



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**

CARRERA DE INGENIERÍA ELECTRÓNICA

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE:
“INGENIERO ELECTRÓNICO”**

**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN
METEOROLÓGICA PORTÁTIL PARA EL MONITOREO DE VARIABLES
AMBIENTALES EN EL RANCHO SAN GABRIEL UTILIZANDO
TECNOLOGÍA RFID”**

AUTOR:

Madelaine Joustine López Llerena

TUTOR:

GUAYAQUIL – ECUADOR

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Madelaine Joustine López Llerena, autor del proyecto técnico: “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN METEOROLÓGICA PORTÁTIL PARA EL MONITOREO DE VARIABLES AMBIENTALES EN EL RANCHO SAN GABRIEL UTILIZANDO TECNOLOGÍA RFID”, declaro que el contenido y desarrollo de este proyecto técnico aquí descrito es de mi exclusiva responsabilidad y autoría. Autorizo a la UNIVERSIDAD POLITECNICA SALESIANA la publicación total del trabajo de titulación y su reproducción sin beneficio de ganancia.

Guayaquil, Marzo del 2022



Madelaine Joustine López Llerena

C.I: 0942115148

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTORES

Por medio de esta declaración, Yo Madelaine Joustine López Llerena, con documento de identificación No. 0942115148, cede a la Universidad Politécnica Salesiana los derechos de propiedad intelectual en calidad de autor del proyecto de titulación titulado " DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN METEOROLÓGICA PORTÁTIL PARA EL MONITOREO DE VARIABLES AMBIENTALES EN EL RANCHO SAN GABRIEL UTILIZANDO TECNOLOGÍA RFID", mismo que ha sido desarrollado para optar por el título de Ingeniero Electrónico mención en Telecomunicaciones, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la Normativa Institucional vigente.

Guayaquil, Marzo del 2022



Madelaine Joustine López Llerena

C.I: 0942115148

CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN

Yo Msc. Mónica María Miranda Ramos certifico que bajo mi dirección y asesoría fue desarrollado el proyecto técnico de titulación “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN METEOROLÓGICA PORTÁTIL PARA EL MONITOREO DE VARIABLES AMBIENTALES EN EL RANCHO SAN GABRIEL UTILIZANDO TECNOLOGÍA RFID”, realizado por Madelaine Joustine López Llerena para la obtención del título en Ingeniería Electrónica.

Por medio de la presente certifico que el documento cumple con los requisitos establecidos en el Instructivo para la Estructura y Desarrollo de Trabajos de Titulación para pregrado de la Universidad Politécnica Salesiana.

En virtud de lo anterior, autorizo su presentación y aceptación como una obra auténtica y de alto valor académico.

Guayaquil, marzo del 2021

A handwritten signature in blue ink, reading "Mónica Miranda", is written over a horizontal line.

Ing. Mónica Miranda, Msc
DIRECTOR DE PROYECTO DE TITULACIÓN

DEDICATORIA

El esfuerzo y la dedicación en una carrera es una ejemplificación y efecto de las personas que permanecen detrás. El esfuerzo llevado a cabo en este trabajo de titulación va primordialmente dedicado a mi mamá, por el apoyo a mi educación y por ser la que siempre estuvo conmigo.

Al final, sin embargo, no menos importante, deseo agradecer a todos aquellos que se involucraron en mi trabajo de titulación, y a la organización educativa que me han permitido generar una aportación más a lo académico.

Madelaine Joustine López Llerena

AGRADECIMIENTO

El presente trabajo investigativo lo dedico principalmente a Dios, por ser el inspirador y darme fuerza para continuar en este proceso de obtener uno de los anhelos más deseados.

A mis padres, por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes he logrado llegar hasta aquí y convertirme en lo que soy. Ha sido el orgullo y el privilegio de ser su hija, son los mejores padres.

A todas las personas que me han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que me abrieron las puertas y compartieron sus conocimientos.

Madelaine Joustine López Llerena

RESUMEN

Año	Título	Alumnos	Tutor	Tema de titulación
2021	Ingeniero Electrónico	Madelaine Joustine López Llerena	Ing. Mónica Miranda	DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN METEOROLÓGICA PORTÁTIL PARA EL MONITOREO DE VARIABLES AMBIENTALES EN EL RANCHO SAN GABRIEL UTILIZANDO TECNOLOGÍA RFID.

El presente trabajo de titulación propone y desarrolla un prototipo de estación meteorología portátil que permite la medición de variables físicas tales como: presión atmosférica, temperatura de ambiente, humedad relativa, cantidad de luz solar, humedad de suelo. Con el objetivo de proporcionar información que ayude a optimizar las condiciones necesarias para mejorar cultivos presentes en el rancho San Gabriel, se desarrolla este prototipo, el cual está potenciando por un Arduino Mega que en su núcleo se encuentra un microchip Atmega2560, que por medio de diferentes sensores y módulos transceptores permite la comunicación envío y recepción de datos en tiempo real, mediante la utilización de la banda ISM de 2.4Ghz. Acoplado los sensores correspondientes a la tarjeta de adquisición de datos para su procesamiento con el algoritmo desarrollado.

Una vez que las variables a monitorear por el prototipo de estación meteorológica son obtenidas de forma correcta, se acoplan las antenas de transmisión inalámbrica RFID, con la cual se realiza pruebas de transmisión y recepción de datos a una distancia horizontal máxima de 300 m aproximadamente en campo abierto con línea

de vista (LOS Acrónimo en inglés de Line-Of-Sight) ya que las antenas con tecnología RFID que se utiliza en este proyecto funcionan de forma óptima a la distancia antes mencionada, sin pérdida de datos ni del enlace de comunicación.

Para la comodidad y portabilidad en la adquisición de los parámetros capturados por el prototipo se emplea un receptor portable con una pantalla para la visualización de cada una de las variables a monitorear, además este prototipo dispone de una alarma sonora anti robo la cual se activa mediante un bypass de relés y está separada del módulo emisor para mayor seguridad, cabe recalcar que el prototipo es energizado por medio de baterías recargables, la estación remota emisora recarga su batería por medio de un panel solar, y la batería del módulo receptor se recarga por medio de un cargador para baterías lipo de 2 celdas.

ABSTRACT

Year	Profession	Students	Director of Technical Project	Technical Project
2021	Electronic Engineer	Madelaine Joustine López Llerena	Ing. Mónica Miranda	DESIGN AND IMPLEMENTATION OF A PORTABLE WEATHER STATION PROTOTYPE FOR MONITORING ENVIRONMENTAL VARIABLES AT RANCHO SAN GABRIEL USING RFID TECHNOLOGY.

This degree work proposes and develops a portable weather station prototype that allows the measurement of physical variables such as: atmospheric pressure, ambient temperature, relative humidity, amount of sunlight, soil moisture. With the aim of providing information that helps optimize the necessary conditions to improve crops present in the San Gabriel ranch, this prototype is developed, which is powered by an Arduino Mega that has an Atmega2560 microchip at its core, which through Different sensors and transceiver modules allow communication, sending and receiving data in real time, through the use of the 2.4Ghz ISM band. Coupling the corresponding sensors to the data acquisition card for its processing with the developed algorithm.

Once the variables to be monitored by the weather station prototype are obtained correctly, the RFID wireless transmission antennas are attached, with which data transmission and reception tests are carried out at a maximum horizontal distance of approximately 300 m in open field with line of sight (LOS Acronym in English for Line-Of-Sight) since the antennas with RFID technology used in this project work optimally at the aforementioned distance, without loss of data or communication link .

For comfort and portability in the acquisition of the parameters captured by the prototype, a portable receiver is used with a screen for the visualization of each of the variables to be monitored, in addition this prototype has an anti-theft sound alarm

which is activated by a relay bypass and is separated from the transmitter module for greater safety, it should be noted that the prototype is powered by rechargeable batteries, the remote transmitter station recharges its battery by means of a solar panel, and the battery of the receiver module is recharged by means of a charger for 2-cell lipo batteries.

ABREVIATURAS

RFID: Radio Frecuency Identification.

PCB: Printed Circuit Board.

TFT: Thin Film Transistor-Liquid Crystal Display.

IDE: integrated development environment.

LiPo:(Litium/ polymer).

MAh: Milliamps.

Opamp: Amplificador operacional.

%RH: Humedad Relativa.

DHT 11: Digital temperature and humidity sensor.

BMP 180: Barometric pressure.

ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN	II
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTORES	III
CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN.....	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
RESUMEN.....	VII
ABSTRACT	IX
ABREVIATURAS	XI
ÍNDICE DE TABLAS.....	XV
ÍNDICE DE FIGURAS.....	XVI
INTRODUCCIÓN.....	XVIII
1. EL PROBLEMA	1
1.1. Planteamiento del problema	1
1.2. Importancia y alcances.....	1
1.3. Delimitación del problema	2
1.3.1. Delimitación espacial	2
1.3.2. Delimitación temporal	2
1.3.3. Delimitación social	3
1.4. Beneficiarios.....	3
1.5. Justificación.....	3
1.6. Descripción de la propuesta	4
1.7. Impacto	4
1.8. Grupo objetivo	5
1.8.1. Población.....	5
1.8.2. Muestra.....	5
1.9. Objetivos	5
1.9.1. Objetivo general.....	5
1.9.2. Objetivos específicos	5
2. FUDAMENTOS TEORICOS	6
2.1. Estación Meteorológica	6
2.1.1. Principales Sensores utilizados en estaciones meteorológicas	7
2.1.2. Tipos de estaciones Meteorológicas	7
2.2. Telemetría	8
2.3. Sistema Fotovoltaico	9
2.4. Proteus Design Suite.....	9
2.5. Pantalla táctil TFT	10
2.6. Sensor DTH22	11
2.7. Sensor BPM 180	12
2.8. Panel Solar.....	13
2.9. Arduino.....	13
2.9.1. Arduino IDE	14
2.9.2. Arduino Mega 2560.....	14

2.10. Sensor de humedad de suelo.....	16
2.11. Convertidor voltaje DC-DC step-down LM596	17
2.12. Regulador de Carga Panel solar	18
2.13. Sensor de gotas de lluvia YL83	19
2.14. Módulo RFID	19
2.15. Sensor MPU6050	21
2.16. Batería Lipo.....	21
2.16.1. Significado de la nomenclatura de una batería LiPo	22
2.16.2. Capacidad indicada en mAh	22
2.16.3. Capacidad de descarga, Numero C	23
La tasa de descarga de un batería tipo se interpreta por el numero C que viene descrito en la etiqueta de información con la cual se puede realizar un cálculo.	23
3. MARCO METODOLOGICO	24
3.1. Metodología	24
3.1.1. Métodos.....	24
3.1.2. Método deductivo	24
3.1.3. Método experimental	24
3.1.4. Técnicas	25
3.1.5. Técnica de investigación.....	25
3.1.6. Observación.....	25
3.1.7. La entrevista	25
3.1.8. Recolección de información	26
3.2. Diseño del prototipo de estación meteorológica	26
3.3. Diseño de tarjetas de circuito impreso.....	26
3.4. Conexión y diagramas esquemáticos	28
3.4.1. Diagrama de conexión de fuente de alimentación módulo emisor.....	29
3.5. Diagrama de flujo de funcionamiento de módulo emisor	30
3.6. Diagrama de flujo de funcionamiento de módulo receptor	31
3.7. Diseño e impresión 3D módulo receptor.....	31
3.7.1. Diseño de estructura de módulo receptor	31
3.7.2. Impresión de estructura de módulo receptor	32
3.8. Pruebas y Conexiones físicas del prototipo	33
3.8.1. Conexión de alarma sonora	34
3.8.2. Conexión de módulo regulador de voltaje	34
3.8.3. Conexión de batería Lipo a modulo receptor.....	35
3.8.4. Conexión de sensores a tarjeta electrónica de fuente de alimentación ..	36
3.8.5. Conexión de Panel solar y controlador de carga.....	36
3.9. Diagramas de flujo de programación de módulo emisor y receptor	38
3.10. Código de programación utilizado	39
4. RESULTADOS	40
4.1. Prototipo de estación meteorológica modulo emisor	40
4.1.1. Sensor de temperatura y humedad.....	41

4.1.2. Panel solar y sensores de lluvia y luz.....	42
4.1.3. Controlador de carga (fuente de alimentación)	43
4.1.4. Sensor de humedad de suelo	44
4.1.5. Módulo receptor	44
4.2. Menú del módulo receptor.....	45
4.2.1. Visualización de datos del sensor de lluvia	45
4.2.2. Visualización de valor humedad de suelo módulo receptor.....	46
4.2.3. Visualización de valor presión atmosférica	46
4.2.4. Visualización de valor humedad, cantidad de luz y temperatura ambiente	47
4.2.5. Resultados de los parámetros finales visualizados en la pantalla del receptor	48
4.3. Comparación de datos	48
CONCLUSIONES.....	49
RECOMENDACIONES.....	50
BIBLIOGRAFIA.....	51
ANEXOS	53
Anexo 1: Código de programación utilizado	53
Anexo 2: Registro fotográfico	75

ÍNDICE DE TABLAS

TABLA I TIPOS DE ESTACIONES METEOROLÓGICAS	7
TABLA II ESPECIFICACIONES TÉCNICAS DE ARDUINO MEGA.	15
TABLA III. COMPONENTES UTILIZADOS EN PCB DE SENSORES	28

ÍNDICE DE FIGURAS

Fig. 1 Ubicación geográfica del Rancho San Gabriel. [1].....	2
Fig. 2 Estación meteorológica Profesional. [3].....	6
Fig. 3 Proceso de sistema telemétrico. [4].....	8
Fig. 4 Proteus Design Suite. [7].....	10
Fig. 5 Shiel de Arduino de pantalla TFT táctil. [8]	10
Fig. 6 Sensor DHT22 utilizado para medir temperatura y humedad del ambiente. [9]	12
Fig. 7 Sensor BPM 180, sensor de presión barométrica de bajo costo. [10].....	13
Fig. 8 Panel solar fotovoltaico. [11].....	13
Fig. 9 Interfaz del IDE de Arduino. [13].....	14
Fig. 10 Tarjeta PCB de Arduino con chip Atmega 2560. [14].....	15
Fig. 11 Sensor de humedad de suelo. [15]	16
Fig. 12 Módulo de convertidor de voltaje DC con chip regulador LM596 [16]	17
Fig. 13 Módulo regulador de voltaje y carga de panel solar [18]	18
Fig. 14 Módulo sensor de lluvia para Arduino [19].....	19
Fig. 15 Módulo de antena RFID de largo alcance. [21].....	20
Fig. 16 Módulo de acelerómetro y giroscopio. [23]	21
Fig. 17 Batería Lipo de alta eficiencia. [24]	22
Fig. 18 Significado de siglas y números impresos en batería Lipo. [26].....	23
Fig. 19 Diseño de tarjeta vista 3D.....	27
Fig. 20 Diseño de PCB de sensores pistas.....	27
Fig. 21 Diagrama esquemático de conexión de sensores hacia Arduino.	29
Fig. 22 Conexión de fuente de alimentación.....	30
Fig. 23 Diagrama de funcionamiento de módulo emisor.	30
Fig. 24 Módulo receptor diagrama de flujo de entradas y salidas.	31
Fig. 25 Vista frontal de diseño en SolidWorks.	32
Fig. 26 Diseño de estructura de módulo receptor en SolidWorks.	32
Fig. 27 Impresión 3D módulo receptor.....	33
Fig. 28 Conexión y prueba de sensores módulo emisor.	33
Fig. 29 Conexión de buzzer (Alarma).	34
Fig. 30 Conexión de Step down en módulo receptor.	35
Fig. 31 Conexión de Batería Lipo.	35
Fig. 32 Tarjeta de conexión de sensores.....	36
Fig. 33 Conexión de módulo de carga y panel solar.	37
Fig. 34 Conexión de sistema fotovoltaico con carga.....	37

Fig. 35 Diagramas de flujo de módulo receptor y emisor.	38
Fig. 36 Modulo emisor.	41
Fig. 37 Prototipo de estación meteorológica instalada en el Rancho San Gabriel (Modulo Emisor).	41
Fig. 38 Sensor DHT11 instalado en el exterior de la estructura.	42
Fig. 39 Panel solar, sensor de lluvia y sensor de luz.	42
Fig. 40 Controlador de carga.	43
Fig. 41 Batería conectada al módulo de carga.	43
Fig. 42 Obteniendo valores de humedad de suelo.	44
Fig. 43 Módulo receptor.	44
Fig. 44 Apreciación del menú que dispone el módulo receptor.	45
Fig. 45 Visualización sensor de lluvia.	45
Fig. 46 Humedad de suelo.	46
Fig. 47 Presión atmosférica.	46
Fig. 48 Cantidad de luz solar.	47
Fig. 49 Sensor de temperatura.	47
Fig. 50 Pantalla en tiempo real del módulo receptor.	48
Fig. 51 Aplicación de tiempo incorporada en los celulares.	48
Fig. 52 Módulo emisor en plantación agrícola.	75
Fig. 53 Integración de partes y componentes a módulo emisor.	75
Fig. 54 Gestión de cables y posicionamiento de componentes internos módulo emisor.	76
Fig. 55 Ventilador de 12 voltios para extraer calor generado por el equipo.	76
Fig. 56 Reconocimiento de terreno.	77

INTRODUCCIÓN

La agrometeorología es un término compuesto de meteorología y agricultura. La agrometeorología estudia la relación del tiempo con la producción de los cultivos agrícolas, incluyendo las enfermedades y plagas que afectan a las plantas; además la influencia de los factores y elementos del clima sobre la fenología de los cultivos. Una de las principales actividades de la agrometeorología, es la búsqueda permanente de caracterizar en tiempo y espacio de la relación entre el clima y el rendimiento de los cultivos.

Se propone el diseño e implementación de un prototipo de estación meteorológica utilizando tecnología inalámbrica y de microcontroladores, el cual obtendrá información sobre; temperatura, humedad relativa, cantidad de luz, humedad de suelo, presión atmosférica, velocidad de viento y también contará con una bocina la cual actuará como alarma antirrobo.

Esta tesis mantiene el orden siguiente: en el primer capítulo se explica el problema, antecedentes, importancia y alcance, delimitación, beneficiarios de la propuesta, propuesta de solución y objetivos. En el segundo capítulo se describe los fundamentos teóricos, estaciones meteorológicas, los diferentes sensores, Arduino, el capítulo tres se refiere al marco metodológico, tipo de investigación. El capítulo 4 trata sobre los resultados de la investigación.

1. EL PROBLEMA

1.1. Planteamiento del problema

En el Rancho San Gabriel se realizan diferentes tipos de cultivos entre los principales se encuentran variedades de pasto y maíz para alimentar el ganado. Así como la siembra y el cuidado de los cultivos también se lo realiza de forma artesanal, sin ningún instrumento o herramienta tecnológica que ayude a conocer el estado de humedad en que se encuentra el suelo, humedad en el ambiente, presión atmosférica entre otras variables meteorológicas que puedan influir en la calidad y cantidad de la cosecha, todos estos parámetros no pueden ser medidos ni cuantificados en la actualidad, solo la experiencia de cada agricultor, que es adquirida empíricamente durante años, es por esta razón que se consideró la necesidad de construir un prototipo de estación meteorológica portátil.

Este tipo de prototipo utilizado para la monitorización de variables atmosféricas aplicados a la agricultura beneficiará de mejor manera a los agricultores y a los diferentes sembríos del Rancho San Gabriel, el cual ayudará en la selección de mejores terrenos para la siembra, mayor cuidado durante la etapa de crecimiento, con la finalidad de obtener un cultivo de calidad, gracias a la tecnificación utilizando dispositivos de bajo costo e inalámbricos, de esta forma se incrementa el nivel de conocimiento acerca de la influencia del clima sobre el crecimiento y desarrollo de cultivos.

1.2. Importancia y alcances

La producción agrícola en Ecuador está propensa a desastres naturales tales como: sequías, inundaciones, ráfagas de viento, olas de calor, etc. Estos pueden tener efectos devastadores sobre la producción de cultivos agrícolas y comprometer seriamente el desarrollo económico de las poblaciones que de ellos dependen.

Al incrementar el nivel de conocimiento acerca de la influencia del clima sobre el crecimiento y desarrollo de cultivos, se pueden crear estrategias en base a la información obtenida por el prototipo estación meteorológica, para mejorar las técnicas y cuidado del cultivo y de esta forma incrementar la calidad del producto y

minimizar pérdidas de producción por no considerar los factores climáticos durante su crecimiento.

1.3. Delimitación del problema

Como se detalla en este documento, el proyecto de titulación actual especifica el lugar donde será instalada la estación meteorológica portátil, y su ubicación geográfica.

1.3.1. Delimitación espacial

El proyecto se desarrolló en el rancho San Gabriel ubicado en la vía a la parroquia Virgen de Fátima, del Cantón Milagro, Provincia del Guayas, tal y como se aprecia en la Fig. 1 se señala la ubicación geográfica del Rancho san Gabriel



Fig. 1 Ubicación geográfica del Rancho San Gabriel. [1]

1.3.2. Delimitación temporal

El presente proyecto se desarrolló durante el transcurso del año 2020 – 2022, cuyo plazo normal de realización es de 12 meses a partir de su aprobación.

Mediante la integración y puesta en práctica de diferentes habilidades y conocimientos adquiridos durante el transcurso de la carrera universitaria se realizó este proyecto, en donde se destacan temas relacionados con programación de

microcontroladores, redes de computadoras entre otros; apoyándose de revistas científicas, repositorios digitales e información científica relacionada al tema de titulación.

1.3.3. Delimitación social

Este proyecto beneficia a todos los agricultores que conforman el Rancho San Gabriel, ya que no necesitan trasladarse personalmente hasta el cultivo para verificar la humedad del suelo, humedad del ambiente etc. y al estar el cultivo en constante monitoreo se tiene un mejor entendimiento de los parámetros atmosféricos presentes en el cultivo durante todo el año.

1.4. Beneficiarios

Los beneficiarios de este proyecto son los trabajadores, y todos quienes conforman el Rancho San Gabriel, ya que, al contar con un sistema de medición de variables meteorológicas en tiempo real, mejoran en gran medida el entendimiento del clima en determinadas épocas del año y en áreas específicas de extensiones de terreno no muy grande.

1.5. Justificación

Con la implementación de un prototipo de estación meteorológica portátil el cual permite obtener valores atmosféricos en tiempo real, entender mejor la influencia del clima sobre los cultivos y prevenir pérdidas por una mala interpretación del clima en cuanto, lluvias, inundaciones, ráfagas de viento, etc.

Mediante este proyecto se busca poder dar mayores herramientas tecnológicas de bajo costo introduciendo el término de agrometrología, y de esta forma asegurar una cosecha de calidad y minimizar las pérdidas en los cultivos.

1.6. Descripción de la propuesta

Se propone diseñar un prototipo de estación meteorológica para realizar mediciones meteorológicas de la atmosfera de una determinada área donde se cultiven diferentes tipos de plantas, en la cual se podrá conocer de forma remota en qué condiciones ambientales se encuentra un cultivo y determinar si necesitará menos horas de riego, esto conlleva a optimar tiempo y recursos en especial el líquido vital (Agua), adicional a estas funciones se incluirá una bocina que actuará como alarma antirrobo para dar una mayor seguridad al prototipo en el caso de que intente ser hurtado .

Este proyecto tendrá como propósito mostrar la influencia del clima en el rendimiento de los cultivos, los beneficios obtenidos al hacer uso de una estación meteorología e introducir el concepto de agrometeorología y su importancia, con la información que se obtendrá de la estación meteorológica se podrá hacer un historial de las condiciones ambientales del área seleccionada a lo largo del tiempo, además de analizar en qué meses del año se tendrá las mejores condiciones ambientales para llevar a cabo un cultivo de calidad, si bien es cierto haciendo uso de internet se puede estar al tanto de las condiciones ambientales de la ciudad , pero estos parámetros son de una extensión territorial muy grande lo que se busca con el presente tema propuesto, es de alguna forma sectorizar y monitorear las condiciones ambientales de una extensión territorial especifica.

1.7. Impacto

Al implementar sistemas compactos de bajo costo para monitorear variables ambientales, como lo son las estaciones meteorológicas portátiles, se incentiva su uso en la población agrícola. Nuevas tecnológicas que consigo taren beneficios al agricultor, ya que al contar con un monitoreo constante de variables meteorológicas los agricultores pueden tomar mejores decisiones en cómo cuidar sus cultivos, no desperdiciar agua entre otros beneficios.

1.8. Grupo objetivo

1.8.1. Población

El personal que labora en el Rancho San Gabriel es la población a la cual es de mucha ayuda controlar y registrar parámetros meteorológicos relacionados con el clima.

1.8.2. Muestra

Los distintos parámetros meteorológicos captados por los sensores de presión atmosférica, humedad de suelo, temperatura entre otros, las cuales serán procesadas en una unidad central, dicha información será utilizada para la respectiva valoración del estado ambiental en determinadas horas de un mismo día o durante todo un año, haciendo análisis diarios.

1.9. Objetivos

1.9.1. Objetivo general

Diseñar e implementar un prototipo de estación meteorológica para el monitoreo de variables ambientales en el Rancho San Gabriel.

1.9.2. Objetivos específicos

- Desarrollar módulo electrónico con microelectrónica de sistemas programables para adquisición de datos e implementación de alarma de seguridad.
- Desarrollar módulos sensores telemétricos con tecnología inalámbrica para monitorear variables meteorológicas.
- Diseñar e imprimir en 3D los componentes del contenedor para el ensamblaje el prototipo.
- Ejecutar pruebas de funcionamiento del prototipo.
- Validar datos obtenidos de la simulación del prototipo con un sitio web meteorológico.

2. FUDAMENTOS TEORICOS

En el siguiente capítulo se encuentra el estudio y la teoría que se utiliza como fundamento teórico para el desarrollo del prototipo de estación meteorológica, con base en fuentes documentales y/o trabajos anteriores.

2.1. Estación Meteorológica

Estación meteorológica se define como una instalación dedicada a medir y registrar regularmente distintas variables del clima donde se encuentre el equipo instalado, dichos datos recolectados se utilizan para la elaboración de predicciones meteorológicas a partir de modelos numéricos y también para estudios climáticos. [2]

En la Fig. 2 se muestra una estación meteorológica profesional, la cual tiene un tamaño robusto y sensores de alta precisión para monitorear variables meteorológicas.



Fig. 2 Estación meteorológica Profesional. [3]

2.1.1. Principales Sensores utilizados en estaciones meteorológicas

En las estaciones meteorológicas comerciales se pueden encontrar estaciones que pueden captar por medio de sus sensores muchas de las variables meteorológicas que se encuentran en el ambiente tales como:

- Humedad
- Temperatura
- Velocidad de viento
- Dirección de viento
- Presión barométrica
- Lluvia
- Humedad de suelo
- Medición de radiación solar

2.1.2. Tipos de estaciones Meteorológicas

Según lo establecido por la Organización Meteorológica Mundial (OMM), las estaciones meteorológicas se las puede clasificar de la siguiente forma, tal y como se puede apreciar en la TABLA I.

TABLA I
TIPOS DE ESTACIONES METEOROLÓGICAS

Según su finalidad	Clasificación
Sinop típica	Climatológicas
	Agrícolas
	Especiales
	Aeronáuticas
	Satélites
De acuerdo con la magnitud de las observaciones	Principales
	Ordinarias
	Auxiliares
Por nivel de observación	Superficie
	Altitud
Según el Lugar de Observación	Terrestre
	Aéreas
	Marítimas

Pero hoy en día las estaciones meteorológicas se pueden también clasificarlas en dos principales grupos: Las realizadas para domicilios llamadas domésticas y las estaciones profesionales.

Una estación doméstica permite conocer los datos como temperatura, humedad y velocidad de viento y en base a estos datos obtenidos se pueden realizar tareas de riego, dichas estaciones son de bajo costo y de estructuras sencillas.

Las estaciones profesionales tienen ya integrados varios sensores, por ejemplo: anemómetro, dirección de viento, presión, estas necesitan de personas que tengan un conocimiento medio acerca de la programación de estos dispositivos, los costos de estas estaciones meteorológicas pueden llegar a ser muy elevados. [2]

2.2. Telemetría

La telemetría es una tecnología, la cual permite que las mediciones y datos recopilados en lugares remotos se transmitan de forma cableada o inalámbrica a una unidad central de control, el cual procesa la información y controla los procesos.

Un sistema de telemetría normalmente sigue un proceso, el cual consta de capturar y procesar la información desde la fuente, en este caso una estación meteorológica que capta los datos por medio de sensores, luego transmite los datos adquiridos a través de radio frecuencia o de forma cableada, se reciben y se visualizan los datos y por último se utilizan plataformas inteligentes para interpretar y ordenar los datos para luego tomar decisiones tal y como se muestra en la Fig. 3, se ilustra el proceso de un sistema telemétrico desde la recolección de información hasta el destino final. [4]



Fig. 3 Proceso de sistema telemétrico. [4]

Ventajas de la telemetría:

- Tener información en tiempo real.
- Tomar buenas decisiones basadas en mediciones y analítica.
- Conocer a tiempo los eventos y alarmas de procesos y activos.
- Eficiencia operacional.
- Mayor productividad.
- Mejor rentabilidad.

2.3. Sistema Fotovoltaico

Un sistema fotovoltaico es el conjunto de equipos eléctricos y electrónicos que producen energía eléctrica a partir de la radiación solar. El principal componente de este sistema es el módulo fotovoltaico, a su vez compuesto por células capaces de transformar la energía luminosa incidente en energía eléctrica de corriente continua. El resto de los equipos incluidos en un sistema fotovoltaico depende en gran medida de la aplicación a la que está destinado. [5]

2.4. Proteus Design Suite

Proteus Design Suite permite el diseño de circuitos y la simulación de los diferentes dispositivos electrónicos que se encuentran comercialmente, el programa antes mencionado es una solución de software que integra también el diseño de PCB, dentro del IDE Proteus interviene dos programas que permiten la realización de circuitos esquemáticos y el diseño de tarjetas PCB los cuales son ARES e ISIS, gracias a estas dos plataformas, el usuario tiene un flujo de trabajo más rápido y cómodo, esto ayuda a que el producto final pueda ser realizado en menor tiempo. [6]



Fig. 4 Proteus Design Suite. [7]

2.5. Pantalla táctil TFT

Este módulo de Pantalla TFT de bajo costo táctil es utilizada en este proyecto para simplificar el uso de botones físicos y a su vez visualizar de forma ordenada todos los datos obtenidos por los sensores que componen el prototipo de estación meteorológica, dicha pantalla se conecta directamente a los pines de entrada del Arduino mega, tal y como se puede apreciar en la Fig. 5, el cual forma parte del módulo receptor. [8]

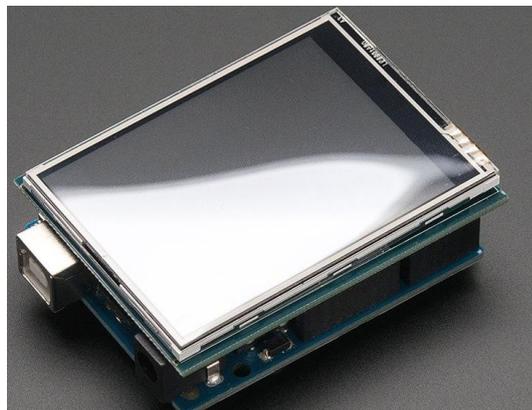


Fig. 5 Shield de Arduino de pantalla TFT táctil. [8]

Especificaciones:

- Resolución de 240x320, color de 18 bits (262.000).

- Pantalla SPI de alta velocidad con controlador de pantalla táctil SPI digital.
- La pantalla utiliza pines digitales 13-9.
- Funciona con cualquier Arduino '328 clásico. Soldar cerró tres puentes para usar el encabezado ICSP para usar con Leonardo o Mega.
- Regulador LDO de 3,3 V a 300 mA integrado.
- 4 retroiluminación LED blancos. Encendido de forma predeterminada, pero puede conectar el transistor a un pin digital para controlar la luz de fondo.
- Pantalla táctil resistiva de 4 cables conectada al controlador STMPE610.

2.6. Sensor DTH22

El Sensor DTH22 como se puede apreciar en la Fig. 6, es un sensor utilizado para medir la temperatura y humedad presentes en el ambiente ya que contiene en su interior un sensor capacitivo de humedad y un termistor para la medición de temperatura, el cual cuenta con un Pin de salida digital por donde envía los datos recolectados a la tarjeta de adquisición de datos (Arduino Mega) de la estación meteorológica. [9]

Especificaciones técnicas:

- Voltaje de Operación: 3V - 6V DC.
- Rango de medición de temperatura: -40°C a 80 °C.
- Precisión de medición de temperatura: $<\pm 0.5$ °C.
- Resolución Temperatura: 0.1°C.
- Rango de medición de humedad: De 0 a 100% RH.
- Precisión de medición de humedad: 2% RH.
- Resolución Humedad: 0.1%RH.
- Tiempo de captura de datos: 2s.
- Interface digital: Single-bus (bidireccional).
- Modelo: AM2302.
- Dimensiones: 20*15*8 mm.
- Peso: 3 gr.
- Carcasa de plástico blanco.



Fig. 6 Sensor DHT22 utilizado para medir temperatura y humedad del ambiente. [9]

2.7. Sensor BMP 180

Sensor de precisión BMP 180 es la mejor solución de detección de bajo costo para medir la presión barométrica y la temperatura del ambiente. Este sensor también puede ser usado como altímetro ya que la presión atmosférica cambia con la altitud respecto con el nivel del mar. El sensor está soldado a una PCB con un regulador de 3.3V, un cambiador de nivel I2C y resistencias pull-up en los pines I2C. [10]

Características:

- V in: de 3 a 5 VCC.
- Lógica: compatible de 3 a 5 V.
- Rango de detección de presión: 300-1100 hPa (9000 m -500 m sobre el nivel del mar).
- Resolución de hasta 0,03 hPa / 0,25 m.
- Rango operativo de -40 a + 85 ° C, precisión de temperatura de + -2 ° C.
- Esta placa / chip utiliza la dirección I2C de 7 bits 0x77.

En la Fig. 7 se aprecia una placa PCB diseñada específicamente para conectar mediante sus pines el sensor BMP 180, sensor de presión barométrica.

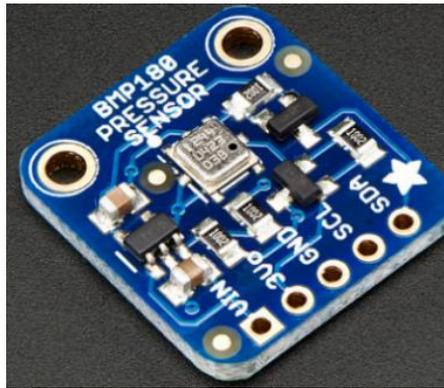


Fig. 7 Sensor BMP 180, sensor de presión barométrica de bajo costo. [10]

2.8. Panel Solar

El panel solar Fig.8, que es utilizado en el mencionado proyecto realiza la medición de lúmenes, la cantidad de luz solar que existe en una determinada área, dependiendo de la cantidad de lúmenes, se puede determinar si es un día soleado, parcialmente nublado entre otros. [11]



Fig. 8 Panel solar fotovoltaico. [11]

2.9. Arduino

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino pueden leer entradas variadas y convertirlo en una salida, activando relé para encender un motor, encendiendo un LED para indicar el estado de encendido o apagado de un equipo. Puede trabajar en conjunto con un tablero industrial enviando un conjunto de instrucciones al

microcontrolador en el tablero. Para hacerlo, utiliza el lenguaje de programación Arduino (basado en Wiring) y el Software Arduino (IDE), basado en Processing. [12]

2.9.1. Arduino IDE

El IDE es un conjunto de herramientas de software que permiten a los programadores desarrollar y grabar todo el código necesario para hacer que la placa Arduino funcione. El IDE de Arduino permite escribir, depurar, editar y grabar el programa (llamados “sketches” en el mundo Arduino) de una manera sumamente sencilla, en gran parte a esto se debe el éxito de Arduino, a su accesibilidad, en la Fig. 9 se puede observar el IDE de Arduino tal y como se mostraría al usuario o programador. [13]

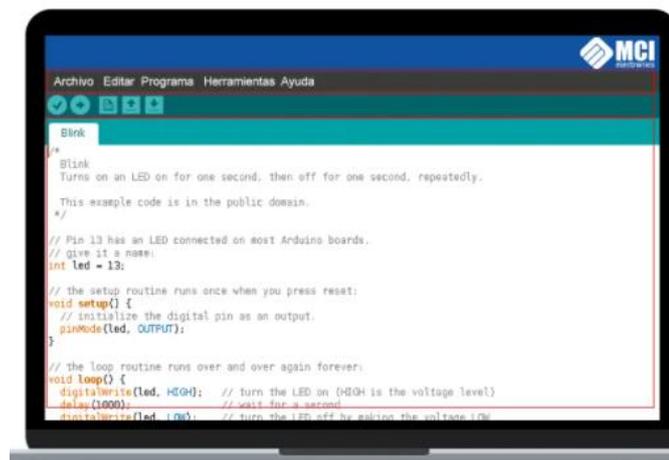


Fig. 9 Interfaz del IDE de Arduino. [13]

2.9.2. Arduino Mega 2560

La placa Arduino Mega 2560 es un PCB con un microcontrolador basado en ATmega2560, este compuesto por 54 pines de entrada/salida digital, los cuales se distribuyen de la siguiente manera, 15 se pueden usar como salidas PWM), 16 entradas analógicas, 4 UART (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP, y un botón de reinicio, sus principales especificaciones técnicas se describen en la Tabla II. [14]

En el desarrollo de este proyecto se utiliza una placa Arduino mega con chip Atmega 2560, tal y como se muestra en la Fig. 10, la cual se encarga de procesar todos los datos recolectados por los sensores como lo son, sensor de temperatura, humedad, luz solar, etc.

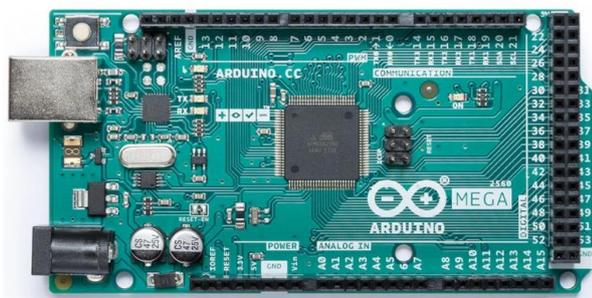


Fig. 10 Tarjeta PCB de Arduino con chip Atmega 2560. [14]

TABLA II
ESPECIFICACIONES TÉCNICAS DE ARDUINO MEGA.

Nombre	Especificaciones técnicas
MICROCONTROLADOR	ATmega2560
TENSIÓN DE FUNCIONAMIENTO	5V
VOLTAJE DE ENTRADA (RECOMENDADO)	7-12 V
VOLTAJE DE ENTRADA (LÍMITE)	6-20 V
PINES DE E / S DIGITALES	54 (de los cuales 15 proporcionan salida PWM)
PINES DE ENTRADA ANALÓGICA	dieciséis
CORRIENTE CC POR PIN DE E / S	20 mA
CORRIENTE CC PARA PIN DE 3.3V	50 mA
MEMORIA FLASH	256 KB de los cuales 8 KB utiliza el gestor de arranque
SRAM	8 KB
EEPROM	13
LARGO	101,52 milímetros
ANCHO	53,3 milímetros
PESO	37 g

En la tabla II se encuentran detalladas las principales características del Arduino Mega, el cual es utilizado en el proyecto tanto en la parte emisora y receptora como unidad central de procesamiento.

2.10. Sensor de humedad de suelo

Este sensor de humedad es utilizado en el proyecto para obtener valores de humedad de suelo y de esta forma decidir si es necesario regar el cultivo, dicho sensor está diseñado para captar las variaciones de humedad del suelo con errores de medición bajos comparados con otro tipo de sensores. [15]

El funcionamiento de este sensor se centra en medir la resistencia entre 2 electrodos insertados en el suelo, la resistencia entre los dos electrodos depende de la humedad en la que se encuentre el suelo, por lo que para un suelo muy húmedo tendremos una resistencia muy baja y para un suelo muy seco la resistencia será muy alta (circuito abierto). [15]

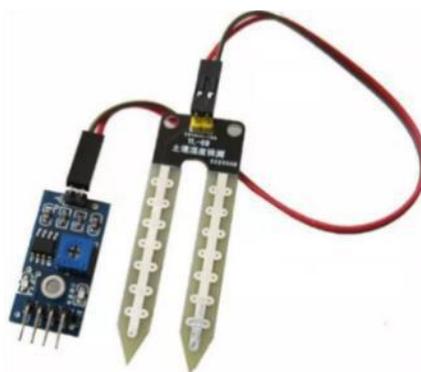


Fig. 11 Sensor de humedad de suelo. [15]

Características:

- Voltaje de alimentación: 3.3V - 5V DC (VCC).
- Corriente de operación: 35mA.
- Voltaje de señal de salida analógico (AO): 0 a VCC.
- Voltaje de señal de salida digital (DO): 3.3V/5V TTL.
- Opamp LM393 en modo comparador, umbral (threshold) regulable por potenciómetro.
- Superficie de electrodo: Estaño.
- Vida útil electrodo sumergido: 3 a 6 meses.
- Dimensiones YL-38: 30*16 mm.

- Dimensiones YL-69: 60*20*5 mm.

2.11. Convertidor voltaje DC-DC step-down LM596

Este tipo de módulo tal y como se muestra en la Fig. 12 son de gran importancia en el prototipo ya que son los encargados de regular el voltaje que proviene de la batería de 12 voltios y proteger los componentes vitales que se conectan en su salida de voltaje regulada.



Fig. 12 Módulo de convertidor de voltaje DC con chip regulador LM596 [16]

El convertidor de voltaje DC-DC Step-Down 3A LM2596 tiene como objetivo entregar un voltaje de salida constante inferior al voltaje de entrada frente a variaciones del voltaje de entrada. Soporta corrientes de salida de hasta 3A, voltaje de entrada entre 4.5V a 40V y voltaje de salida entre 1.23V a 37V. El voltaje de salida se selecciona mediante un potenciómetro multivuelta. [16]

Características:

- Voltaje de entrada: 4.5V a 40V DC.
- Voltaje de salida: 1.23V a 37V DC.
- V. salida ajustable (el voltaje de entrada debe tener al menos 1.5V más que la salida).
- Corriente de Salida: máx. 3A, 2.5A recomendado (usar disipador para corrientes mayores a 2A).
- Potencia de salida: 25W.
- Eficiencia de conversión: 92%.

- Regulación de carga: $S(I) \leq 0.5\%$.
- Regulación de voltaje: $S(u) \leq 0.5\%$.
- Frecuencia de Trabajo: 150KHz.
- Ripple en la salida: 30mV (máx.) 20M bandwidth.
- Protección de sobre-temperatura: SI (apaga la salida).
- Protección de corto circuito: SI (hasta 5A).
- Protección limitadora de corriente: SI.
- Protección frente a inversión de polaridad: NO.
- Dimensiones: 43mm*21mm*13mm.
- Convertidor DC-DC Buck: LM2596.

2.12. Regulador de Carga Panel solar

Un regulador de carga solar básicamente es un dispositivo electrónico que controla constantemente el estado de carga y descarga de las baterías regulando su tensión para realizar un llenado óptimo y alargar su vida útil además de proteger contra sobrecarga y pico de voltaje. Este componente es vital en todo sistema de carga fotovoltaico el cual se puede observar en la Fig. 13. [17]



Fig. 13 Módulo regulador de voltaje y carga de panel solar [18]

Características:

- Color: Negro
- Voltaje nominal: 12 V o 24 V.

- Corriente de carga nominal: 10 A.
- Corriente de carga nominal: 10 A.
- Voltaje de parar fuente de alimentación: * 10.8 V o 21.6 V.
- Voltaje de reanudar fuente de alimentación: * 11.8 V o 23.6 V.
- Voltaje de parar la carga: * 14 V o 28 V.
- Temperatura de trabajo: -20 °C ~ + 60 °C.

2.13. Sensor de gotas de lluvia YL83

En la Fig. 14 se muestra el sensor utilizado para detectar las precipitaciones de lluvia, el cual se compone de electrodos expuestos al ambiente para poder detectar de forma efectiva las gotas de lluvia, se alimenta con un voltaje de 5 voltios, tiene una salida digital y analógica la cual la hace ideal para trabajar con tarjetas embebidas como Raspberry Pi y Arduino. [19]

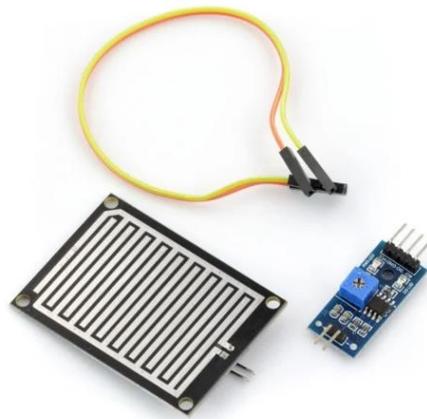


Fig. 14 Módulo sensor de lluvia para Arduino [19]

2.14. Módulo RFID

Este módulo utiliza un transceptor de 2,4 GHz de Nordic Semiconductor, modelo nRF24L01+. En cual opera en la banda de 2,4 GHz (frecuencia libre). Este módulo es compatible con 6 canales de recepción de datos debido a este módulo se pueden obtener procesamientos de señales de alta velocidad en comparación con el protocolo RF, como por ejemplo la retransmisión automática de paquetes perdidos y generación de señales automáticas de contestación. En la Fig. 15 se muestra el módulo de antena RFID montado en una placa con todos los componentes necesarios para funcionar en conjunto con Arduino.

Ideal para proyectos de telemetría, control de periféricos, aplicaciones de mandos de video juegos, juguetes, etc. Trabaja con velocidades máximas de operación de hasta 2Mbps, GFSK eficiencia de modulación, la capacidad anti-interferencia, especialmente adecuado para aplicaciones de control industrial. [20]



Fig. 15 Módulo de antena RFID de largo alcance. [21]

Características

- Fuente de alimentación: 1.9 ~ 3.6V.
- voltaje de funcionamiento del puerto IO: 0 ~ 3.3v / 5v (Tolerante a 5V).
- Consumo Corriente: 115 mA.
- Tasa de trasmisión: +20 dBm.
- Recepción de sensibilidad: $\leq -95\text{dBm}$ a 1Mbps.
- El alcance de transmisión: 1000m en zona abierta.
- Dimensiones: 16.5mmx45.5mm.
- Funcionamiento en la banda ISM de 2,4 GHz, por lo que no necesita licencia y es libre en todo el mundo.
- 3 velocidades de Datos seleccionables: 250Kbps, 1Mbps y 2Mbps.
- Acelerador por hardware del protocolo SPI (ShockBurst TM).
- Consumo energético ultra bajo, capaz de durar años utilizando una batería.

2.15. Sensor MPU6050

El MPU-6050 es el primer dispositivo de seguimiento de movimiento integrado de 6 ejes del mundo que combina un giroscopio, acelerómetro de 3 ejes y un Digital Motion Processor™ (DMP), todo en un pequeño paquete de 4x4x0.9mm. Con su bus de sensor I2C dedicado, acepta directamente entradas desde una brújula externa de 3 ejes y proporciona una salida completa de MotionFusion de 9 ejes. El dispositivo de seguimiento de movimiento MPU-60X0, muestra sus 6 ejes. Este sensor también está diseñado para interactuar con múltiples dispositivos no inerciales sensores digitales, como sensores de presión, en su auxiliar Puerto 2C.

Tal y como se muestra en la Fig. 16 se muestra el módulo MPU-6050, el cual cuenta con tres convertidores de analógico a digital (ADC) de 16 bits para digitalizar las salidas del giroscopio y tres ADCs de 16 bits para digitalizar las salidas del acelerómetro. Para un seguimiento preciso tanto rápido como lento de movimientos, las partes cuentan con un giroscopio programable por el usuario a escala completa de ± 250 , ± 500 , ± 1000 , y ± 2000 ° / seg. (Dps) y un rango de escala completa de acelerómetro programable por el usuario de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$. [22]

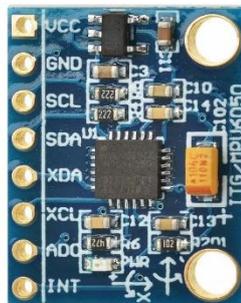


Fig. 16 Módulo de acelerómetro y giroscopio. [23]

2.16. Batería Lipo

Una batería LiPo (Litio y polímero) es un tipo de batería recargable muy habitual en el mundo de los multirrotores. Nacen como una opción aceptable a la utilización de combustibles para realizar vuelos. Son muy recomendables ya que ofrecen unas prestaciones superiores a las NiCd y NiHm.

Una batería LiPo puede almacenar una gran cantidad de energía y puede fabricarse en medidas personalizadas. Ofrecen una tasa de descarga muy alta. En la Fig. 17 se aprecia una batería lipo de alta eficiencia de dos celdas a 7.4v voltios [24]



Fig. 17 Batería Lipo de alta eficiencia. [24]

2.16.1. Significado de la nomenclatura de una batería LiPo

Las baterías LiPo tienen la particularidad de disponer de un mayor voltaje por celda que otros tipos. Pueden llegar a los 4.2 v cuando están completamente cargadas. El número S se corresponde con el número de celdas. Una batería 4S (4 celdas) estaría compuesta por tanto de 4 sub-baterías conectadas en serie. [25]

2.16.2. Capacidad indicada en mAh

Las baterías LiPo tienen la particularidad de disponer de un mayor voltaje por celda que otro tipo de batería a mayor número de miliamperios (mAh) más capacidad de carga. En principio esto se podría entender que recurrir a baterías de mayor capacidad brinda la posibilidad es mejor, pero esto acarrea una dificultad para poder obtener mayor corriente en este tipo de baterías también deben aumentar sus números de celdas lo que provoca que cada vez sean más pesadas, por lo tanto, se debe elegir la mejor configuración de baterías en la aplicación que de desee utilizar. Tal como se muestra en la Fig.18 se detalla las especificaciones del fabricante que imprime como etiqueta en la batería. [25]

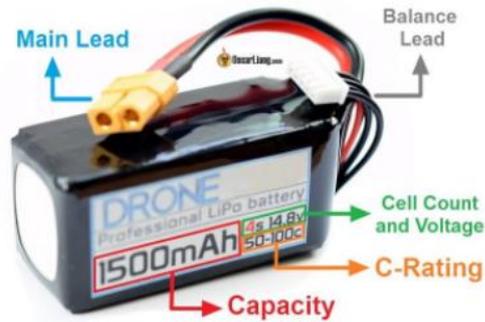


Fig. 18 Significado de siglas y números impresos en batería Lipo. [26]

2.16.3. Capacidad de descarga, Numero C

La tasa de descarga de un batería tipo se interpreta por el numero C que viene descrito en la etiqueta de información con la cual se puede realizar un cálculo sencillo (Máxima descarga = C Rating x capacidad) para obtener la tasa de descarga máxima de una batería LiPo. [26]

3. MARCO METODOLOGICO

3.1. Metodología

3.1.1. Métodos

En el siguiente apartado se muestra detalladamente técnicas métodos, y procedimientos, utilizados en la realización de este proyecto de titulación.

3.1.2. Método deductivo

Se hace uso de esta metodología a durante el diseño e implementación de la parte electrónica del prototipo de estación meteorológica portátil. Se requieren tener conocimientos de las siguientes áreas

- Redes de computadoras.
- Electrónica analógica y digital.
- Programación de microcontroladores

Este método se sustenta básicamente por, la reproducibilidad y la refutabilidad la cuales se describen brevemente a continuación

La reproducibilidad. - Tiene relación con hacer experimentos y ensayos que, bajo ciertas condiciones estructurales, ambientales, etc. tienen la posibilidad de repetir y obtener los mismos resultados en cada repetición.

La refutabilidad. – Involucra que cualquier resultado logrado podría ser discutido por medio de otros experimentos, lo cual implica que los resultados no son incondicionales.

3.1.3. Método experimental

El método experimental se basa en la realización constante de pruebas a una hipótesis, con la finalidad de comprobar, corroborar y refutar dicha hipótesis, las pruebas realizadas pueden darse en ambientes controlados, como laboratorios de ensayos, o en ambientes un poco más inestables en donde los resultados varían según el entorno donde se realice la prueba.

La metodología antes mencionada se utilizó durante la realización del programa que controla las funciones del sistema biométrico para el manejo y utilización del siguiente hardware.

- Programación de Arduino.
- Programación de antenas RFID.
- Programación de Pantalla TFT.

3.1.4. Técnicas

Se define como técnicas al conjunto de procedimientos o pasos para lograr un objetivo, en todo proyecto es necesario definir las técnicas a utilizar antes de llevar a cabo un trabajo. Las técnicas empleadas en este proyecto de titulación son:

3.1.5. Técnica de investigación

Técnica que surge a partir de la observación de un fenómeno con el objetivo de tomar o registrar datos que permitan entender el comportamiento del mismo a través de su análisis. La técnica de investigación permite desarrollar y agilizar la investigación científica la cual cumple un papel fundamental para llevar a cabo la exploración del hecho estudiado.

3.1.6. Observación

Considerada una de las técnicas más antiguas y rudimentarias que existen, permite la extracción de información o datos del fenómeno estudiado permitiendo realizar un análisis con los registros obtenidos.

3.1.7. La entrevista

Técnica que permite la recolección de datos por medio del dialogo entre un entrevistador o científico, con uno o varios individuos que estén en contacto directo con el fenómenos o caso de estudio. La información rescatada de esta técnica dependerá netamente de la información obtenida del entrevistador hacia los entrevistados.

3.1.8. Recolección de información

Con la utilización de las técnicas antes mencionadas, se pueden clasificar la información en diferentes conjuntos según la importancia de la investigación obtenida. Para el presente trabajo científico-técnico se recolectó la información de fuentes primarias y secundarias, de trabajos científicos relacionados, así como sistemas de seguridad disponibles en el mercado.

3.2. Diseño del prototipo de estación meteorológica

Para este proyecto se diseñaron dos dispositivos portátiles uno se encarga de la recolección de datos obtenidos por los sensores para enviarlo al receptor, el cual recibe los datos de temperatura, humedad de suelo, cantidad de luz solar etc. y los muestra en una pantalla de forma ordenada, ambos funcionan con baterías recargables para poder hacer portable y compacto el transporte del prototipo.

El módulo de emisor está conformado principalmente por un módulo de antena RFID conectado Arduino mega 2560 por medio de comunicación serial, el cual mediante sus pines de entrada y salidas tanto digital y analógicas permiten conectar diferentes sensores, como lo son sensores de temperatura, humedad de suelo y ambiente, índice de lluvia, cantidad de luz solar, presión atmosférica, etc. Además, la programación del Arduino Mega se realizó de forma que envíe continuamente todos los datos obtenidos en tiempo real, adicional a esto el módulo emisor del prototipo de estación meteorológica integra un sensor MPU 650, el cual se utiliza para activar una alarma sonora dado el caso si el prototipo intenta ser robado, el sensor MPU 650 detecta los cambios de los valores de posición del prototipo en el espacio, esto ocasiona que se active una alarma sonora en el módulo receptor que notifica al usuario que el prototipo posiblemente está siendo hurtado.

En el módulo receptor está compuesto por: Arduino mega 2650, antena RFID y una pantalla táctil la misma que permite seleccionar y mostrar los valores obtenidos de un sensor específico.

3.3. Diseño de tarjetas de circuito impreso.

En la Fig. 19 se muestra el PCB realizado para poder suministrar voltaje y conectar al Arduino todos los sensores que conforman el prototipo de estación meteorológica

portátil, adicional a esto, el diseño del PCB permite gestionar mejor los cables y la conexión de los sensores.

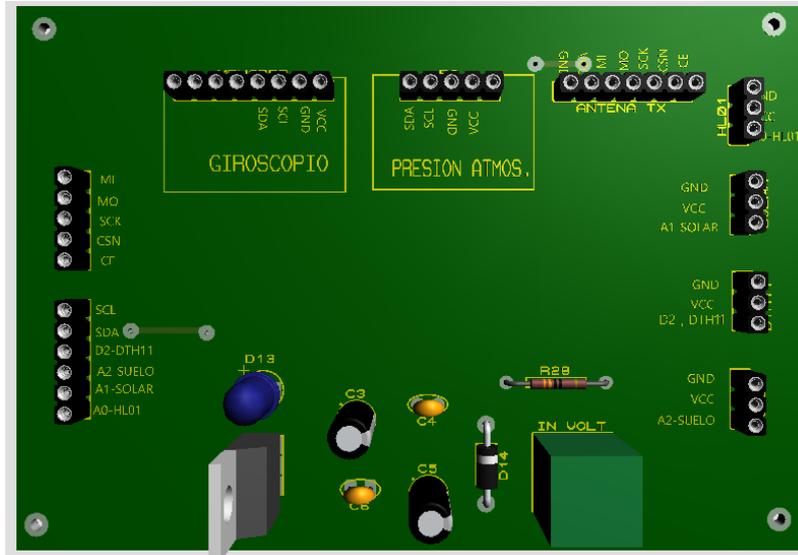


Fig. 19 Diseño de tarjeta vista 3D.

En la Fig.20 se muestra la interconexión, pistas y trayectorias de todos los componentes que van conectados al Arduino Mega, el cual realiza la función de unidad central de proceso en el módulo receptor.

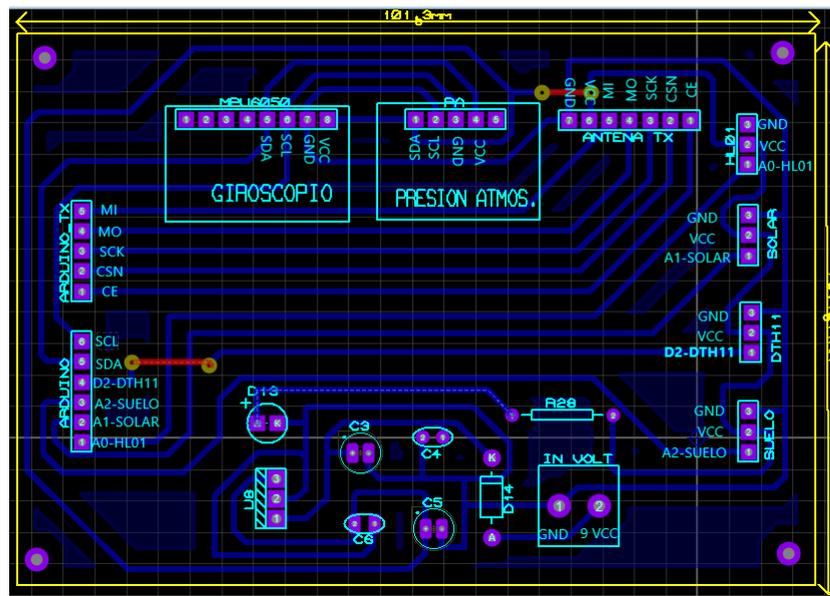


Fig. 20 Diseño de PCB de sensores pistas.

En la Tabla III se detalla todos los componentes utilizados en la elaboración para el PCB de sensores.

TABLA III.
COMPONENTES UTILIZADOS EN PCB DE SENSORES

Componentes electrónicos	
Elemento	Cantidad
LM 7805	1
Diodo rectificador 1N4007	1
Capacitores de 100 nF	1
Capacitores de 220uF	1
Capacitores de 10uF	1
Resistencia de 330 ohmios	1
Diodo LED	1
Espadín hembra	1
MPU 6050	1
Conector Bornera	1
BPM180	1
Antena RFID	1
Panel Solar	1
DHT11	1
YL83	1
Sensor Humedad de suelo	1

3.4. Conexión y diagramas esquemáticos

En este apartado se recopilan y se detallan mediante diagramas esquemáticos las conexiones utilizadas para interconectar los diferentes dispositivos que conforman este prototipo.

El siguiente diagrama esquemático corresponde al PCB realizado para interconectar todos los sensores con Arduino, tal y como se puede apreciar en la Fig. 21, el diagrama esquemático realizado en el programa Proteus.

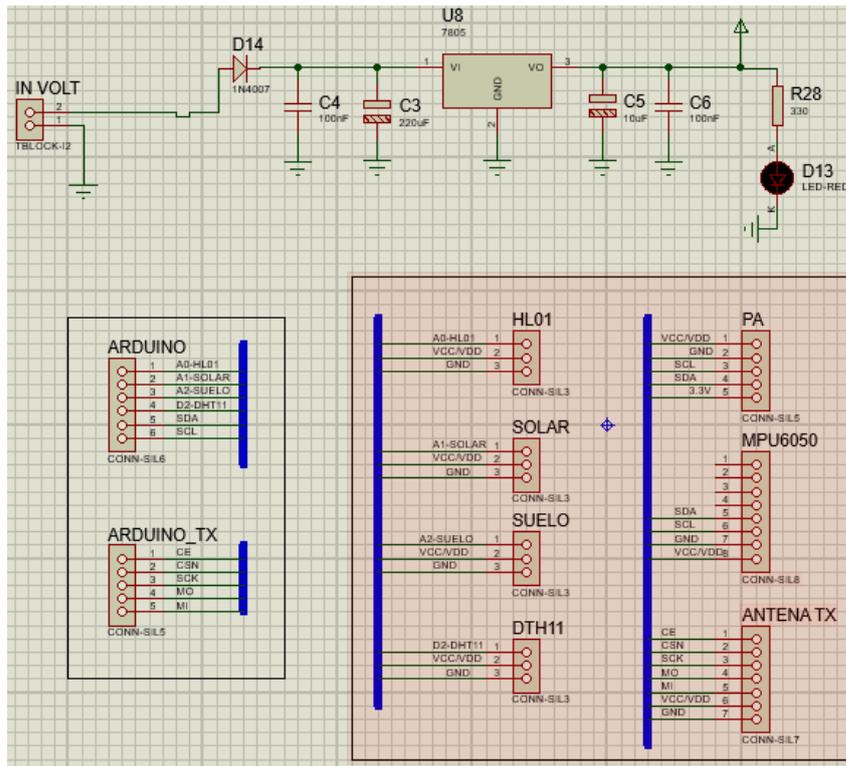


Fig. 21 Diagrama esquemático de conexión de sensores hacia Arduino.

3.4.1. Diagrama de conexión de fuente de alimentación módulo emisor

En la Fig.22 se aprecia las conexiones del sistema fotovoltaico que suministra la energía necesaria para que el prototipo de estación meteorología sea autosustentable, el sistema fotovoltaico se conforma principalmente por el panel solar de 30w, controlador de carga y batería de plomo de 12 voltios. Una vez el panel solar es expuesto al sol, genera un voltaje de aproximadamente 14 voltios DC, el panel solar se conecta en la entrada de alimentación del controlador de carga, el cual posee una entrada de voltaje y dos salidas, una vez el controlador de carga entra en funcionamiento, regula tensión de la salida de voltaje donde va conectada la batería en 13 voltios DC, la cual es la tensión adecuada para cargar la batería sin que esta se deteriore y por último se encuentra los terminales de salida para poder conectar cualquier carga que pueda funcionar con 12 voltios DC, en esta caso se conecta un regulador de tensión para poder conectar, sensores y placa Arduino del modulo emisor.

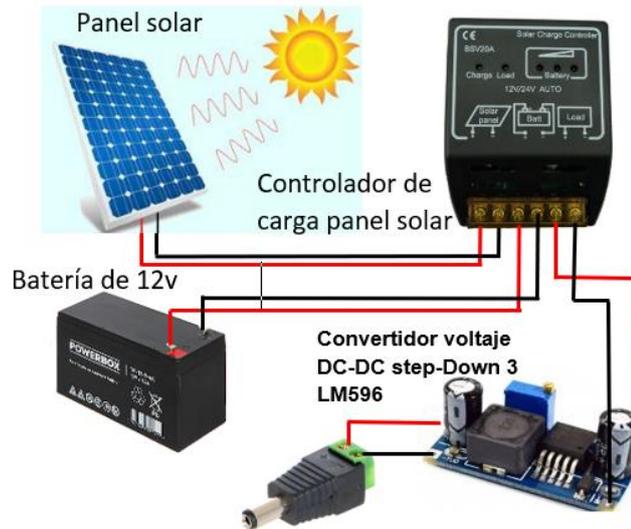


Fig. 22 Conexión de fuente de alimentación

3.5. Diagrama de flujo de funcionamiento de módulo emisor

En la Fig. 23 se observa detalladamente los elementos que conforman el módulo emisor, el cual es el encargado de recolectar y enviar inalámbricamente los datos obtenidos por los sensores hacia el módulo receptor. Todos los sensores están conectados a un PCB diseñado en el Software Proteus, este PCB se diseñó con la finalidad de obtener una mejor gestión de Cables y ser también una interfaz física, entre los sensores y el Arduino Mega 2560, ya que a esta parte del proyecto se conectan muchos sensores se debe considerar que el consumo de corriente será mucho mayor, del que puede entregar la fuente de alimentación de la propia Placa Arduino, es por esta razón que el PBC donde van conectados los sensores posee una fuente alimentación regulada independiente.

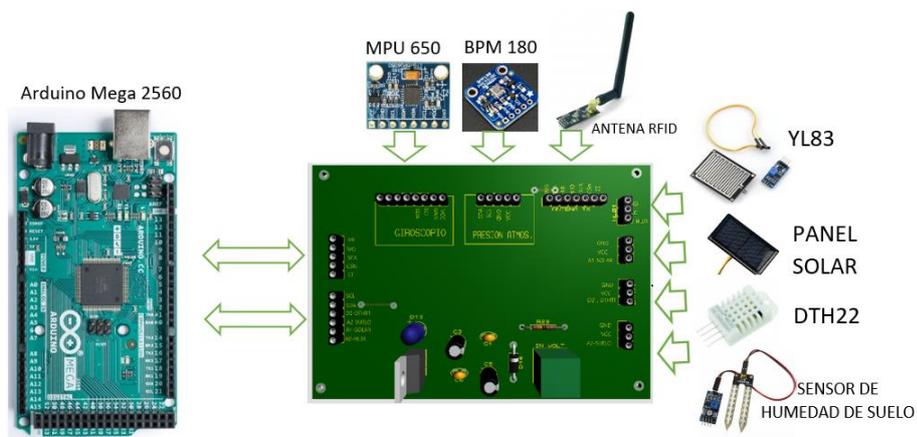


Fig. 23 Diagrama de funcionamiento de módulo emisor.

3.6. Diagrama de flujo de funcionamiento de módulo receptor

Como se detalla en la Fig. 24, se muestra todos los elementos que conforman el módulo receptor, este módulo es el encargado de recibir, procesar y mostrar de forma ordena los datos que recibe.

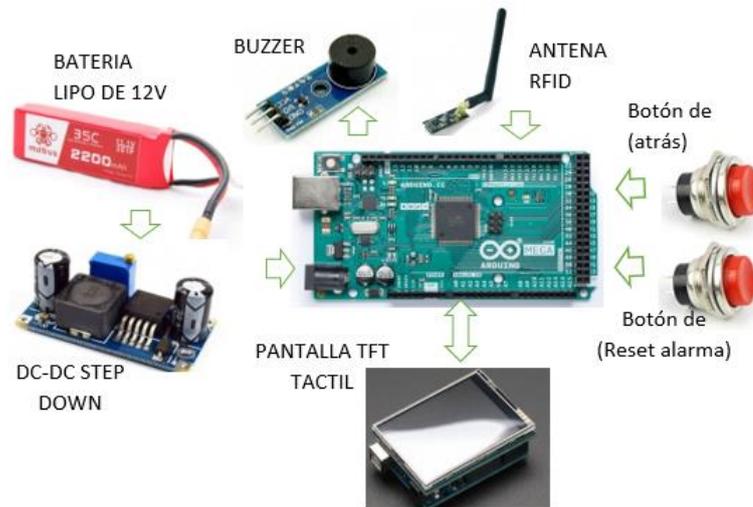


Fig. 24 Módulo receptor diagrama de flujo de entradas y salidas.

3.7. Diseño e impresión 3D módulo receptor.

3.7.1. Diseño de estructura de módulo receptor

Para el diseño de la estructura del módulo receptor se utilizó un software de modelado en 3D AutoCAD, ya que con la ayuda de las herramientas que dispone, se puede exportar el diseño en un archivo apropiado para ser procesado por una impresora 3D, en la Fig. 25 en el diseño se aprecia que se consideró, colocar una tapa frontal removible, para poder acceder a la electrónica que conforma el módulo receptor, además de diseñar espacios para los componentes externos como los son: Pantalla TFT, Voltímetro, botones, antena, interruptor de encendido y conector para cargador.

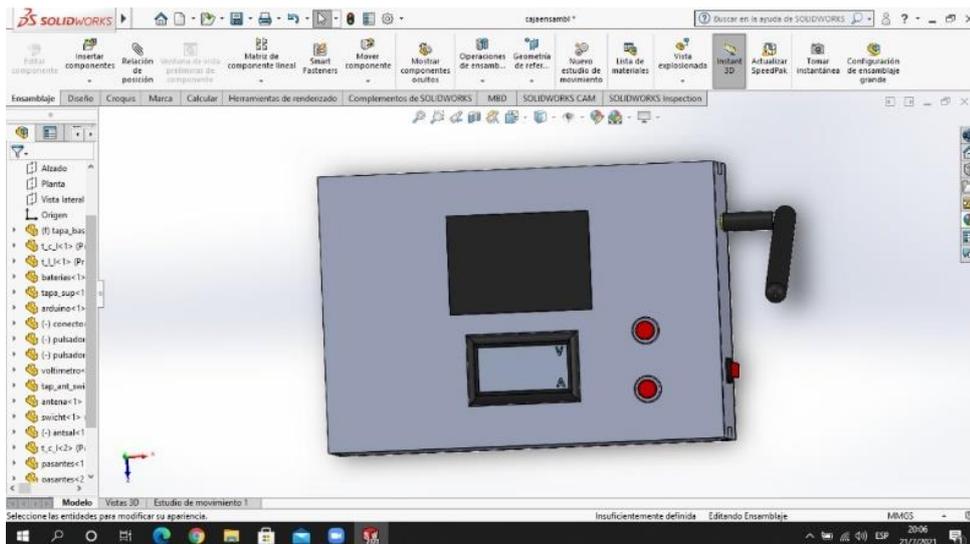


Fig. 25 Vista frontal de diseño en SolidWorks.

Las dimensiones de la caja son de: 6 cm (alto) x 13 cm (ancho) x 6 cm (profundo), estas medidas fueron considerando las dimensiones de los componentes internos del módulo receptor.

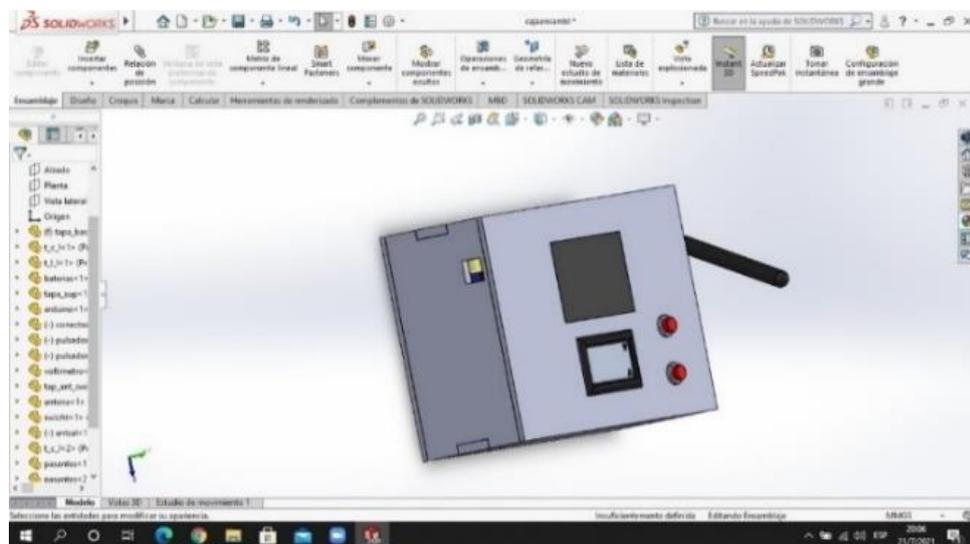


Fig. 26 Diseño de estructura de módulo receptor en SolidWorks.

3.7.2. Impresión de estructura de módulo receptor

La estructura del módulo receptor se realizó de tal forma que las piezas encajen unas con otras y de esta forma evitar colocar adhesivos en las esquinas.



Fig. 27 Impresión 3D módulo receptor.

3.8. Pruebas y Conexiones físicas del prototipo

Conexión y cableado de sensores hacia el Arduino para la prueba y ejecución del algoritmo programado en el IDE de Arduino el cual se puede apreciar el Anexo 1.

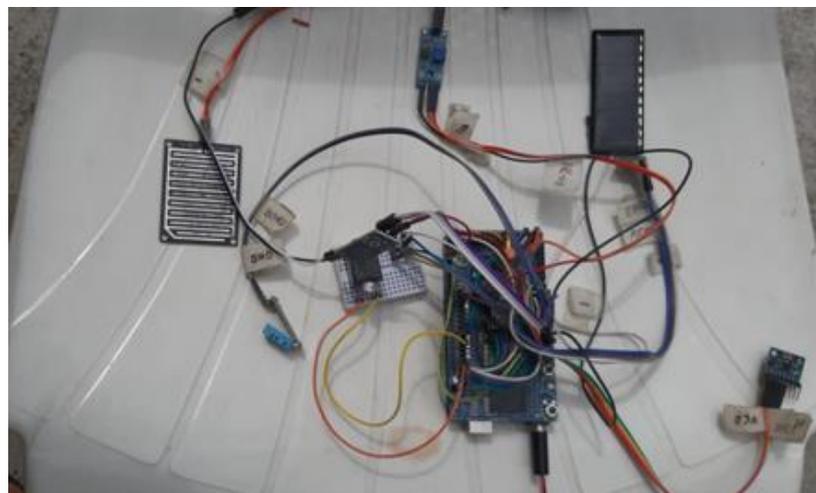


Fig. 28 Conexión y prueba de sensores módulo emisor.

En la Fig.28 se observan las conexiones cableadas de cada uno de los sensores que conforman el modulo receptor tal y como lo son el sensor DHT22, MPU6050, BPM180, panel solar de 5v, YL83, sensor de humedad de suelo, las pruebas preliminares se realizaron conectando todos los sensores hacia Arduino y ejecutando un programa de prueba para verificar el estado de funcionamiento de cada sensor.

3.8.1. Conexión de alarma sonora

Este módulo de Buzzer para Arduino se encuentra en el módulo receptor para notificar de un posible robo de la estación remota.

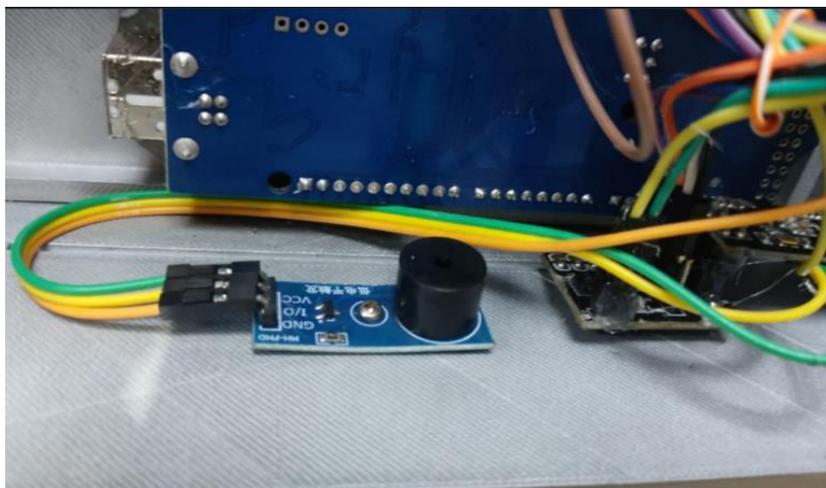


Fig. 29 Conexión de buzzer (Alarma).

En la Fig. 29 se muestra la alarma sonora que trabaja en conjunto con el sensor acelerómetro y giroscopio MPU6050, el cual está constantemente en funcionamiento y es capaz de detectar los cambios en la posición en la que se encuentra el espacio le modulo receptor, es por esta razón que cuando se detecta un cambio brusco en los valores captados, el módulo emisor interpreta estos valores como una alerta para activar la alarma sonora, esto quiere decir que por algún motivo el prototipo de estación meteorológica se ha movido de su posición original.

3.8.2. Conexión de módulo regulador de voltaje

El módulo receptor funciona a un voltaje de 5 voltios regulados es por esta razón que el Arduino Mega2650 debe estar acompañado de un módulo StepDown para poder regular y disminuir el voltaje de 12 voltios DC de la batería LiPo.

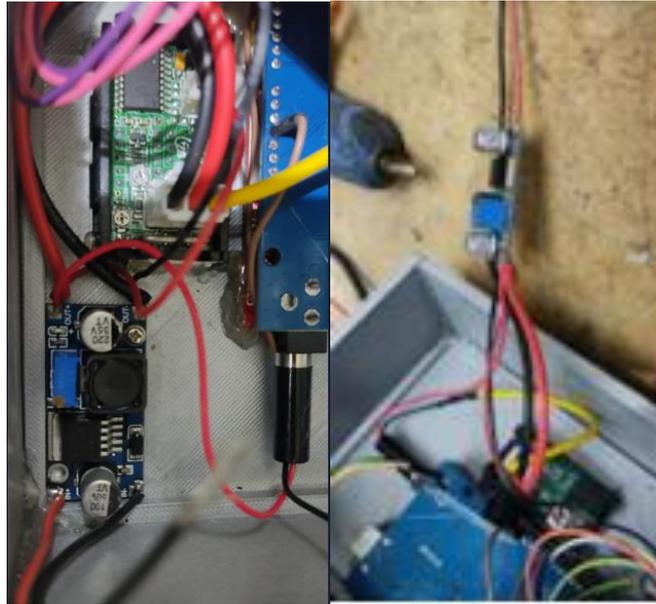


Fig. 30 Conexión de Step down en módulo receptor.

3.8.3. Conexión de batería Lipo a modulo receptor

El módulo receptor necesita ser alimentado por una fuente de alimentación recargable como lo es una batería LiPo de 12 voltios tal y como se muestra en la Fig. 31, se elegio este tipo de batería ya que tiene excelentes características que la hacen superior a otras baterías que se encuentran en el mercado ya que mantiene una tasa de descarga muy baja en comparación a baterías de plomo y son muy compactas para este tipo de aplicaciones.



Fig. 31 Conexión de Batería Lipo.

3.8.4. Conexión de sensores a tarjeta electrónica de fuente de alimentación

Se diseñó una tarjeta electrónica donde se conectan todos los sensores que conforman el prototipo con el fin de gestionar de mejor forma los cables de alimentación y señal que van hacia el Arduino.

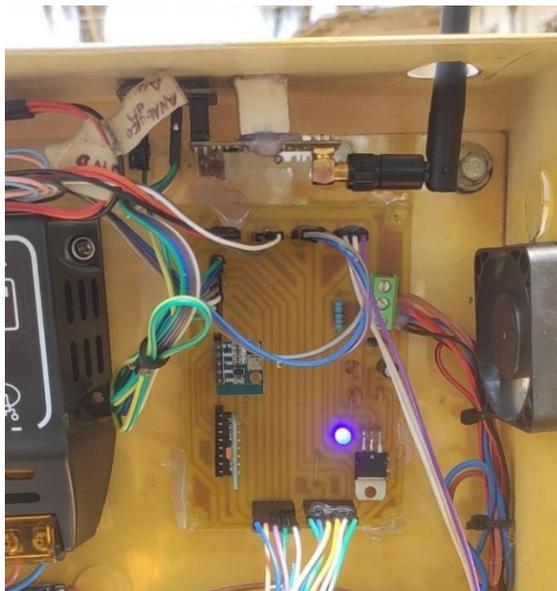


Fig. 32 Tarjeta de conexión de sensores.

En la Fig. 32 se muestra el PCB diseñado para conectar todos los sensores del modulo receptor, se puede apreciar que la tarjeta diseñada, permite tener cables más ordenados y una fuente de alineación regulada independiente de la fuente de alimentación principal.

3.8.5. Conexión de Panel solar y controlador de carga

Tal y como se muestra en la Fig. 33, la conexión del panel hacia el regulador de carga acompañado del paquete de batería, en primera instancia se realiza la prueba de voltaje del sistema fotovoltaico sin carga para comprobar con el instrumento de medición Multímetro que todos los voltajes sean los adecuados.



Fig. 33 Conexión de módulo de carga y panel solar.

En la Fig. 34 se visualiza el sistema fotovoltaico funcionando con carga y energizando correctamente todos los componentes del módulo emisor.



Fig. 34 Conexión de sistema fotovoltaico con carga.

3.9. Diagramas de flujo de programación de módulo emisor y receptor

En los siguientes diagramas de flujo de la Fig. 35 se observa el desglose de la lógica de programación utilizadas tanto en el módulo emisor como receptor.

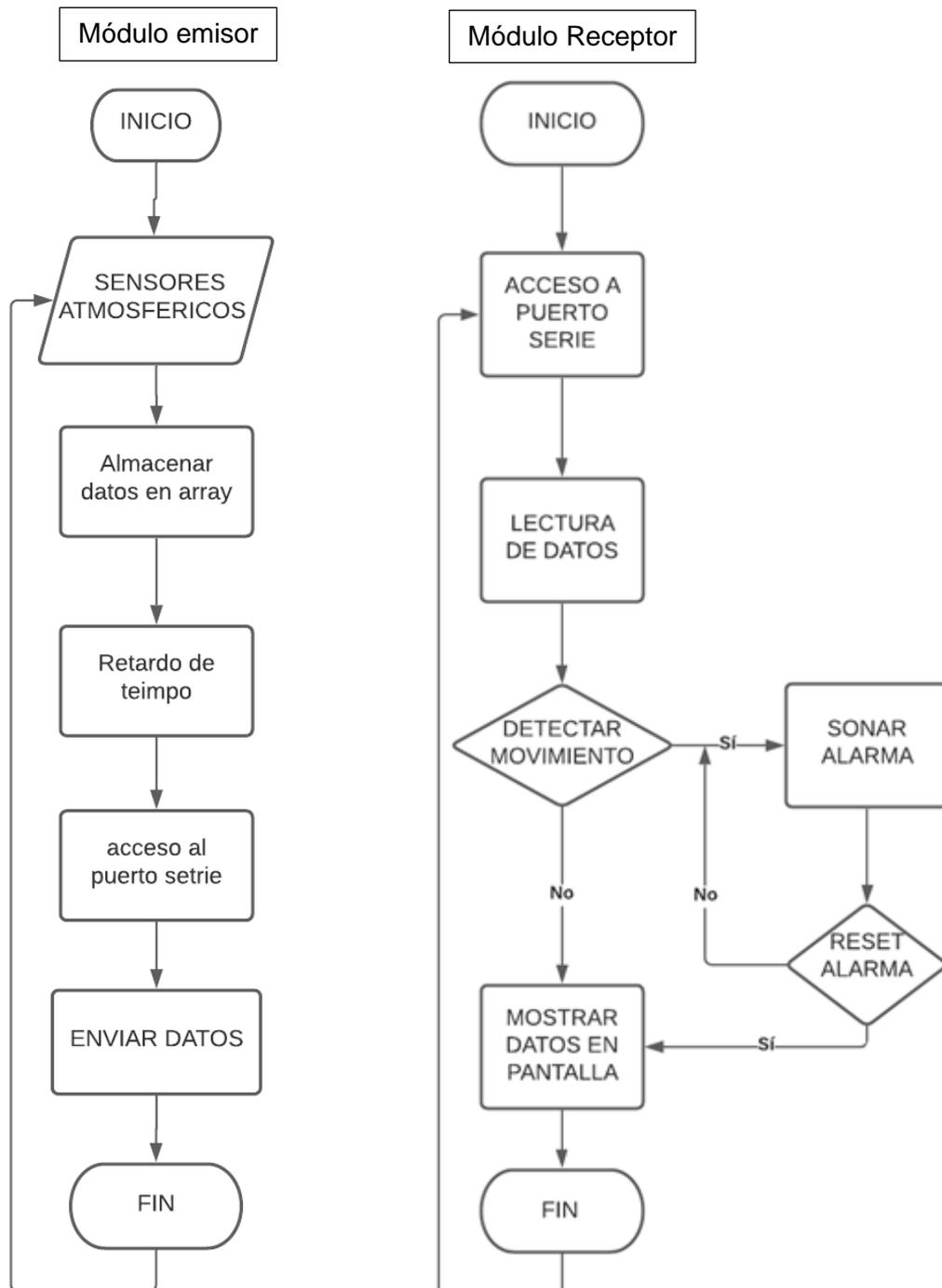


Fig. 35 Diagramas de flujo de módulo receptor y emisor.

3.10. Código de programación utilizado

Luego de probar todos los sensores y módulos de antenas RFID con fragmentos de código que específicamente hacen funcionar cada sensor y verificar que trabajen correctamente, se procede a unir todas las partes de códigos previamente probados, en un solo código fuente tal y como se muestra en el (Anexo 1), el cual será el responsable de hacer que el prototipo funcione como se desea, la programación se realizó en el IDE Arduino y los “Sketchs” realizados tiene como nombre, “Emisor” para la parte del prototipo que envía datos al receptor, y “Receptor” para la parte que se encarga de procesar y mostrar los datos recibidos.

4. RESULTADOS

Con la implementación de este prototipo de estación meteorológica se han obtenido mayor control en los diferentes cultivos que se realizan en el rancho san Gabriel en cuestión de datos meteorológicos, de esta forma se obtuvo como resultado cultivos más productivos.

A continuación, se muestra el análisis y registro de resultados del DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN METEOROLÓGICA PORTÁTIL PARA EL MONITOREO DE VARIABLES AMBIENTALES EN EL RANCHO SAN GABRIEL UTILIZANDO TECNOLOGÍA RFID.

4.1. Prototipo de estación meteorológica modulo emisor

El módulo emisor se conforma de un Arduino mega, módulo de comunicación RFI de largo alcance, batería, controlador de carga, panel solar, sensores de humedad, luz, temperatura, presión atmosférica, lluvia, entre otros tal y como se muestra en la Fig.36. Se encuentra ubicada en el cultivo más cercano del rancho san Gabriel aproximadamente 80 metros, tomando como punto de recepción la casa del propietario del rancho el cual se encarga de enviar todos los datos captados por los sensores al módulo receptor.

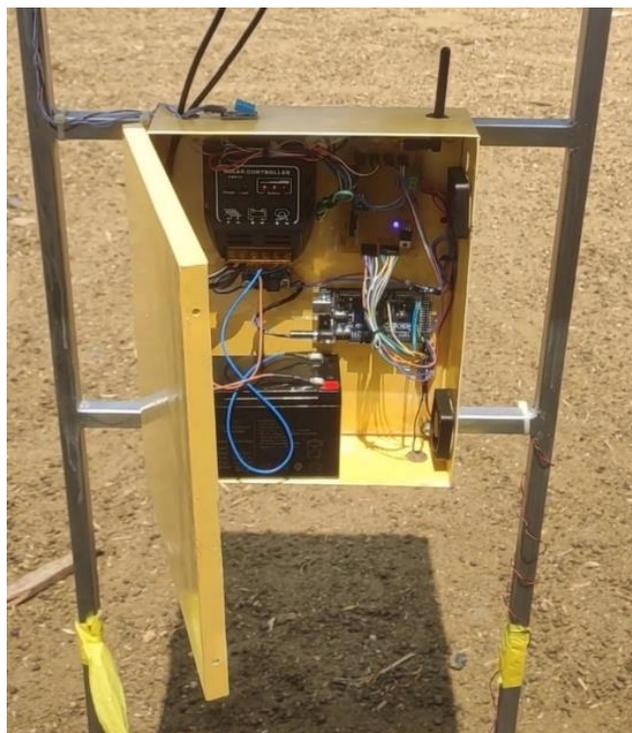


Fig. 36 Modulo emisor.



Fig. 37 Prototipo de estación meteorológica instalada en el Rancho San Gabriel (Modulo Emisor).

4.1.1. Sensor de temperatura y humedad

En la figura 38 se puede apreciar la ubicación estratégica del sensor de temperatura y humedad del ambiente, se instaló en la parte externa del módulo emisor para poder captar la temperatura y humedad con el menor error de medición posible según las prestaciones y características del sensor empleado.



Fig. 38 Sensor DHT11 instalado en el exterior de la estructura.

4.1.2. Panel solar y sensores de lluvia y luz

El panel solar está situado en la parte superior de la estructura del módulo emisor ya que debe recargar la batería interna del mismo, además se puede apreciar en la figura 39 que los sensores de lluvia y cantidad de luz se encuentran instalados encima del panel solar con el propósito de minimizar el error de medición y tener un mayor acceso y visibilidad, de los sensores antes mencionados para su fácil limpieza y mantenimiento, ya que al estar expuestos pueden sufrir deterioro por exceso de polvo, restos de plantas o insectos, los valores medidos llegarán a ser erróneos,

La información obtenida es enviada a la tarjeta de adquisición para su posterior procesamiento.



Fig. 39 Panel solar, sensor de lluvia y sensor de luz.

4.1.3. Controlador de carga (fuente de alimentación)

En el siguiente prototipo es necesario implementar un controlador de carga con su respectiva batería para suministrar la energía necesaria al circuito que es el encargado de adquirir y procesar los datos obtenidos de los sensores y así enviarlos al módulo receptor.



Fig. 40 Controlador de carga.

Encargada de alimentar el circuito del emisor, donde su aproximado de duración es mayor a 12 horas.



Fig. 41 Batería conectada al módulo de carga.

4.1.4. Sensor de humedad de suelo

Se visualiza al sensor de humedad de suelo quien mide en porcentaje la cantidad que se encuentra en el mismo, para de esta forma considerar humedecer el cultivo.



Fig. 42 Obteniendo valores de humedad de suelo.

4.1.5. Módulo receptor

Procesa y muestra los datos que envía el emisor, por lo que se colocó una pantalla para así apreciar los valores de este, de tal forma que se interactúe con el menú de pantalla de forma táctil sin la utilización de botones físicos.



Fig. 43 Módulo receptor.

4.2. Menú del módulo receptor

Se aprecia todas las opciones que dispone el módulo receptor, en donde al presionar de forma independiente se ejecutan cada una de las funciones que se encuentran en el mismo.



Fig. 44 Apreciación del menú que dispone el módulo receptor.

4.2.1. Visualización de datos del sensor de lluvia

Luego de la respectiva prueba de funcionamiento se aprecia el valor obtenido y procesado del sensor de lluvia HL83 en porcentaje el cual indica que valores en el rango de 1 al 100 % en donde 1 en ausencia de lluvia clima seco, y menor o igual a 100% indica índice de precipitación de lluvia.



Fig. 45 Visualización sensor de lluvia

4.2.2. Visualización de valor humedad de suelo módulo receptor

El valor de humedad de suelo se puede apreciar en el módulo receptor y está representado con (%) ya que se trata de humedad relativa, en donde 1% representa ausencia de humedad representa suelo seco y valores mayores de 50% de humedad representa suelo mojado, este dato es importante para decidir si regar o no el cultivo, para de esta forma poder administrar eficientemente el líquido vital (agua) destinado para humedecer el terreno del cultivo.



Fig. 46 Humedad de suelo.

4.2.3. Visualización de valor presión atmosférica

Se observa los datos que se obtiene de la presión atmosférica en determinado lugar (Rancho San Gabriel), la cual indica si es beneficioso la siembra de cierto cultivo, ya que influye en el rendimiento y calidad del producto final.



Fig. 47 Presión atmosférica.

4.2.4. Visualización de valor humedad, cantidad de luz y temperatura ambiente

Dependiendo de la cantidad de luz solar, el agricultor puede predecir con mayor fiabilidad si el clima estará lluvioso o permanecerá soleado durante el transcurso del día, el mismo que va a trabajar junto con el sensor de lluvia, en donde 1500 lux representa a día nublado y 100000 lux representa a día soleado.

El prototipo cuenta también con un sensor DHT11 con la finalidad de visualizar en tiempo real la temperatura y humedad ambiental que se encuentra en el cultivo.



Fig. 48 Cantidad de luz solar.



Fig. 49 Sensor de temperatura.

4.2.5. Resultados de los parámetros finales visualizados en la pantalla del receptor

Entre las funciones que posee el módulo receptor es la de visualizar en una sola pantalla todos los datos que han sido obtenidos en tiempo real por el módulo emisor mediante cada uno de sus sensores.



Fig. 50 Pantalla en tiempo real del módulo receptor.

4.3. Comparación de datos

Por medio de una aplicación de tiempo incorporada en los celulares con sistema Android (Celulares Xiaomi) se pudo comparar los datos obtenidos por el prototipo de estación meteorológica, los mismos que fueron comparados en tiempo real y como resultado se obtuvo que los valores tanto de la estación meteorológica y la aplicación son muy similares.



Fig. 51 Aplicación de tiempo incorporada en los celulares.

CONCLUSIONES

Se efectuó el diseño e implementación de un prototipo de estación meteorológica para monitorear varios parámetros del ambiente con respecto al cultivo de pasto para ganado utilizando sistemas de bajo costo que sean autosustentables.

Se llevó en efecto la implementación del sistema autosostenible en el prototipo mediante paneles solares, la cual permite energizar todo el prototipo mediante energía solar.

Se realizó el respectivo diseño de la tarjeta, impresión de la estructura de protección para los múltiples sensores que irán dentro del receptor, elaboración del prototipo en un pedestal para su factible movilización, armado configuración de sensores que van conectados a la tarjeta.

Se ejecutó la organización del hardware utilizando tarjeta PCB para conectar todos los sensores del módulo de medición para la obtención de parámetros meteorológicos tales como: presión atmosférica, cantidad de luz solar, humedad relativa, temperatura, humedad del suelo.

Todo el prototipo será entregado al dueño del rancho para su comodidad al momento de la siembra del cultivo de pasto para el ganado.

RECOMENDACIONES

Para realizar un mallado que abarque más extensiones de terreno se debe disponer de más materiales tales como: antenas RFID de largo alcance conectados en un mallado mesh, estructura de protección más robusta ya que debe tolerar humedad excesiva, lluvias y ráfagas violentas de viento en ambientes agrícolas.

Para los sensores meteorológicos se aconseja instalar una estructura diferente donde se sitúe la caja principal, ya que en una lluvia puede afectarse la electrónica diseñada, también situar un cimiento fijo y que esté sujeta al suelo debido a las ráfagas violentas de viento que pueden afectar la estabilidad del prototipo meteorológico.

Como incremento para posteriores investigaciones se pueden emplear sensores de mayor precisión para obtener mayor fiabilidad de los resultados en el transcurso del tiempo.

Como recomendación se debería probar el prototipo por los menos meses antes para obtener un histórico de datos ya que en época de lluvias podrían alterar la medición de los sensores para una correcta data que aporte al estudio de este tipo de desarrollos tecnológicos.

Como última recomendación se debe efectuar las respectivas instrucciones al dueño del cultivo para interpretar y comprender la información que se visualizará en la pantalla del receptor para futuras soluciones en caso de haberlas.

BIBLIOGRAFIA

- [1] Google, «Google Maps,» [En línea]. Available: <https://www.google.com/maps/@-2.21822,-79.6343875,186m/data=!3m1!1e3>. [Último acceso: 10 12 2021].
- [2] E. S. I. Palaguachi, «Diseño, desarrollo e implementación de una estación meteorológica basada en una red jerárquica de sensores, software libre y sistemas embebidos para la Empresa ELECAUSTRO en la Minicentral Gualaceo utilizando comunicación MQTT y MODBUS,» 2021. [En línea]. Available: <https://dspace.ups.edu.ec/handle/123456789/16056>. [Último acceso: 17 12 2021].
- [3] SensorGO, «SensorGO,» 2021. [En línea]. Available: <https://sensorgo.mx/estaciones-meteorologicas/>. [Último acceso: 16 12 2021].
- [4] sitrack, «Qué es la telemetría y sus aplicaciones,» [En línea]. Available: <http://landing.sitrack.com/telemetr%C3%ADa-y-sus-aplicaciones>.
- [5] J. G. G. González, «Diseño e implementación de un proyecto telemétrico autosustentable para supervisión de las condiciones hídricas y eficiencia de riego en plantaciones agrícolas,» 2016. [En línea]. Available: <http://dspace.ups.edu.ec/handle/123456789/13464>. [Último acceso: 5 12 2021].
- [6] V. G. Cerna González y J. S. Vargas Sellan, «Diseño e implementación de un sistema inalámbrico de llamado a enfermería a través de una red zigbee con notificación de mensajes de texto utilizando módulo gsm sim900.,» 2019. [En línea]. Available: <http://dspace.ups.edu.ec/handle/123456789/17401>. [Último acceso: 5 12 2021].
- [7] L. Electronics, «Labcenter,» 2021. [En línea]. Available: <https://www.labcenter.com/downloads/>. [Último acceso: 12 11 2021].
- [8] A. Industries, «Adafruit,» [En línea]. Available: <https://www.adafruit.com/product/1651>. [Último acceso: 01 12 2021].
- [9] Adafruit, «Adafruit Industries,» [En línea]. Available: <https://www.adafruit.com/product/385>. [Último acceso: 25 11 2021].
- [10] A. Industries, «Adafruit,» [En línea]. Available: <https://www.adafruit.com/product/1603>. [Último acceso: 30 11 2021].
- [11] N. J. Reyesn Guerrero, «Diseño de un sistema regenerativo de carga de baterías con paneles solares para un vehículo eléctrico Kia Soul,» Cuenca, 2018.
- [12] Arduino, «Arduino,» [En línea]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Último acceso: 01 11 2021].
- [13] Arduino, «Software de Arduino,» [En línea]. Available: <https://arduino.cl/programacion/>. [Último acceso: 14 12 2021].

- [14] V. Antonio, «Proyecto arduino,» [En línea]. Available: <https://proyectoarduino.com/arduino-mega-2560/>. [Último acceso: 12 12 2021].
- [15] «Naylamp Mechatronics SAC,» 2021. [En línea]. Available: <https://naylampmechatronics.com/sensores-temperatura-y-humedad/47-sensor-de-humedad-de-suelo-fc-28.html>. [Último acceso: 05 01 2022].
- [16] « Naylamp Mechatronics SAC,» 2021. [En línea]. Available: <https://naylampmechatronics.com/conversores-dc-dc/196-convertidor-voltaje-dc-dc-step-down-3a-lm2596.html>. [Último acceso: 12 12 2021].
- [17] «Enercity S.A.,» 2019. [En línea]. Available: <https://enercitysa.com/blog/regulador-de-carga-solar-2/>. [Último acceso: 13 11 2021].
- [18] «Amazon,» 2020. [En línea]. Available: <https://www.amazon.com/-/es/20-AMP-Regulador-Protecci%C3%B3n-bater%C3%ADa-cmp12-bsv20/dp/B00LB0WU2E>. [Último acceso: 12 10 2021].
- [19] B. B., «BOTLAND.STORE,» 2012. [En línea]. Available: <https://botland.store/weather-sensors/1732-rain-sensor-module-yl-83.html>. [Último acceso: 07 01 2022].
- [20] «Tecnopura S.A.,» [En línea]. Available: <https://www.tecnopura.com/producto/modulo-transceptor-rf-nrf24l01-pa-lna-de-2-4-ghz-con-antena-sma/>. [Último acceso: 12 9 2021].
- [21] NaylampMechatronics, «NaylampMechatronics,» 2021. [En línea]. Available: <https://naylampmechatronics.com/inalambrico/113-modulo-rf-nrf24l01-pa-lna-antena.html>. [Último acceso: 10 01 2022].
- [22] «TDK INVENSENSE,» 2003. [En línea]. Available: <https://invensense.tdk.com/products/motion-tracking/6-axis/>. [Último acceso: 12 12 2021].
- [23] «Luis Llamas,» 2010. [En línea]. Available: <https://www.luisllamas.es/usar-arduino-con-los-imu-de-9dof-mpu-9150-y-mpu-9250/>. [Último acceso: 24 11 2021].
- [24] «Mobus,» 2012. [En línea]. Available: <https://mobus.es/blog/que-es-una-bateria-lipo/> . [Último acceso: 22 01 2022].
- [25] J. M. S. Jose Saldarrega Andrade, de *Diseño e implementación de dos robots de batalla de 3 libras para el Club de Robótica de la Universidad Politécnica Salesiana*, Guayaquil, 2020, p. 14.
- [26] «Prometec.net,» 2015. [En línea]. Available: <https://www.prometec.net/elegir-bateria-lipo/>. [Último acceso: 26 01 2022].
- [27] «hola,» hola , [En línea]. Available: dsd. [Último acceso: dsds].

ANEXOS

Anexo 1: Código de programación utilizado

```
#include <RF24.h>
#include <RF24Network.h>
#include <SPI.h> //librerias necesarias(SPI y RF24Network).
#include <DHT.h>
#include <DHT_U.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085_U.h>
#include <Adafruit_MPU6050.h>
RF24 radio(7, 8); // pines del nRF24L01 (CE,CSN).
RF24Network network(radio);
const uint16_t this_node = 01; // direccion de este nodo, en forma octal( 04,031, etc).
const uint16_t node01 = 00; //definir nodo al que enviara informacion.
float X0=0,Y0=0,Z0=0;
int alarma=0;
//----Sensor DHT11(Temperatura y humedad)----//
int Sensor_DHT = 2;
float temp;float humed;
DHT dht(Sensor_DHT,DHT11);
//----Sensor HL-83(lluvia o humedad de suelo)----//
int lluvia = A0;
//---Sensor BMP085(Presion atmosferica)---//
Adafruit_BMP085_Unified bmp = Adafruit_BMP085_Unified(10085);
//----Sensor MPU6050(girosopio/acelerometro)----//
Adafruit_MPU6050 mpu;
Adafruit_Sensor *mpu_temp, *mpu_accel, *mpu_gyro;
//Nivel de radiacion solar
float Voltaje;
float dato[13];
void detalles_del_sensor()
{
    sensor_t sensor;
```

```

    bmp.getSensor(&sensor);
    Serial.println("-----");
    Serial.print ("Sensor:  "); Serial.println(sensor.name);
    Serial.print ("Driver Ver:  "); Serial.println(sensor.version);
    Serial.print ("ID unica:  "); Serial.println(sensor.sensor_id);
    Serial.print ("Maximo valor:  "); Serial.print(sensor.max_value); Serial.println(" hPa");
    Serial.print ("Minimo Valor:  "); Serial.print(sensor.min_value); Serial.println(" hPa");
    Serial.print ("margen de error:  "); Serial.print(sensor.resolution); Serial.println("
hPa");
    Serial.println("-----");
    Serial.println("");
    delay(500);
}

void setup() {
  Serial.begin(115200);
  Serial.print("inicial");
  SPI.begin();//inicializar SPI y el radio.
  radio.begin();
  network.begin(90, this_node); //(canal, direccion de este nodo).
  dht.begin();//pin2
  radio.setDataRate(RF24_2MBPS);
  radio.setPALevel(RF24_PA_MAX);
  pinMode(lluvia,INPUT);//A0
  pinMode(A1,INPUT);//PANEL
  pinMode(A2,INPUT);//Sensor de Humedad del suelo
  pinMode(9,OUTPUT);//RELE
  digitalWrite(9,HIGH);
  while (!Serial)
  delay(10);
  if (!mpu.begin()) {
    Serial.println("no se detecta ningun MPU6050");
    while (1){delay(10);}}
  Serial.println("MPU6050 encontrado");
  mpu_temp = mpu.getTemperatureSensor();
  mpu_temp->printSensorDetails();

```

```

mpu_accel = mpu.getAccelerometerSensor();
mpu_accel->printSensorDetails();

mpu_gyro = mpu.getGyroSensor();
mpu_gyro->printSensorDetails();

if(!bmp.begin())
{Serial.print("sensor bmp no detectado");
while(1);}detalles_del_sensor();
for (int i=0;i<=255;i++){
sensors_event_t accel;
sensors_event_t gyro;
sensors_event_t temper;
mpu_temp->getEvent(&temper);
mpu_accel->getEvent(&accel);
mpu_gyro->getEvent(&gyro);
X0=(gyro.gyro.x)*(57,295);
Y0=(gyro.gyro.y)*(57,295);
Z0=(gyro.gyro.z)*(57,295);
Serial.print("X0:");Serial.print(X0);
Serial.print(" Y0:");Serial.print(Y0);
Serial.print(" Z0:");Serial.println(Z0);
}

}

void loop() {
network.update();
RF24NetworkHeader header;
Voltaje=analogRead(A1);
int HSUELO=analogRead(A2);HSUELO=map(HSUELO,0,1023,100,0);
temp= dht.readTemperature();humed=dht.readHumidity();
int HumedadS=analogRead(lluvia);HumedadS=map(HumedadS,0,1023,100,0);
sensors_event_t event; bmp.getEvent(&event);
float temperature; bmp.getTemperature(&temperature);
float seaLevelPressure = 1016.00;

```

```

sensors_event_t accel;
sensors_event_t gyro;
sensors_event_t temper;
mpu_temp->getEvent(&temper);
mpu_accel->getEvent(&accel);
mpu_gyro->getEvent(&gyro);
float X=(gyro.gyro.x)*(57,295);
float Y=(gyro.gyro.y)*(57,295);
float Z=(gyro.gyro.z)*(57,295);

RF24NetworkHeader header2(node01);
dato[0]=temp;
dato[1]=humed;
dato[2]=HumedadS;
dato[3]=event.pressure;
dato[4]=temperature;
dato[5]=(bmp.pressureToAltitude(seaLevelPressure,event.pressure));
dato[6]=temper.temperature;
dato[7]=X;
dato[8]=Y;
dato[9]=Z;
dato[10]=Voltaje;
dato[11]=HSUELO;
dato[12]=alarma;

network.write(header2, &dato, sizeof(dato));
Serial.println("-----");
Serial.println("");
Serial.print("TEMP_E:");Serial.print(temp);
Serial.print(" HUM:");Serial.print(humed);
Serial.print(" LLUV:");Serial.println(HumedadS);
if (event.pressure){
Serial.print("TEMP_I:");Serial.print(temperature);
Serial.print(" PRESS:");Serial.print(event.pressure);
Serial.print(" ALT:");
Serial.println(dato[5]);}
Serial.print("TEMP_R:");Serial.print(temper.temperature);

```

```

Serial.print(" HUM_SU:");Serial.println(HSUELO);

Serial.print("X0:");Serial.print(X0);
Serial.print(" Y0:");Serial.print(Y0);
Serial.print(" Z0:");Serial.println(Z0);

Serial.print("GX:");Serial.print(X);
Serial.print(" GY:");Serial.print(Y);
Serial.print(" GZ:");Serial.println(Z);
Serial.print("PANEL:");Serial.println(Voltaje);
Serial.print("ALARMA:");Serial.println(alarma);
delay(100);
if ((X0+10)<X||(X0-10)>X){digitalWrite(9,LOW);alarma=1;}
if ((Y0+10)<Y||(Y0-10)>Y){digitalWrite(9,LOW);alarma=1;}
if ((Z0+10)<Z||(Z0-10)>Z){digitalWrite(9,LOW);alarma=1;}
}}

```

Codigo utilizado por módulo receptor

```

#include <Adafruit_TFTLCD.h>
#include <pin_magic_MEGA.h>
#include <registers.h>
#include <Adafruit_GFX.h>
#include <TouchScreen.h>
#include <RF24Network.h>
#include <RF24.h>
#include <SPI.h>
#include <SD.h>
int BO1,BO2;
RF24 radio(49, 48);          // nRF24L01 (CE,CSN)
RF24Network network(radio); // Include the radio in the network
const uint16_t this_node = 00; // Address of this node in Octal format ( 04,031, etc)
const uint16_t node01 = 01;   // Address of the other node in Octal format
float datos[13];

// Pines necesarios para los 4 pines del panel tactil
#define YP A3 // Pin analogico A1 para ADC
#define XM A2 // Pin analogico A2 para ADC

```

```

#define YM 9
#define XP 8

// Definimos la presion mxima y minima que podemos realizar sobre el panel
#define MINPRESSURE 1
#define MAXPRESSURE 1000
#define DEBUG

// Para mejor precision de la presion realizada, es necesario
// medir la resistencia entre los pines X+ y X-.
// En Shield TFT 2.4" LCD se mide entre los pines A2 y 6
// Instancia del panel tactil (Pin XP, YP, XM, YM, Resistencia del panel)
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 340);

short TS_MINX = 850; // Coordenadas del panel tactil para delimitar
short TS_MAXX = 110; // y que coincida con el tama±o del LCD
short TS_MINY = 888; // el tama±o de la zona donde podemos presionar
short TS_MAXY = 130;

#define LCD_CS A3 // Definimos los pines del LCD
#define LCD_CD A2 // para poder visualizar elementos graficos
#define LCD_WR A1
#define LCD_RD A0
#define LCD_RESET A4

#if defined __AVR_ATmega2560__ // Para Arduino Uno/Duemilanove, conectamos la
tarjeta SD en los pines del puerto SPI
#define SD_SCK 13 // que se corresponden con los pines MOSI -> 11, MISO ->
12 y SCK -> 13
#define SD_MISO 12
#define SD_MOSI 11
#endif

// Chip Select del bus SPI correspondiente a la conexi3n con la tarjeta SD
#define SD_CS 10

```

```

#define    BLACK  0x0000 // Definimos los colores
#define    BLUE   0x7BCF // que utilizaremos para
#define    RED    0x9800 // el texto y los elementos graficos
#define    GREEN  0x7BCF
#define    CYAN   0x7BCF
#define    MAGENTA 0x7BCF
#define    YELLOW 0x7BCF
#define    WHITE  0xD69A
#define    LIGHTGRAY 0x7BCF
#define    DELFIN 0x7BCF
//0xCC44

```

```

Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET); //

```

Instancia LCD

```

int X; // Variables que almacenaran la coordenada
int Y; // X, Y donde presionemos y la variable Z
int Z; // almacenara la presion realizada
int a = 0; // variable "flag" para control rebotes

void setup(void)
{
  Serial.begin(115200);
  SPI.begin();
  radio.begin();
  network.begin(90, this_node);//(canal,nodo)
  radio.setDataRate(RF24_2MBPS);
  tft.begin(0x9341); // Iniciamos el LCD especificando el controlador ILI9341.
  tft.setRotation(3);
  tft.fillScreen(BLACK);
  pinMode(44,INPUT);
  pinMode(45,INPUT);
  pinMode(13, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(A8,OUTPUT);digitalWrite(A8,HIGH);
  pinMode(21,OUTPUT);digitalWrite(21,LOW);delay(100);digitalWrite(21,HIGH);

```

```

#if defined __AVR_ATmega2560__
if (!SD.begin(SD_CS, SD_MOSI, SD_MISO, SD_SCK ))
{
  tft.println(F("failed!"));    // Definimos los pines de la tarjeta SD para MEGA
  return;
}
#else
if (!SD.begin(SD_CS)) {
  tft.println(F("failed!"));
  return;
}
#endif
//IMG();
MenuInicio();
}

void MenuInicio(){
//320,260

  tft.fillRect(0,0, 320, 260, BLACK);
  tft.drawRect(2,0, 100, 40, GREEN);
  tft.setCursor(23,13.5);
  tft.setTextSize(2);
  tft.setTextColor(GREEN);
  tft.println("DHT11");

  tft.drawRect(2, 50, 100, 40, GREEN);
  tft.setCursor(20,63.5);
  tft.setTextSize(2);
  tft.setTextColor(GREEN);
  tft.println("HL 83");

  tft.drawRect(2,100, 100, 40, GREEN);
  tft.setCursor(18,113.5);
  tft.setTextSize(2);
  tft.setTextColor(GREEN);
  tft.println("BMP085");

```

```
tft.drawRect(2,150, 100, 40, GREEN);
tft.setCursor(10,163.5);
tft.setTextSize(2);
tft.setTextColor(GREEN);
tft.println("MPU6050");
```

```
tft.drawRect(2,200, 100, 40, GREEN);
tft.setCursor(20,213.5);
tft.setTextSize(2);
tft.setTextColor(GREEN);
tft.println("PANEL");
```

```
tft.drawRect(112,200, 100, 40, GREEN);
tft.setCursor(130,213.5);
tft.setTextSize(2);
tft.setTextColor(GREEN);
tft.println("SUELO");
```

```
tft.drawRect(220,200, 100, 40, RED);
tft.setCursor(240,213.5);
tft.setTextSize(2);
tft.setTextColor(RED);
tft.println("TODOS");
```

```
tft.setTextColor(BLACK);
tft.fillRoundRect(110,5,200,100,20,GREEN);
tft.fillRect(223,15,10,2,BLACK);
tft.setCursor(175,15); tft.setTextSize(2); tft.println("Diseno");
tft.setCursor(125,32);tft.println("y Construccion");
tft.setCursor(125,49);tft.println("de un Satelite");
tft.setCursor(165,66);tft.println("en lata");
tft.setCursor(150,83);tft.println("--CanSat--");tft.setTextColor(GREEN);
tft.setCursor(135,110);tft.setTextSize(1);tft.println("-Ing. Monica Miranda");
tft.setCursor(135,120);tft.setTextSize(1);tft.println("-Madelaine Lopez");
loop();
```

```

}
void loop()
{
  A:

  network.update();

  RF24NetworkHeader header;
  network.read(header, &datos, sizeof(datos));
  Serial.println(datos[12]);
  if(datos[12]>=1){digitalWrite(21,LOW);}else{digitalWrite(21,HIGH);}
  lecturaPanel();//detectar presion y coordenadas

  if((X > 0 && X < 20) && (Y > 190 && Y < 260) && (Z > MINPRESSURE && Z <
MAXPRESSURE))
  {

  tft.fillRect(2,0, 100, 40, GREEN);
  tft.setCursor(23,13.5);
  tft.setTextSize(2);
  tft.setTextColor(BLACK);
  tft.println("DHT11");
  delay(500);
  tft.fillRect(3, 1, 98, 38, BLACK);
  tft.setCursor(23,13.5);
  tft.setTextSize(2);
  tft.setTextColor(GREEN);
  tft.println("DHT11");
  DHT11MENU();

  }

  if((X > 55 && X < 95) && (Y > 190 && Y < 260) && (Z > MINPRESSURE &&
Z < MAXPRESSURE))
  {
  tft.fillRect(2, 50, 100, 40, GREEN); // Dibujamos un "boton"
  tft.setCursor(20,63.5); // Colocamos el cursor
  tft.setTextSize(2); // Especificamos el tamaño del texto

```

```

tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("HL 83"); // Escribimos por pantalla
delay(500);
tft.fillRect(3, 51, 98, 38, BLACK);
tft.setCursor(20,63.5); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(GREEN); // Definimos el color del texto
tft.println("HL 83"); // Escribimos por pantalla
    HL83MENU();
}

if((X > 125 && X < 167) && (Y > 190 && Y < 260) && (Z > MINPRESSURE &&
Z < MAXPRESSURE))
{
    tft.fillRect(2,100, 100, 40, GREEN);
    tft.setCursor(18,113.5); // Colocamos el cursor
    tft.setTextSize(2);
    tft.setTextColor(BLACK);
    tft.println("BMP085");
    delay(500); // Ponemos la variable flag en 1
    tft.fillRect(3, 101, 98, 38, BLACK);
    tft.setCursor(18,113.5); // Colocamos el cursor
    tft.setTextSize(2); // Especificamos el tamaño del texto
    tft.setTextColor(GREEN); // Definimos el color del texto
    tft.println("BMP085");
    BMP085MENU();
}

if((X > 205 && X < 240) && (Y > 190 && Y < 260) && (Z > MINPRESSURE &&
Z < MAXPRESSURE))
{

tft.fillRect(2,150, 100, 40, GREEN); // Dibujamos un "boton"
tft.setCursor(10,163.5); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("MPU6050"); // Escribimos por pantalla
    delay(500); // Ponemos la variable flag en 1

```

```

tft.fillRect(3, 151, 98, 38, BLACK);
tft.setCursor(10,163.5); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(GREEN); // Definimos el color del texto
tft.println("MPU6050");
MPU6050MENU();
}
if((X > 280 && X < 320) && (Y > 190 && Y < 260) && (Z > MINPRESSURE &&
Z < MAXPRESSURE))
{
tft.fillRect(2,200, 100, 40, GREEN); // Dibujamos un "boton"
tft.setCursor(20,213.5); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("PANEL"); // Escribimos por pantalla
delay(500);

tft.fillRect(3, 201, 98, 38, BLACK);
tft.setCursor(20,213.5); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(GREEN); // Definimos el color del texto
tft.println("PANEL");
PANELMENU();
}

if((X > 280 && X < 320) && (Y > 98 && Y < 163) && (Z > MINPRESSURE && Z
< MAXPRESSURE))
{
//tft.drawRect(112,200, 100, 40, GREEN);
//tft.setCursor(130,213.5);

tft.fillRect(112,200, 100, 40, GREEN); // Dibujamos un "boton"
tft.setCursor(130,213.5); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("SUELO"); // Escribimos por pantalla
delay(500);

```

```

tft.fillRect(113, 201, 98, 38, BLACK);
tft.setCursor(130,213.5); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(GREEN); // Definimos el color del texto
tft.println("SUELO");
SUELOMENU();
}

if((X > 280 && X < 320) && (Y > 0 && Y < 90) && (Z > MINPRESSURE && Z <
MAXPRESSURE))
{
tft.fillRect(220,200, 100, 40, RED);
tft.setCursor(240,213.5); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("TODOS"); // Escribimos por pantalla
delay(500);

tft.fillRect(221, 201, 98, 38, BLACK);
tft.setCursor(240,213.5); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(RED); // Definimos el color del texto
tft.println("TODOS");
TODOS();
}
goto A;
}
void DHT11MENU(){
tft.fillRect(0,0, 320, 260, BLACK);
tft.fillRect(0,0,320,80, GREEN);
tft.drawRect(0,0,320,240, GREEN);
tft.setCursor(80,20);tft.setTextSize(5);tft.setTextColor(BLACK);
tft.println("DHT11");
tft.drawRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(RED);
tft.println("<-Atras");

```

```

a:lecturaPanel();
network.update();
while ( network.available() ) {
    lecturaPanel();
    RF24NetworkHeader header;
    network.read(header, &datos, sizeof(datos));
    tft.setCursor(5,90);tft.setTextSize(2);tft.setTextColor(GREEN);
    tft.println("TEMPERATURA EXTERNA");
    tft.setCursor(5,110);tft.setTextColor(WHITE);tft.print(datos[0]);tft.println("C");
    tft.setCursor(5,135);tft.setTextColor(GREEN);
    tft.println("HUMEDAD");
    tft.setCursor(5,155);tft.setTextColor(WHITE);tft.print(datos[1]);tft.println("%");
    tft.fillRect(100,110,200,20,LIGHTGRAY);tft.fillRect(102,112,datos[1],16,RED);
    if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);
    tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
    tft.println("<-Atras");delay(200);MenuInicio();
    }
    delay(1000);tft.fillRect(5,110,90,20,BLACK);tft.fillRect(5,150,120,20,BLACK);
}lecturaPanel();
if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);
    tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
    tft.println("<-Atras");delay(200);MenuInicio();
    }
goto a;}
void HL83MENU(){
tft.fillRect(0,0, 320, 260, BLACK);
tft.fillRect(0,0,320,80,BLUE);
tft.drawRect(0,0,320,240,BLUE);
tft.setCursor(80,20);tft.setTextSize(5);tft.setTextColor(BLACK);
tft.println("HL83");
tft.drawRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(RED);
tft.println("<-Atras");
a:
lecturaPanel();

```

```

network.update();
while ( network.available() ) {lecturaPanel();
RF24NetworkHeader header;
network.read(header, &datos, sizeof(datos));
int HS=datos[2];
tft.setCursor(5,90);tft.setTextSize(2);tft.setTextColor(BLUE);
tft.println("PORCENTAJE DE LLUVIA:");
tft.setCursor(5,110);tft.setTextSize(3);
tft.setTextColor(WHITE);tft.print(HS);tft.println("%");
if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
tft.println("<-Atras");delay(200);MenuInicio();
}
delay(1000);tft.fillRect(5,110,200,30,BLACK);
}lecturaPanel();
if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
tft.println("<-Atras");delay(200);MenuInicio();
}
goto a;}
void BMP085MENU(){
tft.fillRect(0,0, 320, 260, BLACK);
tft.fillRect(0,0,320,80,LIGHTGRAY);
tft.drawRect(0,0,320,240,LIGHTGRAY);
tft.setCursor(80,20);tft.setTextSize(5);tft.setTextColor(BLACK);
tft.println("BMP085");
tft.drawRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(RED);
tft.println("<-Atras");
a:
lecturaPanel();
network.update();
while ( network.available() ) {lecturaPanel();
RF24NetworkHeader header;
network.read(header, &datos, sizeof(datos));

```

```

tft.setCursor(5,90);tft.setTextSize(2);tft.setTextColor(LIGHTGRAY);
tft.println("PRESION:");
tft.setCursor(5,110);tft.setTextColor(WHITE);tft.print(datos[3]);tft.println("mbar");
tft.setCursor(5,130);tft.setTextColor(LIGHTGRAY);
tft.println("TEMPERATURA INTERNA:");
tft.setCursor(5,150);tft.setTextColor(WHITE);tft.print(datos[4]);tft.println("C");
tft.setCursor(5,170);tft.setTextColor(LIGHTGRAY);
tft.println("ALTITUD:");
tft.setCursor(5,190);tft.setTextColor(WHITE);tft.print(datos[5]);tft.println("m");
if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
tft.println("<-Atras");delay(200);MenuInicio();
}
delay(1000);tft.fillRect(5,110,200,20,BLACK);tft.fillRect(5,150,200,20,BLACK);
tft.fillRect(5,190,200,20,BLACK);

}lecturaPanel();
if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
tft.println("<-Atras");delay(200);MenuInicio();
}
goto a;}
void MPU6050MENU(){
tft.fillRect(0,0, 320, 260, BLACK);
tft.fillRect(0,0,320,80,DELFIN);
tft.drawRect(0,0,320,240,DELFIN);
tft.setCursor(80,20);tft.setTextSize(5);tft.setTextColor(BLACK);
tft.println("BMP085");
tft.drawRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(RED);
tft.println("<-Atras");
a:
lecturaPanel();
network.update();
while ( network.available() ) {lecturaPanel();

```

```

RF24NetworkHeader header;
network.read(header, &datos, sizeof(datos));
tft.setCursor(5,90);tft.setTextSize(2);tft.setTextColor(DELFIN);
tft.println("TEMPERATURA ACELEROMETRO:");
tft.setCursor(5,110);tft.setTextColor(WHITE);tft.print(datos[6]);tft.println("C");
tft.setCursor(5,170);tft.setTextColor(DELFIN);
tft.println("GIRO:");tft.drawLine(65,140, 65, 210, DELFIN);
tft.setCursor(70,140);tft.print("X:");tft.setTextColor(WHITE);tft.println(datos[7]);
tft.setCursor(70,170);tft.setTextColor(DELFIN);tft.print("Y:");tft.setTextColor(WHITE);
tft.println(datos[8]);
tft.setCursor(70,195);tft.setTextColor(DELFIN);tft.print("Z:");tft.setTextColor(WHITE);t
ft.println(datos[9]);
if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
tft.println("<-Atras");delay(200);MenuInicio();
}
delay(1000);tft.fillRect(5,110,200,20,BLACK);tft.fillRect(90,140,100,70,BLACK);

}lecturaPanel();
if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
tft.println("<-Atras");delay(200);MenuInicio();
}
goto a;}
void PANELMENU(){
tft.fillRect(0,0, 320, 260, BLACK);
tft.fillRect(0,0,320,80,YELLOW);
tft.drawRect(0,0,320,240,YELLOW);
tft.setCursor(20,20);tft.setTextSize(4);tft.setTextColor(BLACK);
tft.println("PANEL SOLAR");
tft.drawRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(RED);
tft.println("<-Atras");
a:
lecturaPanel();

```

```

network.update();
while ( network.available() ) {lecturaPanel();
  RF24NetworkHeader header;
  network.read(header, &datos, sizeof(datos));
  float LUZ= (datos[10])*97.75171065;
  tft.setCursor(5,90);tft.setTextSize(3);tft.setTextColor(YELLOW);
  tft.println("NIVEL DE LUZ:");
  tft.setCursor(5,120);tft.setTextSize(4);
  tft.setTextColor(WHITE);tft.print(LUZ);tft.println("LUX");
  /*if(LUZ>0 && LUZ<2)
  {tft.setCursor(195,90);tft.println("OSCURO");}
  if(LUZ>3 && LUZ<4)
  {tft.setCursor(195,90);tft.println("SOLEADO");}*/
  if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);
  tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
  tft.println("<-Atras");delay(200);MenuInicio();
  }
  delay(1000);tft.fillRect(5,120,240,30,BLACK);
}lecturaPanel();
  if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);
  tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
  tft.println("<-Atras");delay(200);MenuInicio();
  }
goto a;}

void SUELOMENU(){
tft.fillRect(0,0, 320, 260, BLACK);
tft.fillRect(0,0,320,80,BLUE);
tft.drawRect(0,0,320,240,BLUE);
tft.setCursor(80,20);tft.setTextSize(5);tft.setTextColor(BLACK);
tft.println("SUELO");
tft.drawRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(RED);
tft.println("<-Atras");
a:

```

```

lecturaPanel();
network.update();
while ( network.available() ) {lecturaPanel();
  RF24NetworkHeader header;
  network.read(header, &datos, sizeof(datos));
  int HSS=datos[11];
  tft.setCursor(5,90);tft.setTextSize(2);tft.setTextColor(BLUE);
  tft.println("HUMEDAD DEL SUELO:");
  tft.setCursor(5,110);tft.setTextSize(3);
  tft.setTextColor(WHITE);tft.print(HSS);tft.println("%");
  if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);
  tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
  tft.println("<-Atras");delay(200);MenuInicio();
  }
  delay(1000);tft.fillRect(5,110,200,30,BLACK);
}lecturaPanel();
if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);
  tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
  tft.println("<-Atras");delay(200);MenuInicio();
  }
goto a;}

void TODOS(){
tft.fillRect(0,0, 320, 260, BLACK);
tft.fillRect(0,0,320,20,RED);
tft.drawRect(0,0,320,240,RED);
tft.setCursor(20,5);tft.setTextSize(2);tft.setTextColor(BLACK);
tft.println("Sensores");
tft.drawRect(210,190,100,40,RED);
tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(RED);
tft.println("<-Atras");
a:
lecturaPanel();
network.update();
while ( network.available() ) {lecturaPanel();

```

```

RF24NetworkHeader header;
network.read(header, &datos, sizeof(datos));
int HS=datos[2];
float LUZ= (datos[10])*97.75171065;

tft.setCursor(80,25);tft.setTextSize(2);tft.setTextColor(RED);tft.println("TEMPERATURA:"
);
tft.setCursor(5,45);tft.setTextSize(1);tft.setTextColor(RED);tft.println("EXTERNA");
tft.setCursor(100,45);tft.setTextSize(1);tft.setTextColor(RED);tft.println("INTERNA");

tft.setCursor(195,45);tft.setTextSize(1);tft.setTextColor(RED);tft.println("ACELEROMETR
O");

tft.setCursor(5,55);tft.setTextSize(2);tft.setTextColor(WHITE);tft.print(datos[0]);tft.println("
C");

tft.setCursor(100,55);tft.setTextSize(2);tft.setTextColor(WHITE);tft.print(datos[4]);tft.printl
n("C");

tft.setCursor(195,55);tft.setTextSize(2);tft.setTextColor(WHITE);tft.print(datos[6]);tft.printl
n("C");

tft.setCursor(5,75);tft.setTextSize(2);tft.setTextColor(RED);tft.print("HUMEDAD:");
tft.setTextColor(WHITE);tft.print(datos[1]);tft.println("%");
tft.setCursor(5,95);tft.setTextSize(2);tft.setTextColor(RED);tft.print("PORCENTAJE
DE LLUVIA:");
tft.setTextColor(WHITE);tft.print(HS);tft.println("%");
tft.setCursor(5,115);tft.setTextSize(2);tft.setTextColor(RED);tft.print("PRESION:");
tft.setTextColor(WHITE);tft.print(datos[3]);tft.println("mbar");
tft.setCursor(5,135);tft.setTextSize(2);tft.setTextColor(RED);tft.print("ALTITUD:");
tft.setTextColor(WHITE);tft.print(datos[5]);tft.println("m");
tft.setCursor(5,195);tft.setTextSize(2);tft.setTextColor(RED);tft.print("GIRO:");

tft.setCursor(70,175);tft.setTextSize(2);tft.print("X:");tft.setTextColor(WHITE);tft.println(da
tos[7]);

tft.setCursor(70,195);tft.setTextSize(2);tft.setTextColor(RED);tft.print("Y:");tft.setTextColo
r(WHITE);tft.println(datos[8]);

```

```

tft.setCursor(70,215);tft.setTextSize(2);tft.setTextColor(RED);tft.print("Z:");tft.setTextColor
(WHITE);tft.println(datos[9]);

    tft.setCursor(5,155);tft.setTextSize(2);tft.setTextColor(RED);tft.print("NIVEL DE
LUZ:");tft.setTextColor(WHITE);tft.print(LUZ);tft.println("LUX");

    if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);

    tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
tft.println("<-Atras");delay(200);MenuInicio();
    }
delay(1000);
tft.fillRect(5,55,310,20,BLACK);
tft.fillRect(100,75,100,20,BLACK);
tft.fillRect(255,95,30,20,BLACK);
tft.fillRect(100,115,140,20,BLACK);
tft.fillRect(100,135,80,20,BLACK);
tft.fillRect(161,155,120,20,BLACK);
tft.fillRect(90,175,115,55,BLACK);
}lecturaPanel();

if(((X > 245 && X < 320) && (Y > 0 && Y < 100) && (Z > MINPRESSURE && Z <
MAXPRESSURE))||(digitalRead(45)==1)){tft.fillRect(210,190,100,40,RED);

tft.setCursor(216,204);tft.setTextSize(2);tft.setTextColor(BLACK);
tft.println("<-Atras");delay(200);MenuInicio();
    }
goto a;}

void lecturaPanel()
{
    digitalWrite(13, HIGH);
    TSPoint p = ts.getPoint(); // Realizamos lectura de las coordenadas
    digitalWrite(13, LOW);

    pinMode(XM, OUTPUT); // La librería utiliza estos pines como entrada y salida
    pinMode(YP, OUTPUT); // por lo que es necesario declararlos como salida justo
        // despues de realizar una lectura de coordenadas.

    // Mapeamos los valores analogicos leidos del panel tactil (0-1023)
    // y los convertimos en valor correspondiente a la medida del LCD 320x240

```

```
X = map(p.x, TS_MAXX, TS_MINX, tft.width(), 0);
Y = map(p.y, TS_MAXY, TS_MINY, tft.height(), 0);
Z = p.z;
Serial.print("X= ");Serial.print(X);
Serial.print(" Y= ");Serial.print(Y);
Serial.print(" Z= ");Serial.print(Z);
Serial.print(" A= ");Serial.print(a);
BO1=digitalRead(44);
BO2=digitalRead(45);
Serial.print(" BOTON(44)= ");Serial.print(BO1);
Serial.print(" BOTON(45)= ");Serial.println(BO2);
}
```

Anexo 2: Registro fotográfico



Fig. 52 Módulo emisor en plantación agrícola.

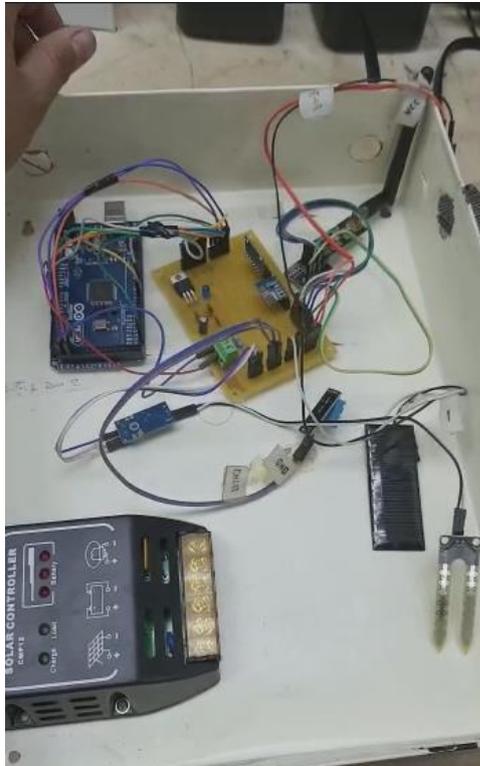


Fig. 53 Integración de partes y componentes a módulo emisor.

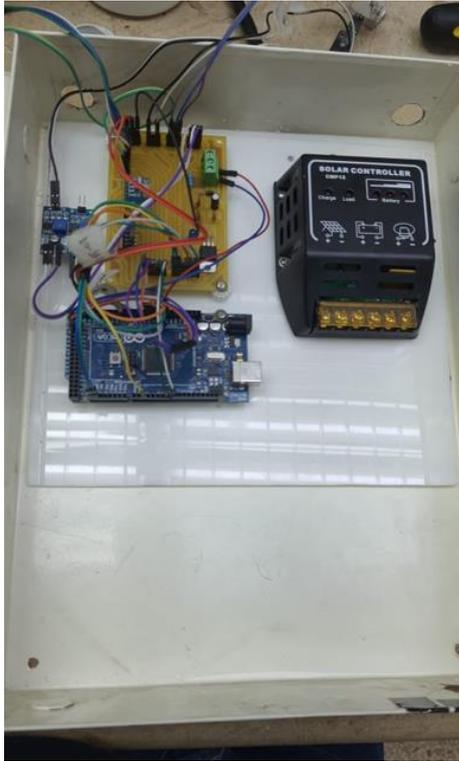


Fig. 54 Gestión de cables y posicionamiento de componentes internos módulo emisor.



Fig. 55 Ventilador de 12 voltios para extraer calor generado por el equipo.



Fig. 56 Reconocimiento de terreno