivos electrónicos	27
3.15. Diagrama de flujo, código de la góndola	30
3.16. Diagrama de recepción de datos, base terrena	31
3.17. Diagrama de flujo, código de la base terrena	32
4.1. Lectura de sensores en la Raspberry Pi	33
4.2. Datos entregados por el BMP180 y PT100	34
4.3. Resultado de los sensores UV en [mV]	34
4.4. Lectura del sensor UV en Índice UV	35
4.5. Fotos Camara Pi, resolución 1280x720(por defecto)	35
4.6. Pruebas de consumo de Raspberry Pi	36
4.7. Pruebas de consumo de transmisión de Raspberry Pi	36
4.8. Ancho de banda inicial	37
4.9. Ancho de banda reconfigurado	37
4.10. Coeficiente de reflexión de las antenas monopolares de 1/4 de onda	38
4.11. Coeficiente de reflexión de las antenas Yagi-Uda	39
4.12. Muestra de potencia del enlace enviando una trama	39
4.13. Fragmento del código secundario para almacenamiento de las tramas	40
4.14. Archivo de texto con la tramas almacenadas	40
4.15. Fragmento del código secundario para almacenamiento de imágenes	41
4.16. Captura de pantalla del guardado de imágenes	41
4.17. Arquitectura mecánica final	42
4.18. Sistema interno de la Góndola	43
4.19. Datos de ubicación GPS Eagle Flight Computer	44
4.20. Interfaz gráfica final	44
4.21. Ubicación de las antenas durante las pruebas	46

4.22. Muestras de las pruebas realizadas en las cuatro ubicaciones	46
4.23. RSSI principal	47
4.24. Calidad de la señal RSSI en decibelios para las pruebas de comunicación . .	48
4.25. Mapa UIS - Aeropuerto Palonegro Radio Mobile	50
4.26. Radio enlace UIS - Aeropuerto Palonegro Radio Link	51
4.27. Mapa Aeropuerto Palonegro - Peaje Picacho Radio Mobile	53
4.28. Radio enlace Aeropuerto Palonegro - Peaje Picacho Radio Link	54
5.1. GoPro Max	58
5.2. Ubicación de las Cámaras para Raspberry Pi	58
5.3. Paquete de cámaras cuadrascópica Arducam	59
5.4. Banco de Baterías de polímero de litio INIU 20000mAh	60
8.1. Espectro de coeficiente de reflexión antena monopolar	65
8.2. Espectro de coeficiente de reflexión antena monopolar 90°	66
8.3. Espectro del coeficiente de reflexión Yagi-Uda 690-2700[MHz]	66
8.4. Espectro del coeficiente de reflexión Yagi-Uda 900[Mhz]	67
8.5. Muestra # 1 de potencia	67
8.6. Muestra # 2 de potencia	68
8.7. Estadio 1° de Marzo Google Earth online	68
8.8. Teatrino-Administración III Google Earth online	69
8.9. Teatrino-Camilo Torres(UIS-75) Google Earth online	69
8.10. Teatrino - Edificio Santa Isabella Google Earth online	70
8.11. RSSI Estadio 1° de Marzo	70
8.12. RSSI Teatrino - Administración III	71
8.13. RSSI Teatrino - Camilo Torres(UIS-75)	71
8.14. RSSI Teatrino - Edificio Santa Isabella	72
8.15. RSSI Teatrino - Camilo Torres(UIS-75) con LNA	72
8.16. RSSI Teatrino - Edificio Santa Isabella con LNA	73
8.17. Foto recibida 1	73
8.18. Foto recibida 2	74
8.19. Foto recibida 1	74
8.20. Foto recibida 2	75
8.21. Foto recibida 1	75
8.22. Foto recibida 2	76
8.23. Foto recibida 1	76

Índice de tablas

2.1. Valores recomendados de RSSI	11
3.1. Variables selectas	14
3.2. Características de lectura BMP180	17
3.3. Características Principales de Módem RFD900+	21
3.4. Características de las antenas	23
3.5. Tabla de consumo máximo	23
3.6. Conexionado de la Raspberry	26
4.1. Presupuesto de potencia de las pruebas	49
5.1. Sensores ideales	56

Resumen

Título: DISEÑO DEL SISTEMA DE SENSORES, CÁMARAS Y TELECOMUNICACIONES DEL GLOBO SONDA ESTRATOSFÉRICO DE LA MISIÓN E3TRATOS*

Autor: David Mantilla Jaramillo**

Palabras clave: Globo estratosférico, Atmósfera, Cambios climáticos, Python.

Descripción:

Durante los últimos años, se ha incrementado la búsqueda de conocimiento respecto a la atmósfera y todas las variables que componen esta capa que protege la vida en la tierra debido a cambios climáticos y afectaciones al medio ambiente. Uno de los métodos para evaluar la atmósfera ha sido denominado: Globo estratosférico o meteorológico, que se encarga de llevar una carga útil hasta cierta altura para obtener información del cómo se comportan dichas variables. Aquí es donde nace la misión del globo estratosférico E3tratos(Altamar Amaranto, Fajardo Ariza, C.A., Rodriguez Ferreira, J.G., and Universidad Industrial de Santander. Escuela de Ingenierías Eléctrica, E. y de T., 2019). Este consistió en incorporar un grupo de sensores y radios de telecomunicación a una Raspberry Pi 3 B+, para leer cambios de las variables estratosféricas programados en Python y continuamente ser transmitidos a tierra donde la base terrena recibe estos datos y los interpreta. En el presente libro se explica el desarrollo y diseño de un prototipo para el sistema electrónico de carga útil que llevará el globo, reestructurando el código para adicionar una cámara al sistema para la transmisión de imágenes que cubrirá el ascenso y descenso. Adicionalmente se rediseña la interfaz de la base terrena en Visual Studio así como también se realizan pruebas y simulaciones de comunicación a larga distancia para validar su correcto funcionamiento.

*Trabajo de grado

**Facultad de Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Ingeniería Electrónica. Director: Julián Gustavo Rodríguez Ferreira.Ph.D Atrofísica. Codirector: Jaime Guillermo Barrero Pérez. Mg. Potencia eléctrica.

Abstract

Title: DESIGN OF THE SENSOR, CAMERA AND TELECOMMUNICATION SYSTEM OF THE STRATOSPHERIC BALLOON PROBE OF THE E3TRATOS MISSION*

Author: David Mantilla Jaramillo**

Key Words: Stratospheric balloon, Atmosphere, Climate change, Python.

Description:

During the last years, the search for knowledge about the atmosphere and all the variables that compose this layer that protects life on earth due to climatic changes and environmental affectations has increased. One of the methods to evaluate the atmosphere has been called: stratospheric or meteorological balloon, which is responsible for carrying a payload to a certain height to obtain information on how these variables behave. This is where the mission of the stratospheric balloon E3tratos(Altamar Amaranto, Fajardo Ariza, C.A., Rodriguez Ferreira, J.G., and Universidad Industrial de Santander. Escuela de Ingenierías Eléctrica, E. y de T., 2019) was born. This consisted of incorporating a group of sensors and telecommunication radios to a Raspberry Pi 3 B+, to read changes in the stratospheric variables programmed in Python and continuously transmitted to the ground where the ground base receives this data and interprets it. This book explains the development and design of a prototype for the electronic payload system that will carry the balloon, restructuring the code to add a camera to the system for the transmission of images that will cover the ascent and descent. Additionally the interface of the ground base is redesigned in Visual Studio as well as tests and simulations of long distance communication to validate its correct operation.

*Degree work

**Faculty of Physicomechanics. School of Electrical, Electronic and Telecommunications Engineering. Electronics Engineering. Director: Julián Gustavo Rodriguez Ferreira.Ph.D Astrophysics. Codirector: Jaime Guillermo Barrero Pérez. Mg. Electrical power.

Introducción

Con el pasar de los años el planeta tierra ha presenciado fenómenos que afectan al medio ambiente y a la vida terrestre debido a incendios forestales masivos, enfermedades pulmonares, oleadas de calor, contaminación ambiental, etc. Lo que ha llevado al ser humano a la ininterrumpida búsqueda de los factores que afectan nuestras condiciones de vida. Varios de estos factores son las alteraciones que puede tener nuestra atmósfera lo que ha llevado a la humanidad a desarrollar métodos para la observación y recopilación de datos sobre los cambios que tienen las diferentes variables atmosféricas allí ubicadas.

Los globos estratosféricos o globos de gran altitud es uno de los métodos que se empeñan en obtener información a alturas comprendidas por la estratosfera, este es un globo que se llena de helio o hidrógeno para ascender alrededor de los 37[km] de altura junto con instrumentos requeridos según el propósito de este, como puede ser la obtención de imágenes y/o lectura a través de sensores para estas condiciones ya sea para fines educativos como científicos.

Aquí es donde entra la misión del globo estratosférico E3Tratos(Altamar Amaranato, Fajardo Ariza, C.A., Rodriguez Ferreira, J.G., and Universidad Industrial de Santander. Escuela de Ingenierías Eléctrica, E. y de T., 2019). El cual, consistió en diseñar un la arquitectura de un conjunto de sensores conectados a una Raspberry Pi 3 B+ para leer variables estratosféricas, enviarlas a los radios de telecomunicación y finalmente transmitirlos la base terrena donde se interpretan estos datos en un interfaz.

El presente libro explica el diseño del sistema electrónico para este globo reestructurando el código de la Raspberry Pi 3 B+ para incorporar una cámara que cubra el ascenso y el descenso durante la misión. Del mismo modo se rediseña la interfaz gráfica programada en Visual Studio en lenguaje Python para la visualización de estos datos de sensores e imágenes.

Para cumplir con el propósito de esta misión se seleccionaron sensores que estarán a condiciones específicas de operación durante el vuelo, sin embargo para este proyecto se realizaron pruebas con sensores que pudieran validar la arquitectura a tener en cuenta de

cada protocolo anteriormente nombrado que trabaja la Raspberry, incluso se adiciona un circuito integrado para que la mini computadora pueda leer sensores analógicos. Adicional a esto se incorporó una cámara Pi, diseñada para trabajar con Raspberry, para validar configuraciones de cámara a tener en cuenta en futuros lanzamientos.

Finalmente se anexan resultados de las pruebas realizadas donde se muestran los datos en la interfaz de la base terrena desarrollada en Tkinter a través de Python, asimismo se explica cómo se guarda la trama de datos y las imágenes recibidas seguido de los valores de RSSI a diferentes distancias y cómo afectaron a las pruebas. Por último se incluyen simulaciones que validan la comunicación a larga distancia para después añadir recomendaciones a tener en cuenta sobre estas.

1. Objetivos

1.1. Objetivo general

Implementar el sistema de obtención de datos de sensores atmosféricos y cámaras que componen la carga útil del globo sonda estratosférico de la misión E3Tratos para luego ser transmitidos a tierra por medio de un radio enlace.

1.2. Objetivos específicos

- Extraer las variables presentes en las dos capas inferiores de la atmósfera como temperatura, presión y altura, que puedan ser medidas a través de sensores comerciales que operen en las condiciones que se tendrán durante la misión, además seleccionar un conjunto de cámaras para la obtención de imagen
- Diseñar la conexión del sistema inteligente basado en los lenguajes manejados por la Raspberry Pi 3 para la transmisión inalámbrica de las variables sensadas
- Implementar un radio enlace que permita en línea de vista directa la recepción y transmisión de datos e imágenes en un rango de 20 [Km] y realizar las pruebas para su correcto funcionamiento.
- Verificar en laboratorio que el sistema inteligente y el radio enlace cumplen con las restricciones que impone el globo sonda.

2. Marco de Referencia

2.1. Estado del arte

A lo largo del tiempo el ser humano ha desarrollado un gran interés sobre la atmósfera llevando a cabo múltiples estudios, uno de estos es por medio de globos sonda estratosféricos los cuales llevan un carga útil hasta cierta altura, para realizar medidas de diferentes variables que allí se encuentren.

Uno de estos estudios realizados es “High-Altitude Balloon-Based Sensor System Design and Implementation” (Wang y col., 2020) donde explican el experimento de un globo de gran altitud con un sistema diseñado por ellos para obtener datos de temperatura, posición, altura, actitud y videovigilancia en el que su implementación dio resultados de vuelo al rededor de 30 h y 53 min.

También se encuentran artículos que aportan a la comunidad como “Passepartout Sherpa – A low-cost, reusable transportation system into the stratosphere for small experiments”(Taraba y col., 2014) presentando un sistema de vuelo establecido reutilizable de hasta 35 km de altitud configurado con su respectivo globo y paracaídas adicionando un reflector de radar, junto con el modulo Sherpa quien alberga la instrumentación para estos experimentos, adicional de un modulo EXP el cual se desarrolla de manera independiente según el usuario.

Asimismo, se pueden encontrar proyectos, como lo es “High altitude balloon testing of Arduino and environmental sensors for CubeSat prototype”(Lay y col., 2022) el cual desarrolla un CubeSat basado en Arduino para estudiantes con fines prácticos de diseño y construcción donde se adapta a un sistema de globo estratosférico para la validación de los instrumentos que miden tanto temperatura, presión y humedad relativa el cual puede ser adaptado para Raspberry.

Incluso se tiene la investigación “ADQUISICIÓN, TRANSMISIÓN Y VISUALIZACIÓN DE LA TELEMETRÍA DEL GLOBO SONDA ESTRATOSFÉRICO DE LA MISIÓN

E3TRATOS”(Altamar Amaranto, Fajardo Ariza, C.A., Rodriguez Ferreira, J.G., and Universidad Industrial de Santander. Escuela de Ingenierías Eléctrica, E. y de T., 2019), el cual fue uno de los proyectos pioneros en esta área de investigación de la misión E3Tratos la cual consiste en enviar la carga útil a la estratosfera para medir ciertas variables atmosféricas finalizando con la comunicación a tierra.

Esta anterior investigación, será la guía para este proyecto teniendo como base la misión E3Tratos con la diferencia de que en este se incorpora una cámara para la toma de imágenes durante el ascenso y descenso donde también se realizan pruebas y simulaciones de comunicación a larga distancia que validan el correcto funcionamiento de este.

2.2. Misión E3Tratos

La radiación solar es la fuente que enriquece al planeta tierra haciendo sostenible la vida en ella, sin embargo, la generación de gases de efecto invernadero al pasar de los años generados por el hombre, han ido creando un desnivel entre el balance de radiación que recibe la tierra y que re-emite al espacio, gracias a que la concentración de estos gases no dejan escapar la radiación re-emitada reflejándola de nuevo a la tierra, generando cambios climáticos y alterando ecosistemas enteros.

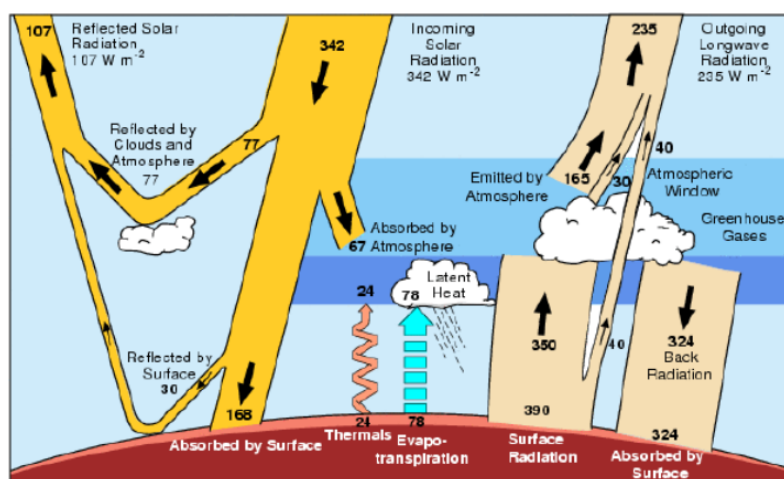


Figura 2.1: Balance de radiación en la atmósfera

Tomado de (Miller y Yates, 2005)

Partiendo de esta problemática, se plantea una solución para monitorear la calidad del aire, aquí nace la Misión E3Tratos la cual lleva a cabo el lanzamiento de un globo sonda estratosférico con la idea de recopilar datos de variables encontradas en las dos primeras capas de la atmósfera (Altamar Amaranto, Fajardo Ariza, C.A., Rodriguez Ferreira, J.G., and Universidad Industrial de Santander. Escuela de Ingenierías Eléctrica, E. y de T., 2019). Dicho globo llevará consigo la carga útil que internamente incorpora estos sensores para el monitoreo del aire, esta es denominada góndola y se puede apreciar en la Figura 2.2

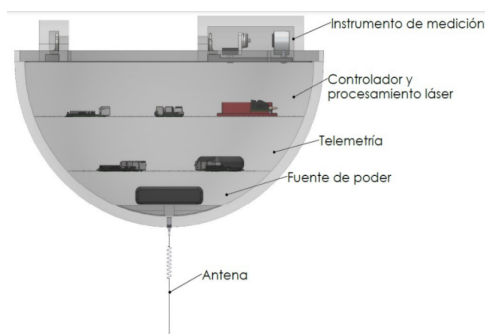


Figura 2.2: Estructura de la Góndola

Tomado de (CUDRIZ y ANDRES, 2019)

2.3. Variables atmosféricas

Desde su creación natural hace miles de años, la atmósfera ha estado constituida por múltiples variables quienes aportan con la estabilidad de la vida terrestre, sin embargo la huella que el hombre va dejando a lo largo de la historia, ha deteriorado la atmósfera cada año, haciendo que estas condiciones a las que se encuentra varíen significativamente generando cambios climáticos afectando miles de vidas y originando daños irreparables a futuro.

2.3.1. Gases de efecto invernadero

Estos son un conjunto de gases que gracias a su producción en exceso por el ser humano alcanzan concentraciones que afectan la atmósfera produciendo calentamiento global, cambios climáticos e incluso se les atribuye el incendio de bosques y por generar cáncer de piel gracias a la fractura de la capa de ozono, quien nos protege de la radiación ultravioleta.

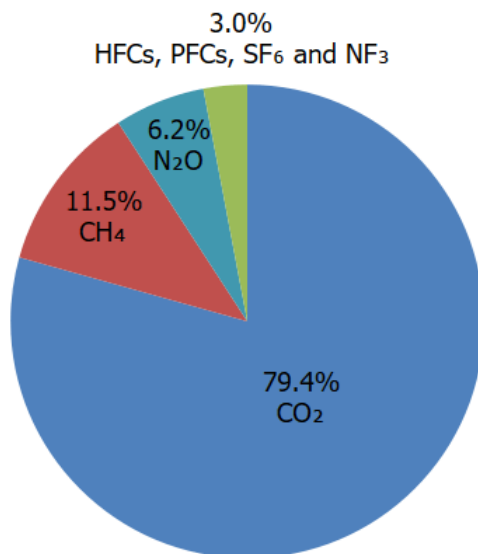


Figura 2.3: Visión general de gases de efecto invernadero

Tomado de:<https://cutt.ly/B7LTikI>

En la Figura 2.3, podemos apreciar que para el año 2020 se tuvo la cantidad porcentual de cada gas contaminante siendo el dióxido de carbono aquel con mas abundancia, seguidos del metano y el oxido nitroso. Estos gases se tomarán en cuenta para la investigación de este libro.

2.4. Raspberry Pi 3B+

Esta es la versión 3 B+ de las computadoras Raspberry Pi el cual fue desarrollada para métodos de aprendizaje y enseñanza de programación, estas mini computadoras del tamaño de una tarjeta de crédito, pueden operar como una personal, cumpliendo funciones

de almacenamiento interno, conexión a internet, bluetooth y múltiples puertos(Harrington, 2015).



Figura 2.4: Raspberry Pi 3B+

En la Figura 2.4 tenemos la versión de este computador(Nath, 2020) y a continuación se adjuntan sus características:

- Procesador: Broadcom BCM2837B0, Cortex-A53 (ARM) 64-bit SOC
- Frecuencia: 1.4 GHz
- Memoria: 1 GB RAM
- Puertos: USB 2.0 = 4
- Video: VideoCore IV 3D. Full-size HDMI
- WIFI: Dual-band 802.11ac wireless LAN (2.4GHz and 5GHz) and Bluetooth 4.2
- Ethernet: Gigabit Ethernet over USB 2.0 (max 300 Mbps). Power-over-Ethernet support (with separate PoE HAT). Improved PXE network and USB massstorage booting
- Interfaces: Camera Serial Interface (CSI), Display Serial Interface (DSI)
- Pines: 40-GPIO (General purpose input/output)
- Protocolos: I2C, SPI and UART.

2.4.1. Raspbian

Como toda computadora, un hardware no sería nada sin su software, en este caso, Raspbian es el sistema operativo de código abierto de las Raspberry Pi, basado en Debian perteneciente a Linux, fácil de manejar para desarrollar el aprendizaje a personas nuevas en programación(Harrington, 2015).

2.5. Radio enlace

Es la interconexión en radio frecuencias entre dos puntos para la transmisión y recepción de datos, dichos puntos pueden ser fijos o móviles(para este trabajo uno de los puntos es móvil), no obstante, un radio enlace puede verse afectado por interferencias generadas por obstáculos, urbanización e incluso la misma geografía de la tierra(Aldaz-Rosas y col., 2016).

2.5.1. Presupuesto de potencia

El presupuesto de potencia del enlace es el cálculo que hay que tener en cuenta para establecer una buena comunicación entre dos puntos.

Para que se de el radio enlace de manera eficiente, se deben conocer las pérdidas del espacio libre, las potencias de transmisión y recepción, las ganancias de las antenas, las pérdidas de línea e interferencia por precipitaciones(Buettrich, 2007).

La ecuación para el presupuesto de potencia es la siguiente:

$$PP[dB] = PIRE[dBm] - FSL[dB] - Pre.[dB] + G.A.RX[dBi] - PL.RX[dB] \quad (2.1)$$

Donde:

PIRE: Es la Potencia Irradiada Isotrópica Efectiva.

FSL: Son las pérdidas en el espacio libre.

Pre: Precipitaciones.

G.A.RX: Ganancia de la antena receptora.

PL.RX: Pérdidas en la línea receptora.

El PIRE es la Potencia Isotrópica Radiada Equivalente y su ecuación esta dada por:

$$PIRE[dBm] = P.TX[dBm] - PL.TX[dB] + G.A.TX[dBi] \quad (2.2)$$

Donde:

P.TX: Es la potencia de transmisión.

PL.TX: Pérdidas en la línea transmisora.

G.A.TX: Ganancia de la antena transmisora.

Ahora se tiene las FSL que son las pérdidas en el espacio libre(Free Space Loss), estas se dan por obstáculos como árboles, construcciones y precipitaciones haciendo que esto genere atenuación durante la transmisión.

$$FSL[dB] = 20Log(f[GHz]) + 20Log(d[km]) + 92,5[dB] \quad (2.3)$$

Donde:

f : Es la frecuencia central.

d : Es la distancia en línea de vista directa entre las antenas.

Para finalizar, se tiene el margen de operatividad, quien define si es eficiente el enlace, esto se hace con el presupuesto de potencia y la sensibilidad del receptor, que sabemos por hoja de datos que es máximo de -121[dBm].

$$Margen[dB] = PP - SR[dBm] \quad (2.4)$$

Donde SR es la sensibilidad del receptor. Una vez se calcula este dato, se prioriza que

esté por encima de los 10[dB] para confirmar que el enlace sea exitoso(CACERES CHANAGA, 2021).

2.5.2. RSSI

Este es el indicador de fuerza de la señal recibida(por sus siglas en inglés:Received Signal Strength Indicator), esto se usa para saber la potencia con la que es recibida la señal y se muestra en escala de decibelios.

En este libro, tomamos una ecuación(Ardupilot.org, 2022) para interpretar el valor RSSI en escala de dBm:

$$RSSI(dBm) = (RSSI/1,9) - 127 \quad (2.5)$$

Adicional a esto se tiene la Tabla 2.1 de referencia con los valores recomendados para validar la calidad de la señal.

Tabla 2.1: Valores recomendados de RSSI

Rango de RSSI	Calidad de la señal
Mejor a -40 dBm	Excepcional
-40 dB a -55 dBm	Muy bueno
-55 dB a -70 dBm	Bueno
-70 dB a -80 dBm	Marginal
menores a -80 dBm	Intermitente o no operacional

Tomado de (Nunez Perez y col., 2017)

3. Arquitectura y Diseño

3.1. Arquitectura

Inicialmente se pone en contexto al lector respecto a la planificación del sistema general interno del globo. Como se puede ver en la Figura 3.1, tenemos dos fases a desarrollar una implica la carga útil de la góndola y otra la estación terrena de recepción de datos.

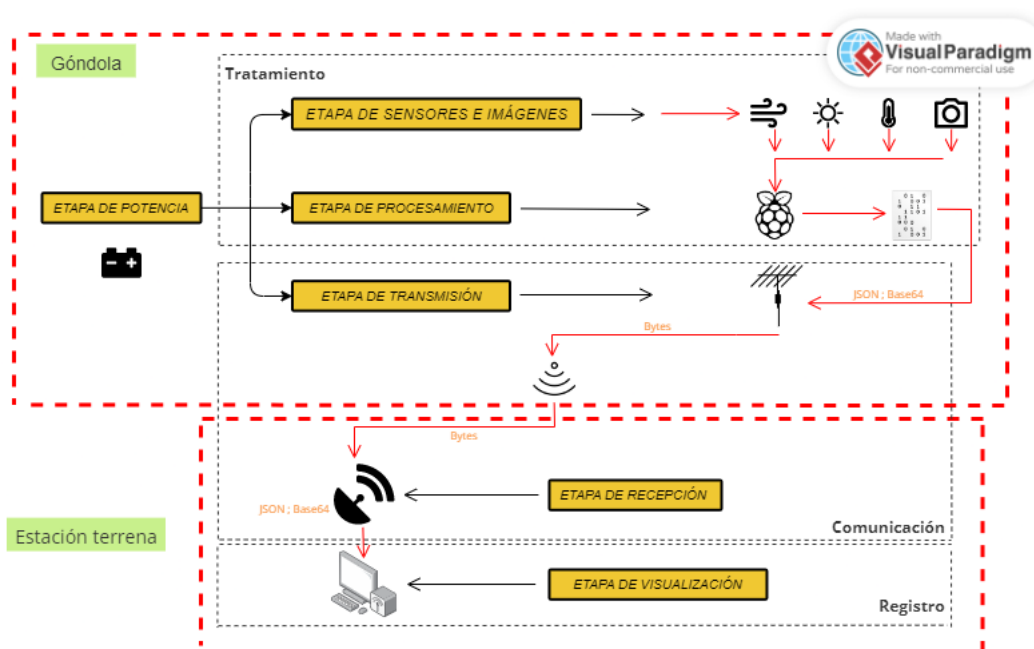


Figura 3.1: Arquitectura del sistema

Diseñado en VisualParadigm por el Autor

3.1.1. Góndola

En la góndola tenemos la etapa de potencia, esta se encargará de alimentar el sistema de la carga útil con cinco voltios durante tres horas aproximadamente que es el tiempo de vuelo en el que el globo llega a la máxima altitud para luego explotar.

A continuación tenemos la sección de tratamiento que cubre dos etapas, la de sensores e imagen y la de procesamiento. La primera se encarga de la toma de imágenes y lecturas de

lo sensores de las variables atmosféricas escogidas; la segunda etapa se encarga de la creación de un código que obtenga, agrupe, codifique y transfiera los datos a los transceptores.

Aquí también tenemos la sección de comunicación que se comparte con la estación terrena, en esta sección para la parte de la góndola se tiene el transceptor que toma los datos codificados y los transmite a tierra.

3.1.2. Estación terrena

Se hace continuación de la sección de comunicación, esta se encarga de la recepción de los datos donde el transceptor toma los datos para su próxima decodificación.

Y para finalizar tenemos la sección de registro que la compone una sola etapa, la de visualización, en esta se toman los datos provenientes de el transceptor y son interpretados en la interfaz gráfica.

El propósito de este capítulo es explicar los pasos que se deben tener en cuenta para cumplir con los objetivos de dicho trabajo.

3.2. Diseño

3.2.1. Variables medibles

Partimos de establecer las variables que se encuentran entre las primeras dos capas de la atmósfera, la troposfera y estratosfera, las cuales abarcan el recorrido que tendrá que cumplir el globo a lanzar. En la Tabla 3.1 se muestran un grupo selecto de variables para próximamente determinar sus sensores.

Tabla 3.1: Variables selectas

Variable	Rango
Temperatura	-60 a 33[°C]
Presión	5 a 1000[hPa]
Humedad	0 a 100[%RH]
Altura	0 a 40[km]
Radiación	200 a 370[nm]
Campo Magnético	-2 a 2[G]
Concentración de CO_2	410 a 430[ppm]
Concentración de O_3	0 a 8[ppm]

Esta tabla fue creada gracias a un estudio realizado por el autor sobre las variables atmosféricas y todo lo que se necesitó para determinarlas. Este estudio lo podemos encontrar en el enlace de Repositorios en el capítulo de anexos, carpeta “Estudio de Variables atmosféricas”, documento “Listado de variables y sensores atmosféricos”.

3.2.2. Cerebro de la misión

Este proyecto requiere de un controlador que pueda procesar los datos de sensores y cámaras para luego transmitirlos al radio que finalmente enviará la información a la base terrena. Para este libro se implementará como cerebro de la misión la Raspberry Pi 3B+ debido a sus características, además de que esta puede ser programada en Python de la misma forma que la base terrena lo cual hará compatible y eficiente el envío y recepción de datos. En la Figura 3.2 se puede ver una imagen de la versión de esta mini computadora con un zoom que se realiza a los GPIO para ver las funciones por defecto de cada uno, aquí podemos apreciar los pines I2C, SPI, UART y alimentación.

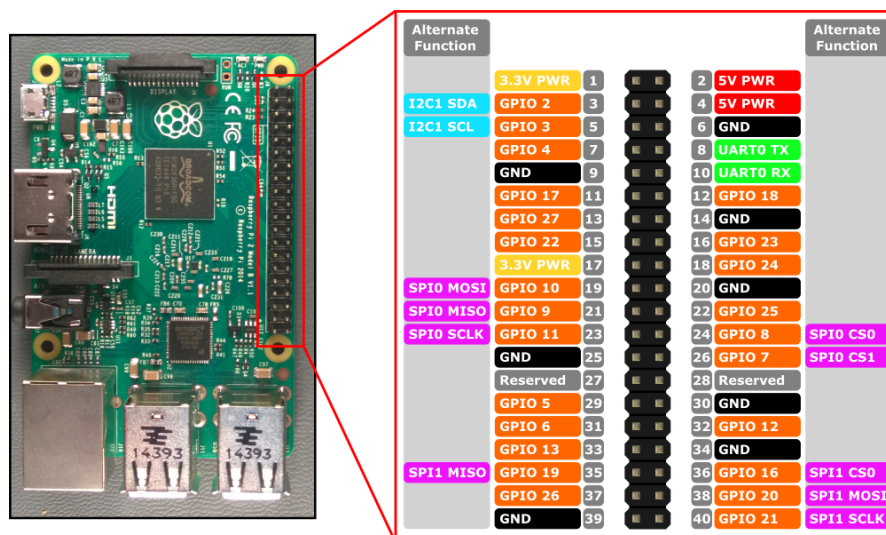


Figura 3.2: Esquema ampliado de pines de Raspberry Pi 3B+

Tomado de: <https://cutt.ly/QMFCMI5>

Ahora se tiene el consumo de esta en diferentes configuraciones. En reposo consume $0.3[A]$, al estar reproduciendo video a 1080p alcanza un consumo de hasta $0.55[A]$ y finalmente en estrés máximo alcanza un consumo de $1.34[A]$.

3.2.3. Sensores usados

3.2.3.1. PT100

Este es un detector de temperatura resistivo (RTD), el cual debe su funcionamiento gracias a la variación de la resistencia (100ohm a 0°C) de este por medio de la temperatura, opera entre -200 a $600[^\circ\text{C}]$, este dispositivo requiere de un módulo adicional llamado MAX31865 el cual interpreta los datos analógicos entregados por la PT100 y los convierte en protocolo SPI. Su funcionamiento va desde 3 a $5[V]$ consumiendo de 2 a $3.5[\text{mA}]$ con una resolución de 15 bits (Maxim, 2012).

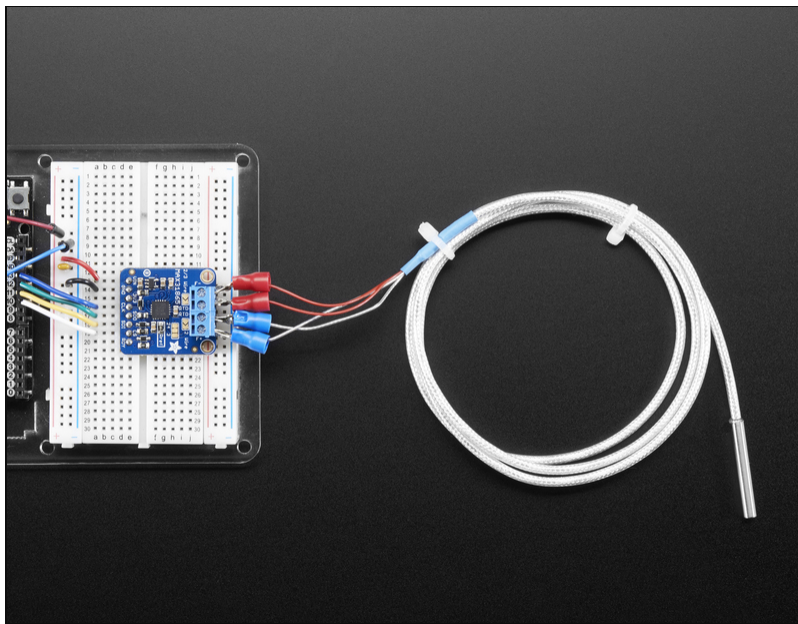


Figura 3.3: Conexión de PT100 con MAX31865

Tomado de («Adafruit MAX31865 RTD PT100 or PT1000 Amplifier», 2016)

3.2.3.2. BMP180

Este es un sensor de presión barométrica y temperatura, también le permite conocer la altitud a la que se encuentra ya que este puede calcularse fácilmente obteniendo las dos variables anteriores. En la Figura 3.4 vemos su módulo.

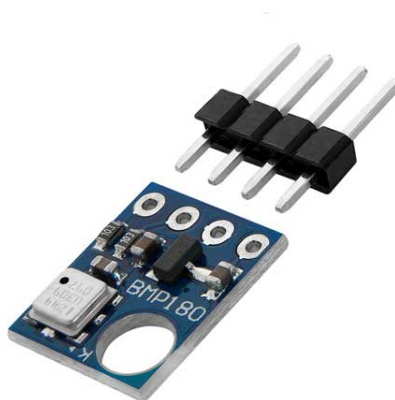


Figura 3.4: Módulo BMP180

Tomado de: <https://cutt.ly/5M3qKj0>

Trabaja con protocolo I2C y se basa en tecnología de efecto piezorresistiva el cual al someterse a deformaciones mecánicas se obtienen variaciones en su resistencia eléctrica. Este dispositivo se puede alimentar desde 3.3 a 5[V], puede llegar a alcanzar 19 bits de resolución. En la Tabla 3.2 tenemos rango y precisión de cada variable.

Tabla 3.2: Características de lectura BMP180

Variable	Mín	Máx	Precisión
Temperatura	-40[°C]	+85[°C]	±1[°C]
Presión	300[hPa]	1.100[hPa]	±1[hPa]
Altitud	0[m]	9.130[m]	±1[m]

Tomado de: <https://programarfacil.com/blog/arduino-blog/bmp180-arduino/>

Este dispositivo tiene diferentes modos de consumo. Ultra bajo consumo a 3[μ A], estándar 5[μ A], alta resolución 7[μ A], ultra alta resolución 12[μ A] y finalmente el modo de resolución avanzada 32[μ A](Bosch, 2013).

3.2.3.3. GUVA-S12SD

Este es un foto-diodo que detecta entre 240-370[nm] del espectro de luz ultravioleta(UV-B), junto con módulo se convierte en un sensor analógico de luz ultravioleta entregando una salida de voltaje que dividido entre 0.1[V](según su hoja de datos) entrega su lectura en índice UV desde 0 a 11+, trabaja con un voltaje de 2.7 a 5.5[V] y podemos apreciar su representación gráfica en la Figura 3.5 (ROITHNER, 2011).

Al ser un sensor analógico necesita de un convertidor a digital, se usa el MCP3008 quien se alimenta desde 2.7 a 5.5[V] y consume en reposo desde 5[nA] hasta 2[μ A] máximo y activo alimentado a 5[V] consume hasta 500[μ A] con una resolución de 10 bits (MICROCHIP, 2001). En este caso se usarán dos sensores UV para realizar pruebas a este convertidor.

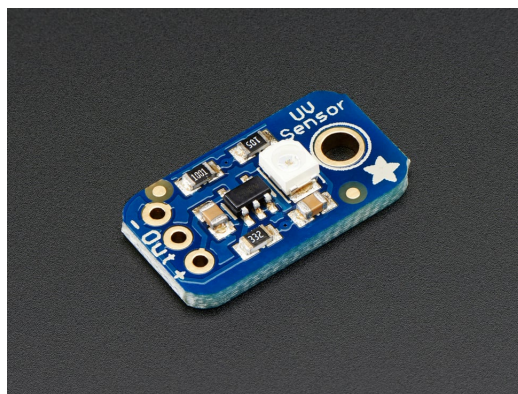


Figura 3.5: Modulo sensor UV

Tomado de: <https://www.adafruit.com/product/1918>

3.2.3.4. IMU MPU9250

Este es un modulo de unidad de medida inercial capaz de entregar nueve grados de libertad o ejes de movimiento por protocolo I2C y SPI. Un giroscopio de tres ejes, un acelerómetro de tres ejes y un magnetómetro de tres ejes todo incluido en un chip (Invensense, 2016). Podemos ver su representación en la Figura 3.6



Figura 3.6: IMU MPU-9250

Tomado de: <https://cutt.ly/J0a1RpJ>

Trabaja con voltajes desde 3.3-5[v] con un consumo de 3.7[mA] máximo y posee

una resolución de 16 bits para mayor precisión. A continuación podemos ver sus rangos de medidas:

- $\pm 250, \pm 500, \pm 1,000$ y $\pm 2,000$ [$^{\circ}/s$] (Giroscopio) 16-bit ADC's, consumiendo 3.2[mA]
- $\pm 2, \pm 4, \pm 8$ y ± 16 [g] (Acelerómetro) 16-bit ADC's, consumiendo 450[μA]
- $\pm 4, \pm 8, \pm 12$, or ± 16 [gauss] (Magnetómetro), consumiendo 280[μA]

3.2.4. Captación de imagen

Una de las ideas principales durante el desarrollo de este libro, es definir un sistema de captación de imagen que cubra el ascenso y descenso del globo sonda, para esto tendremos en cuenta el puerto MIPI CSI, este puerto es para adicionar una cámara a la mini computadora que tome una foto, la guarde en el sistema, la codifique, la transmita y finalmente sea eliminada de la Raspberry para no ocupar espacio en la memoria de esta durante la misión.

3.2.4.1. Cámara de Raspberry Pi

Estas son cámaras de 5[MP] con un tamaño de 25x25x9[mm], grabando vídeo con una resolución de 1080p30, 720p60 y 640x480p60/90. Son especialmente para usarse con las mini computadoras Raspberry Pi. A continuación se muestra una imagen de estas cámaras en la Figura 3.7

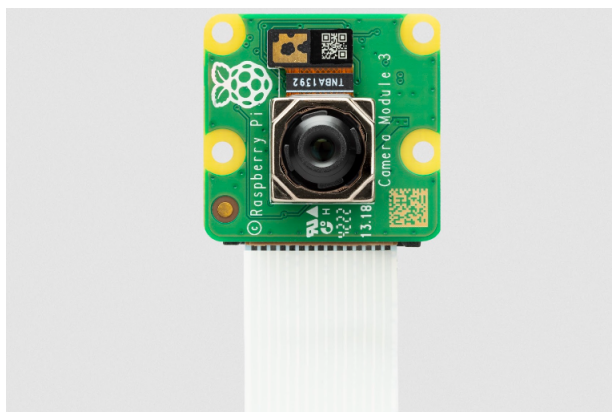


Figura 3.7: Modulo de cámara Pi

Tomado de: <https://cutt.ly/m9JqcDA>

3.2.5. Radios de comunicación

En esta sección se toca un punto clave del diseño general para la comunicación de la góndola y son los radios que se encargarán de enviar y recibir datos durante el trayecto. Hay que tener en cuenta que para la selección de estos dispositivos se tuvo en cuenta la altura a la que llegará la góndola.

3.2.5.1. RFD900+

Estos son los módem transceptores que se encargaran de la transmisión y recepción de datos (RFDesign, 2013), a continuación se adjunta una imagen de estos en la Figura 3.8, seguido de la Tabla 3.3 de características.

Ahora en la Figura 3.9 podemos ver la PCB de los pines de este módem. Los pines que usaremos para la comunicación será 1(GND), 5(VCC), 7(RX) y 9(TX) que saldrán del puerto serial de la Raspberry Pi para enlazarse con este.



Figura 3.8: Módem RFD900+

Tomado de: <https://cutt.ly/Y18d9Z6>

Tabla 3.3: Características Principales de Módem RFD900+

Especificación	Variable
Alcance	40[km]
Frecuencia	902-928[MHz]
Potencia	1[W](+30[dBm])
Tasa de datos aéreos	4-250[kbit/seg]
Tasa de transmisión de datos	2400-115200[baudios]
Sensibilidad	>121[dBm]
Tamaño	30x57x12.8[mm]
Peso	14,5[g]
Alimentación	3.3-5.5[v]
Consumo	800[mA] máx. a +5[v]
Temperatura	-40 a +85[°C]

Tomado de:<http://rfdesign.com.au/products/rfd900-modem/>

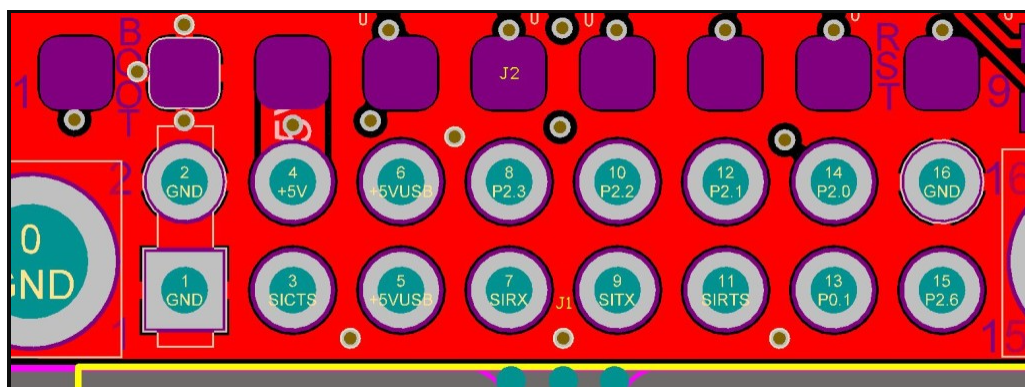


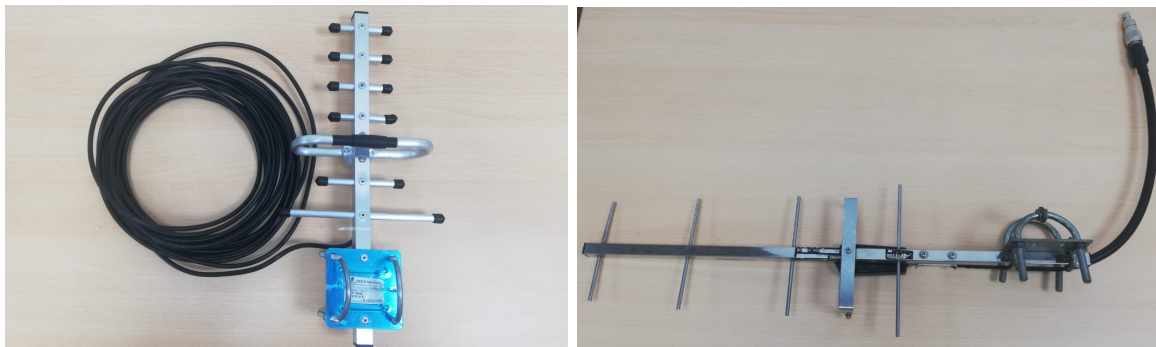
Figura 3.9: PCB RFD900+ pinout

Tomado de:<https://cutt.ly/g168tS4>

Estos dispositivos tienen puerto doble RP-SMA para conectar diversidad de antenas que operan a esta frecuencia. Para este libro nos enfocamos en dos tipos.

3.2.5.2. Yagi-Uda

Este tipo de antenas direccionales inventadas en 1932 deben su eficiencia al diseño de un dipolo junto con directores y reflectores que permiten una alta distancia de comunicación. A continuación tenemos la Figura 4.21 donde vemos las Yagi-Uda que trabajan a estas frecuencias de operación de los Módem.



(a) Yagi-Uda 690-2700[MHz]

(b) Yagi-Uda 900[Mhz]

Figura 3.10: Antenas direccionales para la recepción de datos

Tomado de: Autor

3.2.5.3. Monopolar 1/4 de onda

Este tipo de antena son de propagación omnidireccional a diferencia de la anterior, y para este proyecto se usa como transmisora, podemos ver esta antena en la Figura 3.11 sacada de la hoja de datos del módem RFD900+.



Figura 3.11: Antena monopolar de 1/4 de onda

Tomado de: [http://files.rfdesign.com.au/Files/documents/RFD900 %20DataSheet.pdf](http://files.rfdesign.com.au/Files/documents/RFD900%20DataSheet.pdf)

A continuación se presenta la Tabla 3.4 con las antenas que presentamos junto con sus características.

Tabla 3.4: Características de las antenas

Antena	Frecuencia	Ganancia	Tipo
Monopolar 1/4 onda	900 MHz	2.1 dBi	Omnidireccional
Monopolar 1/4 onda 90°	900 MHz	2.1 dBi	Omnidireccional
Yagi-Uda	690 - 2700 MHz	9-11 dBi	Direccional
Yagi-Uda	900 MHz	6-8 dBi	Direccional

3.2.6. Alimentación

En esta sección se analiza el consumo general de los dispositivos conectados a la Raspberry y de esta misma para poder determinar la carga que tendrá la batería conectada y próximamente asegurar tiempo de vuelo. Para esto se toman las corrientes máximas de consumo de cada sensor y radios de comunicación que llevará esta mini computadora mostrados en la Tabla 3.5. Luego se suma la corriente de consumo de la misma Raspberry reproduciendo vídeo, y en estrés máximo.

Tabla 3.5: Tabla de consumo máximo

Dispositivo	Consumo de corriente
MAX31865	3.5[mA]
BMP180	32[μ A]
GUVA-S12SD	500[μ A]
IMU MPU9250	3.7[mA]
RFD900+	800[mA]

La suma de estas corrientes de consumo de la tabla anterior es de 807.732[mA]. Sumadas con el consumo de la Raspberry reproduciendo vídeo, se tiene un carga de ≈ 1.36 [A],

pero sumadas al consumo de la Raspberry en estrés máximo, se tiene una carga de $\approx 2.15[A]$.

Una vez se tienen estos valores de consumo, se continua con la selección de la batería que suplirá esto, para luego determinar sus horas de trabajo en los dos casos de carga. En este proyecto se usó la siguiente.

3.2.6.1. VTA 10000[mAh]

Este es un banco de baterías portátil de polímero de litio de 10000[mAh] con cuatro puertos, dos de entrada, uno USB tipo C y otro Micro USB para cargar la batería con 5[V] a 2[A] y dos de salida USB tipo A con 5[V] a 1[A] ó 2[A] máximo. Además tiene la capacidad de carga inalámbrica Qi ó también llamada carga por inducción de 5[W] para alimentar dispositivos con el mismo sistema de carga. En la Figura 3.12 podemos ver una imagen de esta batería portátil.

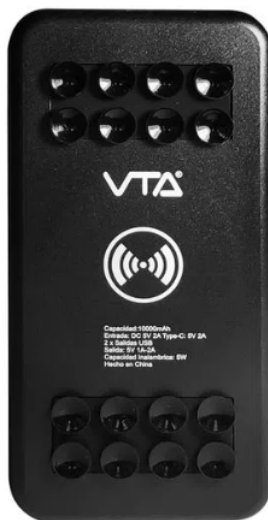


Figura 3.12: Banco de baterías de polímero de litio VTA 10000[mAh]

Tomado de: <https://cutt.ly/K7UTSZH>

Ahora determinamos la autonomía de esta batería con las dos cargas que presentamos anteriormente. Primero está la Ecuación 3.1 donde evaluamos el consumo grabando

vídeo de la Raspberry.

$$Autonomía(h) = \frac{10000[mAh]}{1,36[A]} = 7,4[h] \quad (3.1)$$

Aquí se puede ver un tiempo estimado de siete horas con veinticuatro minutos(7[h] y 24[min]). Ahora está la Ecuación 4.1 pero en este caso tenemos el consumo en estrés máximo.

$$Autonomía(h) = \frac{10000[mAh]}{2,15[A]} = 4,7[h] \quad (3.2)$$

En este caso se tiene un tiempo estimado de cuatro horas con cuarenta y dos minutos(4[h] y 42[min]).

3.2.7. PCB

Debido a que se tienen dispositivos compartiendo puertos de la Raspberry como lo son los sensores I2C, los puertos de alimentación y las tierras comunes, se implementa una placa de circuito impreso(PCB, por sus siglas en inglés), con el fin de extender sus conexiones en paralelo y a futuro adicionar dispositivos I2C.

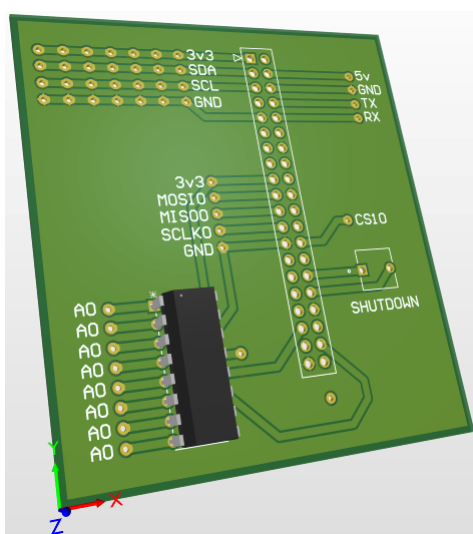


Figura 3.13: PCB de extensión de puertos

Diseñado en AltiumDesigner por el Autor

Como se ve en la Figura 3.13, existen dos pistas adicionales dirigidas hacia el “Shutdown”. Este es un botón que permite apagar de manera segura la Raspberry para evitar que la micro SD que tiene toda la información y el sistema operativo, entre en corrupción.

En la Tabla 3.6 tenemos un listado de los elementos conectados directamente con la Raspberry, donde se aprecia los puertos a los que están conectados cada uno.

Tabla 3.6: Conexión de la Raspberry

Dispositivo	Puerto
BMP180	I2C, Pines: 1, 3, 5 y 9
IMU MPU9250	I2C, Pines: 1, 3, 5 y 9
MAX31865(PT100)	SPI0, Pines: 17, 19, 21, 23, 25 y 26
MCP3008(GUVA-S12SD)	SPI1, Pines: 17, 25, 35, 38 y 40
Cámara Pi	MIPI-CSI
RFD900+	UART, Pines: 4, 6, 8 y 10
VTA 10000[mAh]	Micro Usb
Botón de apagado("Shutdown")	Pines: 30 y 32

Luego tenemos la Figura 3.14 donde vemos el conexionado de todos los dispositivos electrónicos que lleva el sistema interno de la góndola, seguido de se explica una lista explicando el protocolo y los colores de los cables que se aprecian en el Diagrama de conexiones.

Representación de colores para el Diagrama de conexiones:

- Alimentación: 5[V](Blanco), 3.3[V](Rojo), GND(Verde), Analógico(Negro).
- I2C: 3.3[V](Rojo), GND(Verde), SDA(Amarillo), SCL(Azul).
- SPI: 3.3[V](Rojo), GND(Verde), MOSI(Naranja), MISO(Cian), SCLK(Morado), CS(Rosa).
- UART: 5[V](Blanco), GND(Verde), TX(Ocre), RX(Marrón).
- Shutdown: GND(Verde), GPIO 12(Gris).

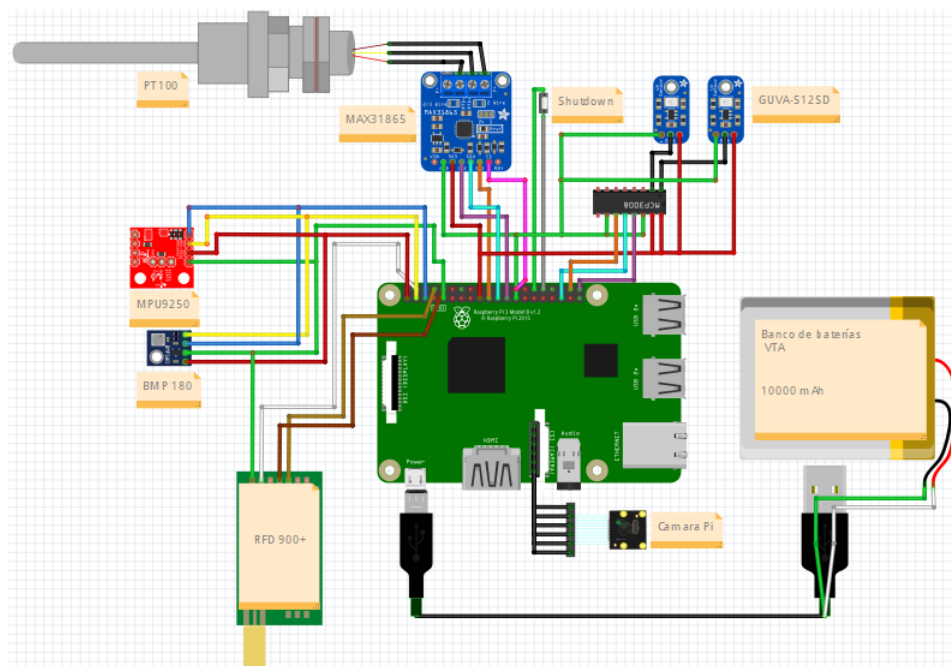


Figura 3.14: Diagrama de conexiones de dispositivos electrónicos

Diseñado en Fritzing por el Autor

Una vez se recopila toda la electrónica de la góndola, ahora pasamos a el diseño de obtención de datos para su próxima comunicación, primero se crea el código de lectura para transmitirlos a los radios seguido de el código de visualización e interpretación de datos de la base terrena.

3.2.8. Código de la góndola

Antes de empezar a describir el código de la góndola se realiza un paso a paso para anexar cada uno de los dispositivos como se mostró en la Figura 3.14 siguiendo las configuraciones de la mini computadora.

Primero: se debe tener en cuenta que antes de utilizar cualquier puerto de la Raspberry Pi hay que dar permisos para que estos se activen, como los puertos I2C el cual solo viene uno por defecto, de la misma forma para los puertos SPI, serial UART y cámara Pi.

Segundo: para incluir cada sensor, se descargan las librerías correspondientes de cada uno de estos las cuales se encargan de tomar los datos generados por el sensor, procesarlos

y finalmente etiquetarlos a una variable, luego en la creación del código general se importan dichas librerías donde solo se llaman estas etiquetas, así podremos apreciar las lecturas de los sensores en las unidades que se esperan.

Tercero: aunque algunos dispositivos necesiten de manera externa incluir librerías, dentro de Raspbian se incluyen librerías básicas para el funcionamiento de algunos puertos específicos de la Raspberry. Este es el caso de la cámara ya que no sólo necesita de permisos para activar el puerto MIPI-CSI sino que también usa su correspondiente librería para poder agregar configuraciones a la captura de imagen, como resolución, brillo, velocidad, etc.

Cuarto: como se incluyó en el primer punto, hay ciertos GPIO por defecto para usar estos protocolos de comunicación, sin embargo, se le recuerda al lector que estos puertos de la Raspberry son de propósito general, por ende pueden configurarse. Este es el caso de los SPI quienes en este libro se necesita habilitar dos: uno para los sensores analógicos con el MCP3008, y otro para la RTD con su convertidor. Esto se puede solucionar con la librería “Spidev” quien nos permite usar hasta tres puertos SPI en los GPIO.

Quinto: finalmente, se realizan configuraciones avanzadas de la Raspberry para su funcionamiento directo con la batería, es decir, durante todo el proceso la mini computadora fue operada con pantalla, ratón y teclado para realizar el código completo de lectura y escritura de cada sensor, sin embargo, cuando todo este sistema se desplace a la góndola, no usaremos ninguno de los tres elementos que fueron mencionados anteriormente, por lo que se configura esta para que al estar alimentada, encienda y pueda comenzar a correr el código junto con el sistema operativo, de esta forma el código completo se ejecuta automáticamente enviando los datos a la base terrena.

3.2.8.1. Diagrama de flujo de la góndola

A continuación se muestra la Figura 3.15. Esta describe cómo procede el código a ejecutarse una vez encendida la Raspberry. Note que uno de los procesos es tomar la trama de datos de los sensores y codificarla con el módulo “json.dumps()” con la idea de

dejarlos como formato de cadena (“string”, en ingles) y así poder transmitirlos.

Además, se tiene la variable “N” la cual puede cambiar y modificarse si así el lector lo prefiere. Cabe resaltar que, como los datos de lectura de todos los sensores son cada segundo, sí se llegase a poner $N = 60$, transmitirá 60 tramas de datos en 60 segundos y luego tomará una foto, la enviará y volverá a empezar.

Finalmente termina el bucle cuando “ccc” sea igual a “N”, lo que permite que entre al ciclo de captura de imagen, la codifique en “Base64” y luego de un tiempo la borre del almacenamiento de la mini computadora, para volver a entrar de nuevo en el bucle .

El código completo se podrá encontrar en el enlace de Repositorios, carpeta “Códigos”, “Código góndola” llamado “Prueba_Comm2.py”.

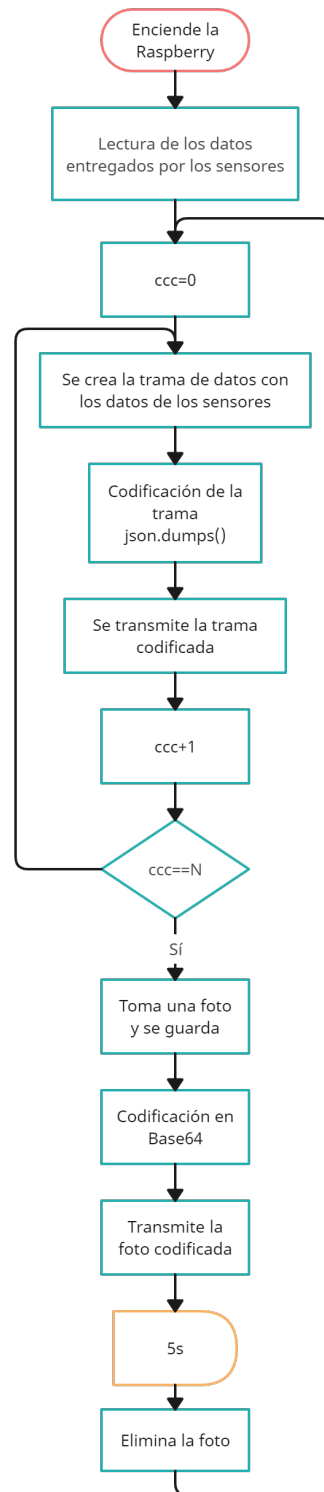


Figura 3.15: Diagrama de flujo, código de la góndola

Diseñado en createely por el Autor

3.2.9. Código de la base terrena

En la sección presente, se explica la recepción de los datos y como se visualizarán estos. En la Figura 3.16 podemos ver el esquema de como se recibirán estos datos.

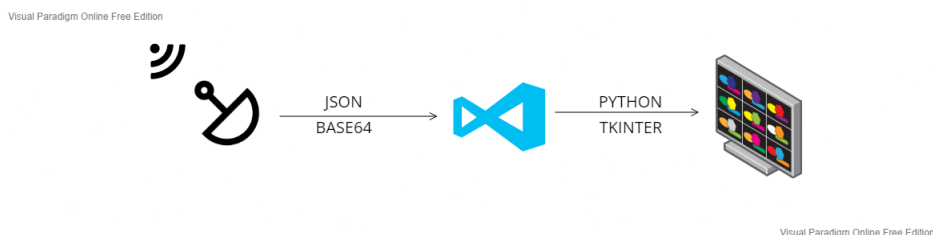


Figura 3.16: Diagrama de recepción de datos, base terrena

Diseñado en VisualParadigm por el Autor

3.2.9.1. Conexión de la antena con la base terrena

Aquí se explica como está compuesta la base terrena para que el lector comprenda su arquitectura. Este consta de la conexión entre el módem RFD900+ y una computadora portátil, los cuales se enlazan con un cable FTDI-USB, donde el puerto FTDI va conectado al radio y el USB conectado a uno de los puertos del portátil. Recordar que al conectar este cable al computador, él tomará un nombre de etiqueta (COM#) y este será utilizado para enlazar el puerto serial al código.

Debido a que se requiere visualizar estos datos en una interfaz basada en la librería Tkinter en Python 3, el código de la base terrena será desarrollado en Visual Studio, esta herramienta permitirá programar en tiempo real el código mientras se realizan pruebas para validar su funcionamiento. Este cumple su objetivo con dos hojas de código donde la primera crea la ventana de visualización y la segunda es quien procesa los datos de llegada, los decodifica, y los anexa a unas etiquetas que luego serán llamadas por el primer código.

3.2.9.2. Diagrama de flujo de la base terrena

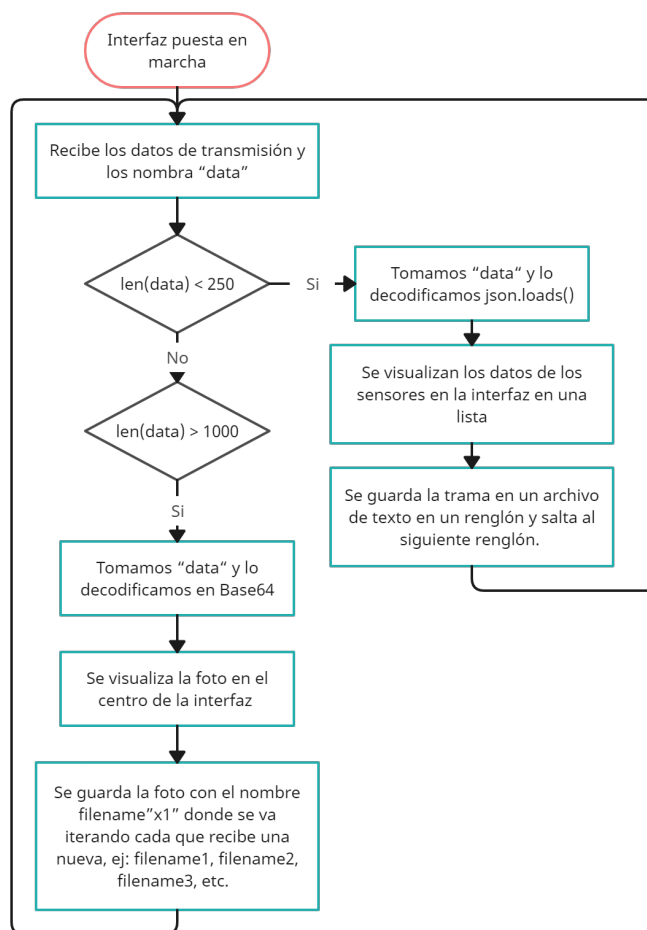


Figura 3.17: Diagrama de flujo, código de la base terrena

En la Figura 3.17 se tiene el diagrama de ejecución del código de la base terrena. El código de esta parte se encuentra en la misma dirección de repositorios pero en este caso en la carpeta "Códigos", "Código la base terrena" llamados "main.py" y "mainframe.py". En caso de correr esta interfaz, se debe ejecutar el primer código.

4. Resultados

4.1. Pruebas de sensores

Una vez se ejecuta el código de la Raspberry aparece la ventana “Shell” mostrada en la Figura 4.1 donde se tiene la lectura de los sensores. Los primeros tres renglones pertenecen al BMP180, la cuarta de la PT100, la quinta del sensor UV, desde la sexta hasta la décima tercera línea son datos del giroscopio y finalmente el último renglón avisando que la imagen ha sido tomada y guardada.

```
Shell
Y Rotation: -85.86934462821188
Foto tomada y guardada

Temperature is 30.6
Pressure is 100291
Altitude is 86.44
Temperature: 32.576C
UV--: 0.000
Gx : -252 escala : -1.9236641221374047
Gy : -4 escala : -0.030534351145038167
Gz : 82 escala : 0.6259541984732825
Acel. X: 16388 escala: 1.000244140625
Acel. Y: 292 escala: 0.017822265625
Acel. Z: -1084 escala: -0.066162109375
X Rotation: 1.018558004392988
Y Rotation: -86.08114566233239
Foto tomada y guardada

Temperature is 30.6
Pressure is 100280
Altitude is 87.37
Temperature: 32.508C
UV--: 0.000
Gx : -276 escala : -2.1068702290076335
Gy : -11 escala : -0.00396946564885496
Gz : 54 escala : 0.4122137404580153
Acel. X: 16488 escala: 1.00634765625
Acel. Y: 224 escala: 0.013671875
Acel. Z: -1044 escala: -0.063720703125
X Rotation: 0.7767963939250973
Y Rotation: -86.2947083953469
Foto tomada y guardada

Python 3.9.2 (/usr/bin/python3)
>>>
```

Figura 4.1: Lectura de sensores en la Raspberry Pi

Viendo la figura anterior se explica lo siguiente, primero, la configuración que tiene la pt100 es de 2 hilos, esto agrega una desviación de 2[°C] por encima de la lectura real que entrega el BMP180; esto es debido a que durante la redacción del código, cuando se configuraba de 3 hilos este generaba un error de lectura, entregando valores de -500[°C] o datos incoherentes como se ve en la Figura 4.2.

```
BMP180
Temperature is 30.5
Pressure is 100335
Altitude is 82.75
PT100
Temperature: -57.878C

BMP180
Temperature is 30.5
Pressure is 100331
Altitude is 83.09
PT100
Temperature: -57.812C

Python 3.9.2 (/usr/bin/python3)
>>>
```

Figura 4.2: Datos entregados por el BMP180 y PT100

Ahora, en segunda medida, tenemos al sensor UV que como se observó en la Figura 4.1 aparece en cero. Esto se da ya que la luz tiene que incidir directamente en el sensor, a continuación se muestran las pruebas de dos sensores UV donde se reflejaba la luz solar con un espejo y se apuntaba directamente a estos.

```
1: 0.000
2: 17.00
1: 20.00
2: 0.000
1: 21.00
2: 0.000
1: 21.00
2: 0.000
1: 23.00
2: 0.000
1: 24.00
2: 0.000
```

Figura 4.3: Resultado de los sensores UV en [mV]

En la Figura 4.3 se ven los numerales 1 y 2, de cada sensor respectivamente en milivoltios. Ahora se ven los resultados en Índice UV en la Figura 4.4. Se notan muestras en cero, pero esto debido a que el haz de luz no era constante.

```

1: 0.00
1: 2.35
1: 0.00
1: 0.00
1: 0.00
1: 0.00
1: 0.00
1: 0.00
1: 0.00
1: 0.00

```

Figura 4.4: Lectura del sensor UV en Indice UV

4.2. Pruebas a la cámara

Para las pruebas de las cámaras se jugó con las configuraciones que estas tienen, desde velocidad de disparo, resolución, brillo, etc. A continuación se ve la Figura 4.5 donde en la imagen 4.5a se configura un brillo de 40, y en la imagen 4.5b un brillo de 70.



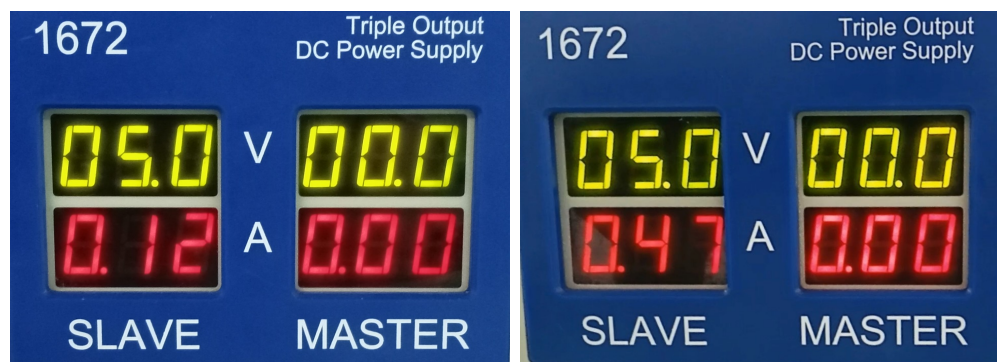
(a) Foto tomada con brillo bajo (b) Foto tomada con brillo alto

Figura 4.5: Fotos Camara Pi, resolución 1280x720(por defecto)

Cabe resaltar que la iluminación que tomen las fotos va condicionada con la velocidad de disparo que se le configure a la cámara, esto es un punto importante por si se requiere mejorar el sistema de captura de imagen.

4.3. Potencia de consumo de la Raspberry Pi

Se tomaron cuatro pruebas de consumo, la primera se muestra en la Figura 4.6a donde aparece el consumo de la Raspberry en reposo, y como se puede apreciar tiene un consumo total de 120[mA] a 5[V] constante. La segunda muestra el consumo de la Raspberry conectada con todos los dispositivos electrónicos operando. En esta ocasión el consumo es de 470 [mA], observe la Figura 4.6b.



(a) En reposo

(b) En operación

Figura 4.6: Pruebas de consumo de Raspberry Pi



(a) Enviando una Trama

(b) Enviando una Imagen

Figura 4.7: Pruebas de consumo de transmisión de Raspberry Pi

Ahora en las dos pruebas restantes, tenemos el consumo para la transmisión de los datos, en la Figura 4.7a tenemos la corriente de consumo al enviar una trama de lecturas de sensores alrededor de 610[mA], mientras que en la Figura 4.7b muestra el máximo consumo que tiene la transmisión de imágenes llegando hasta 1.02[A] a cortas distancias.

Tomando los datos anteriores y aplicando la ecuación de autonomía de la batería usada durante estas pruebas tenemos

$$Autonomía(h) = \frac{10000[mAh]}{1,02[A]} = 9,8[h] \quad (4.1)$$

En este caso se tiene un tiempo estimado de nueve horas con cuarenta y ocho minutos(9[h] y 48[min]). lo que nos entrega más del doble de tiempo presupuestado en teoría.

4.4. Configuración de comunicación

Primero se configura el ancho de banda de los radios en laboratorio ya que inicialmente se encontraba desde 915-928[MHz] como se muestra en la Figura 4.8, por lo que se desperdicia la mitad de capacidad de ancho de banda, a lo que se reconfiguró desde 902 - 928[MHz] como se ve en la Figura 4.9.

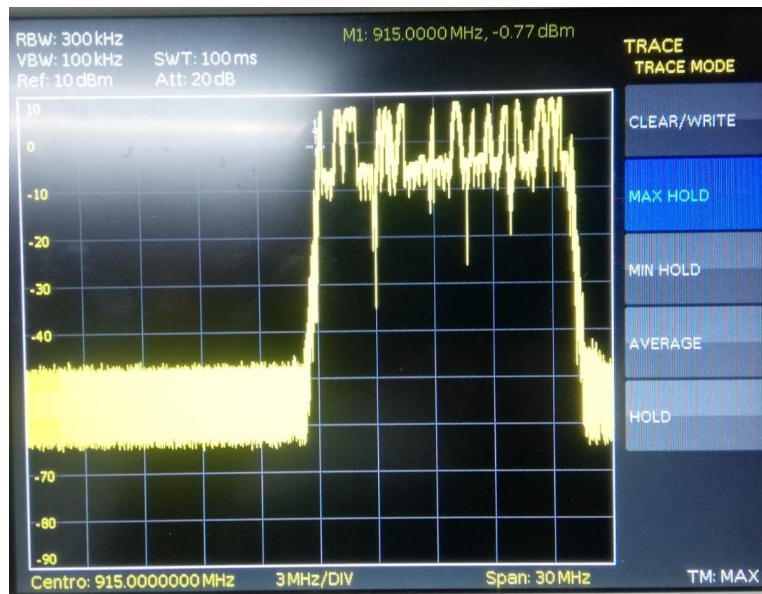


Figura 4.8: Ancho de banda inicial

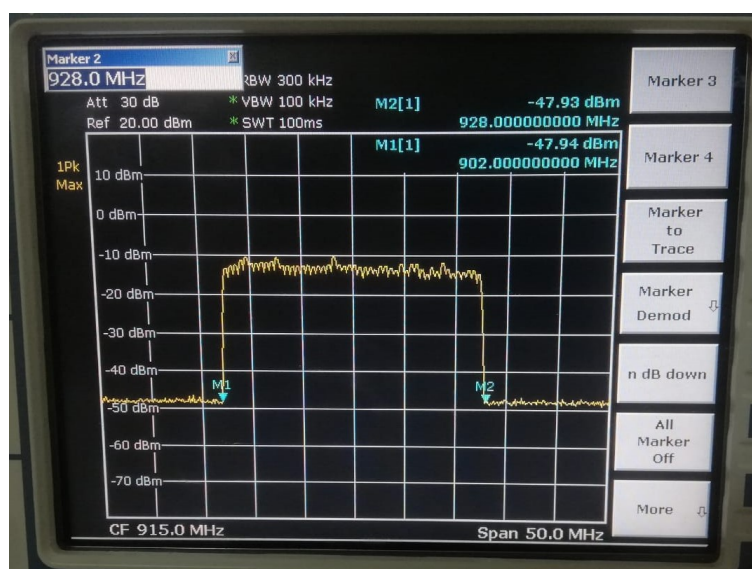
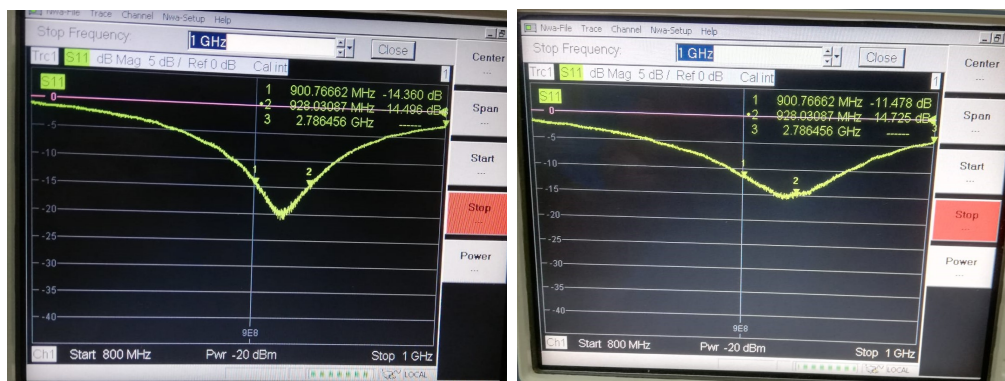


Figura 4.9: Ancho de banda reconfigurado

Ahora, se analizan las antenas para este trabajo donde se observa el coeficiente de reflexión o parámetro s11 el cual debe tener un valor por debajo de $-10[\text{dB}]$ para que garantice aproximadamente un 10 % de potencia reflejada y 90 % de potencia transmitida (Amor Jiménez y Paredes, 2011).



(a) Monopolar de 1/4 de onda

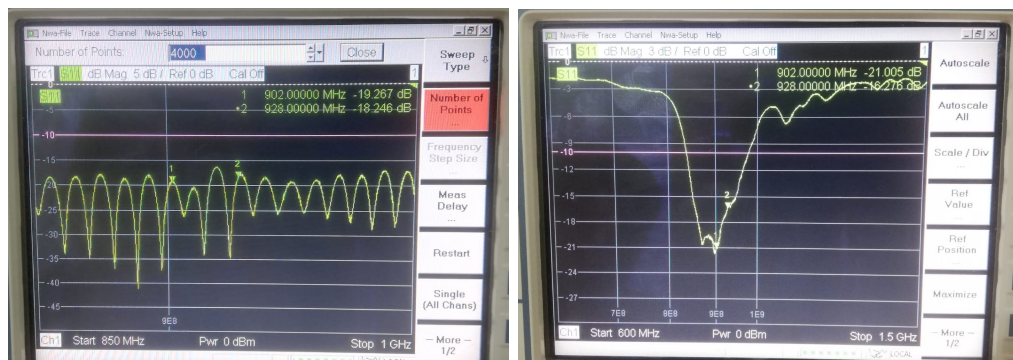
(b) Monopolar de 1/4 de onda de 90°

Figura 4.10: Coeficiente de reflexión de las antenas monopolares de 1/4 de onda

Como se tiene en la Figura 4.10, los coeficientes de reflexión de cada antena están por debajo de $-10[\text{dB}]$ en su frecuencia operativa en este caso entre 900-928[MHz], sin embargo, el coeficiente es más pronunciado en la gráfica 4.10a, por lo tanto, esta será utilizada como antena receptora por su capacidad.

Ahora se muestran los coeficientes de reflexión para las dos antenas Yagi-Uda mostradas anteriormente. En esta ocasión la antena de la Figura 4.11a presentan picos pronunciados en diferentes frecuencias, esto se debe a la longitud de la línea que presenta anteriormente mostrado en la Figura 3.10a. Aún así, esta antena tiene un coeficiente de reflexión menor a $-10[\text{dB}]$. Seguido de esto también se tiene la Figura 4.11b la cual tiene un rango de frecuencias operativa limitado, pero su coeficiente de reflexión alcanza a llegar a $-21[\text{dB}]$.

Finalmente, se realizan pruebas de la potencia del enlace. La Figura 4.12 se tomó en laboratorio, con una distancia corta entre las antenas, con una atenuación de $30[\text{dB}]$. Las antenas están configuradas a $30[\text{dBm}]$ por lo que no existe pérdidas según la imagen.



(a) Yagi-Uda 690-2700[MHz]

(b) Yagi-Uda 900[Mhz]

Figura 4.11: Coeficiente de reflexión de las antenas Yagi-Uda

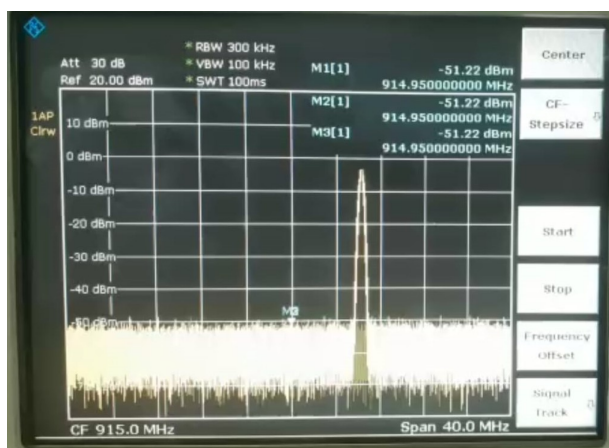


Figura 4.12: Muestra de potencia del enlace enviando una trama

4.5. Almacenamiento de datos

Una de las cosas a tener en cuenta fue el desarrollo de un método que permitiera realizar el almacenamiento de la información que llegaba tanto de sensores como de cámara, esto debido a que cuando finalice la misión, se pueda acceder a estos y posteriormente realizarles un análisis minucioso.

4.5.1. Trama de datos

En el código secundario se tienen las líneas mostradas en la Figura 4.13, donde en la primera se le indica a la variable "file1" abra esa dirección que aparece entre comillas,

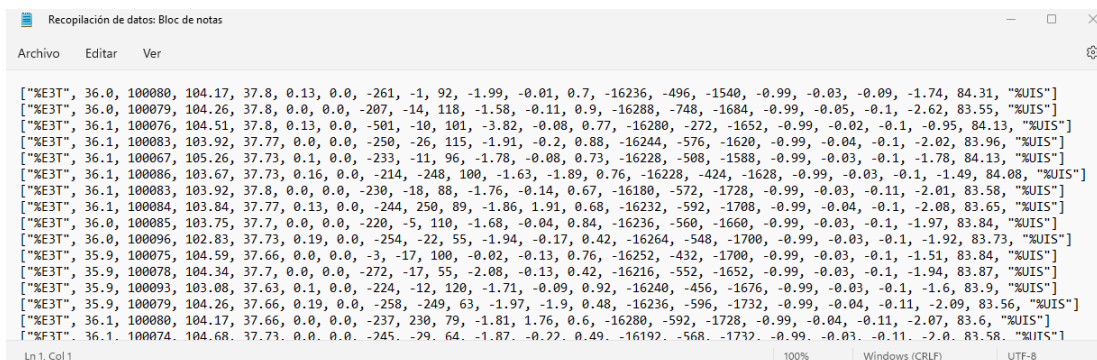
y agregue lo siguiente. La segunda le indica a la variable “sensor1_1” que tome el valor decodificado de la trama que llega que en este caso es denominado “sensor1”. En la tercera línea se aprecia como a la variable inicial se le dice que escriba los datos de “sensor1_1” sumando “os.linesep” que significa que tan pronto entre otro dato de la trama se escriba en el siguiente renglón. Por último se le da orden de cerrado a la variable inicial.

```
file1= open("c:/Users/edins/Documents/deber1python/Recopilación de datos.txt","a")
sensor1_1=json.dumps(sensor1)
file1.write(sensor1_1 + os.linesep)
file1.close()
```

Figura 4.13: Fragmento del código secundario para almacenamiento de las tramas

Ahora se agrega la Figura 4.14. Aquí se muestra que las tramas se van agregando en un archivo de texto llamado “Recopilación de dato” cada trama nueva se adiciona a un nuevo renglón.

Las tramas están compuestas por un vector fila. Inicialmente por las siglas de la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, seguido de los datos transmitidos por los sensores y finalmente las siglas de la Universidad Industrial de Santander.



```
["E3TR", 36.0, 100080, 104.17, 37.8, 0.13, 0.0, -261, -1, 92, -1.99, -0.01, 0.7, -16236, -496, -1540, -0.99, -0.03, -0.09, -1.74, 84.31, "E3TR"]
["E3TR", 36.0, 100079, 104.26, 37.8, 0.0, 0.0, -207, -14, 118, -1.58, -0.11, 0.9, -16288, -748, -1684, -0.99, -0.05, -0.1, -2.62, 83.55, "E3TR"]
["E3TR", 36.1, 100076, 104.51, 37.8, 0.13, 0.0, -501, -10, 101, -3.82, -0.08, 0.77, -16280, -272, -1652, -0.99, -0.02, -0.1, -0.95, 84.13, "E3TR"]
["E3TR", 36.1, 100083, 103.92, 37.77, 0.0, 0.0, -250, -26, 115, -1.91, -0.2, 0.88, -16244, -576, -1620, -0.99, -0.04, -0.1, -2.02, 83.96, "E3TR"]
["E3TR", 36.1, 100067, 105.26, 37.73, 0.1, 0.0, -233, -11, 96, -1.78, -0.08, 0.73, -16228, -508, -1588, -0.99, -0.03, -0.1, -1.78, 84.13, "E3TR"]
["E3TR", 36.1, 100086, 103.67, 37.73, 0.16, 0.0, -214, -248, 100, -1.63, -1.89, 0.76, -16228, -424, -1628, -0.99, -0.03, -0.1, -1.49, 84.08, "E3TR"]
["E3TR", 36.1, 100083, 103.92, 37.8, 0.0, 0.0, -230, -18, 88, -1.76, -0.14, 0.67, -16180, -572, -1728, -0.99, -0.03, -0.11, -2.01, 83.58, "E3TR"]
["E3TR", 36.1, 100084, 103.84, 37.77, 0.13, 0.0, -244, 250, 89, -1.86, 1.91, 0.68, -16232, -592, -1708, -0.99, -0.04, -0.1, -2.08, 83.65, "E3TR"]
["E3TR", 36.0, 100085, 103.75, 37.7, 0.0, 0.0, -220, -5, 110, -1.68, -0.04, 0.84, -16236, -560, -1660, -0.99, -0.03, -0.1, -1.97, 83.84, "E3TR"]
["E3TR", 36.0, 100096, 102.83, 37.73, 0.19, 0.0, -254, -22, 55, -1.94, -0.17, 0.42, -16264, -548, -1700, -0.99, -0.03, -0.1, -1.92, 83.73, "E3TR"]
["E3TR", 35.9, 100075, 104.59, 37.66, 0.0, 0.0, -3, -17, 100, -0.02, -0.13, 0.76, -16252, -432, -1700, -0.99, -0.03, -0.1, -1.51, 83.84, "E3TR"]
["E3TR", 35.9, 100078, 104.34, 37.7, 0.0, 0.0, -272, -17, 55, -2.08, -0.13, 0.42, -16216, -552, -1652, -0.99, -0.03, -0.1, -1.94, 83.87, "E3TR"]
["E3TR", 35.9, 100093, 103.08, 37.63, 0.1, 0.0, -224, -12, 120, -1.71, -0.09, 0.92, -16240, -456, -1676, -0.99, -0.03, -0.1, -1.6, 83.9, "E3TR"]
["E3TR", 35.9, 100079, 104.26, 37.66, 0.19, 0.0, -258, -249, 63, -1.97, -1.9, 0.48, -16236, -596, -1732, -0.99, -0.04, -0.11, -2.09, 83.56, "E3TR"]
["E3TR", 36.1, 100080, 104.17, 37.66, 0.0, 0.0, -237, 230, 79, -1.81, 1.76, 0.6, -16280, -592, -1728, -0.99, -0.04, -0.11, -2.07, 83.6, "E3TR"]
["E3TR", 36.1, 100074, 104.68, 37.73, 0.0, 0.0, -245, -79, 64, -1.87, -0.22, 0.49, -16192, -568, -1732, -0.99, -0.03, -0.11, -2.0, 83.58, "E3TR"]
```

Figura 4.14: Archivo de texto con la tramas almacenadas

4.5.2. Imágenes

Ahora se tiene la Figura 4.15, donde la primera línea decodifica la imagen en Base64 y la agrega a la variable “f2”. La segunda le indica a la variable “img” que abra como imagen

la variable “f2” almacenados en bytes dentro de un búfer. En la tercera línea se le indica a “F” que tome la variable “img” y la almacene con la librería de Tkinter como imagen. En la cuarta línea se le indica a “x1” que tome la cadena de texto de la variable “x”. Para la quinta línea le decimos “img” que se guarde con el nombre “filename” más el valor de “x1” para que así se explique cual es la imagen 1, 2, 3 y así sucesivamente, y finaliza con “.jpg”. En la séptima línea se le dice a la variable “F0” que abra la imagen que ha sido guardada anteriormente, de esta forma cada que la variable “x” aumente como un la ultima línea, la séptima va a mostrar la imagen que ha sido guardada en el número en el que se encuentra.

```
f2=base64.b64decode(data)
img=Image.open(io.BytesIO(f2))
F=ImageTk.PhotoImage(img)
x1=str(x)
img.save('filename'+x1+'.jpg')
#print(type(F))
F0=Image.open('filename'+x1+'.jpg')
#F=ImageTk.PhotoImage(F0)
#self.Ph.set(img)
x += 1
```

Figura 4.15: Fragmento del código secundario para almacenamiento de imágenes

Ahora en la Figura 4.16 se muestra la carpeta general que contiene los datos de almacenamiento tanto de la trama de datos recopilada como las imágenes, además de los códigos realizados en Python y logos de la interfaz. Esto indica que las imágenes son guardadas por defecto en la carpeta en el que está guardado el proyecto.

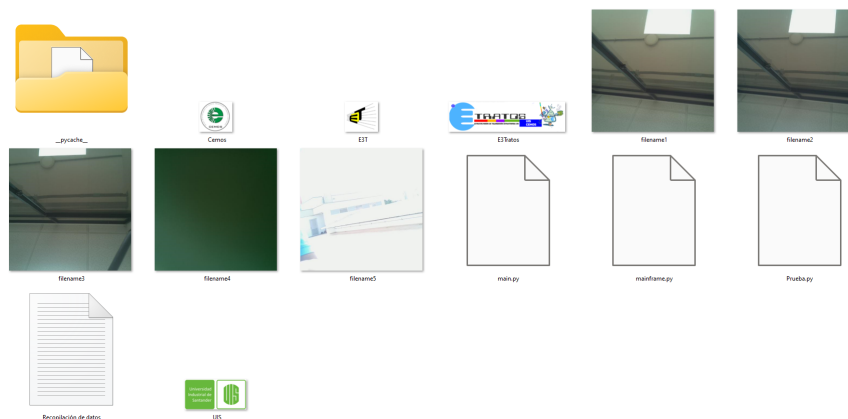


Figura 4.16: Captura de pantalla del guardado de imágenes

El funcionamiento de este sistema es el siguiente, entra una primera imagen con el nombre “filename1”, la segunda con nombre “filename2” y así sucesivamente se van guardando en la carpeta inicial, pero cada vez que sea ejecutado el código y empiece a recibir imágenes, se volverá a crear de nuevo una imagen con el nombre “filename1” reescribiendo las imágenes que anteriormente tenían este nombre debido a que la variable “x” vuelve a su estado inicial, se le recomienda al lector siempre que se realice una prueba, tomar estas y guardarlas en otra carpeta.

4.6. Montaje final

En la siguiente figura se aprecia el montaje final de este prototipo, a la derecha la estructura del sistema interno de la góndola, a la izquierda, en un trípode tenemos la antena Yagi-Uda de 900[MHz] y el radio receptor conectado a la computadora portátil la cual está ejecutando el código donde muestra la interfaz.

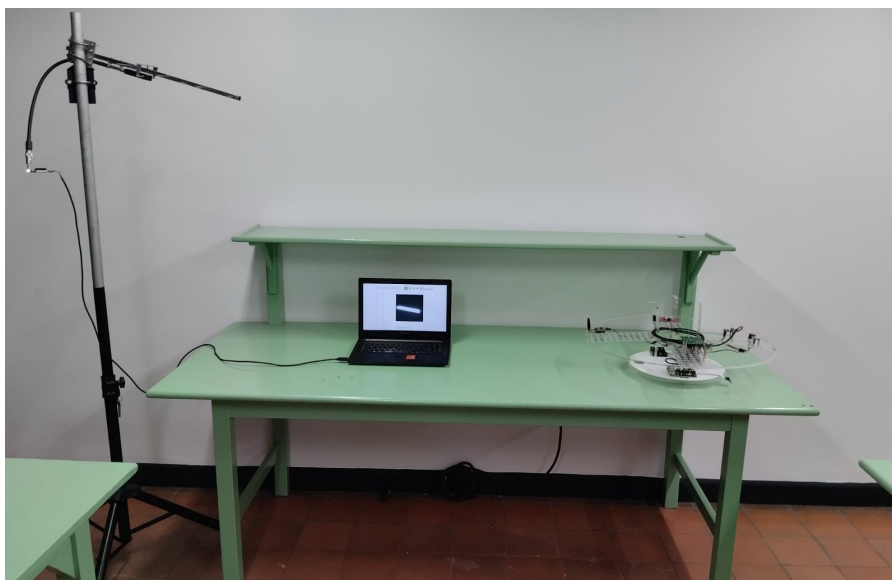


Figura 4.17: Arquitectura mecánica final

En la siguiente figura, mostramos una imagen tomada de cerca al sistema de la electrónica montada en la góndola.

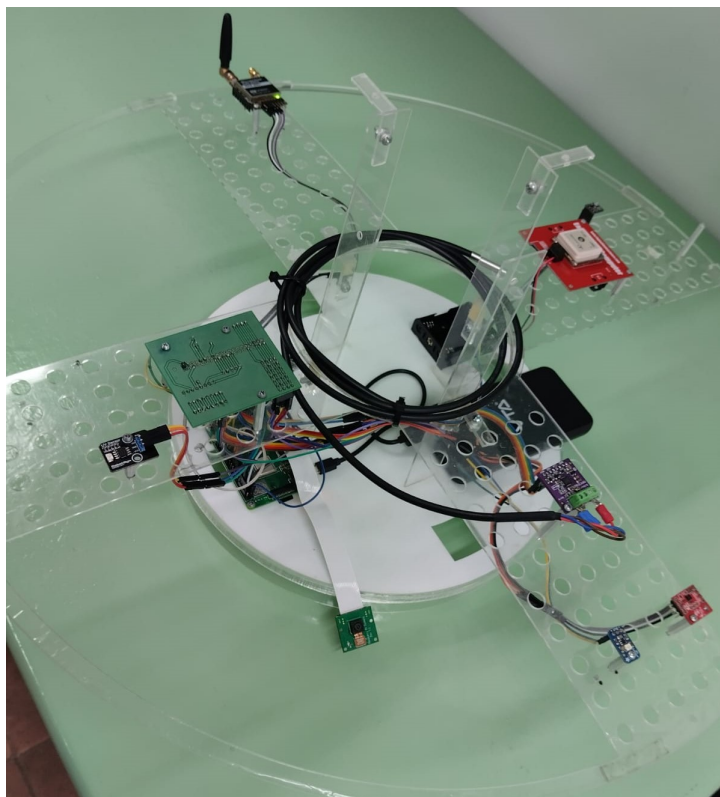


Figura 4.18: Sistema interno de la Góndola

En la Figura 4.18 se ve todos los dispositivos electrónicos conectados a la mini computadora. Adicional a esto, también aparece el módulo GPS Eagle Flight Computer de High Altitude Science que para este proyecto no sé utilizó debido a que era autónomo y no dependía de la Raspberry, sin embargo, a continuación se muestra la Figura 4.19 que muestra la prueba realizada a este donde nos indica: fecha UTC, hora UTC, latitud, longitud, el cabezal, la velocidad [km/h], Altura [m] y el bloqueo con el satélite.

Este dispositivo es alimentado por tres pilas alcalinas AA de 1.5[V], y este almacena los datos en una micro SD donde guarda las ubicaciones por donde estuvo como un tipo de huella. Además se pueden adicionar dos sensores de la misma marca, ya sean de presión o de temperatura.

```

Thank you for flying with High Altitude Science - EXPLORE YOUR WORLD!

Firmware Version 1.5

S1 Data Port - No Sensor Detected
S2 Data Port - No Sensor Detected

UTC Date,UTC Time,Latitude ,Longitude ,Head,Km/h,Alt-m ,Lock,
06/26/12,02:14:48,+000.00000,+000.00000,0000,0000,+000000,0001,
04/18/35,02:14:54,+000.00000,+000.00000,0000,0000,+000000,0001,
04/18/35,02:15:00,+000.00000,+000.00000,0000,0000,+000000,0001,
04/18/35,02:15:06,+000.00000,+000.00000,0000,0000,+000000,0001,
04/18/35,02:15:12,+000.00000,+000.00000,0000,0000,+000000,0001,
04/18/35,02:15:18,+007.10145,-073.12211,0000,0000,+000944,0003,
04/18/35,02:15:24,+007.10145,-073.12212,0000,0000,+000943,0003,
04/18/35,02:15:30,+007.10145,-073.12212,0000,0000,+000944,0003,
04/18/35,02:15:36,+007.10145,-073.12213,0000,0000,+000944,0003,
04/18/35,02:15:42,+007.10145,-073.12213,0000,0000,+000944,0003,
04/18/35,02:15:48,+007.10145,-073.12213,0000,0000,+000944,0003,
04/18/35,02:15:54,+007.10145,-073.12213,0000,0000,+000944,0003,
04/18/35,02:16:00,+007.10145,-073.12213,0000,0000,+000944,0003,
04/18/35,02:16:06,+007.10146,-073.12212,0000,0000,+000944,0003,
04/18/35,02:16:12,+007.10146,-073.12212,0000,0000,+000944,0003,
04/18/35,02:16:18,+007.10146,-073.12212,0000,0000,+000944,0003,
04/18/35,02:16:24,+007.10146,-073.12212,0000,0000,+000944,0003,
04/18/35,02:16:30,+007.10146,-073.12212,0000,0000,+000944,0003,
04/18/35,02:16:36,+007.10146,-073.12212,0000,0000,+000944,0003,
04/18/35,02:16:42,+007.10146,-073.12212,0000,0000,+000945,0003,
04/18/35,02:16:48,+007.10146,-073.12212,0000,0000,+000945,0003,
04/18/35,02:16:54,+007.10146,-073.12212,0000,0000,+000945,0003,
04/18/35,02:17:00,+007.10146,-073.12212,0000,0000,+000945,0003,
04/18/35,02:17:06,+007.10146,-073.12212,0000,0000,+000945,0003,
04/18/35,02:17:12,+007.10146,-073.12212,0000,0000,+000946,0003,
04/18/35,02:17:18,+007.10146,-073.12212,0000,0000,+000946,0003,
    
```

Figura 4.19: Datos de ubicación GPS Eagle Flight Computer

4.7. Interfaz de la Base terrena

En la Figura 4.20 se tiene una captura de pantalla ampliada de los datos de llegada que se detallaban en la figura anterior.



Figura 4.20: Interfaz gráfica final

Donde:

1. Muestra los datos de sensores que envía la góndola.
2. La ubicación de la imagen que va llegando, esta se actualiza cada que recibe otra.
3. En esta sección se ubica el título que tiene la interfaz.
4. Aquí se sitúan los logos de la Universidad Industrial de Santander, el semillero de investigación CEMOS, la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, y finalmente el de la misión E3Tratos.
5. Aquí ponemos el nombre de la universidad anteriormente nombrada junto con el año.
6. Por último este es el nombre que obtiene la ventana de Tkinter.

4.8. Ubicación y resultados de las pruebas de comunicación

Durante las pruebas de comunicación, se ubicaron las antenas en cuatro puntos adentro y fuera de la Universidad Industrial de Santander en Bucaramanga con el fin de tomar referencias para la extrapolación de los datos y finalmente validar la transmisión y recepción a más de 20[km]. Las ubicaciones de estos puntos fueron los siguientes:

- Estadio 1° de Marzo
- Edificio Teatrino - Edificio de Administración III
- Edificio Teatrino - Camilo Torres (UIS-75)
- Edificio Teatrino - Edificio Santa Isabella(A las afueras de la UIS)

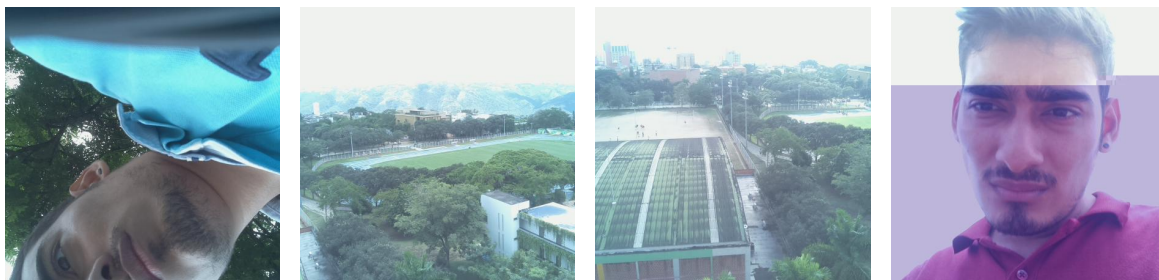
La ubicación de las últimas tres pruebas priorizaron el edificio Teatrino para situar la parte transmisora, mientras que la parte receptora se ubicó en los edificios de Administración III, Camilo Torres (UIS-75) y Santa Isabella. En la Figura 4.21a y 4.21b se ven las ubicaciones de las antenas para la comunicación de la primera y segunda prueba respectivamente.



(a) Estadio 1° de Marzo

(b) Edificio Administración III

Figura 4.21: Ubicación de las antenas durante las pruebas



(a) Estadio

(b) Administración

(c) Camilo T.

(d) S. Isabella

Figura 4.22: Muestras de las pruebas realizadas en las cuatro ubicaciones

Analizando la Figura 4.22 vemos que las muestras de las fotos tomadas en tres de las cuatro ubicaciones llegaron sin pérdidas, sin embargo, la última ubicación fue la que generó pérdida de pocos bits en su transmisión, para entender esto se analizaron los valores de RSSI y presupuestos de potencia.

4.9. Valores de RSSI según la distancia

Inicialmente en laboratorio se determinó que los datos prevalecían con una potencia aproximada a 30[dBm] a una distancia entre antenas de 10[cm]. La gráfica que se ve en

la Figura 4.23 nos muestra la señal de RSSI, donde “x” es tiempo y “y” es el valor RSSI quien para ser obtenido en decibelios se utiliza la Ecuación 2.5. Al reemplazar se obtiene la Ecuación 4.3.

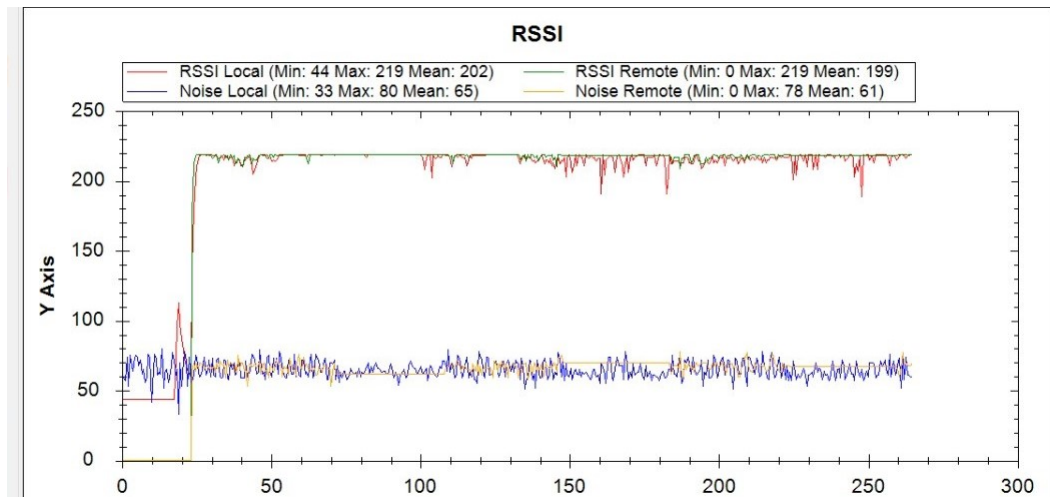


Figura 4.23: RSSI principal

$$RSSI(dBm) = (219/1,9) - 127 \quad (4.2)$$

$$RSSI(dBm) = -11,7 \quad (4.3)$$

Ahora se observa la Tabla 2.1, donde nos indica que la señal de $-11,7[\text{dBm}]$ presenta calidad excepcional para la comunicación.

Luego se importa una gráfica de análisis para indicar los valores de RSSI de las cuatro pruebas de comunicación, esta se aprecia en la Figura 4.24.



Figura 4.24: Calidad de la señal RSSI en decibelios para las pruebas de comunicación

Al hacer un examen de estos datos sobre la calidad de la señal, se verifica que la transmisión y recepción son favorables, por lo que ahora se realizará un análisis a los presupuestos de potencia para cada una de estas pruebas y así descartar que sean problemas de enlace entre los dos puntos.

4.10. Pruebas de presupuesto de potencia

En esta sección se toman los puntos en los que se ubicaron las antenas y sus distancias, fijando como parámetro una sensibilidad de $-90[\text{dBm}]$ aún sabiendo que la máxima es de $-121[\text{dBm}]$, debido a que este es el límite de sensibilidad en el que el margen de operatividad sigue siendo exitoso para la prueba más lejana.

A continuación se muestra la tabla de resultados reemplazando valores en las Ecuaciones 2.1, 2.2, 2.3 y 2.4.

Tabla 4.1: Presupuesto de potencia de las pruebas

Ubicación	PIRE	FSL	PP	Margen
Estadio 1° de Marzo (185m)	31,1[dBm]	77[dB]	-37,9[dB]	52,1[dB]
Teatrino - Admon. III (285m)	31,1[dBm]	80,8[dB]	-41,7[dB]	48,3[dB]
Teatrino - Camilo T. (460m)	31,1[dBm]	84,98[dB]	-45,88[dB]	44,12[dB]
Teatrino - Santa Isabella (1km)	31,1[dBm]	91,73[dB]	-52,63[dB]	37,37[dB]

Teniendo ya estas pruebas y validando tanto teórico como práctico los valores de la calidad de señal y el éxito del enlace, se le arraiga este problema a dos factores.

Primero, el modem RFD900+ está conectado directamente a el puerto UART y el GPIO 5V PWR de la Raspberry, este GPIO puede llegar a entregar la misma corriente de la fuente de la mini computadora, sin embargo, esta a largas distancias puede no entregar la potencia necesaria que exija el radio, por lo que la solución fue conectar este de manera externa a la Raspberry, directamente al otro puerto USB del banco de baterías VTA.

Segundo, los cables utilizados para estas pruebas fueron cables tipo Dupont llamados comúnmente “Jumpers”, estos cables son capaces de soportar hasta 1[A] a temperatura ambiente. A largas distancias el radio RFD900+ demanda cierta cantidad de corriente que quizás estos cables no puedan soportar como lo visto en la Figura 4.7b.

Dicho lo anterior, se le recomienda al lector tener en cuenta estos factores a la hora de un lanzamiento, con la intención de evitar falencias en la misión.

4.11. Pruebas en Radio Mobile para comunicación

Radio-Link es una herramienta que viene con el programa Radio Mobile el cual nos permite ver las simulaciones realistas de redes configuradas para observar la calidad del radio enlace, los obstáculos presentes, el perfil de elevación, la distancia entre las antenas, etc. Esto con la intención de validar la comunicación a largas distancias.

4.11.1. Facultad de Ingeniería Fisicomecánicas - Aeropuerto Palonegro

Para estas pruebas de larga distancia se tiene la antena receptora Yagi-Uda en la ubicación del mirador en el Aeropuerto Palonegro de Bucaramanga, y la antena monopolar de 1/4 de onda transmisora ubicada en el edificio de la Facultad de Ingeniería Fisicomecánicas (Laboratorios Pesados) con una distancia de 6.57[km].

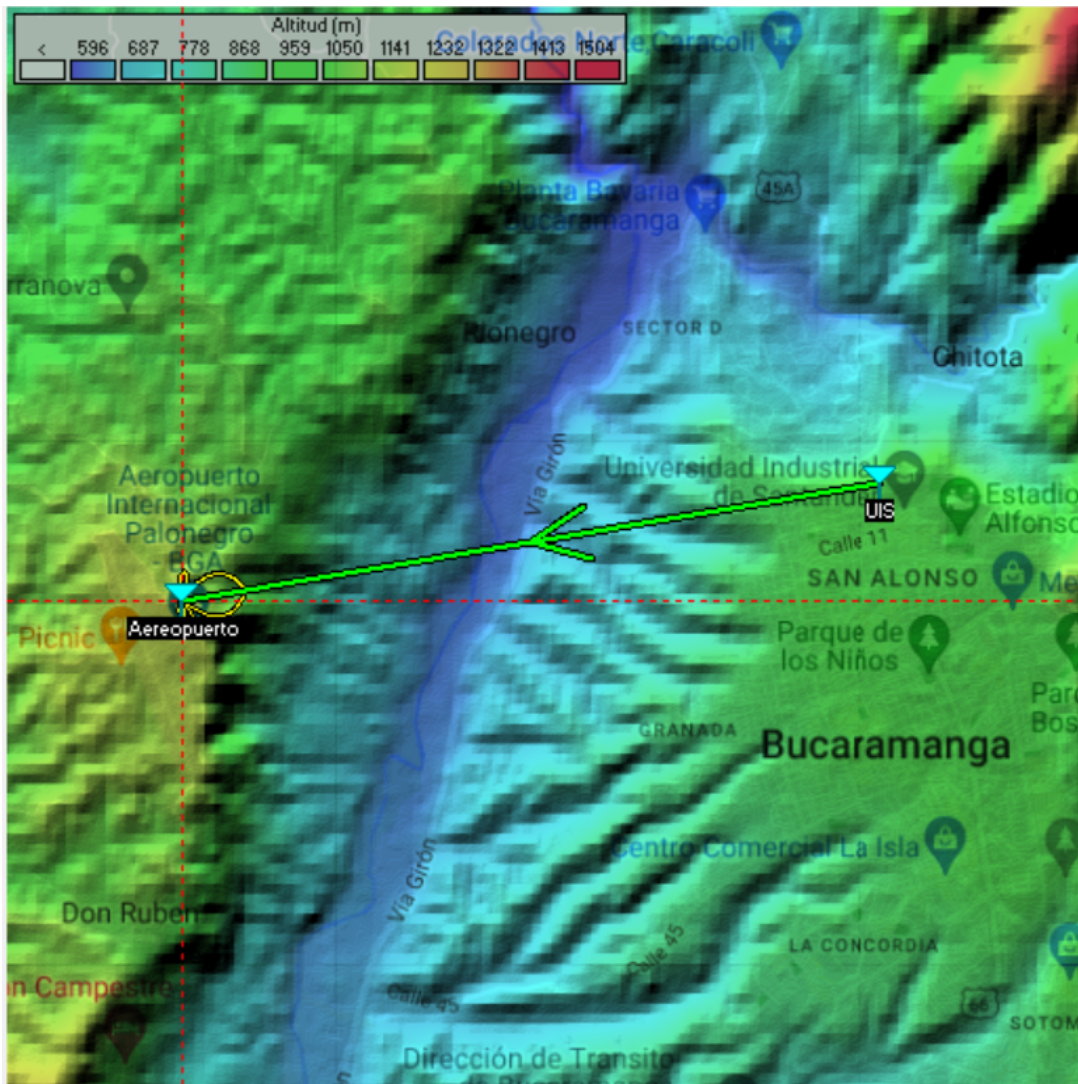


Figura 4.25: Mapa UIS - Aeropuerto Palonegro Radio Mobile

En la Figura 4.25 se puede apreciar el patrón de radiación de la antena Yagi-Uda apuntando hacia la omnidireccional ubicada en la UIS y la flecha de color verde indicando hacia donde se transmiten los datos.

Se aprecia que en la Figura 4.26 se encuentra todo lo relacionado con el radio enlace creado donde aparece la potencia de transmisión, la ganancia de las antenas, la sensibilidad que se ha estado configurando e incluso la altura a las que están ubicadas las antenas, además se ve la distancia a la que están las dos antenas y el margen de operatividad que en este caso es el “RX relativo”.

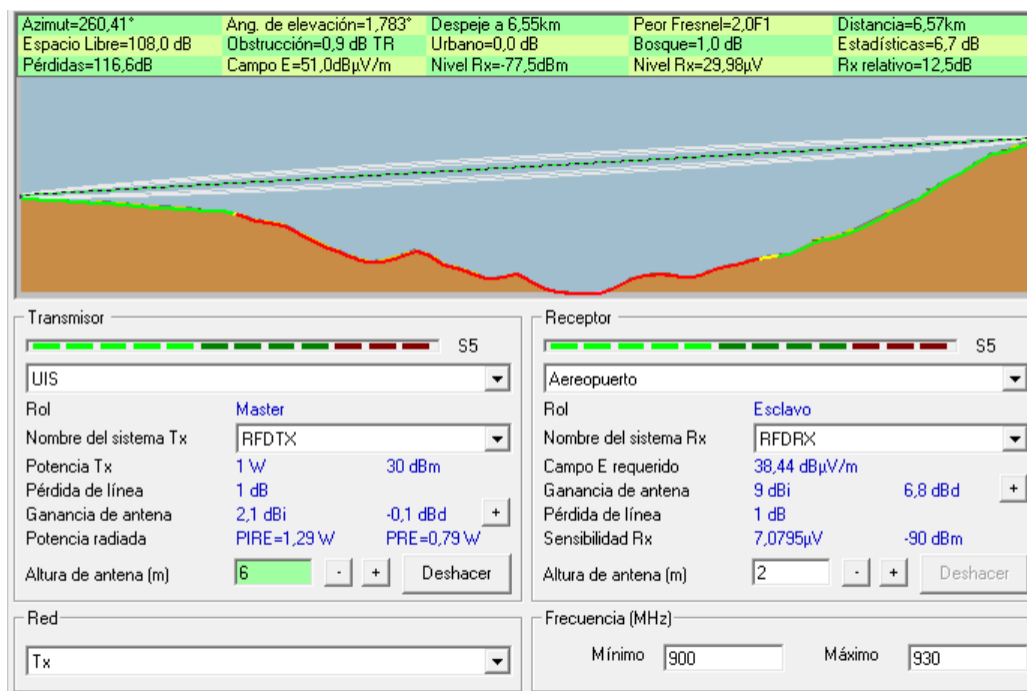


Figura 4.26: Radio enlace UIS - Aeropuerto Palonegro Radio Link

Ahora, visto lo anterior verificamos que el margen de operatividad sea el mismo obtenido en simulación, por lo tanto se reemplazan las ecuaciones de potencia para saber los resultados y compararlos con lo anterior.

$$PIRE = 30[dBm] - 1[dB] + 2,1[dBi] = 31,1[dBm] \quad (4.4)$$

El PIRE sigue siendo igual a los casos anteriores.

$$FSL = 20Log(0,915) + 20Log(6,57) + 92,5[dB] = 108,08[dB] \quad (4.5)$$

$$PP = 31,1[dB] - 108,08[dB] + 9[dBi] - 1[dB] = -68,98[dB] \quad (4.6)$$

$$Margen = -68,98[dB] - (-90[dBm]) = 21,02[dB] \quad (4.7)$$

Ya teniendo el resultado de margen de operatividad, ahora lo comparamos con el anterior obtenido en simulación y se percata que hay una incoherencia a la hora de ver estos valores y es que, en la simulación tenemos un “Rx relativo” de 12.5[dB] a diferencia de el margen obtenido por ecuaciones que es de 21.02[dB], por lo que se hace un análisis de las pérdidas que nos muestra la simulación y esto es debido a que Radio-Link adiciona datos de pérdida como obstrucción, urbanización, bosques y estadísticos iguales a -116,6[dB], por lo tanto se modifican en las pérdidas del espacio libre y verificamos que estos datos son exactos.

$$PP = 31,1[dB] - 116,6[dB] + 9[dBi] - 1[dB] = -77,5[dB] \quad (4.8)$$

$$Margen = -77,5[dB] - (-90[dBm]) = 12,5[dB] \quad (4.9)$$

Una vez corregido, validamos que en efecto estos datos teóricos comparados con los de simulación concuerdan, a lo que damos la deducción de que para estas distancias siendo aún así el margen de operatividad de 12.5[dB], es exitoso el radio enlace.

El lector debe recordar que se ha estado usando una sensibilidad de -90[dBm] inferior a la máxima que pueden tener los radios de comunicación usados en este proyecto.

4.11.2. Aeropuerto Palonegro - Peaje Picacho

Para ir finalizando esta sección se desarrolla la última simulación de prueba para llegar al objetivo de distancia propuesto para el éxito de este proyecto, en este caso los puntos de ubicación fueron el Aeropuerto Palonegro(Antena monopolar de 1/4 de onda transmisora) y

el Peaje Picacho(Yagi-Uda) cumpliendo con una distancia de 23.16[km]

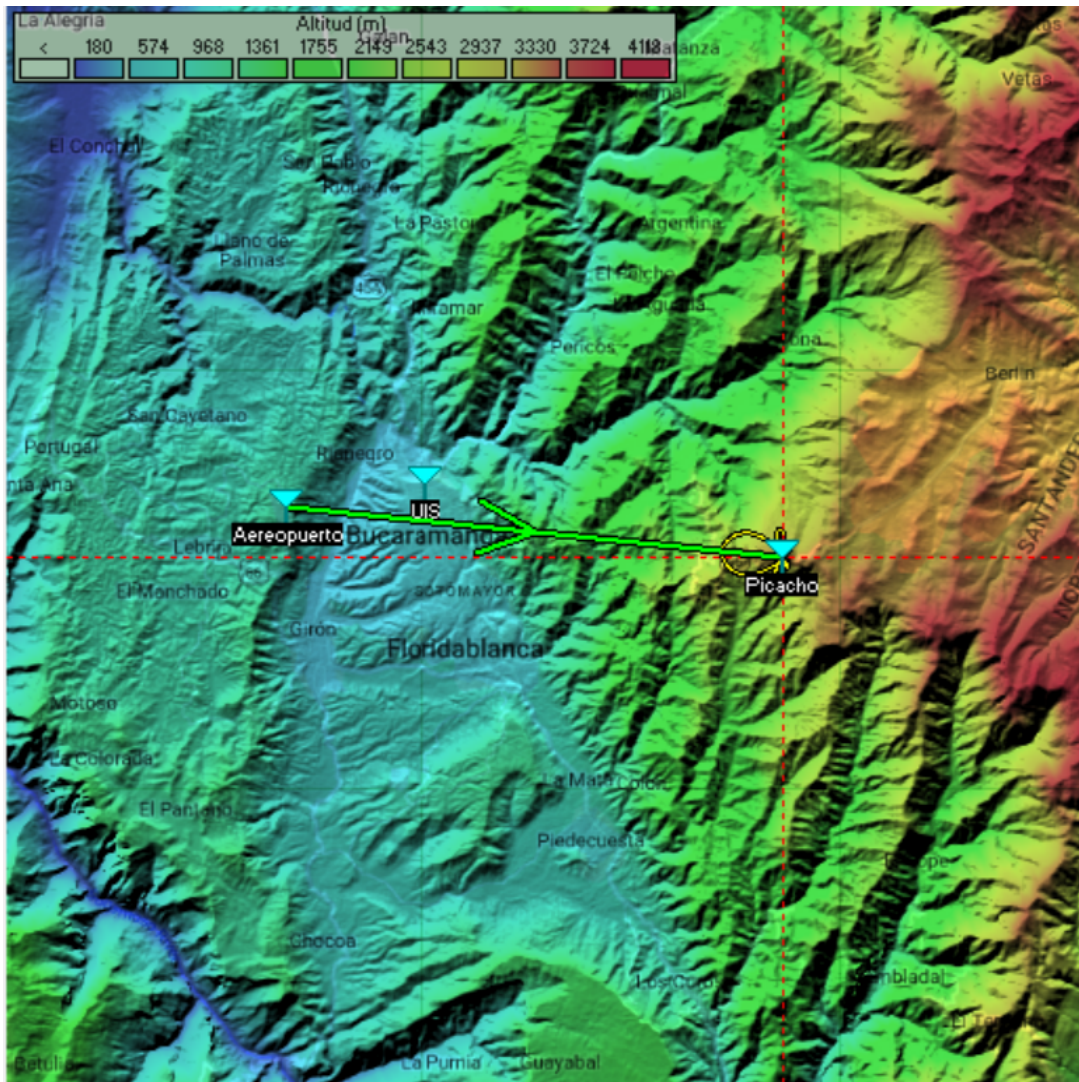


Figura 4.27: Mapa Aeropuerto Palonegro - Peaje Picacho Radio Mobile

Se puede ver en la Figura 4.27 los puntos ubicados en el mapa donde la flecha verde indica hacia donde viajan los datos, también vemos que está el punto anterior “UIS” que en este caso es usado de referencia para tener una idea de las distancias.

Ahora en la Figura 4.28 se visualizan los datos en simulación del radio enlace donde se aprecia que no hay interferencias en las zonas de Fresnel. El “Rx relativo” en este caso es de 3.3[dB], así que ahora lo comparamos con los datos teóricos.

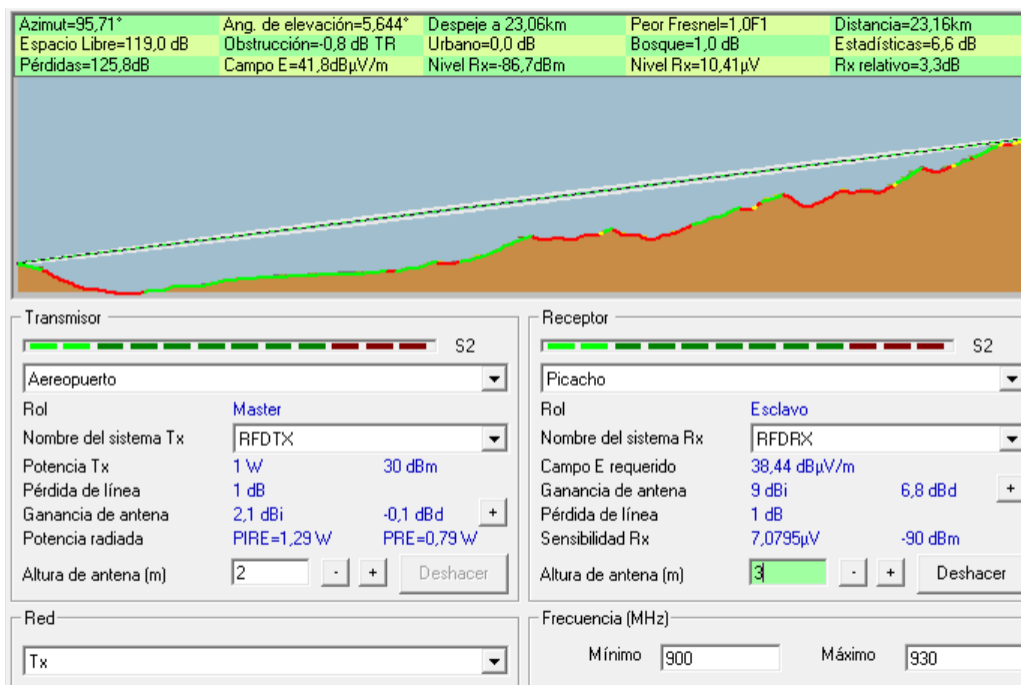


Figura 4.28: Radio enlace Aeropuerto Palonegro - Peaje Picacho Radio Link

Se inicializa de nuevo con el PIRE, valor que se sigue manteniendo.

$$PIRE = 30[dBm] - 1[dB] + 2,1[dBi] = 31,1[dBm] \quad (4.10)$$

$$FSL = 20Log(0,915) + 20Log(23,16) + 92,5[dB] = 119[dB] \quad (4.11)$$

$$PP = 31,1[dB] - 119[dB] + 9[dBi] - 1[dB] = -79,9[dB] \quad (4.12)$$

$$Margen = -79,9[dB] - (-90[dBm]) = 10,1[dB] \quad (4.13)$$

Esta es la prueba más lejana, y aquí es donde se explica el por qué de haber decidido $-90[dBm]$ de sensibilidad cuando se sabe que la máxima es $-121[dBm]$, y es que con este valor aún sigue siendo eficiente el enlace tomado de las ecuaciones de presupuesto de potencia,

sin embargo, en simulación genera un valor por debajo de 10[dB], así que se desarrolla el mismo paso de restar las pérdidas de la simulación al margen de operatividad para validar que funciona.

$$PP = 31,1[dB] - 125,8[dB] + 9[dBi] - 1[dB] = -86,7[dB] \quad (4.14)$$

$$Margen = -86,7[dB] - (-90[dBm]) = 3,3[dB] \quad (4.15)$$

Como se aprecia, se ha llegado a los datos de “Rx relativo” y aunque en este caso para la simulación el margen dice que no es exitoso el radio enlace, hay que tener en cuenta que se trabajó con una sensibilidad por debajo de los límites a propósito para que el lector tenga en cuenta que cuando estos datos teóricos pasan a ser pruebas en un ambiente práctico, influyen muchas variables que pasan desapercibido a simple vista y por ende no entregan los datos exactos que se obtuvieron en la teoría. Si se requiere que el enlace sea exitoso hasta para esta distancia, tendría que tener una sensibilidad de -96,8[dBm] en adelante.

Ahora se afirma lo siguiente, como se ha dicho desde un inicio, estos radios son de comunicación de línea de vista directa y fueron diseñados para dispositivos aéreos no tripulados, en este caso al realizar pruebas en tierra se está limitando el potencial que tienen estos gracias a las pérdidas e interferencias que pueden incidir entre la comunicación de estos dos dispositivos, como ya se ha visto en el transcurso de todas estas pruebas.

5. Trabajo futuro

Este capítulo es incluido especialmente para aquellos lectores que deseen profundizar en este proyecto, donde se proponen dispositivos que pueden llegar a satisfacer el desempeño del globo sonda de la misión E3Tratos. En este libro estos no fueron agregados debido a razones tanto técnicas como económicas, sin embargo, fueron seleccionados luego de una investigación realizada a las variables atmosféricas y sus parámetros.

5.1. Sensores

Tabla 5.1: Sensores ideales

Referencia	Variable	Protocolo	Rango
PT100	Temperatura	SPI	-200 a 650[°C]
TMP116	Temperatura	I2C	-55 a 125[°C]
LM 75B	Temperatura	I2C	-55 a 125[°C]
MS 8607	Humedad	I2C	0 a 100[% RH]
HDC 2010	Humedad	I2C	0 a 100[% RH]
MS5611-01BA03	Presión	SPI/I2C	1 a 120[kPa]
MS 8607	Presión	I2C	300 a 1200[mbar]
OPT300	Radiación	I2C	0,01[lux] a 83[klux]
VEML 3328	Radiación	I2C	320 a 410[nm]
VEML 6070	Radiación	I2C	320 a 410[nm]
DRV5055	Campo magnético	I2C	± 21 a ± 169 [mT]
HMC5883L	Campo magnético	I2C	± 8 [G]
DS0001 DYNAMENT	Concentración de CO ₂	UART	0 a 500[ppm]
MQ 131	Concentración de O ₃	Analog./TTL	200 a 10000[ppm]

Ahora se muestra la Tabla 5.1, donde podemos encontrar estos sensores con su referencia, variable que mide, el protocolo de comunicación y el rango de medida.

5.2. Cámaras

En este libro se llevo a cabo el diseño con una cámara de Raspberry, para validar funcionamiento y transmisión de ellas por medio de los radios RFD900+, sin embargo, se le recomienda como trabajo futuro al lector implementar las siguientes opciones de diseño.

5.2.1. GoPro Max360

Esta es la cámara que se pensaría llevar de manera externa, más específicamente en un brazo saliente de la góndola, la idea es que se pueda obtener una vista de 360° durante el ascenso del globo, sin embargo, al adicionar un brazo extra a la estructura de la góndola, su centro de masa cambiará, si se desea hacer esto, se recomienda realizar un estudio para cambios estructurales de este. Sus características se muestran a continuación.

- Captura tradicional HERO de GoPro y captura de 360° Estabilización de vídeo Max HyperSmooth
- Reducción del ruido del viento avanzada con procesamiento de 6 micrófonos | Audio estéreo de rendimiento superior | Audio de 360° | Captura de audio RAW (modo HERO, formato .wav)
- Wi-Fi y Bluetooth | Compatible con GPS | Conexión a la aplicación GoPro
- Vídeo de 360°: Original de 6K/combinado de 5,6K, original/combinado de 3K; Vídeo HERO: 1440p, 1080p a 60fps con duración de 1,5 a 2 horas.
- Fotos esféricas, foto HERO, PowerPano, más ajustes de foto
- TimeWarp de 360°, TimeWarp HERO, Vídeo secuencial de 360°, Foto secuencial de 360°, Vídeo secuencial HERO, Foto secuencial HERO

En la siguiente Figura 5.1 podemos ver una imagen de esta cámara



Figura 5.1: GoPro Max

Tomado de: <https://gopro.com>

5.2.2. Camarray Hat Arducam

Esta es una tarjeta diseñada por Arducam la cual permite aumentar la cantidad de cámaras que puede usar la Raspberry gracias a sus cuatro módulos sincronizados MIPI para finalmente conectarse a la mini computadora por el puerto individual que trae. La idea principal, una vez conectadas las cuatro cámaras, es ubicarlas en dirección a los cuatro puntos cardinales alrededor de la góndola como se ve en la Figura 5.2.

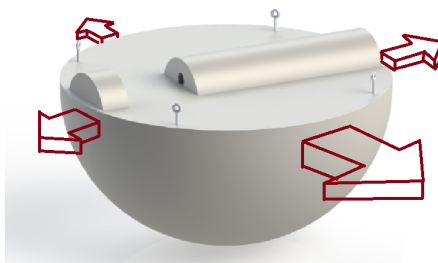


Figura 5.2: Ubicación de las Cámaras para Raspberry Pi

Tomado de (Altamar Amaranto, Fajardo Ariza, C.A., Rodriguez Ferreira, J.G., and Universidad Industrial de Santander. Escuela de Ingenierías Eléctrica, E. y de T., 2019)

Si se llega a tener esta configuración de cámaras durante la misión, se cubriría la visión horizontal de esta y así registrar mejor el ascenso y descenso. A continuación vemos una imagen de esta tarjeta en la Figura 5.3.



Figura 5.3: Paquete de cámaras cuadrascópica Arducam

Tomado de: <https://cutt.ly/31xqhfB>.

5.3. Alimentación

Durante las pruebas se usó el banco de baterías VTA, sin embargo, se encontró un banco de baterías con mayor capacidad de almacenamiento que podría suplir la arquitectura electrónica de la góndola mejor que la usada en pruebas.

5.3.1. INIU 20000[mAh]

Este es un banco de baterías de polímero de litio con capacidad de 20000[mAh]/74.5[Wh] con entrada de 5[V] a 2[A] de 20[W] y con salida de 5[V] a 3[A] entregando 22.5[W]. Incluye tres puertos, dos USB tipo A de salida y un USB tipo C, este último de entrada y salida. Adicional a esto, cuenta con función de linterna y una pantalla siete segmentos para indicar

el porcentaje de carga que posee la batería. En la Figura 5.4 se puede ver este banco de baterías.



Figura 5.4: Banco de Baterías de polímero de litio INIU 20000mAh

Tomado de:<https://cutt.ly/R1AWmrt>

A continuación se analiza la autonomía de la batería con los datos de consumo expuestos en la sección de alimentación del capítulo 4. Primero evaluamos con el consumo de la Raspberry reproduciendo vídeo en la Ecuación 5.1

$$\text{Autonomía}(h) = \frac{10000[mAh]}{1,36[A]} = 14,7[h] \quad (5.1)$$

Aquí se puede ver un tiempo estimado de siete horas con cuarenta y dos minutos(7[h] y 42[min]). Este valor duplica el estimado de horas que puede proporcionar la autonomía de la batería VTA.

Ahora se evalúa con el consumo de la Raspberry en estrés máximo, y como se aprecia la Ecuación 5.2, se ve que al tener este modo de consumo, la Raspberry junto con los dispositivos seleccionados, alcanza un tiempo estimado de nueve horas con dieciocho minutos(9[h] y 18[min]).

$$\text{Autonomía}(h) = \frac{20000[mAh]}{2,15[A]} = 9,3[h] \quad (5.2)$$

Al tener el doble de capacidad, esta batería supe el doble de horas que la anterior permitiendo una autonomía más alta que la usada en este libro.

Teniendo en cuenta las pruebas realizadas de consumo, tomamos la autonomía para este banco de baterías y se aprecia el doble de horas de trabajo de esta en la Ecuación 5.3 siendo 19 horas con treinta y seis minutos (19[h] y 36[*min*]).

$$Autonomía(h) = \frac{20000[mAh]}{1,02[A]} = 19,6[h] \quad (5.3)$$

6. Recomendaciones

- Un aspecto a tener en cuenta sobre la comunicación que se realiza es ser cuidadoso con el tema de antenas, en este caso para la transmisión de datos usamos una monopolar de 1/4 de onda con 2.1[dBi], si se desea un alcance mas alto, puede buscar una antena con mejor ganancia e incluso usar otro tipo de antena que especifique la hoja de datos de estos radios usados.
- Al realizar pruebas con la tarjeta Raspberry Pi 3B+ hay que ser precavido, siempre realizar una copia de seguridad en la cual pueda recuperar todo en caso de que la memoria micro SD entre en corrupción, como ocurrió en el desarrollo de este proyecto. La razón, debido a que las Raspberrys no tienen un botón de encendido ni apagado nativo, durante las pruebas de este proyecto simplemente se conectaba y desconectaba el cable de alimentación, esto hace que la Raspberry Pi se apague forzosamente lo que daña con el tiempo los datos de la micro SD y entran en corrupción ya que al ser una memoria tipo flash, tiene límite de borrados y escrituras, siendo casi imposible recuperarlos. La solución que se tomó para este problema, fue adaptar un botón de apagado denominado “Shutdown” con el fin de apagar de manera correcta la mini computadora y así evitar daños a la memoria. Existe otra opción para que esto no vuelva a pasar y es configurar la Raspberry en modo “Solo lectura” el cual no afecta de manera considerable la micro SD.
- Otro punto que no hay que dejar pasar, es que este proyecto fue realizado como un prototipo para realizar pruebas en tierra de toda la estructura, sin embargo, para un lanzamiento final, se recomienda tener en cuenta el estudio realizado en repositorios y lo siguiente:
 - La temperatura: esta variable afecta a la electrónica de manera considerable, ya que la mayoría de sensores, utilizados en este proyecto, alcanzan a trabajar hasta $-40[^\circ\text{C}]$, además, la batería también podría verse afectada. Para esto se escogieron

los sensores del capítulo de trabajo futuro, allí encontrará algunos sensores que pueden llegar a operar a muy bajas temperaturas.

- El conexionado: uno de las recomendaciones que se le hace al lector es el tema del cableado de todos los dispositivos internos en la Góndola, esto debido a que para estas pruebas se utilizaron “jumpers”, sin embargo, para un lanzamiento estos cables no son seguros ni inmunes a interferencias, para esto se recomienda utilizar cables apantallados para evitar estas EMI.
- La estructura: aclaramos que la estructura de este prototipo está diseñada en acrílico, este es muy frágil, incluso cuando se trabajó para este proyecto se quebró en dos puntos. Lo que se recomienda si se sigue trabajando con este material, es no dejar ninguna esquina sin bordes, ya que las esquinas con ángulos de 90° pueden generar una grieta y partir fácilmente este, además, si incrementa el grosor de este acrílico, puede llegar a conseguir mejor rigidez. En el caso de no seguir utilizando este material, enfocarse en lo principal, que sea resistente y ligero.

7. Conclusiones

- Analizando el capítulo de resultados, específicamente en las pruebas de comunicación y al tener en cuenta que estas fueron desarrolladas en ubicaciones con suficiente cantidad de interferencias junto con las pérdidas de espacio libre, se llega a la conclusión que durante el desarrollo de estas hay que ser precavidos sobre los puntos a los que se situarán los receptores y los transmisores ya que de no ser así, se podrán tener fallas durante el radio enlace o pérdidas de información que afecten estas pruebas.
- Gracias a que toda la arquitectura de código tanto primario como secundario fueron desarrollados en Python y con librerías del mismo, este proyecto es totalmente editable, desde como se transmiten los datos de comunicación hasta el como se llegan a visualizar en la interfaz desarrollada.
- También es válido resaltar que este proyecto fue llevado a cabo con una interfaz desarrollada en Python que a diferencia de las tesis anteriores, no depende de un servidor ni de estar conectado directamente a la red, solo basta con tener Visual Studio, descargar el código, actualizar las librerías (las cuales son sugeridas directamente por el programa), modificar el puerto “COM” de la computadora portátil en el cual se conecta el radio receptor y por último ser ejecutado. Al no depender de estar conectado a la red de internet, podemos tener la ventaja de ubicar nuestra base terrena en el punto que sea requerido, fuera de interferencias en zonas de campo abierto.
- Finalmente se corrobora que este proyecto es adaptable a un sistema de captura de imágenes cómo lo desarrollado en este libro, enviando varias tramas de datos seguido de una imagen. El tiempo de transmisión dependerá de la calidad, resolución y otros factores que incrementan el tamaño de esta.

8. Anexos

8.1. Repositorios

En el siguiente enlace se encuentra toda la información referente a esta investigación:
https://drive.google.com/drive/folders/1516v87xRdL5aZRPBB6IetfG7uEITAqP9?usp=share_link

8.2. Coeficiente de reflexión de cada antenna usada



Figura 8.1: Espectro de coeficiente de reflexión antenna monopolar

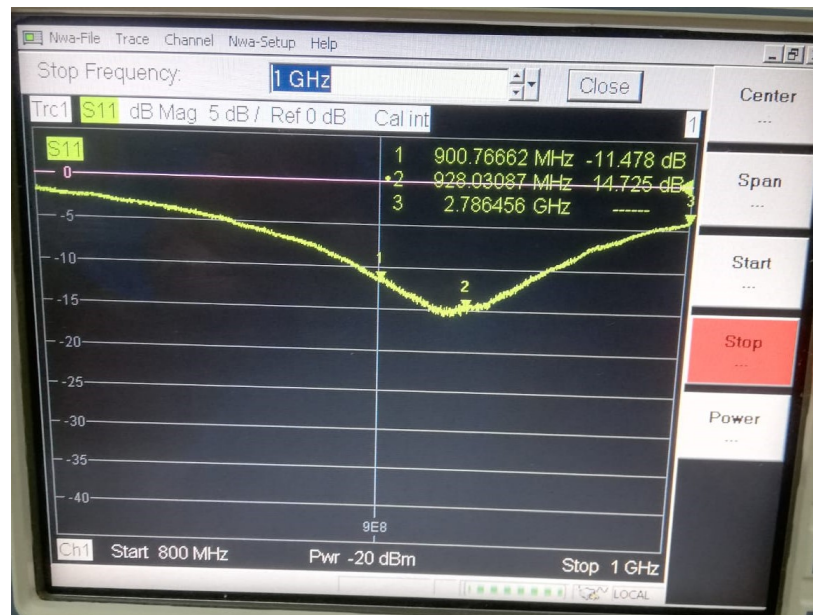
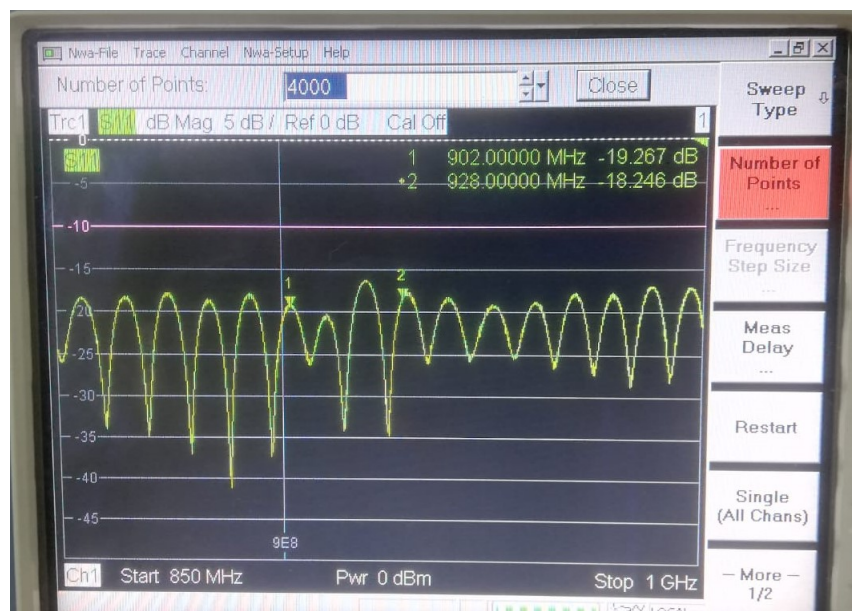
Figura 8.2: Espectro de coeficiente de reflexión antena monopolar 90° 

Figura 8.3: Espectro del coeficiente de reflexión Yagi-Uda 690-2700[MHz]

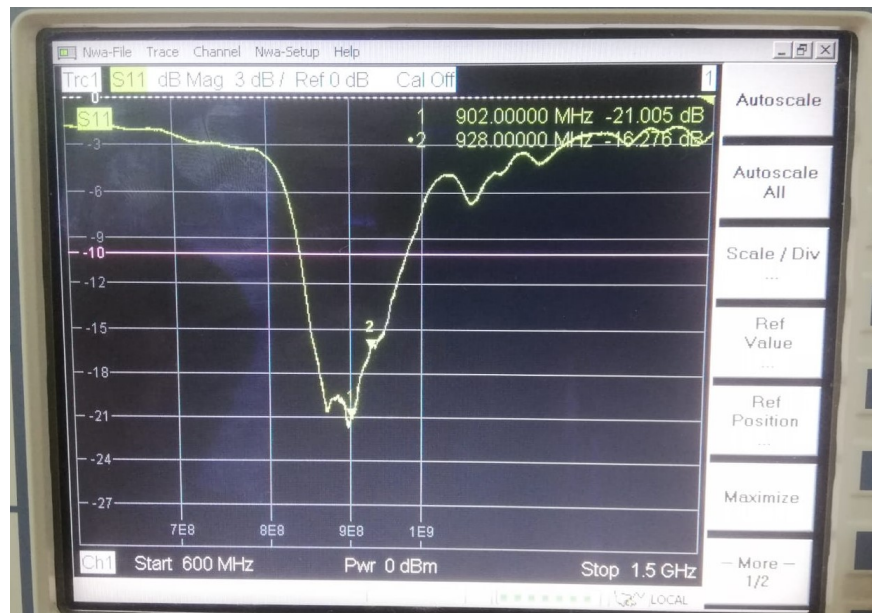


Figura 8.4: Espectro del coeficiente de reflexión Yagi-Uda 900[Mhz]

8.3. Potencia de los datos de transmisión a corta distancia (10[cm])

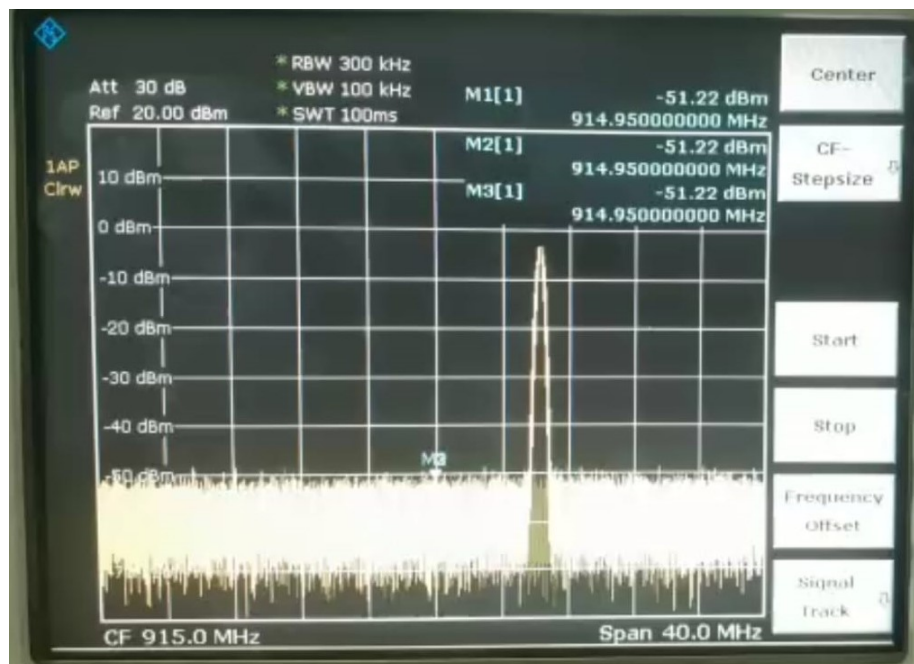


Figura 8.5: Muestra # 1 de potencia

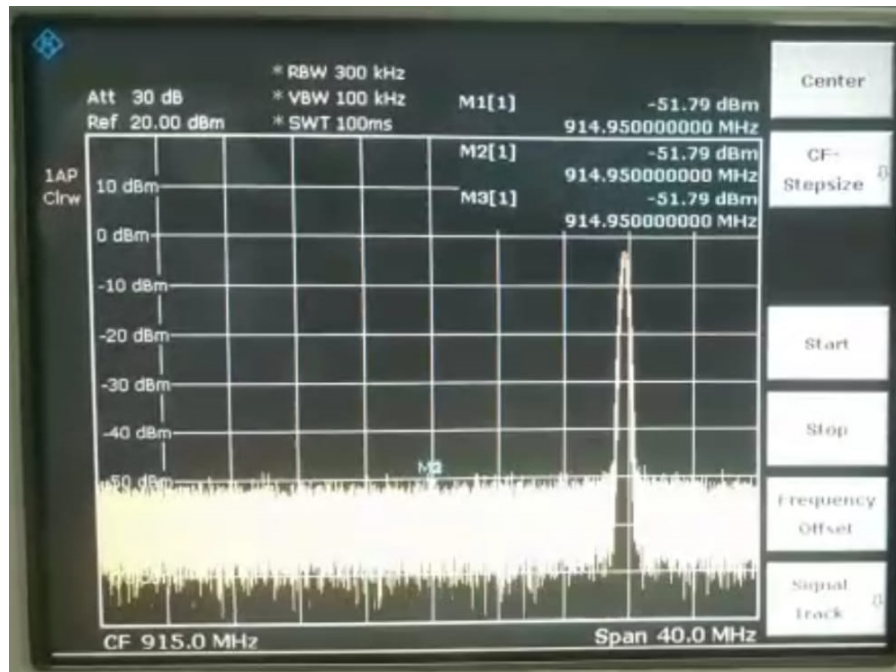


Figura 8.6: Muestra # 2 de potencia

8.4. Distancias de las pruebas mostrados en Google Earth

8.4.1. Estadio 1° de Marzo



Figura 8.7: Estadio 1° de Marzo Google Earth online

8.4.2. Teatrino - Administración III



Figura 8.8: Teatrino-Administración III Google Earth online

8.4.3. Teatrino - Camilo (UIS-75)



Figura 8.9: Teatrino-Camilo Torres(UIS-75) Google Earth online

8.4.4. Teatrino - Edificio Santa Isabella

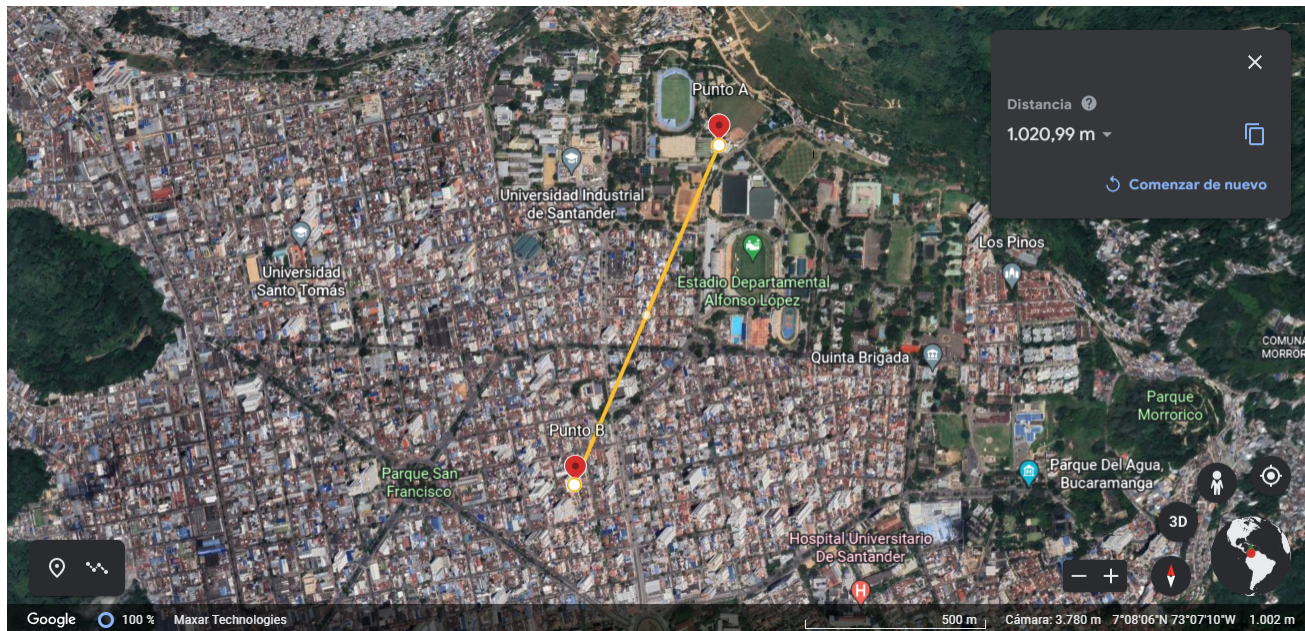


Figura 8.10: Teatrino - Edificio Santa Isabella Google Earth online

8.5. Gráficas de RSSI para cada prueba

8.5.1. Estadio 1° de Marzo

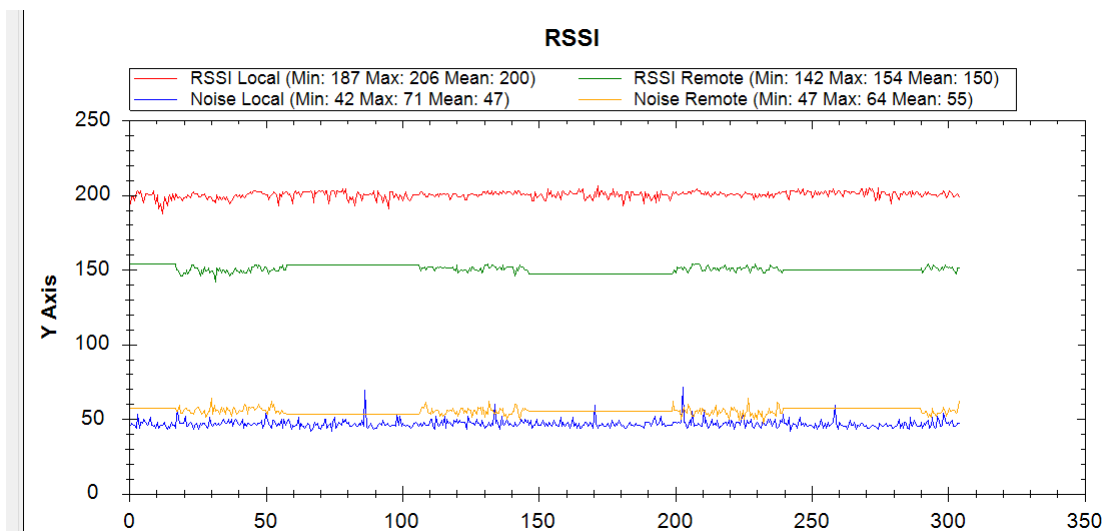


Figura 8.11: RSSI Estadio 1° de Marzo

8.5.2. Teatrino - Administración III

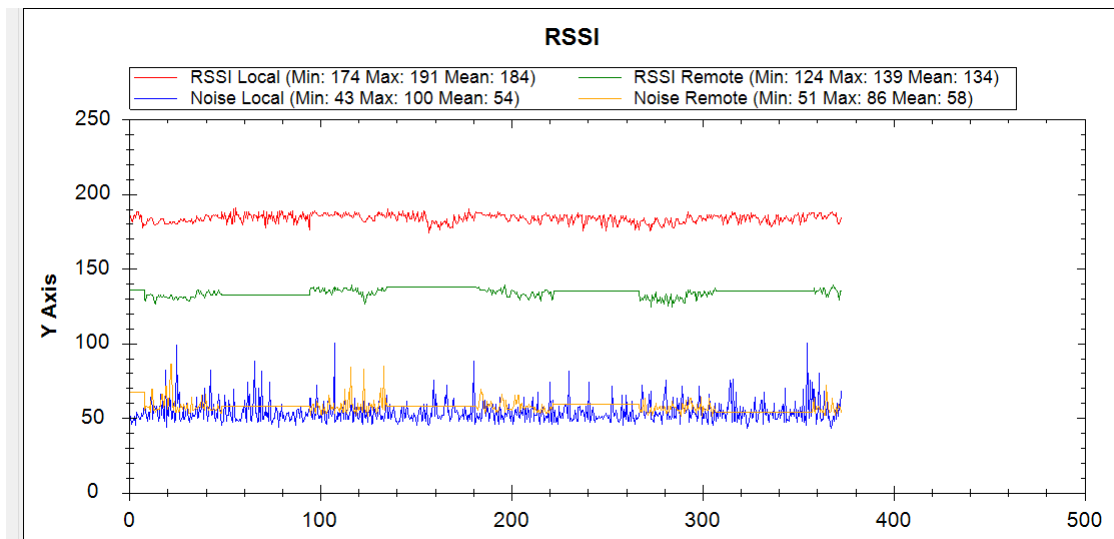


Figura 8.12: RSSI Teatrino - Administración III

8.5.3. Teatrino - Camilo (UIS-75)

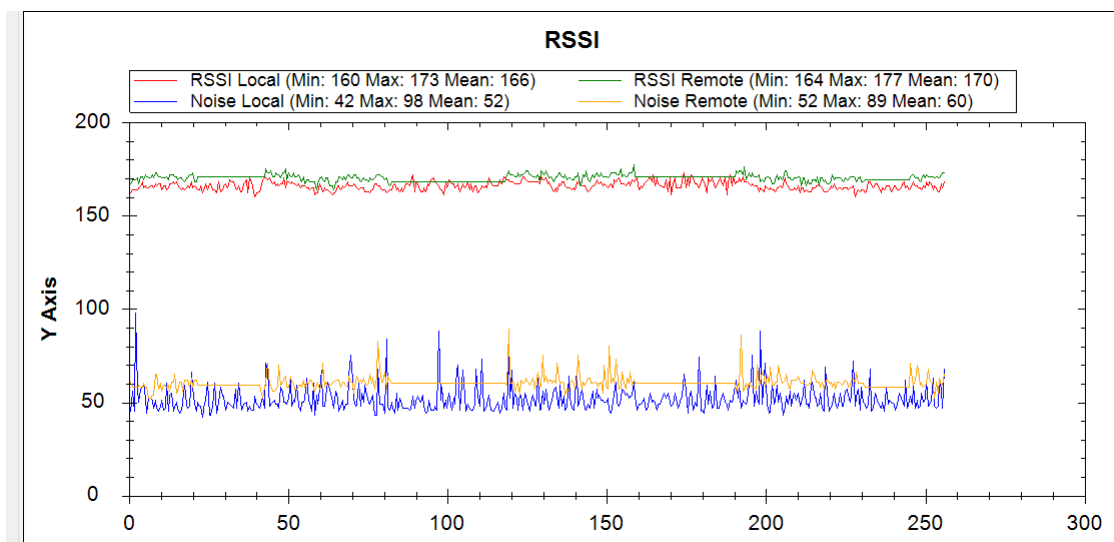


Figura 8.13: RSSI Teatrino - Camilo Torres(UIS-75)

8.5.4. Teatrino - Edificio Santa Isabella

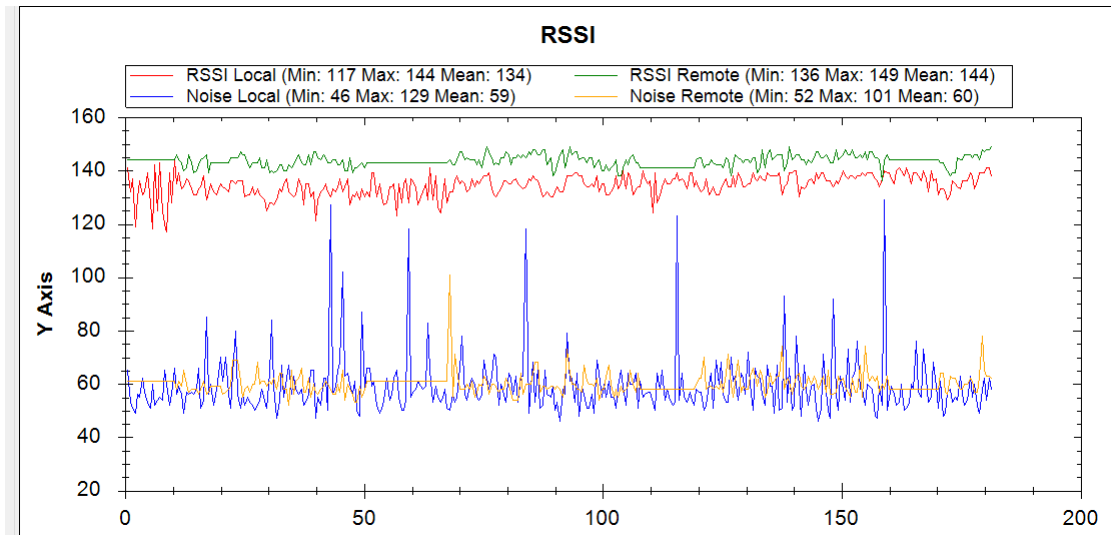


Figura 8.14: RSSI Teatrino - Edificio Santa Isabella

8.6. Gráficas de RSSI con LNA para las últimas dos pruebas

8.6.1. Teatrino - Camilo (UIS-75)

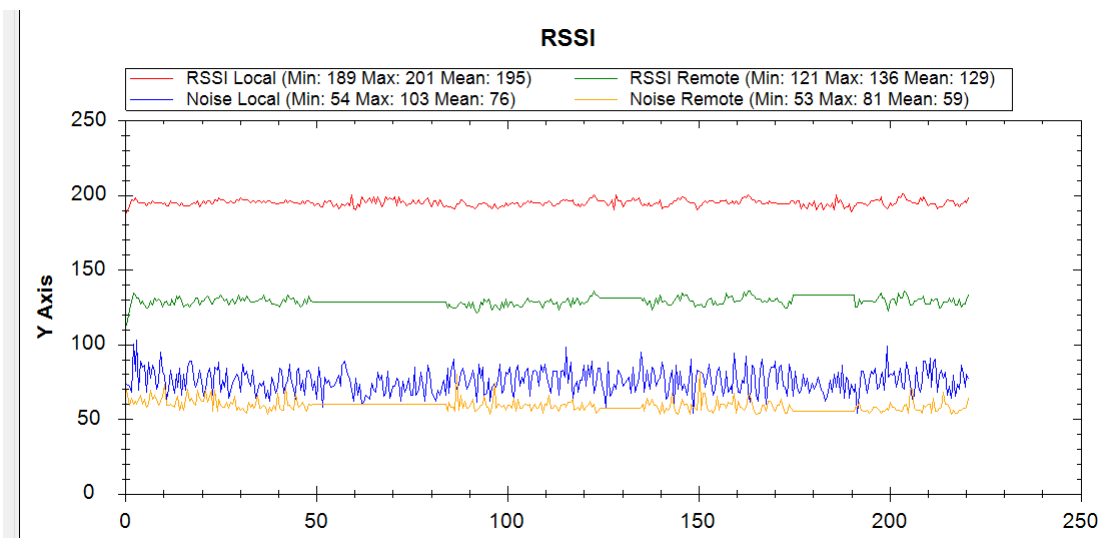


Figura 8.15: RSSI Teatrino - Camilo Torres(UIS-75) con LNA

8.6.2. Teatrino - Edificio Santa Isabella

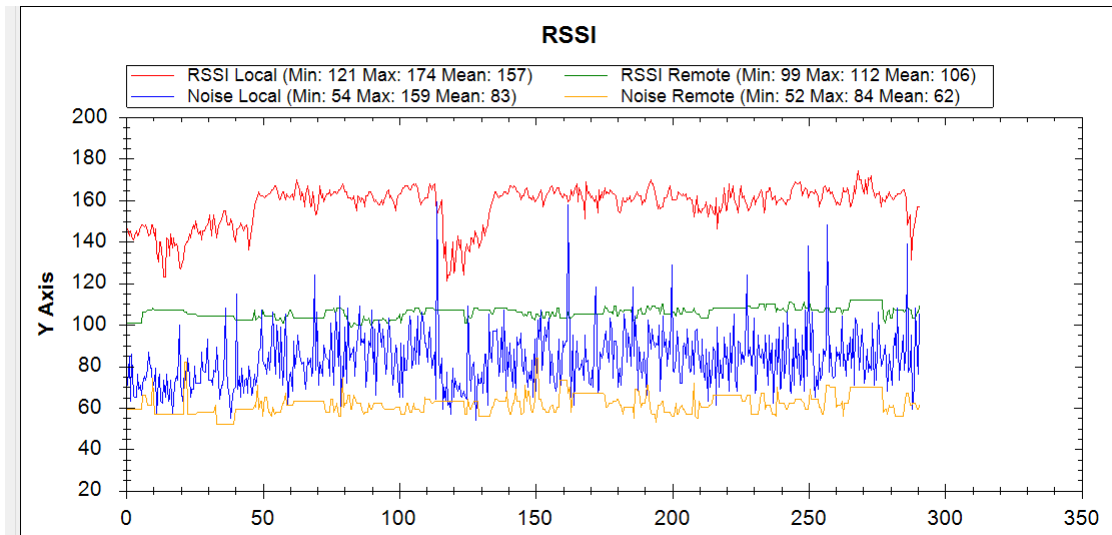


Figura 8.16: RSSI Teatrino - Edificio Santa Isabella con LNA

8.7. Fotos tomadas

8.7.1. Estadio 1° de Marzo

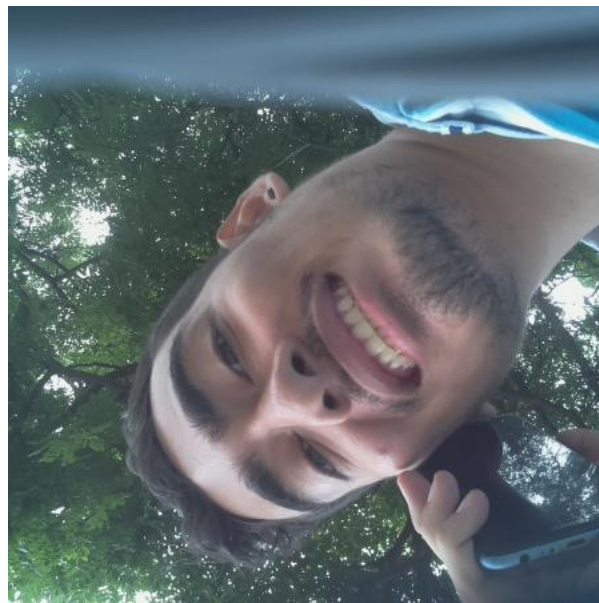


Figura 8.17: Foto recibida 1

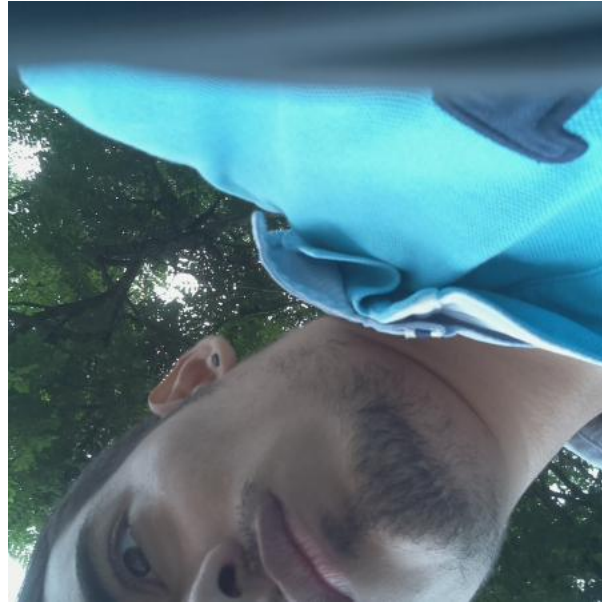


Figura 8.18: Foto recibida 2

8.7.2. Teatrino - Administración III



Figura 8.19: Foto recibida 1



Figura 8.20: Foto recibida 2

8.7.3. Teatrino - Camilo (UIS-75)

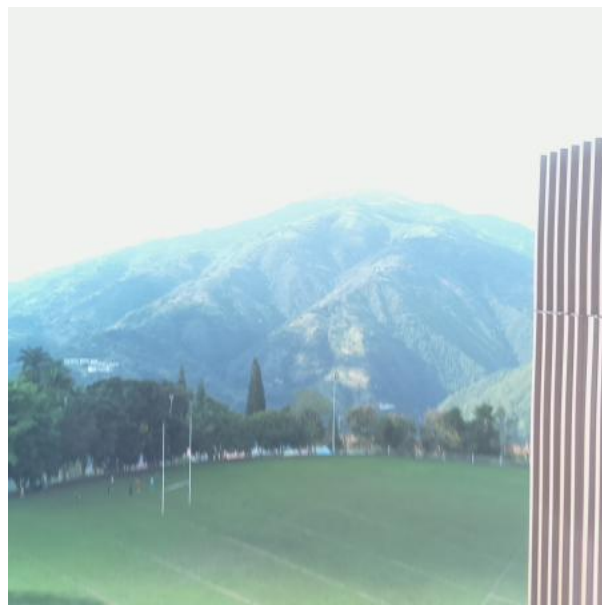


Figura 8.21: Foto recibida 1



Figura 8.22: Foto recibida 2

8.7.4. Teatrino - Edificio Santa Isabella

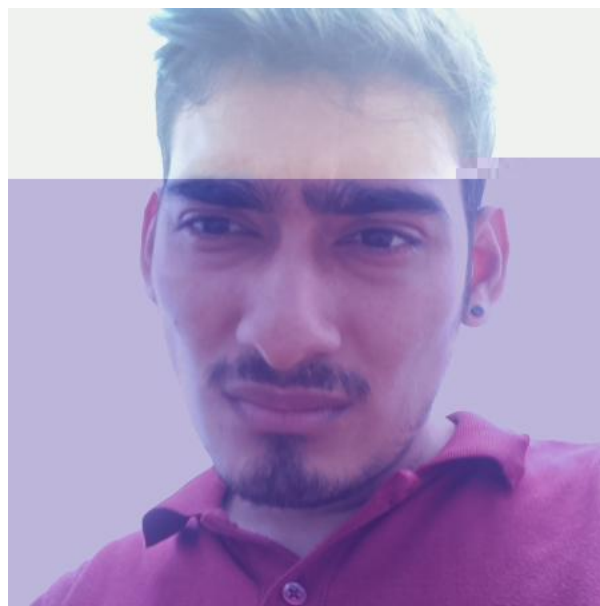


Figura 8.23: Foto recibida 1

Bibliografía

- Adafruit MAX31865 RTD PT100 or PT1000 Amplifier. (2016). <https://learn.adafruit.com/adafruit-max31865-rtd-pt100-amplifier?view=all>
- Aldaz-Rosas, J. C., Hernández-Constante, J. & Cedano-Ol, M. (2016). Cálculo de Radio Enlace Terrestre. *SOMI XXXI Congreso de Instrumentación*, (November).
- Altamar Amaranto, Fajardo Ariza, C.A., Rodriguez Ferreira, J.G., and Universidad Industrial de Santander. Escuela de Ingenierías Eléctrica, E. y de T. (2019). ADQUISICIÓN, TRANSMISIÓN Y VISUALIZACIÓN DE LA TELEMETRÍA DEL GLOBO SONDA ESTRATOSFÉRICO DE LA MISIÓN E3TRATOS.
- Amor Jiménez, C. & Paredes, F. (2011). Dipolo impreso para aplicaciones Wi-fi y caracterización de la ganancia por medio de la celda TEM.
- Ardupilot.org. (2022). SiK Radio — Advanced Configuration - Blimp documentation. [online] Available at: <<https://ardupilot.org/blimp/docs/common-3dr-radio-advanced-configuration-and-technical-information.html>> [Accessed 27 August 2022].
- Bosch. (2013). *BMP180 digital pressure sensor*. BOSCH. <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>
- Buettrich, S. (2007). Unidad 06: Cálculo de radioenlace. *Recuperado de http://itrainonline.org/itrainonline/mmtk/wireless_es/files/06_es_calculo-de-radioenlace_guia_v02.pdf*.
- CACERES CHANAGA, J. F. (2021). *ANALISIS TECNICO PARA LA IMPLEMENTACION DE UN SISTEMA DE TRANSMISION DE DATOS MEDIANTE EL DESPLIEGUE DE LA TECNOLOGIA TVWS EN ZONAS RURALES* [recurso electrónico]. UIS.
- CUDRIZ, E. & ANDRES, S. (2019). DISEÑO Y CONSTRUCCION DE LA ESTRUCTURA MECANICA, ESCUDO TERMICO, SISTEMA DE ELEVACION Y SISTEMA DE DESCENSO DE UN GLOBO SONDA ESTRATOSFERICO.

- Harrington, W. (2015). *Learning raspberry*. Packt Publishing Ltd.
- Invensense. (2016). MPU-9250 Product Specification Revision 1.1, MPU-9250. <https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>
- Lay, K. S., Li, L. & Okutsu, M. (2022). High altitude balloon testing of Arduino and environmental sensors for CubeSat prototype. *HardwareX*, 12, e00329. <https://doi.org/https://doi.org/10.1016/j.ohx.2022.e00329>
- Maxim. (2012). Max31865 PDF, Max31865 description, Max31865 Datasheet, Max31865 View :: Alldatasheet :: <https://pdf1.alldatasheet.com/datasheet-pdf/view/492628/MAXIM/MAX31865.html>
- MICROCHIP. (2001). MCP3008 Data Sheet, MCP300X 10-Bit Analog-to-Digital Converters. <https://pdf1.alldatasheet.es/datasheet-pdf/view/92963/MICROCHIP/MCP3008.html>
- Miller, K. & Yates, D. (2005). Climate Change and Water Resources: A Primer for Water Utilities.
- Nath, O. (2020). Review on raspberry pi 3b+ and its scope. *Int. J. Eng. Appl. Sci. Technol.*, 4(9), 157-159.
- Nunez Perez, J. C., Rodriguez, A. & Téllez, A. (2017). Estimación del alcance de radiotransmisores Xbee. *Research in Computing Science*, 142, 39-46. <https://doi.org/10.13053/rcs-142-1-4>
- RFDesign. (2013). RFD900 Radio Modem Data Sheet, RFD900+ Data Sheet. <http://files.rfdesign.com.au/Files/documents/RFD900%5C%20DataSheet.pdf>
- ROITHNER. (2011). Guva-S12SD PDF, Guva-S12SD descripción electrónicos, Guva-S12SD Datasheet, guva-S12SD view :: Alldatasheet :: <https://pdf1.alldatasheet.es/datasheet-pdf/view/712047/ROITHNER/GUVA-S12SD.html>
- Taraba, M., Fauland, H., Turetschek, T., Stumptner, W., Kudielka, V., Scheer, D., Sattler, B., Fritz, A., Stingl, B., Fuchs, H., Gubo, B., Hettrich, S., Hirtl, A., Unger, E., Soucek, A., Frischauf, N. & Grömer, G. (2014). Passepartout Sherpa – A low-cost, reusable

transportation system into the stratosphere for small experiments. *Advances in Space Research*, 54(11), 2259-2273. [https://doi.org/https://doi.org/10.1016/j.asr.2014.07.030](https://doi.org/10.1016/j.asr.2014.07.030)

030

Wang, Z., Huang, M., Qian, L., Zhao, B. & Wang, G. (2020). High-Altitude Balloon-Based Sensor System Design and Implementation. *Sensors*, 20(7). <https://doi.org/10.3390/s20072080>