



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMÁTICO
DE CONTROL MICROCLIMATOLÓGICO PARA
INVERNADEROS UTILIZANDO MICROCONTROLADOR ARM
CORTEX

TESIS

PRESENTADA POR:

YHONATAN QUISPE CONDORI

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PUNO – PERÚ

2023



NOMBRE DEL TRABAJO

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMÁTICO DE CONTROL MICROCLIMATOLÓGICO PARA INVERNADEROS UTILIZANDO MICROCONTROLADOR ARM CORTEX

AUTOR

YHONATAN QUISPE CONDORI

RECuento DE PALABRAS

19268 Words

RECuento DE CARACTERES

114535 Characters

RECuento DE PÁGINAS

132 Pages

TAMAÑO DEL ARCHIVO

4.0MB

FECHA DE ENTREGA

Jun 4, 2023 8:02 PM GMT-5

FECHA DEL INFORME

Jun 4, 2023 8:04 PM GMT-5

● **15% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos

- 15% Base de datos de Internet
- 6% Base de datos de publicaciones
- Base de datos de Crossref
- Base de datos de contenido publicado de Crossref
- 8% Base de datos de trabajos entregados

● **Excluir del Reporte de Similitud**

- Material bibliográfico
- Material citado
- Material citado
- Coincidencia baja (menos de 12 palabras)


JAMES ROBERTO FERRONONDO MAMANI
C.T. 30113401
SUB DIRECCIÓN DE INVESTIGACIÓN
E.P. INGENIERÍA ELECTRÓNICA


Maximo A. Montalvo Atco
ING. ELECTRÓNICO
CIR 100188

Resumen



DEDICATORIA

Este trabajo de investigación le dedico principalmente, a mi padre y a mis hermanos, que siempre estuvieron presentes en los momentos más críticos y a todas las personas que estuvieron animándome con este trabajo de investigación.

Yhonatan. Q. C.



AGRADECIMIENTOS

Un agradecimiento especial a todos los que estuvieron apoyándome en toda mi formación profesional, a mis familiares, a mis amigos y docentes que compartieron todos sus conocimientos para mi formación profesional.

Yhonatan. Q. C.



ÍNDICE GENERAL

DEDICATORIA

AGRADECIMIENTOS

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ACRÓNIMOS

RESUMEN 16

ABSTRACT 17

CAPITULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA 19

1.1.1. Descripción de la situación problemática..... 19

1.2. FORMULACIÓN DEL PROBLEMA 20

1.2.1. Problema específico 21

1.3. JUSTIFICACIÓN DE PROBLEMA 21

1.4. OBJETIVO DE LA INVESTIGACIÓN..... 22

1.4.1. Objetivo General..... 22

1.4.2. Objetivo Especifico..... 22



CAPITULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE PROYECTO	23
2.2. SUSTENTO TEÓRICO	25
2.2.1. Invernaderos.....	25
2.2.1.1. Tipos de invernadero	26
2.2.1.1.1. Invernadero plano o tipo parral	26
2.2.1.1.2. Invernadero de capilla.....	27
2.2.1.1.3. Invernadero asimétrico o inacral	28
2.2.1.1.4. Invernadero en raspa y amagado	30
2.2.1.1.5. Invernadero tipo túnel o semicilindro.....	31
2.2.1.1.6. Invernadero de cristal o tipo venlo	33
2.2.2. Parámetros de control de un invernadero	33
2.2.2.1. Humidificación del suelo.....	33
2.2.2.2. Temperatura.....	34
2.2.2.3. Iluminación o luz solar	35
2.2.2.4. Potencial de hidrógeno del suelo	36
2.2.2.5. Conductividad del suelo	36
2.2.3. Microprocesadores ARM córtex.....	37
2.2.3.1. Tecnología thumb2	38
2.2.3.2. Microcontrolador ARM cortex-m	39
2.2.3.2.1. Clasificación de los ARM corte-m.....	40



2.2.3.3. FreeRTOS	42
2.2.3.4. Interface de comunicación serial	44
2.2.4. Modbus RS485	45
2.2.5. El modelo TCP/IP.....	46
2.2.6. Sensores	47
2.2.7. Lenguaje de programación	48
2.2.7.1. Assembler	48
2.2.7.2. Lenguaje de programación C/C++	50
2.2.8. Hipótesis	51
2.2.8.1. Hipótesis general.....	51
2.2.8.2. Hipótesis específica	51

CAPITULO III

MATERIALES Y MÉTODO

3.1. DISEÑO METODOLÓGICO DE LA INVESTIGACIÓN	52
3.1.1. Diseño de investigación	52
3.1.2. Nivel de la investigación.....	52
3.1.3. Tipo de investigación.....	52
3.1.4. Población y muestra de la investigación.....	53
3.1.4.1. Población	53
3.1.4.2. Muestra	53
3.1.5. Ubicación y descripción de la investigación	53
3.1.5.1. Ubicación	53



3.1.5.2. Descripción de la investigación	53
3.1.6. Técnicas e instrumentos de recolección de Datos	54
3.1.6.1. Instrumentos.....	54
3.1.7. Técnicas de procesamiento de datos.....	55
3.1.7.1. Diferencia de error	55
3.1.7.2. Media de error.....	55
3.1.7.3. Varianza	55
3.2. DESARROLLO DEL PROTOTIPO	56
3.2.1. Selección de los elementos de control y actuadores.....	56
3.2.1.1. Microcontrolador	56
3.2.1.2. Sensor de temperatura y humedad del aire	57
3.2.1.3. Sensor de humedad del suelo.....	59
3.2.1.4. Sensor de intensidad luminosa	60
3.2.1.5. Modulo para IoT	61
3.2.1.6. Bomba de agua.....	62
3.2.1.7. Ventilador	63
3.2.1.8. Calefactor.....	64
3.2.1.9. Nebulizador ultrasónico.....	65
3.2.2. Sistema de control.....	66
3.2.3. Estructura del invernadero	67
3.2.3.1. Instalación y ubicación de los sensores y actuadores en el prototipo.....	68



3.2.4. Sistema de control.....	74
3.2.4.1. Elección de estrategia de control	74
3.2.4.1.1. Fuente de alimentación AC-DC	75
3.2.4.1.2. Potencia.....	75
3.2.4.1.3. Circuito de protección.....	77
3.2.4.1.4. Stm32f103vc.....	78
3.2.5. Fundamentación para la programación por riego convencional	79
3.2.6. Resumen del procedimiento	80
3.2.7. Programación del microcontrolador	80
3.2.7.1. Código C/C++ sensor humedad y temperatura del aire	83
3.2.7.2. Código sensor humedad del suelo	84
3.2.7.3. Código sensor intensidad luminosa	87
3.2.7.4. Código para actuadores.....	90
3.2.7.5. Código IoT WEB.....	92
3.2.7.6. Control proporcional.....	95
3.2.8. Prueba de funcionamiento	96

CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1. CONTRASTACIÓN DE HIPÓTESIS.....	101
4.2. RESULTADOS	104
V. CONCLUSIONES.....	106



VI. RECOMENDACIONES	107
VII. REFERENCIAS BIBLIOGRÁFICAS.....	108
ANEXOS	114
Anexo 01.....	114
Anexo 02.....	120

Área : Microelectrónica

Tema : Electrónica Aplicada

Fecha de Sustentación: 14 de junio de 2023



ÍNDICE DE FIGURAS

Figura 1: Invernadero tipo parral.....	26
Figura 2: Invernadero tipo capilla.....	28
Figura 3: Invernadero tipo asimétrico	29
Figura 4: Perfil de un invernadero raspa	30
Figura 5: Invernadero tipo túnel5	32
Figura 6: Estructura del invernadero tipo venlo	33
Figura 7: Procesador A5	38
Figura 8: Arquitectura Thumb.....	39
Figura 9: Periféricos de un ARM Cortex-M7.....	42
Figura 10: Posibles estados en la que puede estar un subproceso.....	44
Figura 11: Distintos dispositivos conectados al bus RS485.....	46
Figura 12: Programación en assembler	50
Figura 13: Error máximo de humedad relativa.....	58
Figura 14: Configuración de pines AM2320.....	58
Figura 15: Sensor de Humedad de suelo	59
Figura 16: Respuesta espectral pico del LDR	61
Figura 17: Resistencia como función de la iluminación. $1Ftc = 10.76$ lumen	61
Figura 18: Modulo ESP8266	62
Figura 19: Turbina de bombeo o turbina de vértice	62
Figura 20: Estructura interna	63
Figura 21: Ventilador metálico.....	64
Figura 22: Nebulizador ultrasónico	65
Figura 23: Proceso de formación de niebla por oscilación.....	66
Figura 24: Diagrama en bloques del sistema a controlar	67



Figura 25: Aspecto y sensores del prototipo	68
Figura 26: Medidas en milímetros de la maqueta	68
Figura 27: Vista general	69
Figura 28: Uniones, completamente desarmables.....	69
Figura 29: Vista frontal con el tablero de control.....	70
Figura 30: Ubicación el sensor LDR	70
Figura 31: Ubicación sensor de humedad de suelo.	71
Figura 32: Ubicación del sensor de temperatura y humedad del aire.....	71
Figura 33: Ubicación del calefactor y humidificador ultrasónico.	72
Figura 34: Ubicación del ventilador.	72
Figura 35: Tablero de control con el STM32f103vc.	73
Figura 36: Diagrama en bloques del sistema a controlar	74
Figura 37: Circuito de potencia del electrobomba	76
Figura 38: Circuito de control para la señal PWM.....	76
Figura 39: Circuito de potencia del calefactor	77
Figura 40: Circuito para el nebulizador	77
Figura 41: Configuración de pines y esquemático	78
Figura 42: Esquemático stm32f103vc	79
Figura 43: Ruteo de conexiones Stm32f103vc.....	79
Figura 44: Entorno de programación STMicroelectronics	81
Figura 45: Entorno de configuración de periféricos del MCU	82
Figura 46: Configuración del reloj del sistema.	82
Figura 47: Habilitación y configuración del I2C.....	83
Figura 48: Funciones para el sensor AM2322.....	84
Figura 49: Configuración del ADC1	85



Figura 50: Configuración de la frecuencia del ADC1.....	86
Figura 51: Declaración de variables ad [2].....	87
Figura 52: Guardando el resultado en la variable adin_HS.....	87
Figura 53: Incluyendo librería math.h y declaración de variables.	88
Figura 54: Voltaje que transformado a digital por nuestro MCU	89
Figura 55: Ecuaciones del LDR en C/C++.....	90
Figura 56: Habilitación del TIM1 y configuración del canal 1	91
Figura 57: Configuración de entradas y salidas digitales.....	92
Figura 58: Distintas graficas de monitoreo.	93
Figura 59: Parte del Código en C/C++.....	94
Figura 60: Implementando control proporcional en C/C++	95
Figura 61: Valores de referencia de nuestro microclima.....	97
Figura 62: Datos recibidos de 30 muestras dentro de las 9:00 pm.....	97
Figura 63: Datos recibidos de 30 muestras dentro de las 10:00 pm.....	98
Figura 64: Datos recibidos de 30 muestras dentro de las 2:00 pm.....	99
Figura 65: Datos recibidos de 30 muestras dentro de las 3:00 pm.....	100
Figura 66: Datos recibidos de 30 muestras dentro de las 4:00 pm.....	100
Figura 67: Datos para análisis de contrastación.	102



ÍNDICE DE TABLAS

Tabla 1: Respuesta de adsorción según el pH	36
Tabla 2: Tabla de rendimiento de humedad relativa	58
Tabla 3: Clasificación de la luz solar en función de la longitud de onda	60
Tabla 4: Datos obtenidos en la madrugada.....	102



ÍNDICE DE ACRÓNIMOS

- ADC** : Conversión de digital a analogico
- ANSI** : Instituto Nacional Estadounidense de Estándares
- CAD** : Diseño asistido por computadora
- CE** : Conductividad electrica
- CPLD** : Dispositivo lógico programable complejo
- CPU** : Unidad central de procesamiento
- DMA** : Acceso a memoria directa
- HAL** :Capa de abstraccion de hardware
- I2C** : Circuito inter-integrado
- IoT** : Internet de la cosas
- LUX** : Intencidad de iluminación
- PLC** : Control logico programable
- PWM** : Modulación por ancho de pulso
- RFA** : Radiación fotosintéticamente activa
- RISC** : Grupo reducido de instrucciones para computadoras
- SDK** : Kit de Software de Desarrollo
- UART** : Universal asynchronous receiver / transmitter



RESUMEN

Un sistema de control automático Microclimático, es implementado, utilizando un microcontrolador de 32 bit y el otro con TCP/IP para poder integrar al sistema de información de “Internet de la Cosas” (IoT), las distintas variables que intervienen en un ecosistema, son enviados y representadas en graficas mediante una página Web. El modulo electrónico que se desarrollara contiene un microcontrolador ARM Cortex-M3 como una parte de sus componentes para el control, este envía información de las distintas variables que intervienen en un sistema Microclimático para invernaderos, a la página Web ThingSpeak de la página MathWorks la cual utiliza los protocolos MQTT Y REST y es configurado para mostrar ocho graficas que representan el estado del invernadero. Las distintas variables climatológicas cuyo propósito es mantener a las plantas con los recursos naturales necesarios para el crecimiento y desarrollo que este, debe de contar con la tecnología suficiente para alcanzar el balance correcto de sus parámetros para un determinado tipo de cultivo o planta. La ayuda de los distintos tipos de control, comunicación y sensores electrónicos nos permitieron mantener las variables climáticas en un determinado punto referencial. La lógica, el algoritmo y el desarrollo de la programación de control, mostraron resultados de correcciones rápidas existentes entre las variaciones climáticas del ambiente exterior con la del invernadero, principalmente en el transcurso de la noche, por las rápidas caídas de temperatura causadas por las heladas naturales andinas. La rápida respuesta correctiva controlada por el microcontrolador a las variaciones climáticas detectadas por los distintos sensores muestran una rápida compensación, por mantener en los niveles mínimos establecidos.

Palabras Claves: ARM CORTEX-M3, Invernadero, IoT, Microclimático, ThingSpeak.



ABSTRACT

An automatic Microclimatic control system is implemented, using a 32-bit microcontroller and the other with TCP/IP to be able to integrate the different variables that intervene in an ecosystem into the "Internet of Things" (IoT) information system. Are sent and represented in graphs through a Web page. The electronic module that will be developed contains an ARM Cortex-M3 microcontroller as part of its control components, it sends information on the different variables involved in a Microclimatic system for greenhouses, to the ThingSpeak Web page of the MathWorks page, which It uses the MQTT and REST protocols and is configured to show eight graphs that represent the state of the greenhouse. The different climatological variables whose purpose is to maintain the plants with the natural resources necessary for their growth and development, must have sufficient technology to achieve the correct balance of their parameters for a certain type of crop or plant. The help of the different types of control, communication and electronic sensors allowed us to maintain the climatic variables at a certain reference point. The logic, the algorithm and the development of the control programming, showed results of rapid corrections existing between the climatic variations of the external environment with that of the greenhouse, mainly during the night, due to the rapid drops in temperature caused by frost. Andean natural. The fast corrective responses controlled by the microcontroller to the climatic variations detected by the different sensors show a rapid compensation, by maintaining the minimum established levels.

Keywords: ARM CORTEX-M3, Greenhouse, IoT, Microclimatic, ThingSpeak



CAPITULO I

INTRODUCCIÓN

Una de las actividades más antiguas y vitales en la historia de la humanidad es la agricultura, que eliminó casi por completo a la necesidad de “migrar de un lugar a otro en busca de alimentos” (Gutiérrez Carbajal & Magaña Magaña, 2017), esto es gracias al conocimiento acumulado a lo largo de los años sobre el comportamiento y los recursos naturales que se necesitan para la siembra y su cultivo, de una cierta clase de planta. Para un determinado lugar con su clima natural determinado por las cuatro estaciones del año, se logra la producción de cultivos agrícolas que se puede cultivar. Con estos conocimientos esenciales dio la posibilidad de solucionar la necesidad vital de alimentarse para sobrevivir y la creación de grandes ciudades, dando así la posibilidad de dedicar más tiempo en otras actividades que luego se volverían esenciales para el desarrollo de la tecnología. Pero con el aumento de la población de una determinada ciudad, los recursos naturales no son suficientes para cubrir la siembra y el cultivo de alimentos, para poder solucionar de alguna forma este problema, se tiene que, al menos variar algunas “variables que intervengan para la mayor producción” (AgroSpray, 2021), que pueden ser: aumentar la producción por año, aumentar más área agrícola, o la transformación genética de plantas, cada una de las soluciones mencionadas anteriormente intervienen la manipulación de al menos una variable que intervienen en el crecimiento de la planta.

En este proyecto se implementa un sistema prototipo, que pueda aumentar la producción por año manipulando las variables climáticas que se producen en cada estación del año, estas variables son captadas por distintos sensores, como son: sensor de humedad del aire, de humedad del suelo, de temperatura, de CO₂, PH, y de conductividad suelo. Estas variables son “censadas por un microcontrolador” (Cuji Cáceres & Jara



Barreto, 2022) de 32 bits de la empresa STmicroelectronics, para luego accionar los distintos actuadores que compensaran la falta como; agregando o disminuyendo el recurso natural de alguna variable captada por los sensores, a su vez este sistema será integrada al ecosistema de telecomunicaciones a través de internet de la cosas (IoT). El microcontrolador encargado realizar el envío de la información por internet es el ESP8266 que fue creado por la empresa Tensilica este chip con CPU RISC de 32 bits cuenta con todo el recurso necesario para la comunicación TCP/IP a través de la red inalámbrica WIFI.

En el Capítulo I se presenta toda la generalidad del proyecto como: Ámbito de influencia de la tesis, identificación del problema, hipótesis, objetivos, justificación y la metodología de la investigación.

En el Capítulo II se presenta los conocimientos y las teorías necesarias para cumplir los objetivos planteados. Se enfocará principalmente en la teoría en aplicaciones con invernaderos y microcontroladores.

En el Capítulo III se presenta. Los materiales y equipos utilizados para desarrollar el prototipo, así como los lenguajes de programación utilizados y la lógica de funcionamiento de invernadero.

En el Capítulo IV se presenta los resultados de las comparaciones realizados por el diseño CAD y las muestras realizadas por el prototipo. Como también se verá en detalle realizado por el prototipo y el variable tiempo.

1.1. PLANTEAMIENTO DEL PROBLEMA

1.1.1. Descripción de la situación problemática

Los cambios climáticos, la contaminación ambiental, y el aumento de zonas urbanas han originado el desabastecimiento en forma considerable a la producción de



alimentos, principalmente la contaminación por residuos sólidos y del aire, sus efectos son; cambios drásticos de estaciones del año, temperaturas ambiental extremas nunca antes vistas en un lugar determinado, heladas, plagas y enfermedades en las plantas, la cual limitan el área para la siembra y cultivos de alimentos. Una de la muchas formas existente de crear más áreas de cultivo es, implementando invernaderos que tengan la capacidad de proteger de los distintos cambios bruscos climáticos en lo mayor posible que este pueda, también el invernadero proporciona un aislamiento de su interior al exterior de plagas y enfermedades, esta función del invernadero de aislar del medio ambiente da la posibilidad de crear un área Microclimático de cultivo, dando la posibilidad de cultivar diversas variedades de plantas, que no serían posibles con el ambiente climático natural de dicha zona.

El ciclo de vida de una determinada planta como son; germinación, crecimiento, reproducción o germinación y dispersión de las semillas ocurren en distintas estaciones del año, es decir que en cada etapa de vida de una planta requiere una cantidad acotado diferente de; nutrientes, agua, temperatura, humedad, luz, salinidad y acides del suelo, conociendo estos valores y controlándolos, se podrá realizar la fruticultura para luego obtener un rédito económico.

1.2. FORMULACIÓN DEL PROBLEMA

De las diversas variedades de plantas existentes, solo nos centraremos en una determinada planta de nuestro interés y de su botánica, extrayendo y recolectando toda la información climática que necesita para todas las etapas de su desarrollo. En tal sentido, nos permitimos formular el problema mediante la siguiente interrogante:



¿Cómo crear microclimas sustentables que puedan sujetar la vida de una determinada planta de nuestro interés, con la ayuda de un microcontrolador ARM CORTEX-M3?

1.2.1. Problema específico

- ¿Qué sensores climatológicos disponibles se utilizarán en el prototipo?
- ¿Cómo reunir todas las variables presentes en un invernadero, para luego enviarlo a la WEB?
- ¿Qué tipo de control automático necesitará en invernadero, para poder mantener un clima necesario para la planta?

1.3. JUSTIFICACIÓN DE PROBLEMA

El gran avance de la tecnología en microelectrónica está cada vez más desarrollándose en distintas áreas, en aplicaciones que requieren mayor integración de funciones como; controlabilidad, monitorización, recolección y actualización de datos. Muchos sistemas que incluyen sensores y actuadores son controlados con Microcontroladores que tienen muy pocos recursos de hardware, y frecuencia (Hz) de operación del CPU muy bajos dando como resultado una limitada respuesta en el tiempo de procesos automáticos o utilizan un controlador lógico programable (PLC) que normalmente requieren muchas licencias para liberar todas las funcionalidades del PLC, todo estos limitantes aumentan el presupuesto y dificultan para ser implementas en proyectos que tienen mucha réplicas del mismo sistema de control electrónico, este tipo de microcontroladores de 32 bits, que son la tecnología más utilizada en países con mayor avance tecnológico y que tienen más funciones especiales de comunicación que mejora la obtención de informaciones y actualización de datos en el mismo lugar del evento, y la facilidad de poder integrar a internet y así poder aprovechar el avance tecnológico de las



telecomunicaciones que mejorada en gran parte a la obtención de información y actualización de datos a largas distancias, por el medio de servidores Web, dando como resultado en la acción y decisión del cliente a una situación o acontecimiento particular.

Con este proyecto de tesis aplicado a invernaderos utilizando un microcontrolador de bajo coste y gran poder de procesamiento se estaría motivando a crear más áreas de cultivo de distintas plantas frutales, y generar un rédito económico adicional.

1.4. OBJETIVO DE LA INVESTIGACIÓN

1.4.1. Objetivo General

Diseñar e implementar un sistema automático de control Microclimático, utilizando los periféricos embebidos de un microcontrolador ARM CORTEX-M3

1.4.2. Objetivo Especifico

- Implementar un sistema automático de control Microclimático
- Utilizar los distintos periféricos embebidos de un microcontrolador ARM CORTEX-M3
- Generar un entorno Microclimático en el interior del invernadero



CAPITULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE PROYECTO

El desarrollo de invernaderos inteligentes está teniendo cada vez intereses en las zonas alto andinas como una nueva fuente económica para sus pobladores, en proyecto de “DISEÑO E IMPLEMENTACION DE UN SISTEMA AUTOMATIZADO PARA RIEGO TECNIFICADO BASADO EN EL BALANCE DE HUMEDAD DE SUELO CON TECNOLOGÍA ARDUINO EN EL LABORATORIO DE CONTROL Y AUTOMATIZACIÓN EPIME 2016” (Apaza Mamani & La torre Javier, 2016) implementa un sistema de control, utilizando una placa de desarrollo Arduino que cuenta con un microcontrolador Atmega328 de la empresa ATmel Cord, antes de ser adquirida por Microchip Tecnology Inc. Este microcontrolador de 8 bit de arquitectura RISC (Computación reducida del conjunto de instrucciones) que esta implementada con periféricos básicos con: entradas analógicas, PWM, protocolos de comunicación serial y una operación de frecuencia CPU de 20Mhz.

Otras de las investigaciones realizadas fue en la Universidad Nacional de Huancavelica con el título “SISTEMA DE CONTROL DE LA HUMEDAD RELATIVA PARA UN INVERNADERO, UTILIZANDO EL CONTROLADOR LÓGICO PROGRAMABLE” (Aliaga Mendoza & Quispe Bolaños, 2015) utiliza el PLC SIMATIC S7-200 de la marca Siemens para el cultivo de tomates. Este autómata cuenta con 14 entradas y 10 salidas, y hasta dos módulos de ampliación, para la programación se utilizó su Software STEP7-MICRO/WIN. El algoritmo de la respuesta de los sensores y actuadores son llevados al control PID, para luego ser cargados en la CPU del S7-200 y de esa manera poder cambiar de variables de salida deseada que controlan la activación y desactivación de las cargas encargadas de aumentar (humidificadores) o disminuir



(calefactor) la humedad según sea requerido, obteniendo resultado de regulación dentro del rango útil del 60 al 70 % de la humedad relativa del aire en el invernadero durante el cultivo del tomate.

Las condiciones climáticas necesarias para un cultivo, no siempre determinan la productividad eficiente del cultivo, en; “EVALUACIÓN DE TÉCNICAS HIDROPÓNICAS DE PRODUCCIÓN EN EL CULTIVO DE FRESA (FRAGARIA X ANANASSA) BAJO INVERNADERO” (Zaragoza Nieto, 2013), realiza cultivos sin suelos naturales, esta nueva técnica se utiliza en lugares donde hay muy poca precipitaciones pluviales o en zonas áridas, en donde el suelo es reemplazado por sustratos enriquecidos con minerales, agua, oxígeno y los nutrientes esenciales para su crecimiento. La principal razón de aplicar esta técnica HIDROPÓNICAS se debe a la necesidad de retener la humedad o inyectar la cantidad necesaria de agua, controlada por un sistema electrónico.

La cantidad de sensores en un invernadero y si existen varios invernaderos en diferentes lugares, se hace difícil monitorizar el correcto funcionamiento de sus parámetros Microclimático, en; “DISEÑO DE UNA SOLUCIÓN IOT PARA MONITOREO DE LA CONDICIONES ÓPTIMAS DE UN CULTIVO HIDROPÓNICO DE FLORES EN INVERNADERO” (BRICEÑO JEJÉN & CUBIDES PORRAS, 2020), presenta una solución empleando el embebido RaspberryPi, que tiene en su núcleo el Procesador ARM CORTEX A53, que se encarga de enviar todos los parámetro de estado de cada sensor presente del invernadero a un servidor con MySQL. En dicha investigación también se implementa una página Web que muestra todo el panorama del invernadero en tiempo real así como también la cantidad de ventiladores que se puede soportar el RaspberryPi.



2.2. SUSTENTO TEÓRICO

2.2.1. Invernaderos

Que se define como “recinto cerrado, cubierto y acondicionado para mantener una temperatura regular que proteja las planta de las inclemencias extremas propias del tiempo invernal, como frio intenso, heladas, viento, etc.” (Oxford University Press, 2022), las diferentes formas y materiales utilizadas son elegidos de acuerdo al lugar y a la resistencia de fuerzas naturales que estos deben de resistir, los materiales con mayor frecuencia en su construcción deben de soportar:

- Deben de tener la estructura con suficiente fuerza mecánica para resistir como; los ventarrones, al peso de la nieve y a otras fuerzas externas.
- La cubierta ha de ser resistente al deterioro por la luz natural, como a su vez debe de dejar pasar la mayor cantidad del espectro electromagnético que son esenciales para el crecimiento y desarrollo de la planta.
- la geometría tridimensional del invernadero debe de ser lo suficientemente grande como para permitir una mecanización rápida y cómodamente.
- El diseño debe de ser en lo mayor posible eco amigable, con mayor aprovechamiento de los recursos naturales exteriores, con una geometría que aproveche en lo máximo la energía almacenada en su interior del invernadero.

Las diferentes formas más utilizadas en invernaderos según al tipo de cultivo que contendrá, determinan las características de sus elementos constructivos como:

- Tipo de suelo que utilizara, que pueden ser naturales o sustratos artificiales.
- La topología mayormente utilizada son lugares con pequeña pendiente orientada de norte a sur.

2.2.1.1. Tipos de invernadero

2.2.1.1.1. Invernadero plano o tipo parral

Normalmente implementado en zonas áridas, donde las precipitaciones fluviales son escasas, la estructura es simple, constituido por partes verticales y horizontales, las verticales son construidos por estructuras rígidos como vigas de concreto o algún material resistente a la humedad, la cubierta lo forma un material plástico transparente con poca inclinación, en la siguiente figura 01 se muestra un ejemplar de este tipo de invernadero.

Figura 1: Invernadero tipo parral



Fuente: (Ortega Álvarez, 2017, pág. 13)

Sus desventajas son:

- Mala ventilación
- La instalación de ventanas cenitales es bastante difícil
- Poco volumen de aire
- Poco o nada aconsejable en los lugares lluviosos
- Peligro de destrucción del plástico y de la instalación por su vulnerabilidad al viento.

Algunas de sus principales ventajas son:



- Bajo presupuesto para su construcción
- Su gran adaptabilidad a la geometría del terreno
- Mayor resistencia al viento
- Aprovecha el agua de lluvia en periodos secos
- Mayor adsorción de la luz solar.

2.2.1.1.2. Invernadero de capilla

La diferencia más notoria de este invernadero figura 02, es la inclinación o pendiente del techo que este tiene, que normalmente oscila entre 15 a 35 grados centígrados de inclinación, esta configuración permite variar la radiación y pluviometría que mejor se adapte para el cultivo.

Sus principales ventajas son:

- Mayor área de ventilación, debido a los arcos o tijerales
- Mayor resistencia a precipitaciones pluviales
- Es muy adaptable para todo tipo de plástico como cubierta
- Es de fácil construcción y de fácil conservación

Figura 2: Invernadero tipo capilla



Fuente: (Mora López , y otros, 2015, pág. 129)

2.2.1.1.3. Invernadero asimétrico o inacral

La razón para construir la cubierta o techo de forma asimétrica se debe fundamentalmente para poder capturar en lo posible más luz solar, por esta razón, en la construcción y diseño, los invernaderos son construidos con una orientación y con una inclinación de mayor captación de luz solar. Por estas razones la cara que está en la orientación al Sur es aumentado en lo posible, esto hace la construir invernaderos con una orientación en sentido Este-Oeste, paralelo aparente del recorrido del Sol, en la figura 03 se ve este tipo de invernaderos.

Figura 3: Invernadero tipo asimétrico



Fuente: (Carrillo Reveles & Vázquez Minjares, 2008, pág. 23)

Las magnitudes y dimensiones que debe de tener es de acuerdo a las características climáticas de la zona, según el proyecto AUTOMATIZACION DE UN INVERNADERO CON EL PLCS7-200:

La inclinación de la cubierta debe ser aquella que permita que la radiación solar incida perpendicularmente sobre la cubierta al mediodía solar durante el solsticio de invierno, época en la que el sol alcanza su punto más bajo. Este ángulo deberá ser próximo a 60° , pero ocasiona grandes inconvenientes por la inestabilidad de la estructura a los fuertes vientos. Por ello se han tomado ángulo comprendidos entre los 8° y 11° en la cara sur y entre los 18° y 30° en la cara norte.

La altura máxima de la cumbre varía entre 3.0 m y 5.0 m, y su altura mínima de 2.3 m a 3.0 m. La altura de las bandas oscila entre 2.15 m y 3.0 m. La separación de los apoyos interiores suele ser de 2.0 m x 4.0 m. En la Figura 2.3 se muestra una ilustración de este invernadero (Carrillo Reveles & Vázquez Minjares, 2008, pág. 23).

Sus principales ventajas son:

- Buen aprovechamiento de la luz en la época invernal.

- Elevada inercia térmica debido a su gran volumen unitario.
- Buena ventilación.
- Permite la instalación de ventilación cenital a sotavento.

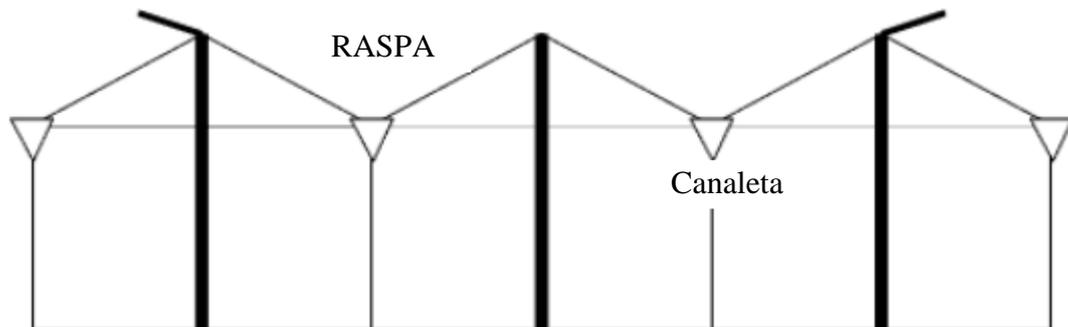
Sus desventajas:

- Se dificulta el cambio del plástico de la cubierta.
- Difícil de aprovechar el agua de lluvia
- Tiene más pérdidas de calor a través de la cubierta debido a su mayor superficie.

2.2.1.1.4. Invernadero en raspa y amagado

Una forma de solucionar la generación de bolsas de agua en los techos de un invernadero por causas fluviales es aumentar el número de vértices para luego unir la parte más baja por una canaleta, que servirá para evacuar el agua, estas uniones de las caras en la parte más alta, dan un aspecto de raspa como se muestra en la figura 04.

Figura 4: Perfil de un invernadero raspa



Fuente: (Barrera Martin, Herrero Martin, & Meraz Garía, 2014, pág. 9)

Generalmente son construidas con las siguientes dimensiones:

Su estructura es muy similar al tipo parral, pero varía la forma de la cubierta. Se aumenta la altura máxima del invernadero en la cumbre, que oscila entre 3.0 m y .2 m, formando lo que se conoce como “raspa”. En la parte más baja, conocida como “amagado”, se unen las mallas de la cubierta al suelo mediante postes y horquillas de



hierro que permite colocar los canalones para el desagüe de las aguas pluviales. La altura del amagado oscila de 2.0 m a 2.8 m, la de las bandas entre 2.0 m y 2.0m. La separación entre apoyos y los postes del amagado es de 2.0 m x 4.0 m y el ángulo de la cubierta oscila entre 6° y 20°, siendo este último el valor óptimo (Pumasupa Paro N. E., 2014).

Ventajas principales son:

- Su estructura se basa principalmente metal galvanizado, lo que reduce el costo.
- Ofrece mejor control del clima interior
- Permite la instalación de ventanas cenitales
- Montaje rápido

Sus desventajas son:

- No aprovecha las aguas fluviales.
- Se dificulta el cambio del plástico de la cubierta.
- Mayor pérdida de calor, debido a la mayor superficie de plástico.

2.2.1.1.5. Invernadero tipo túnel o semicilindro

El nombre característico de estos invernaderos se debe a su apariencia estructural, que se parecen a grandes semi-tubos acostados en la superficie del suelo, donde la parte frontal o lado de la entrada tiene un perfil semicircular formado por algún material resistente. La cubierta y sus laterales del invernadero son completamente constituidos por plástico, esta forma geométrica le da al invernadero una gran resistencia a fuertes vientos y fácil aparcamiento al ser estructuras prefabricadas.

Los soportes son de tubos de hierro galvanizado y tienen una separación interior de 5.0 m x 8.0 m o 3.0 m x 5.0 m. La altura máxima de este tipo de invernaderos oscila

entre 3.5m y 5.0m. En las bandas laterales se adoptan alturas de 2.5 m a 4.0 m. El ancho de estas naves está comprendido entre 6.0 m y 9.0 m y permiten el adosamiento de varios módulos. La ventilación es mediante ventanas cenitales que se abren hacia el exterior del invernadero (Pumasupa Paro N. E., 2014).

Figura 5: Invernadero tipo túnel



Fuente: (Barrera Martin, Herrero Martin, & Meraz Garía, 2014, pág. 12)

Sus ventajas son:

- Buena ventilación
- Buena estanqueidad a la lluvia y al aire
- Estructura con pocos obstáculos en su diseño
- Permite la instalación de ventilación cenital o sotavento y facilita su accionamiento mecanizado
- Buena absorción de radiación solar
- Fácil instalación

Mientras que sus principales desventajas son:

- Elevado costo
- No aprovecha agua de pluvial

2.2.1.1.6. Invernadero de cristal o tipo venlo

Son invernaderos (figura 6) que tienen como cubierta un cristal transparente, generalmente es de vidrio, en su perímetro que generalmente es de concreto debe de contener cualquier tipo de ventarrones naturales.

El techo de este invernadero industrial descansa sobre los canales de recogida de pluviales y sobre un conjunto de barras transversales. La anchura de cada módulo es de 3.2 m. Desde los canales hasta la cumbrera hay un solo panel de vidrio de una longitud de 1.65 m, y anchura que varía desde 0.75 m hasta 1.6 m. La separación entre columnas en la dirección paralela a las canales es de 3.0 m. En sentido transversal está separadas 3.2 m si hay una línea de columnas debajo de cada canal, o 6.4 m si se construye algún tipo de viga en celosía (Carrillo Reveles & Vázquez Minjares, 2008).

Figura 6: Estructura del invernadero tipo venlo



Fuente: (Barrera Martin, Herrero Martin, & Meraz Garía, 2014, pág. 13)

2.2.2. Parámetros de control de un invernadero

2.2.2.1. Humidificación del suelo

Si los demás parámetros están en las proporciones adecuadas para iniciar el nacimiento de una determinada planta, la humidificación del suelo da inicio a la



germinación y ha de estar presente en todo el ciclo de vida de la planta, además de disolver las sustancias y proporcionar conductividad este parámetro ha de estar controlado, ya que un exceso podría causar asfixia radicular y causar la pudrición de raíces y su escasez afectara en su desarrollo.

Las diferentes tecnologías de riego, son aplicados de acuerdo a las dimensiones del campo del cultivo, también la cantidad del recurso agua disponible. Las más utilizadas son: el riego por gravedad, en donde el cultivo es inundado o por surcos; el riego por aspersión, en donde el agua es comprimida para que luego este pase por unas boquillas que dispersan en pequeñas gotas de agua, simulando las condiciones de lluvia; y el riego localizado, o comúnmente llamado riego por goteo.

2.2.2.2. Temperatura

Los diversos lugares dieron diversas variedades de productos vegetales y frutas, esto se debe en gran medida a la temperatura ambiental de dicha zona, como por ejemplo: para la producción de fresas, estas deben de cultivarse con temperaturas que estén en el rango de 2 °C a 6 °C.

La fresa es considerada como una especie que soporta temperaturas hasta de 2°C durante el reposo vegetativo, y para interrumpir el estado durmiente de las yemas se requiere temperaturas inferiores a 6°C. Las estructuras vegetativas son altamente resistentes a las heladas pero sus flores se dañan con temperaturas menores a los 0°C (Bianchi, 1986). Por otro lado, con temperaturas superiores a los 40°C se induce la producción de frutos de mala calidad (Maroto, Pascual, Alargada, & López, 1986), además de planchado de los frutos que es un síntoma de deshidratación (Folquer, 1986). Sin embargo, existen variedades adaptadas a zonas cálidas y soportan temperaturas de hasta 55 °C. (Zaragoza Nieto R. D., 2013).



2.2.2.3. Iluminación o luz solar

La energía vital de toda planta que realiza fotosíntesis, el reemplazo por otras fuentes artificiales luz requerirá un profundo conocimiento fisiológico de la planta a cultivar, como por ejemplo utilizar mayor luminosidad artificial requerida aumentar la temperatura del invernadero, por el contrario, si se disminuye la luminosidad puede que requiera descender la necesidad de otros factores.

La fuente natural de luminosidad es la luz solar, para mejorar la adsorción por el invernadero es necesario mejorar algunos factores:

- Orientación adecuada del invernadero
- Acolchado aislamiento del suelo con material blanco
- Materiales que reducen las sombras interiores
- Materiales de cubierta con buena transparencia
- Aumento del ángulo de incidencia de los rayos solares sobre la cubierta

Si a de utilizarse luz artificial, esta deberá de emitir en lo posible toda la radiación del espectro electromagnético de la luz solar con una distribución de energía proporcional a la intensidad luminosa, si alguna onda electromagnética tiene mayor energía que las otras podría esterilizar el cultivo.

Es interesante destacar el uso del blanqueo, el cual tiene la función de filtro para evitar algunos tipos de rayos solares, tales como los rayos UV; ya que esta labor está en función del desarrollo del cultivo y de las temperaturas, y tiene efectos contradictorios que hay que conocer para hacer un correcto uso. Hay que saber que la planta sombreada se ahíla y se producen abortos de flores en determinadas especies sensibles a la luz (especialmente tomate, pimiento y berenjena), por lo que el manejo del riego y de la solución nutritiva tiene que ir unida al efecto que produce el blanqueo. Los plásticos

sucios o envejecidos provocan el mismo efecto que el blanqueo (Pumasupa Paro N. E., 2014).

2.2.2.4. Potencial de hidrógeno del suelo

La acidez y la alcalinidad otro de los factores importantes a considerarse en un invernadero, con un adecuado potencial de hidrógeno (pH) se aumenta en forma considerable la adsorción de nutrientes para la planta.

Cuando el pH desciende en exceso, es decir, por debajo de 5, la planta tiene problemas para capturar otro tipo de sustancias. El principal obstáculo se da en la absorción de molibdeno, pero con un pH bajo también se obstaculiza la asimilación de nitrógeno y azufre. A valores escasos de pH puede producirse una intoxicación por manganeso y aluminio, debido a que la planta los absorbe con mayor facilidad (Anthura, 2022).

La siguiente tabla resume lo expuesto hasta ahora:

Tabla 1: Respuesta de adsorción según el pH

Difícil o imposible de absorber a pH ≤ 5	Difícil o imposible de absorber a pH ≥ 6
Nitrógeno (N)	Boro (B)
Molibdeno (Mo)	Fosfatos (PO)
Azufre (S)	Cobre (Cu)

Fuente: (Anthura, 2022, pág. 1)

2.2.2.5. Conductividad del suelo

La salinidad que es un compuesto por distintos minerales abióticos que contribuyen en el desarrollo del cultivo, determinan la calidad y fertilidad del suelo del invernadero, la forma de medir salinidad en el suelo es por medio de la Conductividad Eléctrica (CE).

La CE mide la capacidad del suelo, para conducir corriente eléctrica al aprovechar la propiedad de las sales en la conducción de esta; por lo tanto, la CE mide la concentración de sales solubles presentes en la solución del suelo. Su valor es más alto cuanto más fácil se mueva dicha corriente a través del mismo suelo por una concentración más elevada de sales. Las unidades utilizadas para medir la CE son dS/m (decisiemens por metro) (Intagri S.C., 2021).

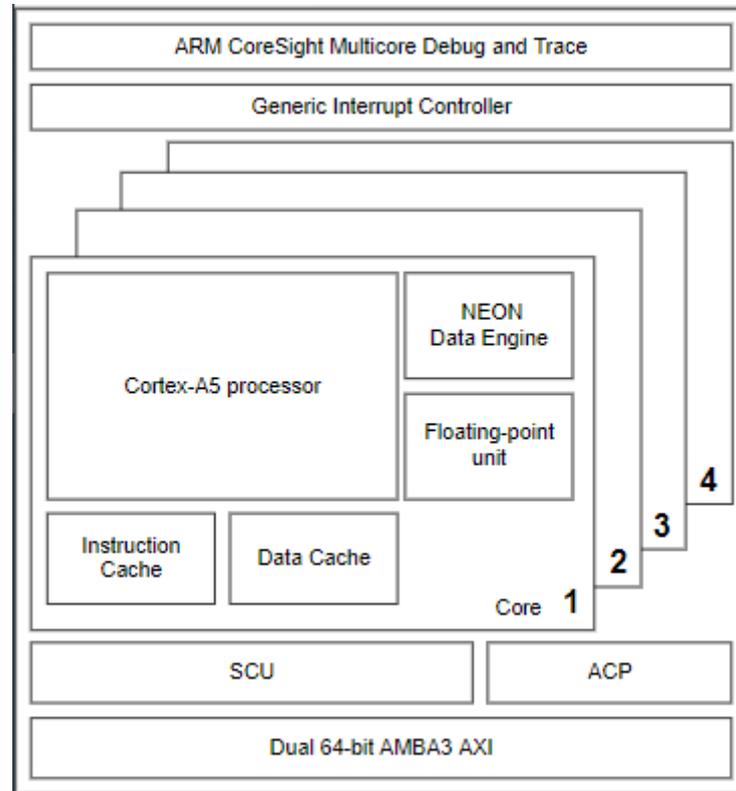
2.2.3. Microprocesadores ARM córtex

La tecnología en arquitectura ARM (Advanced Risc Machine), es la más utilizada en sistemas embebidos, debido a su alto poder de procesamiento y a su alta integración, de las distintas aplicaciones los microcontroladores se clasifican como:

- ARM cortex-M, es un microcontrolador diseñado para aplicaciones en general, su consumo de energía es muy bajo y con una amplia integración de periféricos.
- ARM cortex-R, es una familia de núcleos de procesador RISC ARM de 32 y 64 bits con licencia de Arm Holdings. Los núcleos están optimizados para aplicaciones críticas de seguridad y en tiempo real. Los núcleos de esta familia implementan el perfil ARM Real-time (R), que es uno de los tres perfiles de arquitectura, los otros dos son el perfil de aplicación (A) implementado por la familia Cortex-A y el perfil de microcontrolador (M) implementado por el Cortex-M familia. La familia de microprocesadores ARM Cortex-R actualmente consta de ARM Cortex-R4(F), ARM Cortex-R5(F), ARM Cortex-R7(F), ARM Cortex-R8(F), ARM Cortex-R52(F) , y ARM Cortex-R82(F) (ARM Developer, 2022).
- ARM Cortex-A, utilizada para aplicaciones que requieren mayor memoria RAM y ROM con un alto poder de procesamiento en aplicaciones de alto rendimiento,

entre sus mayores aplicaciones resalta en teléfonos celulares inteligentes, en la figura 7 muestra sus características.

Figura 7: Procesador A5



Fuente: (ARM Developer, 2022, pág. 1)

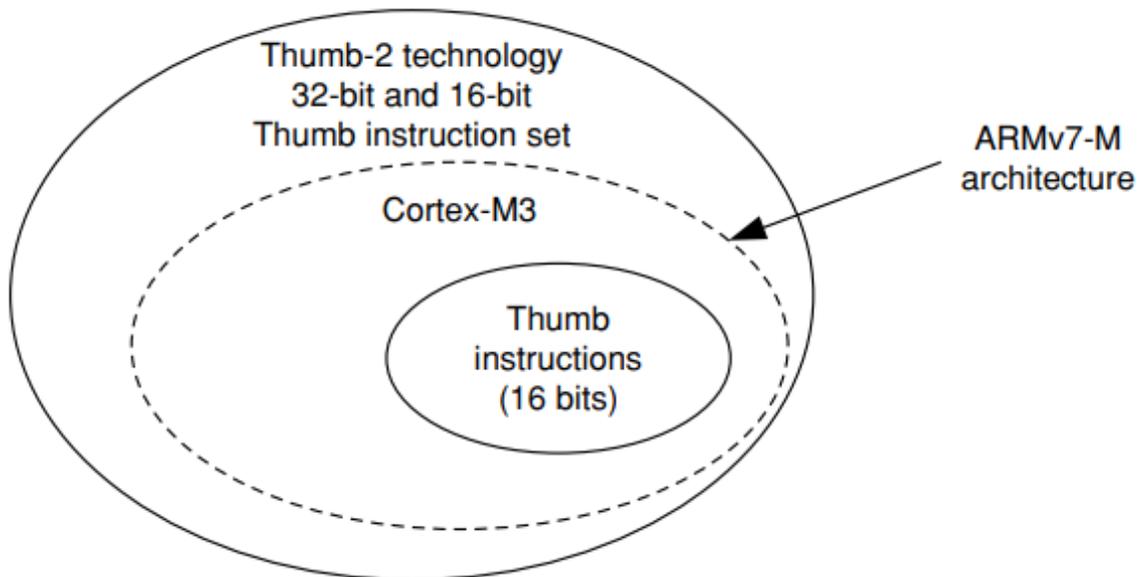
2.2.3.1. Tecnología thumb2

Thumb2 es una tecnología implementado para reducir el tamaño del código de instrucción en cada ciclo de ejecución (Fetch), si el procesador es de 64 bits, por consecuencia ejecuta instrucciones de 64 bits. Lo que hace la tecnología Thumb2 es reducir el código de instrucciones de 64 a 16 bits abreviadas compactadas como un sub conjuntos de instrucciones equivalentes, de esta forma se reduce el tamaño de memoria.

Thumb2 es una tecnología extendido de Thumb que mejora y trabaja en ambos tipos de instrucciones de 16 bits y 32 bits como se muestra en la figura 8, sin la necesidad de cambiar el modo de funcionamiento del procesador entre Thumb y ARM instrucciones. La nueva tecnología de núcleo Thumb-2 está construida sobre la base de la

tecnología de compresión de código Thumb de ARM líder en la industria, conservando todas las cualidades excepcionales del código compacto y la compatibilidad completa del código con las soluciones ARM existentes, al tiempo que proporciona un mayor rendimiento y eficiencia energética a partir del código compacto (Huanca Chuquicallata, 2022).

Figura 8: Arquitectura Thumb



Fuente: (Yui, 2010, pág. 8)

2.2.3.2. Microcontrolador ARM cortex-m

Normalmente su núcleo es de 32 bits de alto rendimiento, y es diseñado para aplicaciones compactas o sistemas embebidos (para aplicaciones específicas), está construida sobre la arquitectura Armv7.

El procesador Cortex-M3 está desarrollado específicamente para plataformas de alto rendimiento y bajo costo para una amplia gama de dispositivos, incluidos microcontroladores, sistemas de carrocería automotriz, sistemas de control industrial y redes inalámbricas y sensores (ARMDeveloper, 2022).



Entre sus principales características son:

- Arquitectura Armv7-M
- Bus AHB (Harvard bus architecture) para el sistema y ATB para los periféricos.
- Cuenta con tecnología Thumb/Thum2
- En 3 etapas, en ejecutar una instrucción
- Controlador de interrupciones sectorizado anidada
- 8 regiones MPU (Unidad de Múltiples Procesos) opcionales con subregiones y región de fondo
- Instrucciones de procesamiento de campo de bits integradas y bandas de bits a nivel de bus
- Controlador de interrupción de despertar de la CPU
- Multiplicación de hardware de ciclo único (32x32), división de hardware (2-12 ciclos), compatibilidad con ajuste saturado
- Puertos opcionales JTAG y Serial Wire Debug. Hasta 8 punto de interrupción y 4 puntos de observación
- Instrucción opcional (ETM), seguimiento de datos (DWT) y seguimiento de instrumentación (ITM)

2.2.3.2.1. Clasificación de los ARM corte-m

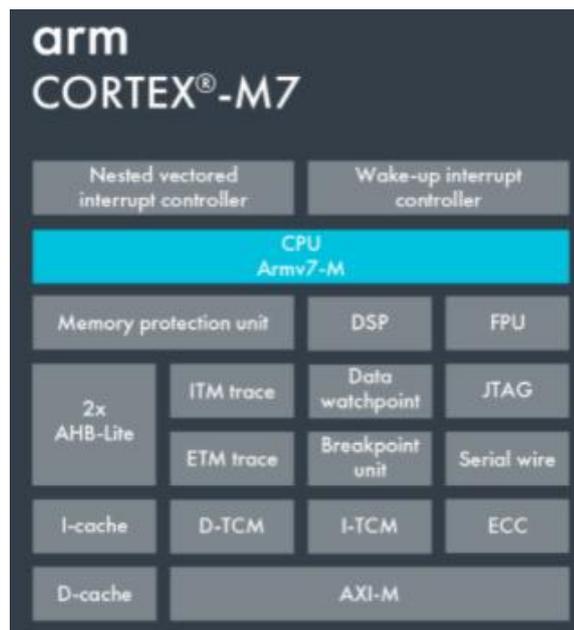
La extensa familia, se distribuye a lo largo de las aplicaciones que van desde el uso general hasta la aplicación militar-aeroespacial, algunos de ellos se muestran a continuación:

- CORTEX-M0, cuenta con todos los periféricos necesarios y se puede encontrar el más pequeño posible en tamaño.



- CORTEX-M1, diseñado especialmente a ser implementados en un FPGA (Field-Programmable Gate Array).
- CORTEX-M3, es el más usado por las industrias tecnológicas, por tener un alto rendimiento y de bajo costo.
- CORTEX-M4, además de contar con todos los beneficios de sus antecesores, cuenta con la capacidad de procesar señales.
- CORTEX-M7, permite construir la variedad más sofisticada de MCU y SoC integrados. Ha sido diseñado para ofrecer un nivel muy alto de rendimiento, manteniendo al mismo tiempo la excelente capacidad de respuesta y la facilidad de uso de la arquitectura Armv7-M. Sus interfaces de sistema flexibles y de alto rendimiento líderes en la industria son ideales para una amplia variedad de áreas de aplicación, incluidas la automoción, la automatización industrial, los dispositivos médicos, el audio de alta gama, el procesamiento de imágenes y voz, la fusión de sensores y el control de motores (ARMDeveloper, 2022).

Figura 9: Periféricos de un ARM Cortex-M7



Fuente: (ARM Developer, 2022, pág. 1)

2.2.3.3. FreeRTOS

La arquitectura ARM es utilizada por muchas empresas fabricantes de chip licenciadas por ARM Holding, todos los chip que son implementados con esta CPU ARM comparten las mismas instrucciones, y para poder acelerar las aplicaciones se diseñaron pequeños sistemas operativos (OS) libres y de pago reduciendo el tiempo y costo en el desarrollo de proyectos. FreeRTOS es uno de los muchos sistemas operativos que administra las distintas tareas que debe ejecutar la CPU en un determinado tiempo, es pequeño y simple, implementado especialmente para microcontroladores, el código está compuesto por 3 archivos en lenguaje C, para administrar a las distintas rutinas y múltiples subprocesos FreeRTOS utiliza distintas funciones para administrar el estado de cada proceso o tarea en ejecución los estados en la que puede estar son:

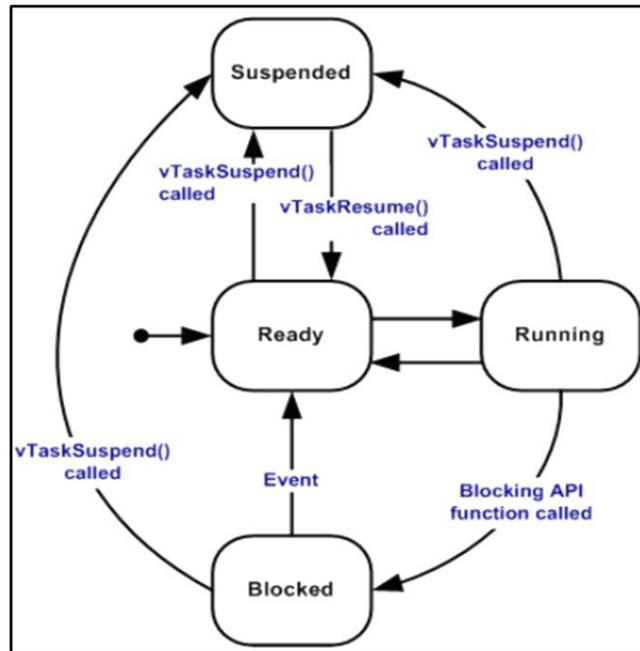
- Running. Cuando un subproceso está ejecutándose y asignado el recurso de la CPU.



- **Ready.** Cuando la tarea esta lista para entrar en modo running, esto quiere decir que todos los recursos como; memoria y periféricos están listos, y están esperando que el “scheduler” (programador de tareas) del sistema operativo le asigne la CPU.
- **Blocked.** El "scheduler" puede llevar una tarea a este estado si detecta que está esperando por un evento temporal (una espera de tiempo con la función), esperando por un evento externo (Interrupción), o a que un recurso compartido sea liberado (colas, semáforo, notificaciones). Cuando la espera termina, el "scheduler" puede regresar la tarea a estado "ready" (FreeRTOS, 2021).
- **Suspended.** Muy similar a "blocked" pero se diferencia en que una tarea solo puede ingresar o salir de este estado cuando es utilizada de forma específica la función (vTaskSuspend () o vTaskResume ()) respectivamente. No es el "scheduler" del sistema operativo el que toma la decisión de cambiar el estado de una tarea a "suspended".

La decisión de la cual tarea entregar la CPU va a depender de las políticas de planificación seleccionada para el "scheduler". Si varias tareas están en estado ready, siempre la que tenga mayor prioridad será la que pase primero a estado "running", en la figura 10 muestra el diagrama de bloques de los distintos estados en la que puede estar un proceso (FreeRTOS, 2021).

Figura 10: Posibles estados en la que puede estar un subproceso



Fuente: (FreeRTOS, 2021, pág. 1)

2.2.3.4. Interface de comunicación serial

Todo sistema de control, que son conformados por distintas partes que cumplen una función específica, necesitan comunicarse continuamente entre ellos, para ello utiliza distintas formas y recursos de hardware, la más simple y económico es la comunicación serial que solo requiere como mínimo 3 líneas de conexión para una comunicación full-dúplex, los microcontroladores Cortex-M3 cuentan con el protocolo RS232.

RS-232 (Recommended Standard 232, en español: "Estándar Recomendado 232"), también conocido como EIA/TIA RS-232C, es una interfaz que designa una norma para el intercambio de datos binarios serie entre un DTE (Data Terminal Equipment, "Equipo Terminal de Datos"), como por ejemplo una computadora, y un DCE (Data Communication Equipment, "Equipo de Comunicación de Datos"), por ejemplo, un módem. Existen otros casos en los que también se utiliza la interfaz RS-232 (WIKIPEDIA La enciclopedia Libre, 2022)



2.2.4. Modbus RS485

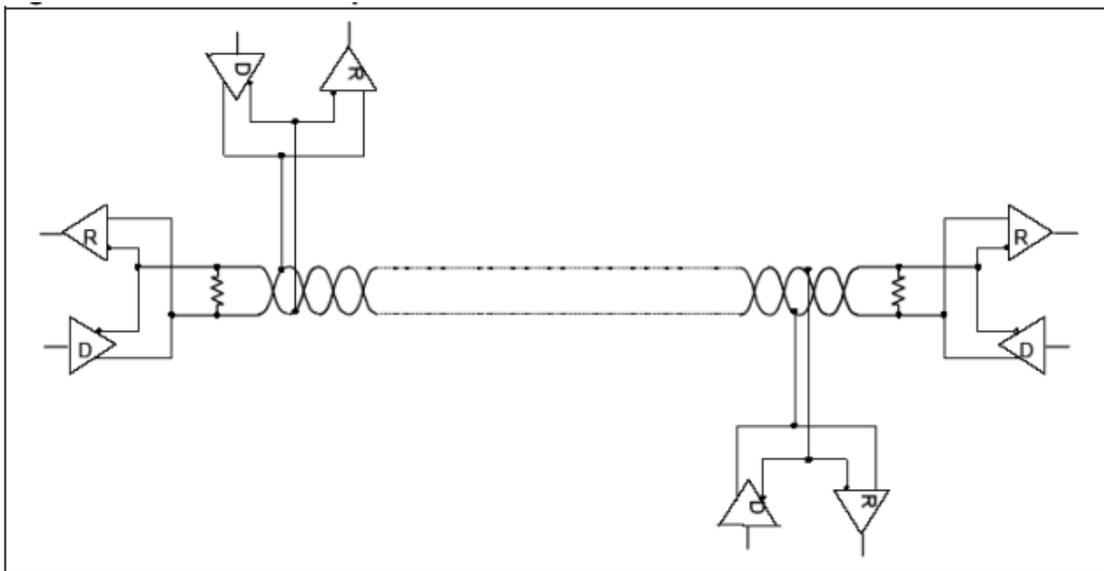
El ambiente donde operan muchos equipos electrónicos y equipos de alta potencia, es bastante contaminada con radiación electromagnética, ya sea de energía estática o inductiva emitidos por los motores con rotores de corriente continua. Para minimizar o superar el problema, la empresa Modicon desarrollo en 1979 el “protocolo de comunicaciones situado en los niveles 1, 2 y 7 del Modelo OSI, basado en la arquitectura maestro/esclavo (RTU) o cliente/servidor (TCP/IP), diseñado para su gama de controladores lógicos programables (PLCs)” (WIKIPEDIA La enciclopedia libre, 2021).

Esta comunicación serial opera dentro del protocolo RS232 derivado de la arquitectura Maestro-Esclavo. Para implementar en un microcontrolador, este deberá de contar con el periférico rs232 más un driver que opere en el nivel 1 del modelo OSI y un firmware que gestionará toda la mensajería de comunicación.

Para que la comunicación sea lo suficientemente robusto, la mayoría de los procesos industriales son implementados con el protocolo RS485 y el enlace son de comunicación Half-Duplex (comunicaciones en una sola dirección en un determinado tiempo) como se muestra en la figura 11, que ofrece una fácil forma de implementar la capa física en redes industriales con protección ESD (descarga electrostática).

- Dispositivo Maestro, es el encargado de iniciar la comunicación enviando y recibiendo mensajes, e implementa las tareas de coordinación entre diferentes esclavos.
- Dispositivo Esclavo, son los dispositivos conectados al bus y que atienden a las peticiones del maestro, proporcionando información o ejecutando tareas.

Figura 11: Distintos dispositivos conectados al bus RS485



Fuente: (STMicroelectronic, 2007, pág. 4)

2.2.5. El modelo TCP/IP

Es un protocolo que enlaza la comunicación entre dos computadoras estandarizada, bajo cualquier situación, independientemente de las condiciones de cualquier nodo o red en particular del internet Word, sus cuatro capas del protocolo TCP/IP se detallan a continuación.

- Capa de aplicación: se ubica por encima de todas las demás capas que maneja protocolos de alto nivel, aspectos de representación, codificación y control de diálogo. El modelo TCP/IP combina todos los aspectos relacionados con las aplicaciones en una sola capa y garantiza que estos datos estén correctamente empaquetados para la siguiente capa.
- Capa de transporte: cuya función es asegurar la calidad del servicio con respecto a la confiabilidad, el control de flujo y la corrección de errores. En donde el protocolo para el control de la transmisión (TCP), ofrece maneras flexibles y de alta calidad para crear comunicaciones de red confiables, sin problemas de flujo y con un nivel de error bajo, mediante o segmentación.



- Capa de Internet: El propósito de la capa de Internet es enviar paquetes origen desde cualquier red en el internet y que estos paquetes lleguen a su destino, independientemente de la ruta y de las redes que recorrieron para llegar hasta allí. El protocolo específico que rige esta capa se denomina Protocolo Internet (IP).
- Capa de acceso de red: También se denomina capa de host a red. Es la capa que se ocupa de todos los aspectos que requiere un paquete IP para realizar realmente un enlace físico y luego realizar otro enlace físico. Esta capa incluye los detalles de tecnología LAN y WAN y todos los detalles de las capas físicas y de enlace de datos del modelo OSI.

2.2.6. Sensores

Para poder saber la magnitud física de algún elemento natural o mecánicos, es necesario transformarla a magnitudes eléctricas como voltaje, corriente, resistencia, energía y potencia. Esta etapa de transformación deberá de ser; acondiciona, amplificar o linealidad dependiendo del elemento que lo transforma, la señal analógica o digitalmente, para luego ser procesada por alguna computadora.

La calidad y fidelidad de un sensor está determinado por las siguientes características:

- Precisión. Define la máxima desviación de la salida con respecto al valor real medido de un sensor en determinadas condiciones y una escala específica.
- Rango. Es el rango de valores de la magnitud de entrada comprendido entre el máximo y el mínimo detectables por un sensor, con una tolerancia de error aceptable.



- Linealidad. Llamada también correlación lineal o pendiente constante de proporcionalidad que relaciona, que para un aumento de valor en la variable de entrada debe existir un aumento en la variable de salida.
- Sensibilidad. Característica que indica la mayor o menor variación de la salida por unidad de la magnitud de entrada. Un sensor es tanto más sensible cuanto mayor sea la variación de la salida producida por una determinada variación de entrada.
- Ruido. Es la variación aleatoria del propio transductor o del sistema de medida causada por perturbaciones no lineales del sensor, que produce una desviación de la salida con respecto al valor teórico.
- Margen de error. Es la diferencia entre el valor medido y el valor real.
- Resolución. Es medida más pequeña que se pueda detectar.

2.2.7. Lenguaje de programación

Para que un sistema o maquina inteligente realice tareas complejas, es necesario traducir estas actividades a una secuencia de pasos o actividades codificadas e interpretables por el microcontrolador, esta traducción que normalmente lo realiza un software específicamente creado, tendrá que comprobar la correcta codificación del código fuente y de realizar la síntesis del programa para optimizar recursos de hardware. Para que un sistema realice grandes y complejas tareas el código fuente es dividido en bloques, y son tratados como objetos, consiguiendo una mayor entendimiento y facilidad de futuras de mejorar el código que controla la máquina.

2.2.7.1. Assembler

La representación simbólica de una palabra maquina “0” y “1” (ceros y unos) mejoran el entendimiento que realiza el programa. Es un lenguaje de programación de

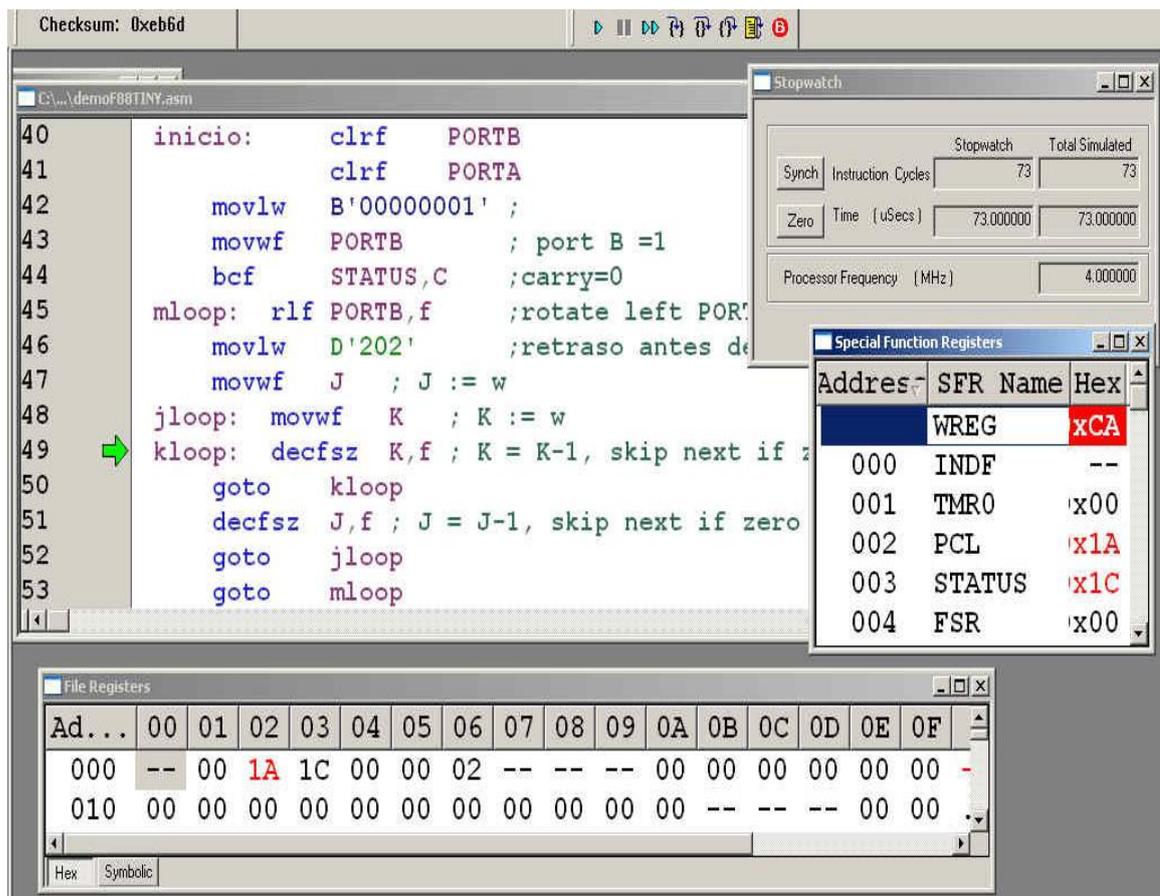


bajo nivel, que tiene casi un total acceso a los registros y memoria de una microcomputadora, que en un lenguaje de alto nivel.

Se denominan lenguajes de bajo nivel porque dependen de la arquitectura del procesador en el que queremos ejecutar el programa y porque no disponen de sentencias con una estructura lógica que faciliten la programación y la comprensión del código para el programador, sino que están formados por una lista de instrucciones específicas de una arquitectura.

Lenguaje simbólico que se ha definido para que se puedan escribir programas con una sintaxis próxima al lenguaje de máquina, pero sin tener que escribir el código en binario, sino utilizando una serie de mnemónicos más fáciles de entender para el programador. Para ejecutar estos programas también es necesario un proceso de traducción, generalmente denominado ensamblaje, pero más sencillo que en los lenguajes de alto nivel. (Alberto Orenga & Enrique Manonellas, 2000)

Figura 12: Programación en assembler



Fuente: (MICROCHIP, 2022, pág. 2)

2.2.7.2. Lenguaje de programación C/C++

Se trata de un lenguaje de alto nivel o intermedio, donde unos conjuntos de códigos son entrelazados para realizar una determinada acción, y son encapsulados y tratados como objetos para poder restringir todo el algoritmo y procesos de algunas funciones y variables privadas que requieren un mayor conocimiento de la máquina, los métodos de programación más utilizadas es la programación procedimental y la orientada a objetos.

La diferencia fundamental entre la programación procedimental y la orientada a objetos está en la forma de tratar los datos y las acciones. En la primera aproximación ambos conceptos son cosas distintas, se definen unas estructuras de datos y luego se



define una serie de rutinas que operan sobre ellas. Para cada estructura de datos se necesita un nuevo conjunto de rutinas. En la programación orientada a objetos los datos y las acciones están muy relacionadas. Cuando definimos los datos (objetos) también definimos sus acciones. En lugar de un conjunto de rutinas que operan sobre unos datos tenemos objetos que interactúan entre sí.

Objetos y mensajes Un Objeto es una entidad que contiene información y un conjunto de acciones que operan sobre los datos. Para que un objeto realice una de sus acciones se le manda un mensaje. Por tanto, la primera ventaja de la programación orientada a objetos es la encapsulación de datos y operaciones, es decir, la posibilidad de definir Tipos Abstractos de Datos. De cualquier forma la encapsulación es una ventaja mínima de la programación orientada a objetos. Una característica mucho más importante es la posibilidad de que los objetos puedan heredar características de otros objetos. (Taleng Oliag, 1995)

2.2.8. Hipótesis

2.2.8.1. Hipótesis general

Las diferentes variables climáticas que intervienen en la producción de un determinado cultivo podrán ser monitorizados y controlables con un microcontrolador ARM CORTEX-M3.

2.2.8.2. Hipótesis específica

- El microcontrolador ARM CORTEX-M3, podrá atender toda la demanda, de todos los sensores.
- El sistema de control, podrá mantener las condiciones climáticas necesarias, para un determinado cultivo.



CAPITULO III

MATERIALES Y MÉTODO

3.1. DISEÑO METODOLÓGICO DE LA INVESTIGACIÓN

3.1.1. Diseño de investigación

En la presente investigación se está realizando una forma de establecer un invernadero que pueda mantener condiciones climáticas para un determinado cultivo y la documentación que existe para lograrlo es muy poca o inentendible, por consiguiente el plan o estrategia de investigación es **experimental** y **descriptiva**; experimental porque para responder la pregunta principal es necesario someter el prototipo a un experimento bajo condiciones reales en campo, y descriptiva porque para lograr los objetivos específicos es necesario conocer los aspectos más importantes de las tecnologías de las que podemos disponer para la conclusión del prototipo.

3.1.2. Nivel de la investigación

Para entender los diferentes propósitos de porque utilizar un microcontrolador ARM CORTEX-M3, para el control de las diferentes variables climatológicas, la cual tiene un alcance **explicativo** y en cierto modo es **exploratoria** pues se está utilizando un microcontrolador de 32 bits, utilizado en otras aplicaciones.

3.1.3. Tipo de investigación

El tipo de investigación a desarrollarse es aplicada, pues busca resolver en cierta medida la cultivación de distintas plantas en diferentes zonas geográficas, en las que sería imposible su cultivo sin la intervención y manipulación de las variables climáticas. Este estudio y análisis en torno a este problema, ampliará el conocimiento que se tiene de este, dependiendo del enfoque que se le quiere dar al problema.



3.1.4. Población y muestra de la investigación

3.1.4.1. Población

La población definido como el conjunto de entidades o cosas respecto a las cuales se realiza el estudio de una investigación, en nuestro caso la población está definida por las variaciones de los valores climáticos dentro del invernadero respecto a todo el tiempo en que se encuentre activo.

3.1.4.2. Muestra

Es aquella cantidad seleccionada al azar que representara a la población, es decir un segmento que tiene las características y propiedades de la población. Para nuestra investigación será la medición de las variaciones climáticas dentro invernadero en un intervalo de tiempo determinado.

3.1.5. Ubicación y descripción de la investigación

3.1.5.1. Ubicación

El proyecto de investigación aplicativo se realizó en el departamento de Puno, Provincia de Puno, Distrito de Puno.

3.1.5.2. Descripción de la investigación

Para poder recrear el ambiente natural para un cultivo determinado se requiere suministrar, limitar y aislar los elementos necesarios para germinación, crecimiento y producción de la planta. El prototipo deberá tener la capacidad de controlar y proteger el exceso de cualquier elemento natural. Para lograrlo se utilizará diversos sensores electrónicos y una estructura resistente a las inclemencias meteorológicas que resguardaran a la planta.



3.1.6. Técnicas e instrumentos de recolección de Datos

En la recolección de datos se refiere a cómo y qué medios se usan para la obtención de la información, que será de utilidad para la comprobación de nuestra hipótesis, para ello utilizaremos varios instrumentos homologados de tal manera que servirán para la comparación y así poder determinar la diferencia (Error) y la calibración del prototipo.

3.1.6.1. Instrumentos

Los instrumentos que intervinieron para la recolección de datos para este proyecto de investigación fueron:

- Cinta métrica. Este instrumento se utilizó para la construcción de prototipo invernadero de acuerdo de las dimensiones del diseño.
- Multímetro digital. Que se utilizó para calibrar y medir los componentes electrónicos utilizados es este proyecto de investigación.
- Vernier o pie de rey de la marca Makawa, este tiene una precisión de 0.02mm que se utilizó para realizar la medida milimétrica de los diseños del invernadero.
- ID STM32cubeID. Utilizado para programar el microcontrolador stm32f103.
- Hoja de observación para la calibración y funcionamiento de ensayo error.
- Software de diseño de circuitos electrónicos.
- Autodesk Inventos. Utilizado para el diseño de la estructura del prototipo.
- Laptop de la marca HP con sistema operativo Windows 10.
- Módulo Stm32f103vct6
- Microcontrolador ESP8266
- Componentes electrónicos pasivos y activos.

3.1.7. Técnicas de procesamiento de datos

El procesamiento de datos se basa en la diferencia del valor reales leídos por los sensores y el valor deseado, estas diferencias denominada error son recolectados y sometidos a operaciones estadísticas que tienen relación con la diferencia de error, media y varianza.

3.1.7.1. Diferencia de error

La diferencia de error es una forma de representar, de que tanto se aproxima las condiciones climáticas dentro de invernadero a las condiciones naturales que debe de tener para la planta. Se realizaron pruebas de diferencia con la siguiente ecuación.

$$Error = VT - VE \quad (1)$$

Donde:

VT: Valor teórico.

VE: Valor experimental.

3.1.7.2. Media de error

Definido por la siguiente ecuación:

$$x = \frac{\sum \text{muestras}}{n} \quad (2)$$

Donde:

x : Media.

n : Número de muestras.

3.1.7.3. Varianza

Definido por la siguiente ecuación:

$$\sigma_x^2 = \sum (xi - \bar{x})^2 / n \quad (3)$$



Donde:

σ_x^2 : Varianza.

x_i : I-enésima diferencia de error.

\bar{x} : media.

n : Número de muestras.

3.2. DESARROLLO DEL PROTOTIPO

Para cumplir con los requerimientos y características con todos los objetivos de la investigación, se buscó los sensores y actuadores que tengan las características necesarias para detectar y variar una determinada variable climática. Con toda la información recopilada de los diferentes estudios realizados en este campo, se procedió a la selección de sensores y actuadores.

3.2.1. Selección de los elementos de control y actuadores

La transformación de señales climática naturales a señales que pueden ser procesadas, es la principal función que el sensor debe de realizar. El microcontrolador que procesará dicha información deberá de soportar todo el tráfico generado por los sensores y contar con la interfaz de comunicación con dichos sensores.

3.2.1.1. Microcontrolador

El controlador deberá de ser capaz de comunicarse y establecer la comunicación con los diferentes sensores, también deberá de procesar los datos en un tiempo acotado, de lo contrario el cultivo podría sufrir los cambios bruscos externos, este microcontrolador deberá de ser también el más utilizado por los diferentes equipos tecnológicos del ahora.



Se seleccionó el STM32f103vt de la empresa STmicroelectronics. Su arquitectura se basa en; “ordenador con conjuntos de instrucciones reducidos” (Wikipedia la enciclopedia , 2022) (ARM). Su ancho de procesamiento de palabra es de 32bits con una frecuencia máxima de trabajo de 72Mhz, algunas funciones que utilizaremos de este microcontrolador son:

- 256 a 512 Kbyte de memoria flash
- Hasta 64 Kbyte de memoria SRAM
- 3x12bits, de 1Msps del conversor analógico digital
- DMA(acceso a memoria directa) para liberar trabajo del CPU
- Hasta 11 temporizadores para la generación del PWM
- Hasta 13 interfaces de comunicación (utilizaremos principalmente el USART).

3.2.1.2. Sensor de temperatura y humedad del aire

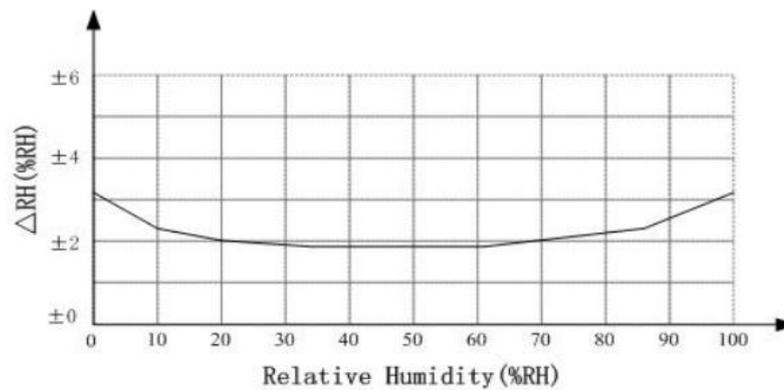
La variación de estos elementos influye mucho en el desarrollo de la planta, “un aire muy húmedo contribuye directamente a la propagación de enfermedades de las raíces y las hojas” también un aire muy seco causara más tiempo en obtener el tamaño adecuado para la venta del producto (PROMIX, 2022), todo esto está relacionado con la temperatura del invernadero, para ello utilizaremos el sensor AM2320 de la empresa AOSONG Electronics, este sensor cuenta con interfaz de comunicación serial I2C y rs232 protocolos estándar de comunicación, las características se muestra en la tabla 2.

Tabla 2: Tabla de rendimiento de humedad relativa

parámetro	condición	min	típico	Max	unid
Resolución			0.1		%RH
rango		0		99.9	%RH
precisión	25°C		+/-3		%RH
Repetitividad			+/-0.1		%RH
intercambiabilidad	Completamente intercambiable				
Tiempo respuesta			<5		S

Fuente: (AOSONG MICROELECTRONICS, 2020, pág. 2)

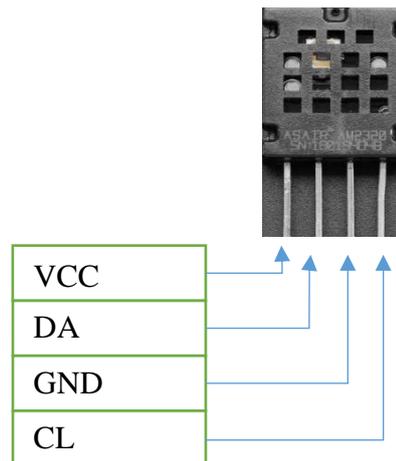
Figura 13: Error máximo de humedad relativa.



Fuente: (AOSONG MICROELECTRONICS, 2020, pág. 2)

Las características de sus pines se muestran en la figura 14.

Figura 14: Configuración de pines AM2320



Elaboración propia

3.2.1.3. Sensor de humedad del suelo

Esta variable de vital importancia logra que las raíces tengan algún tipo de conexión con todos los nutrientes que posee el suelo, como también de sostener el peso de los frutos. Demasiada humedad provocaría la asfixia de las raíces y un poco humedad afectaría directamente al crecimiento y desarrollo del cultivo. El sensor que se utilizara en el proyecto se muestra en la figura 15, este sensor de bajo costo de la empresa OBSOIL está implementado con unos circuitos integrados que filtran y amplifican las señales de corriente producidos por sus electrodos, este sensor es de fácil instalación y es resistente a la corrosión.

Figura 15: Sensor de Humedad de suelo



Elaboración propia

Sus características del sensor son:

- Alta sensibilidad
- Voltaje de operación de 3.3v a 12v
- Modo de salida dual, salida digital y analógica
- Electrodo resistente a la corrosión

Para determinar la cantidad de agua presente en el suelo se aplicará las siguientes ecuaciones:

$$\text{lectura del electrodo puesto en agua} = \text{Contenido de agua al 100\%} \quad (4)$$

$$\text{lectura del electrodo puesto en suelo seco a } 100^{\circ}\text{c} = \text{contenido de agua } 0.1\% \quad (5)$$

La capacidad de campo, que se define como la capacidad de retener el agua por el suelo contra la fuerza de gravedad después de un riego, es utilizado para determinar la cantidad en porcentaje que el electrodo debe de medir. “Para un suelo arcilloso el contenido de agua debe de estar entre 25 a 35%” (PROAIN Tecnología agrícola, 2022).

3.2.1.4. Sensor de intensidad luminosa

Esta medida ayudara a determinar la cantidad de radiación que entra al invernadero y ayudara a elegir la cubierta que debe de tener, y la orientación del invernadero, la fuente de luz natural contiene diferentes longitudes de onda. “La radiación fotosintéticamente activa (RFA) es el tipo que favorece a la fotosíntesis” (INTAGRI, 2022), de acuerdo a la tabla 3 la longitudes de onda esta entre 400-700nm

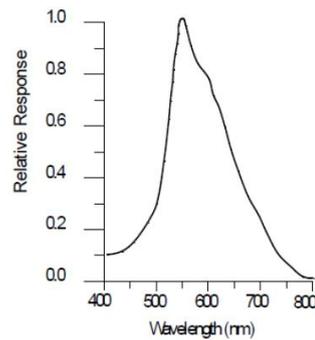
Tabla 3: Clasificación de la luz solar en función de la longitud de onda

UV	Visible	RFA	IR cercano	IR lejano	Solar Total
300-380nm	380-760nm	400-700	760-2500nm	2500-40000nm	300-2500nm

Fuente: (INTAGRI, 2022, pág. 1)

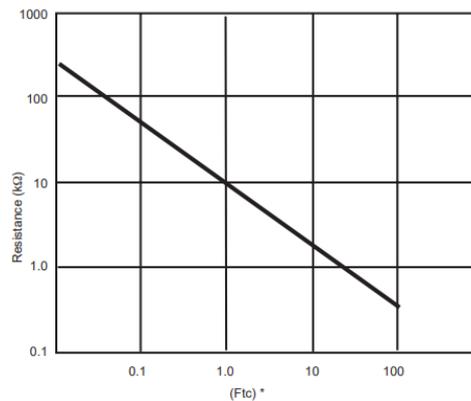
El resistor dependiente de la luz (LDR) se utilizará en este proyecto, debido a su respuesta al espectro luminoso, figura 16. Este sensor es de bajo costo y de fácil implementación, gracias a su función de resistencia respecto a la intensidad luminosa este sensor se comporta casi lineal como se muestra en la figura 17.

Figura 16: Respuesta espectral pico del LDR



Fuente: (LUNA OPTOELECTRONICS, 2016, pág. 6)

Figura 17: Resistencia como función de la iluminación. $1F_{tc} = 10.76$ lumen



Fuente: (LUNA OPTOELECTRONICS, 2016, pág. 5)

3.2.1.5. Módulo para IoT

Para enviar los datos censados de nuestro prototipo utilizamos el microcontrolador ESP8266MOD de bajo costo que integra un procesador Xtensa de 32 bits arquitectura RISC, que tiene integrada TCP/IP y el protocolo para WLAN MAC, con un generador de reloj de frecuencia 2.4 GHz para recibir y transmitir la señal RF. La empresa que fabrica este microcontrolador con memoria ROM separado es ESPRESSIF, en la figura se muestra un módulo que integra este microcontrolador.

Figura 18: Modulo ESP8266



Fuente: (ESPRESSIF, 2022, pág. 1)

Su kit de Software de Desarrollo (SDK) para el microcontrolador está disponible para los sistemas operativos Windows y Linux y una bastante información con aplicaciones con el entorno de programación Arduino.

3.2.1.6. Bomba de agua

Los más utilizados son a combustión y a electricidad, para nuestro prototipo que tiene un área pequeña utilizaremos el de electricidad de una potencia de 0.5 HP, esta electrobomba que cuenta con una turbina de latón como se muestra en la figura 19 y 20, para bombear el agua, puede proporcionar 30 litros por minuto.

Figura 19: Turbina de bombeo o turbina de vértice



Fuente: (PEDROLLO the spring ok life, 2021, pág. 6)

Figura 20: Estructura interna



Fuente: (PEDROLLO the spring ok life, 2021, pág. 3)

Para el control de este motor, utilizaremos un sistema electrónico con control simple implementado con elementos electrónicos pasivos y relé que funcionaran como interruptores electrónicos controlados por alguna señal lógica.

3.2.1.7. Ventilador

Uno de los inconvenientes producidos por aislar el interior del invernadero es la falta de dióxido de carbono (CO_2), para sobre llevar el problema existe la ventilación natural y la forzada. En la ventilación natural se requiere de alguna forma exponer el interior al ambiente natural, esta acción puede ocasionar la transmisión de enfermedades por insectos al cultivo, la más segura es la ventilación forzada electricidad, que consiste en introducir aire filtrada por algún sistema mecánico impulsado por electricidad, además de introducir aire limpia sin insectos, también ayuda a distribuir uniformemente la humedad por todo el interior del invernadero.

El ventilador que instalamos requiere de 24 voltios de voltaje continuo, la cantidad de flujo de aire es controlado por una señal PWM generado por el microcontrolador, en la figura 21, se muestra el ventilador utilizado en este proyecto.

Figura 21: Ventilador metálico



Fuente: (TRU TECHNOLOGIES (SINGAPORE) PTE LTD, 2022, pág. 1)

Sus características principales son:

- Potencia de 77 Watt
- velocidad 2700 rpm
- Flujo de aire, 1010m³/Hr
- Ruido (dB), de 61

3.2.1.8. Calefactor

Las alturas, las heladas y los cambios bruscos de clima obligan a instalar elementos que generen energía térmica para proteger al cultivo de estos cambios irregulares. El más habitual es el calefactor o estufa resistiva que transforma energía eléctrica a térmica, por el principio del efecto Joule para nuestro proyecto que no tiene mucha área a calentar, utilizaremos la estufa de Cuarzo Imaco QH800 que cuenta con tres niveles de potencia.

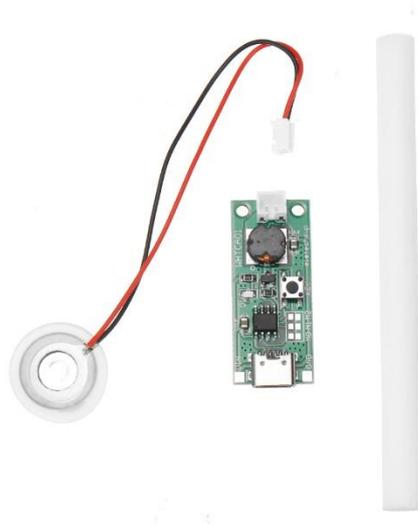
El elemento principal del calefactor lo constituye un circuito resistivo de muy o casi cero de reactancia inductiva, esta característica facilita el control de potencia con

dispositivos semiconductores controlados por compuerta (GATE), como el Mosfet. Para nuestro proyecto circuito el circuito de control esta implementado con este elemento y opto acopladores de protección para el microcontrolador.

3.2.1.9. Nebulizador ultrasónico

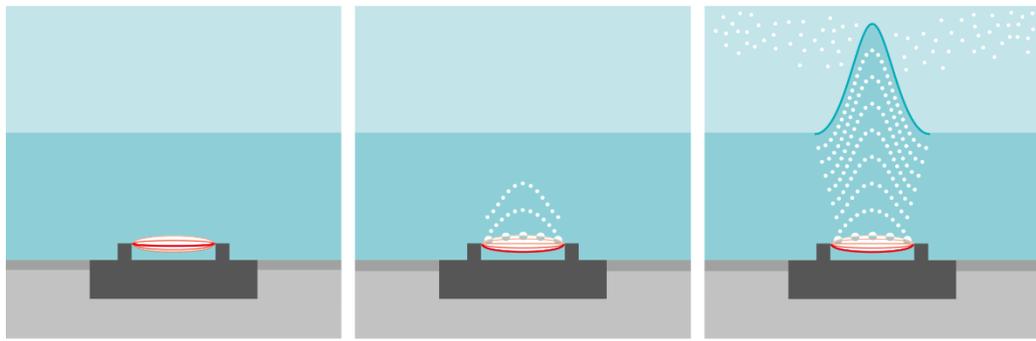
Para evitar el excesivo consumo de nutrientes debido a una baja humedad del aire, en la figura 24 se muestra humificador ultrasónico que gracias al efecto piezoeléctrico el agua es vaporizada generando niebla, esto es posible por la alta frecuencia de oscilación de unos 1.6MHz que experimenta el agua, en la figura 22 se muestra el principio de funcionamiento. Este nebulizador ultrasónico nos proporcionará la humedad faltante de nuestro invernadero

Figura 22: Nebulizador ultrasónico



Elaboración propia

Figura 23: Proceso de formación de niebla por oscilación



Fuente: (STULZ, 2022, pág. 1)

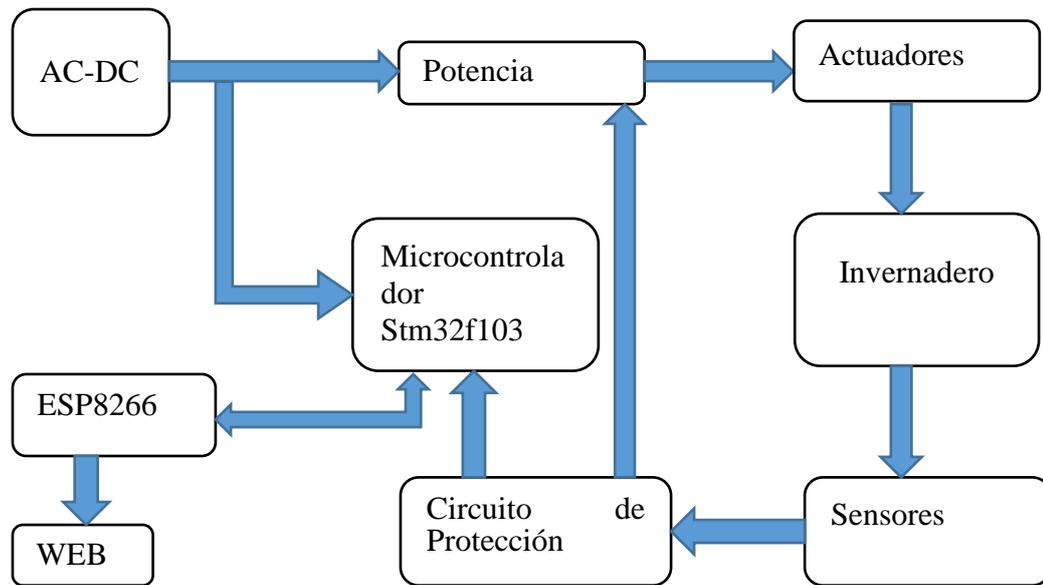
Este pequeño nebulizador cuenta con:

- Una tarjeta de control, que genera el voltaje y la alta frecuencia
- El piezoeléctrico, que vibrara a 1.6MHz para evaporar el agua
- Y una fuente de alimentación de 5vdc

3.2.2. Sistema de control

Para poder diseñar nuestro sistema de control necesitamos representar gráficamente todo nuestro sistema en bloque, estos bloques representan funciones que dependen de algún tipo de variable e interconectados por una secuencia de etapas de amplificación, retroalimentación y compensación que debe de pasar la señal, y así poder ser controlados. También se incluyen las etapas de potencia, rectificación y filtración que alimentaran a los distintos sensores y actuadores del proyecto. En la figura 24 siguiente se muestra las etapas de control electrónico en bloques de nuestro invernadero.

Figura 24: Diagrama en bloques del sistema a controlar



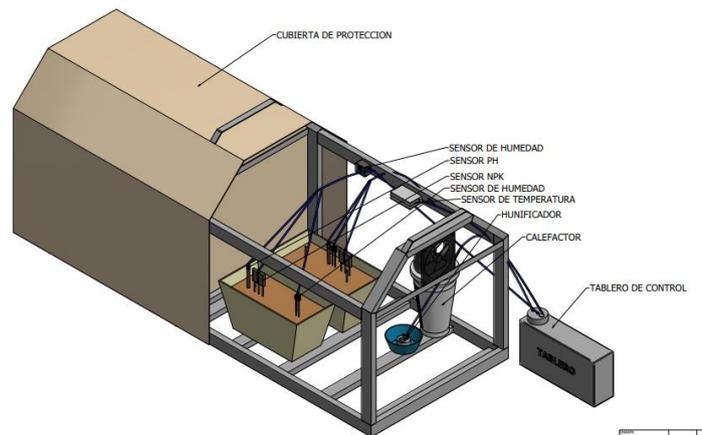
Elaboración propia

3.2.3. Estructura del invernadero

De los distintos tipos de invernaderos, el tipo capilla es la más ideal para nuestra investigación, por ser fácil en implementar los sistemas de ventilación, fácil construcción y de fácil conservación.

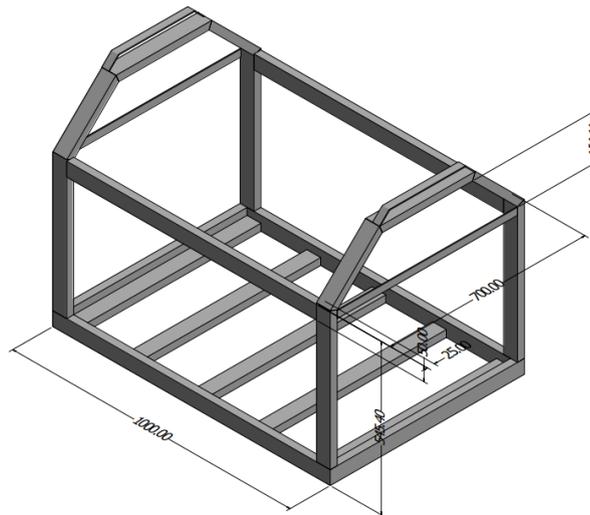
Para la construcción de invernadero se utilizó el aluminio como su soporte estructural por ser de bajo coste y de peso liviano. La forma geométrica con sus respectivas medidas, primeramente se realizaron con el software de Autodesk Inventor, la que nos permitió proyectar la forma y el aspecto que obtendrá el invernadero antes de llevarlo a la construcción, en la figura 25 y 26 se muestra la presentación final que tendrá el prototipo y la conexión de los sensores disponibles en el mercado, las partes de la maketa está compuesta por: la estructura tipo capilla, que protegerá al huerto de exterior; y la estructura de control, donde se procesara toda la información de los sensores instalados en el invernadero para luego poder enviarlo a la WEB.

Figura 25: Aspecto y sensores del prototipo



Elaboración propia

Figura 26: Medidas en milímetros de la maqueta



Elaboración propia

3.2.3.1. Instalación y ubicación de los sensores y actuadores en el prototipo

En el desarrollo del prototipo se tuvieron que agregar algunos elementos esenciales, para que pueda soportar la cubierta protectora, también se prefirió realizar la estructura, ser totalmente desarmable, también que tuvieron que quitar partes que no ayudaban a encerrar herméticamente por la cubierta protectora. En la figura 27 a la figura

28, se muestra vista general de la maqueta ensamblada con algunos sensores y actuadores instalados.

Figura 27: Vista general



Elaboración propia

Figura 28: Uniones, completamente desarmables.



Elaboración propia

Figura 29: Vista frontal con el tablero de control



Elaboración propia.

La instalación de los sensores se ubica en el lugar más cercano al cultivo, en las siguientes de las figuras 30 a la figura 31, se detallan lugar de instalación de los sensores

Figura 30: Ubicación el sensor LDR



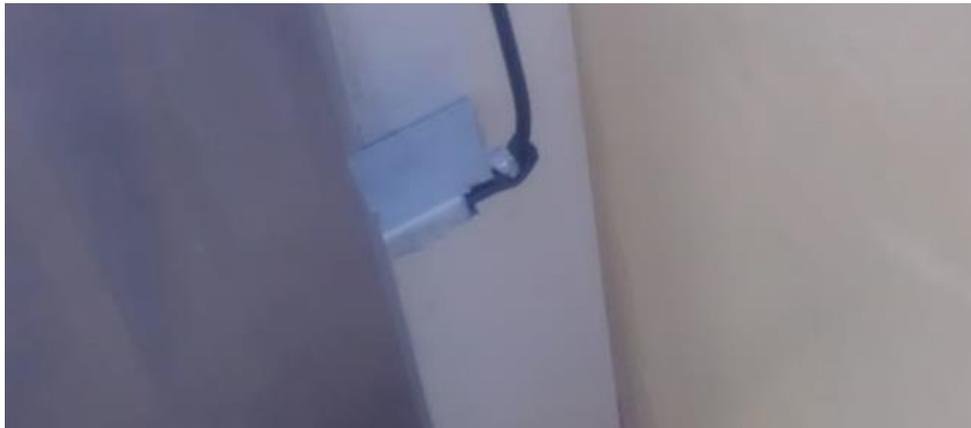
Elaboración propia.

Figura 31: Ubicación sensor de humedad de suelo.



Elaboración propia.

Figura 32: Ubicación del sensor de temperatura y humedad del aire.



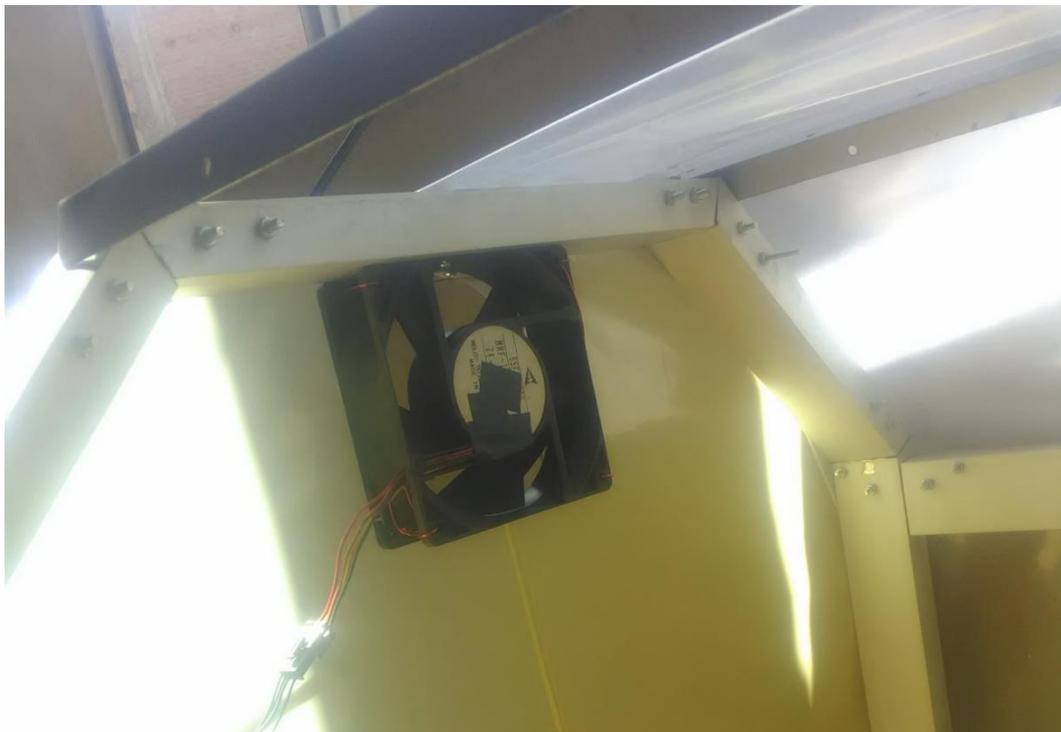
Elaboración propia.

Figura 33: Ubicación del calefactor y humidificador ultrasónico.



Elaboración propia.

Figura 34: Ubicación del ventilador.



Elaboración propia.

Figura 35: Tablero de control con el STM32f103vc.

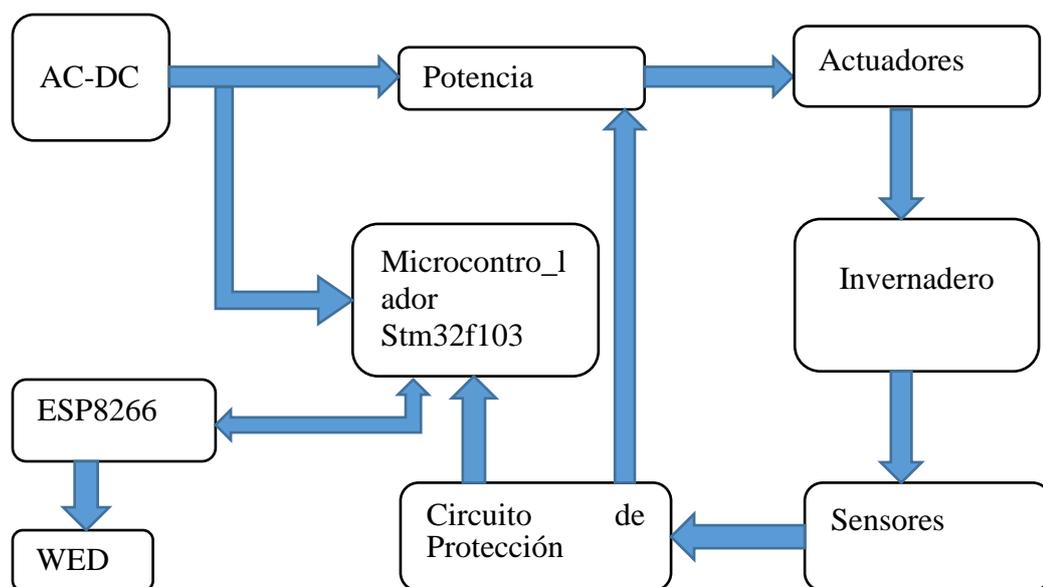


Elaboración propia.

3.2.4. Sistema de control

Para poder diseñar nuestro sistema de control necesitamos representar gráficamente todo nuestro sistema en bloque, estos bloques representan funciones que dependen de algún tipo de variable e interconectados por una secuencia de etapas de amplificación, retroalimentación y compensación que debe de pasar la señal, y así poder ser controlados. También se incluyen las etapas de potencia, rectificación y filtración que alimentaran a los distintos sensores y actuadores del proyecto. En la figura 36 siguiente se muestra las etapas de control electrónico en bloques de nuestro invernadero.

Figura 36: Diagrama en bloques del sistema a controlar



Elaboración propia

3.2.4.1. Elección de estrategia de control

Esta elección es mayormente dependiente de la respuesta en el tiempo que el sistema debe de tener, para sistemas de automatización y procesos industriales el tiempo máximo de respuesta que debe de tener es menor a 2 milisegundos. Para nuestro propósito



de tesis no se requiere respuestas rápidas, utilizando un controlador proporcional será casi suficiente para estabilizar el sistema en la mayoría de los bloques del sistema, o en todo caso se implementará un control PID si lo necesite.

3.2.4.1.1. Fuente de alimentación AC-DC

El bloque AC-DC cuenta con dos salidas de alimentación de 24Vdc y 3Vdc que se encargan de mantener una constante fuente de voltaje, que alimentara a la etapa de potencia del sistema, y al microcontrolador.

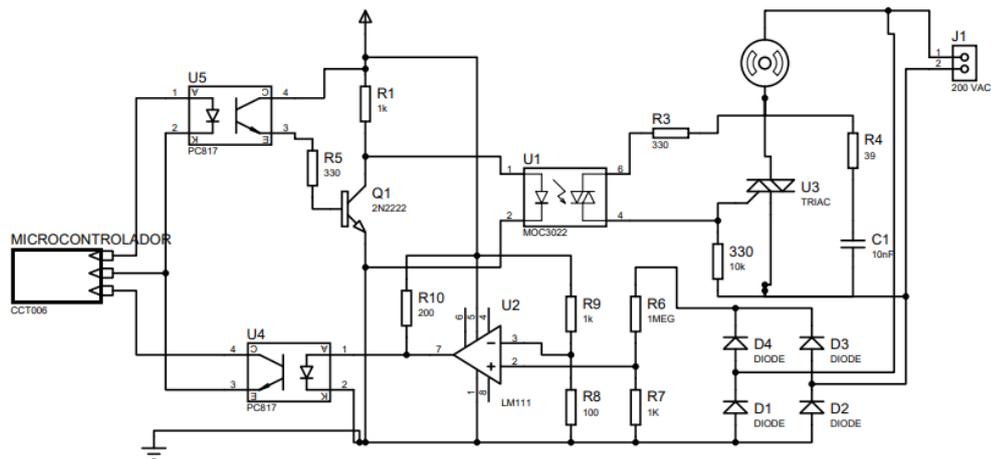
Este bloque es totalmente independiente del microcontrolador, por lo tanto no será controlado.

3.2.4.1.2. Potencia

Los circuitos que conforman este bloque, es de acuerdo a los siguientes actuadores:

- Electrobomba. Para que opere se requiere fuente de voltaje alterna y componentes electrónicos analógicos, también las señales digitales deben de estar aislados con opto acopladores, el circuito electrónico se muestra en la figura 37, con respectivas señales de control y la entrada de alimentación.

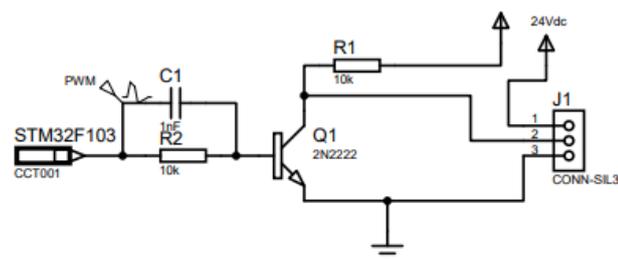
Figura 37: Circuito de potencia del electrobomba



Elaboración propia

- Ventilador. Este actuador cuenta con una etapa de potencia incorporada, solo se requiere aislar y proteger la señal digital de control. En la figura 38 se muestra el circuito implementado para el ventilador donde J1 es terminal de conexión hacia el ventilador y Stm32f103 es el que genera la señal.

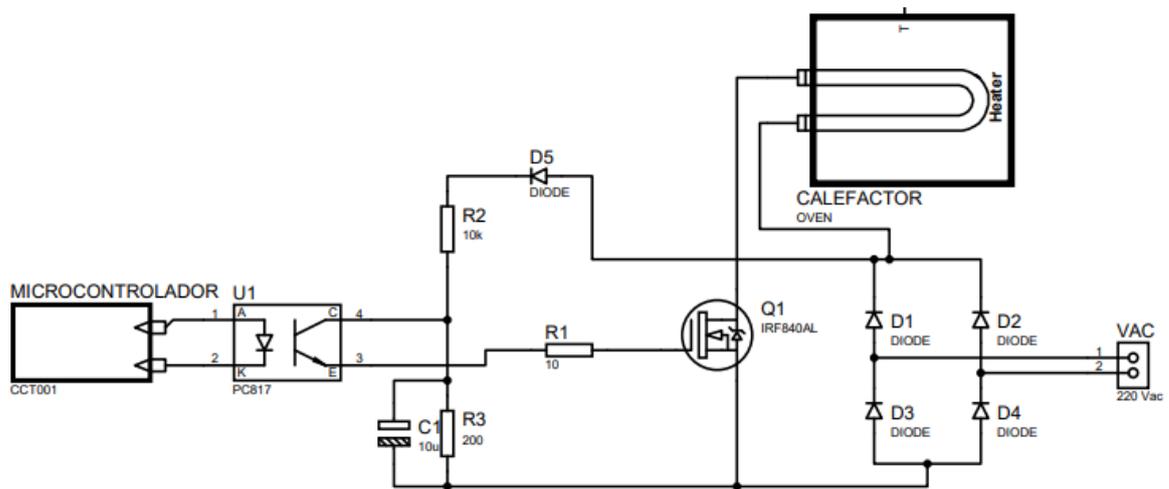
Figura 38: Circuito de control para la señal PWM



Elaboración propia

- Calefactor. Este transductor tiene muy baja reactancia inductiva, esto facilita el control de potencia al calefactor con semiconductores de una sola dirección como los transistores de potencia, en la figura 39 se muestra el diseño para nuestro transductor. En donde el voltaje AC de polariza para que el transistor mosfet pueda operar como interruptor

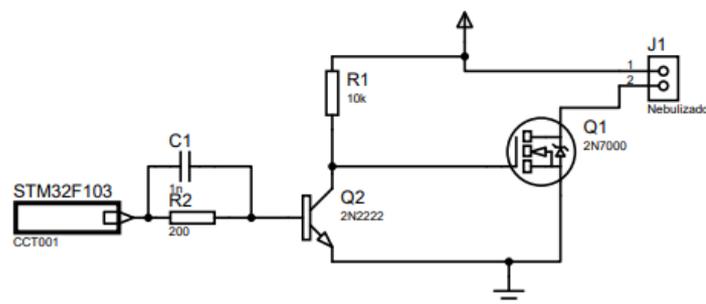
Figura 39: Circuito de potencia del calefactor



Elaboración propia

- Nebulizador. La potencia de este transductor es baja, solo se requiere que el circuito electrónico libere toda la corriente que necesite el nebulizador

Figura 40: Circuito para el nebulizador



Elaboración propia

3.2.4.1.3. Circuito de protección

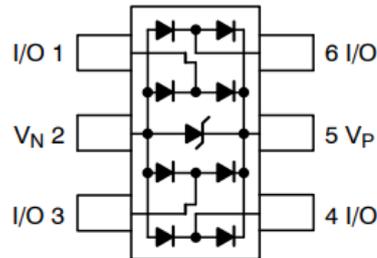
En este bloque mayormente lo conforman componentes que aíslan las señales provenientes de los sensores hacia el microcontrolador, en nuestro circuito de protección incorporamos chip de Descarga Electroestática (ESD) SRV05-4 de la empresa ON Semiconductor, con una operación en alta velocidad. Algunas de sus características más importantes son:

- IEC61000-4-2(ESD) +- 15Kv(aire), +-8Kv (contacto)

- IEC61000-4-4 (EFT) 40A

Este chip es diseñado especialmente para señales digitales su diagrama se muestra en la figura 41.

Figura 41: Configuración de pines y esquemático

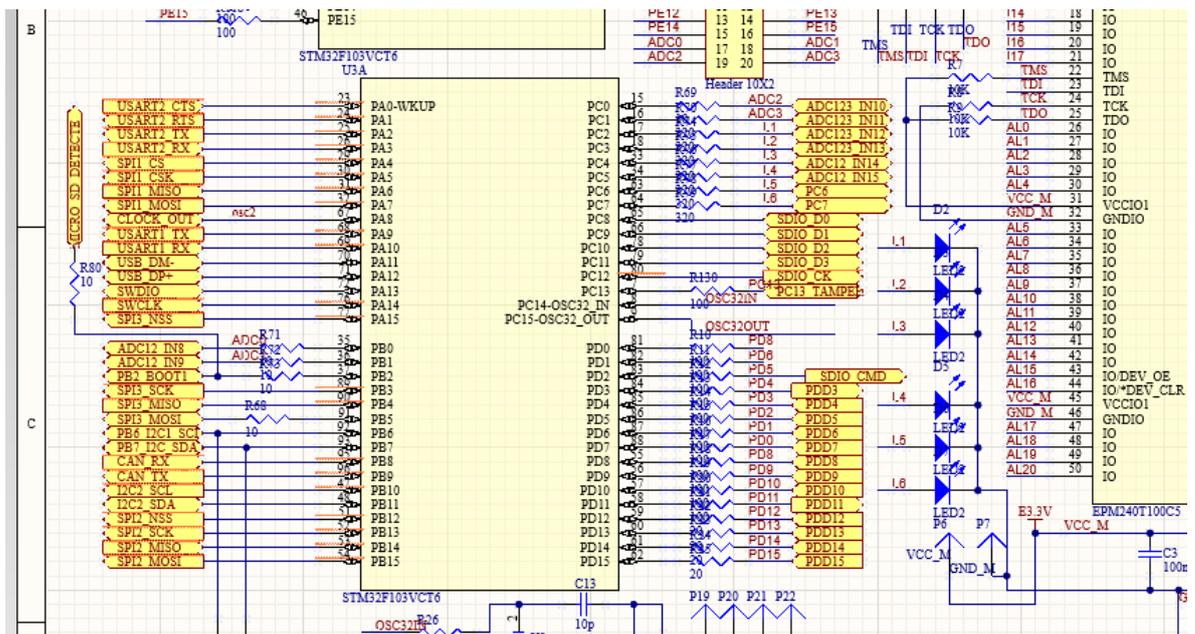


Fuente: (ON Semiconductor, 2017, pág. 1)

3.2.4.1.4. Stm32f103vc

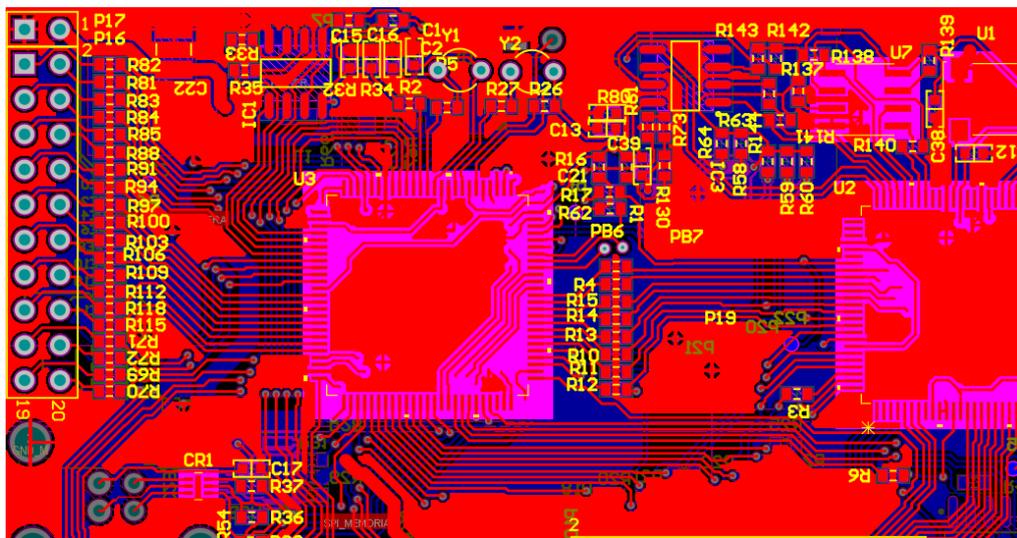
El componente más principal de nuestro proyecto es este chip de la empresa STmicroelectronics, su diagrama de conexiones y esquemático se realizó en el CAD Altium Designer, especializado en diseño de circuitos de alta densidad, sus componentes pasivos y reguladores de voltaje se muestra en la figura 32 y figura 33, también se incorpora una memoria flash de la empresa WINBOND, un Dispositivos lógicos programables complejos (CPLD) de la antigua marca de ALTERA, para futuras pruebas.

Figura 42: Esquemático stm32f103vc



Elaboración propia

Figura 43: Ruteo de conexiones Stm32f103vc



Elaboración propia

3.2.5. Fundamentación para la programación por riesgo convencional

Para realizar una correcta programación del microcontrolador, se requiere definir las variables y constantes que intervienen de acuerdo a referencias de algunos criterios de campo. El riesgo convencional será muestra base para nuestro sistema de algoritmos que



procesara el microcontrolador para sostener un clima dentro del invernadero, los parámetros base se detallan a continuación.

3.2.6. Resumen del procedimiento

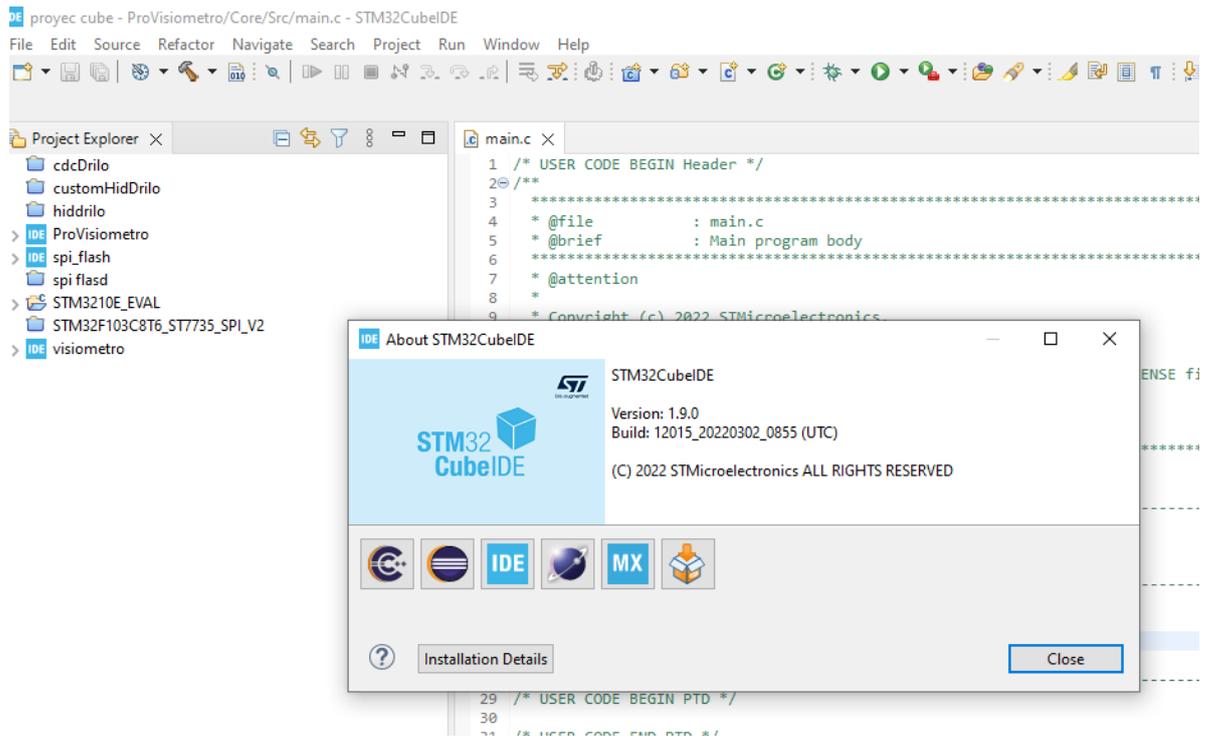
Iniciamos con el área de nuestro prototipo, que tiene 100cmx70cm de área con una altura de 65cm, como nuestra primera constante, seguidamente se extrae mediante un sensor nuestra primera variable importante del suelo, la humedad. Variable dependiente de la cantidad de agua existente en el suelo, mediante una electrobomba y electroválvulas controladas por nuestra tarjeta de control, suministran la cantidad necesaria. Seguidamente se extrae mediante un sensor, la temperatura del ambiente de nuestro prototipo, variable dependiente de la cantidad de energía presente en nuestro invernadero, es controlada por nuestra tarjeta de control, mediante dos actuadores; el ventilador, que distribuirá por toda el área la energía concentrada en zonas muy expuestas a la radiación solar o al calefactor eléctrico. Si, la temperatura cae por debajo del límite inferior, nuestro sistema control, activara el calefactor eléctrico, que suministrara el calor faltante dentro del invernadero. Una vez terminada este proceso, seguidamente se extrae la humedad del aire, variable importante por causas mencionadas en capítulos anteriores, variable dependiente por la cantidad de evaporación del agua, su control se realizará mediante el ventilador; para la extracción del exceso de humedad y el unificador para compensar lo faltante todos controlados por la tarjeta de control.

3.2.7. Programación del microcontrolador

El conjunto de instrucciones que ejecutara el microcontrolador, fue desarrollado en el lenguaje C/C++, lenguaje por defecto, soportado por el entorno de programación de STmicroelectronics el “STM32CubeID”, mostrado en la figura 44. Tiene la capacidad de simplificar la configuración y administrar los distintos recursos, de hardware también

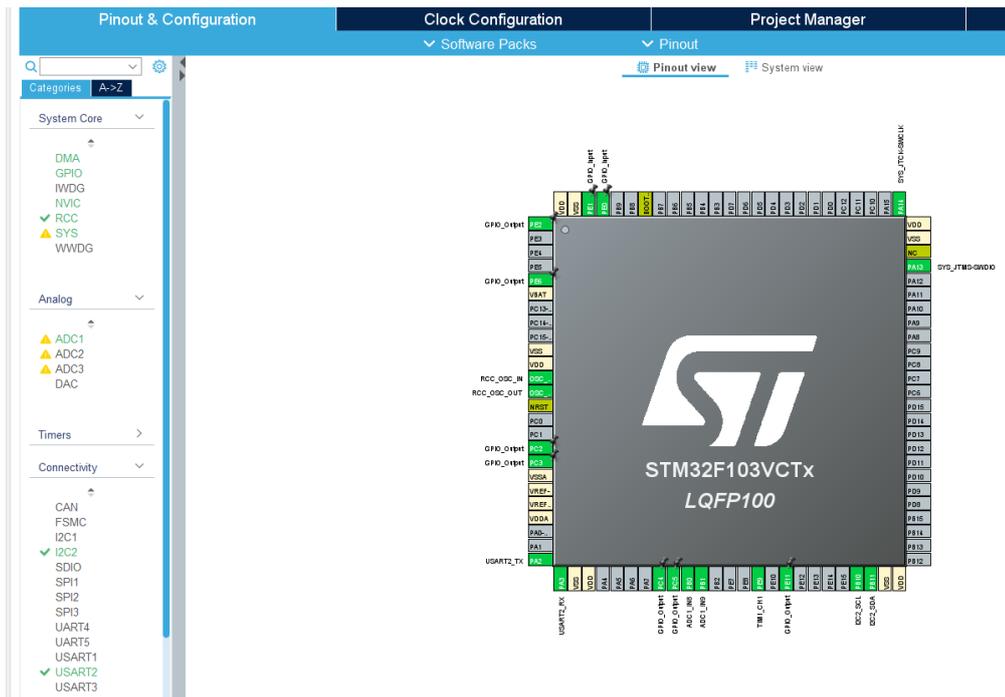
cuenta con un diagrama de interconexión del reloj oscilador que alimenta a todos los periféricos, en las figuras 45 y figura 46 se muestran una parte de la configuración para nuestro proyecto.

Figura 44: Entorno de programación STMicroelectronics



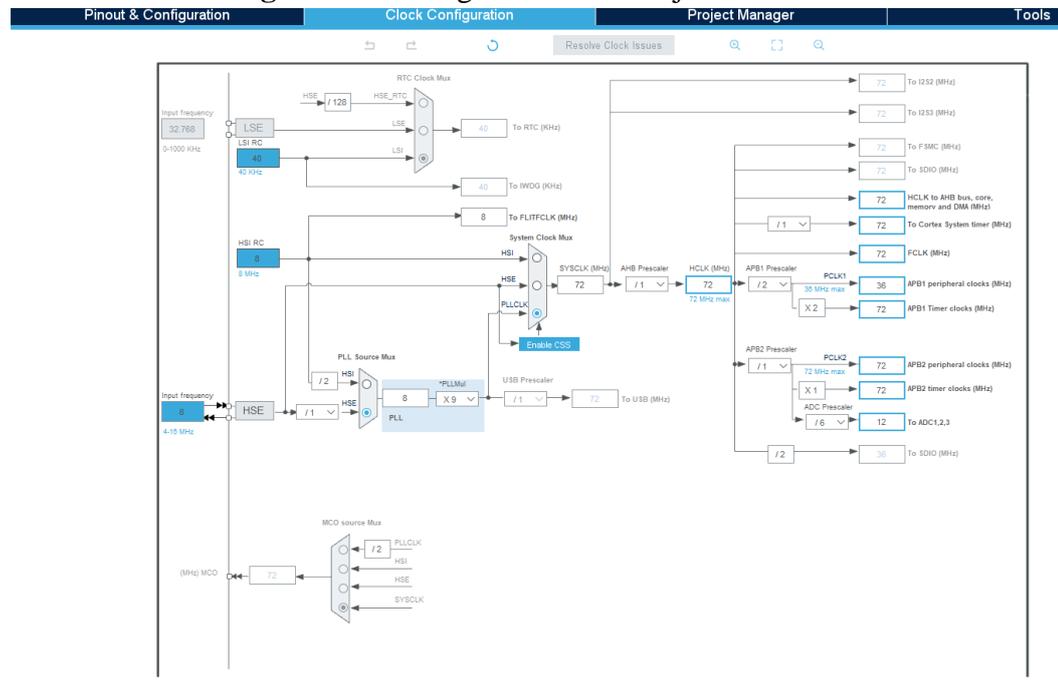
Elaboración propia.

Figura 45: Entorno de configuración de periféricos del MCU



Elaboración propia.

Figura 46: Configuración del reloj del sistema.

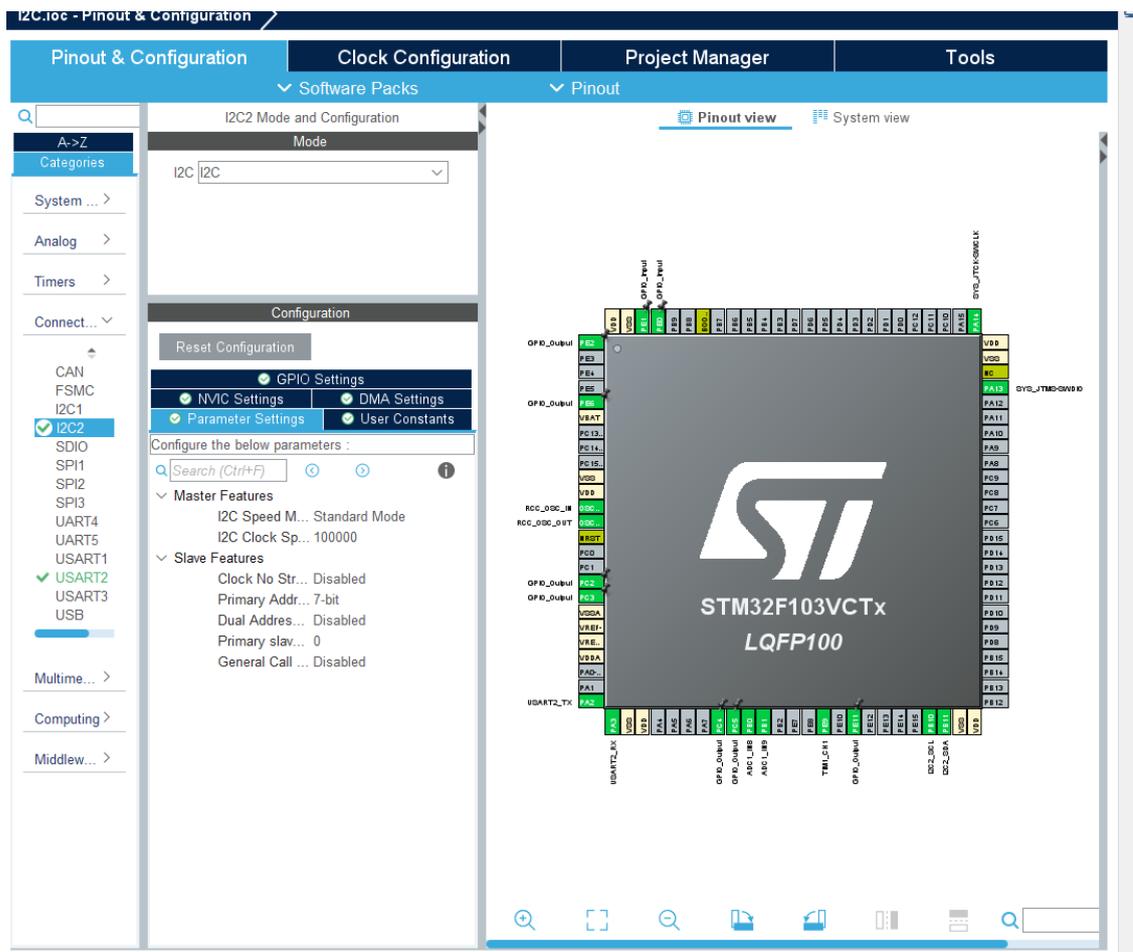


Elaboración propia.

3.2.7.1. Código C/C++ sensor humedad y temperatura del aire

Para controlar este sensor, se utilizar los periféricos de conectividad y activar los el I2C como se muestra en la figura 47, este periférico consume los pines PB10 y PB11 del microcontrolador, también se requiere configurar los bloques internos que controlan al I2C, como la velocidad de transferencia que puede soportar el sensor.

Figura 47: Habilitación y configuración del I2C



Elaboración Propia.

El código C/C++ de control para este sensor es; primero crear un objeto de tipo manipulable de tipo sensor Am3220 (Am2320_HandleTypeDef) con todas las propiedades que el sensor tenga como su dirección de esclavo sensor (#define AM2320_ADDRESS 0x5C<<1 // = 0XB8), y las demás funciones son para la inicialización

y lectura de la temperatura y humedad del aire, en la figura 38 se muestra la declaraciones y funciones para el sensor, como también las variables temperatura, humidity ambas de tipo float.

Figura 48: Funciones para el sensor AM2322.

```
142 MA_USART12_USART_Init();
143 MX_I2C2_Init();
144 MX_ADC1_Init();
145 MX_TIM1_Init();
146 /* USER CODE BEGIN 2 */
147 HAL_GPIO_WritePin(GPIOC, GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5, GPIO_PIN_SET);
148 HAL_Delay(1000);
149 HAL_GPIO_WritePin(GPIOC, GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5, GPIO_PIN_RESET);
150 Am2320_HandleTypeDef Am2320_2;
151 Am2320_2 = am2320_Init(&hi2c2, AM2320_ADDRESS);
152 float temperature, humidity;
153 double lux, res_LDR;
154 int luxInt;
155 int tm, hu;
156 HAL_GPIO_WritePin(GPIOE, GPIO_PIN_11, GPIO_PIN_SET);
157 HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
158 TIM1->CCR1=1;
159 /* USER CODE END 2 */
160
161 /* Infinite loop */
162 /* USER CODE BEGIN WHILE */
163 while (1)
164 {
165     /* USER CODE END WHILE */
166
167     /* USER CODE BEGIN 3 */
168     StatusADC1 = HAL_ADC_Start_DMA(&hadc1, ad, 2);
169     am2320_GetTemperatureAndHumidity(&Am2320_2, &temperature, &humidity);
170     tm=(int)temperature;
171     hu=(int)humidity;
172     adin_HS=((4095-((int)ad[0]))*100)/2495;//humedad suelo
173     i_LDR=(4096-ad[1])/1;//hallando la corriente del LDR. 4095:=3.2V, 1k
174     res_LDR=((ad[1]*1000)/i_LDR);//hallando la resistencia del LDR
175     lux=pow((res_LDR/5000.0),1.6);
176     lux=(10/lux);
177     luxInt=(int)(lux);
178     /*funciones para testeo de variables//
179     //my_printf("=====\r\n");
180     //my_printf("Temperature: %.1f°C\r\n", temperature);
181     //my_printf("Humidity: %.1f%\r\n", humidity);
182     //HAL_Delay(100);
183     }
```

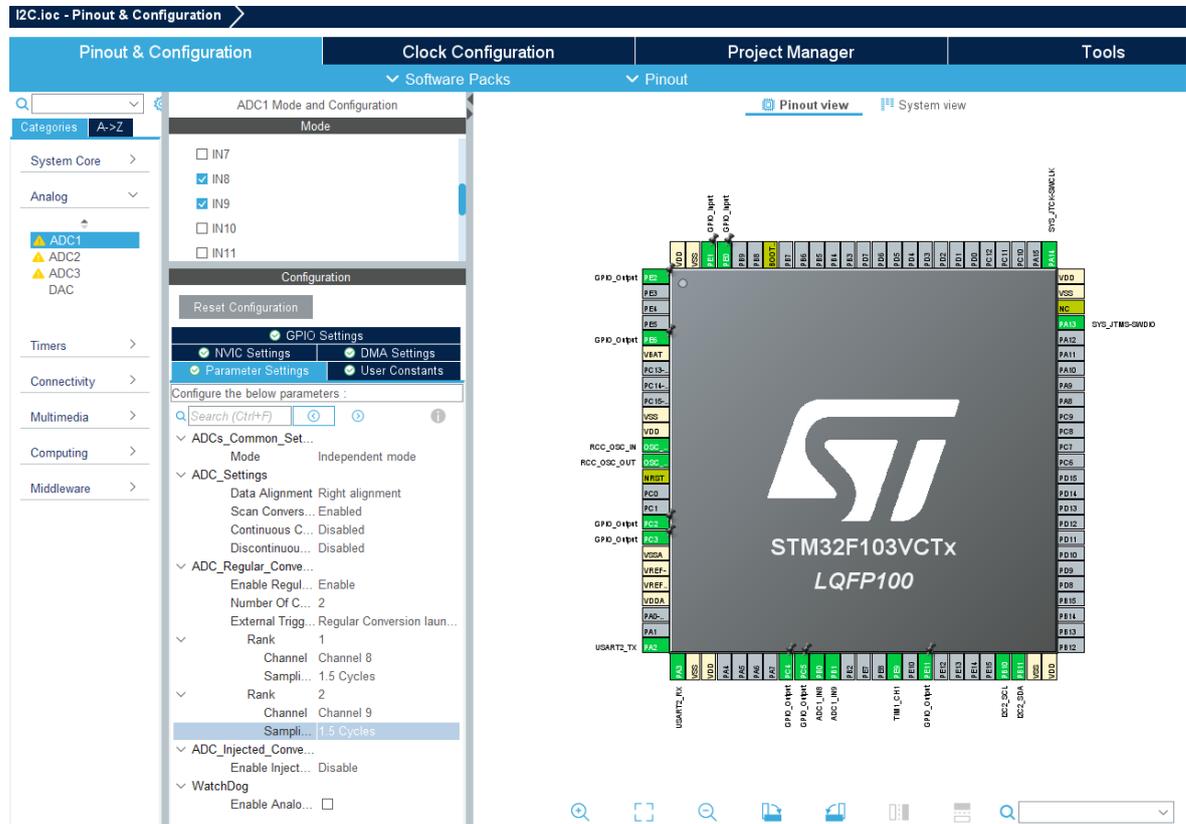
Elaboración propia.

3.2.7.2. Código sensor humedad del suelo

Este sensor es analógico, para ello utilizaremos el periférico ADC1 (Convertidor de Analógico Digital), habilitamos y configurar sus funciones como de muestra en la figura 49, la velocidad máxima de conversión de la señal a digital es de 1 micro segundo, el ADC1 es de 12 bit de resolución con 14 ciclos de reloj, entonces la frecuencia máxima que debe de entrar al periférico ADC1 es 14Mhz, en la figura 50 se muestra la frecuencia de entrada al ADC1 igual a 12Mhz.

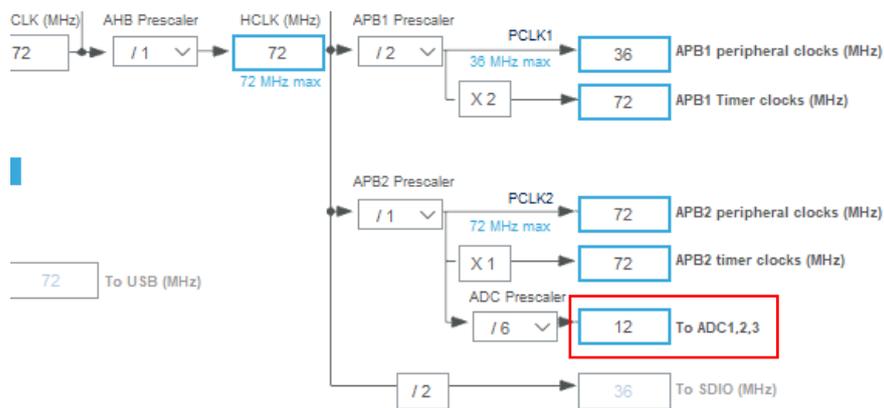
Los pines que se configuraran como entradas analógicas son; PB0 y PB1, nombrados como IN8, IN9 en el lenguaje C/C++.

Figura 49: Configuración del ADC1



Elaboración Propia

Figura 50: Configuración de la frecuencia del ADC1.



Elaboración propia

El código C para extraer la señal analógica se muestra en la figura 51, donde se declara una variable Array de rango 3 de 32 bits (uint32_t ad [2] ;) que almacenaran la conversión. Para iniciar una conversión nos apoyamos en la función implementada por STmicroelectronics, que solo se requiere enviar la dirección de memoria de la variable ad [2] (StatusADC1 = HAL_ADC_Start_DMA (&hadc1, ad, 2) (STMicroelectronic, 2016)), que está en la capa de abstracción de Hardware (HAL). Con el valor obtenido se procesa con los parámetros del sensor para obtener en porcentaje de 0% para un suelo seco y 100% para el suelo de pura agua, 2495 es el valor obtenido por el sensor en agua y 4095 para suelo seco, con unas operaciones matemáticas implementadas en C/C++ mostrado en la figura 52, se obtiene el nuevo dato que será guardado en la variable declarada adin_HS (adin_HS=((4095-((int)ad[0]))*100)/2495; //humedad suelo), para luego ser envía al exterior por el puerto UART.

Figura 51: Declaración de variables ad [2].

```
5 int main(void)
6 {
7     /* USER CODE BEGIN 1 */
8     HAL_StatusTypeDef StatusUart, StatusADC1;
9     uint8_t rxBuffer[]={ "NTA015SETB036SETC035SETD01\0xxx" }; //
10
11     uint8_t sub_rxBuffer[]={ "\0\0\0\0\0\0\0\0" };
12     uint32_t ad[2];
13     int adin_HS, i_LDR; //HS: humedad suelo, LX: lux, corriente del LDR
14                               //resistencia del LDR
15
16     ad[0]=23;
17     ad[1]=111;
18     int SETA=15;
19     int SETB=36;
20     int SETC=35;
21     int SETD=1;
22     /* USER CODE END 1 */
23
24     /* MCU Configuration-----
```

Elaboración propia.

Figura 52: Guardando el resultado en la variable adin_HS.

```
StatusADC1 = HAL_ADC_Start_DMA(&hadc1, ad, 2);
am2320_GetTemperatureAndHumidity(&Am2320_2, &temperature, &humidity);
tm=(int)temperature;
hu=(int)humidity;
adin_HS=((4095-((int)ad[0]))*100)/2495;//humedad suelo
i_LDR=(4096-ad[1])/1;//hallando la corriente del LDR. 4095:=3.2V, 1k
res_LDR=((ad[1]*1000)/i_LDR);//hallando la resistencia del LDR
lux=pow((res_LDR/5000.0),1.6);
lux=(10/lux);
luxInt=(int)(lux);
/*funciones para testeo de variables*/
//my_printf("=====\r\n");
//my_printf("Temperature: %.1f°C\r\n", temperature);
//my_printf("Humidity: %.1f%%\r\n", humidity);
//HAL_Delay(100);
my_printf("corriente %u\r\n",i_LDR );
my_printf("resistencia %f\r\n",res_LDR );
my_printf("VA%f\r\n",lux );
my_printf("adc1: %d %d %d\r\n", ad[0], ad[1], StatusADC1);
*/
if(tm>100){
my_printf("TA%3uHA%3uHS%03uIL%06uSO%02u\r\n",tm,hu,adin_HS, luxInt, SETD);
```

Elaboración propia.

3.2.7.3. Código sensor intensidad luminosa

En esta parte del código C/C++ utilizaremos la librería <math.h> del estándar ANSI C, porque el sensor LDR tiene respuesta logarítmica a la resistencia de corriente respecto a la intensidad de iluminación (LUX), propias de los elementos físicos de la

construcción del sensor, las ecuaciones (4) y (5) mencionas más abajo, son establecidas por el fabricante del sensor.

$$y = \log \left(\frac{R_{10}}{R_{100}} \right) \quad (6)$$

Donde R10 Y R100 resistencia de la celda entre 10 LUX A 100LUX.

$$R_l = R_0 * \left(\frac{L_0}{L} \right)^y \quad (7)$$

Donde Rl se determinar por nuestro código, R0 obtenido por el fabricante del sensor, L0 es la intensidad luminosa para la resistencia de la celda en R0, e y es determinado por la ecuación (6).

La implementación en el lenguaje C/C++, es como sigue; primeramente, incluimos la librería math en la cabecera (#include <math.h>) de nuestro programa y unas dos variables de punto flotante para precisión, como se muestra en la figura 43.

Figura 53: Incluyendo librería math.h y declaración de variables.

```
-----  
19 /* Includes -----  
20 #include "main.h"  
21  
22 /* Private includes -----  
23 /* USER CODE BEGIN Includes */  
24 #include <stdio.h>  
25 #include <stdlib.h>  
26 #include <stdarg.h>  
27 #include <string.h>  
28 #include <math.h>  
29 #include "am2320.h"  
30 #define TXBUFFERSIZE (COUNTOF(aTxBuffer) - 1)  
31 /* Size of Reception buffer */  
32 #define RXBUFFERSIZE TXBUFFERSIZE  
33 #define COUNTOF(__BUFFER__) (sizeof(__BUFFER__) / sizeof(*(__BUFFER__)))  
34 /* USER CODE END Includes */  
35 float temperature, humidity;  
36 double lux, res LDR;  
37 /* Private typedef -----  
38 /* USER CODE BEGIN PTD */  
~~
```

Elaboración propia

Para determinar la resistencia de la celda conectada a una resistencia de 1K en serie como se muestra en la figura 54, primeramente la corriente que circula por la celda,

para ello utilizamos el voltaje que detecta el pin del ADC de nuestro MCU por la siguiente ecuación (8).

$$i_{LDR} = (4096 - ad[1])/1k \quad (8)$$

Luego determinamos la resistencia que presenta la celda por la ecuación (9).

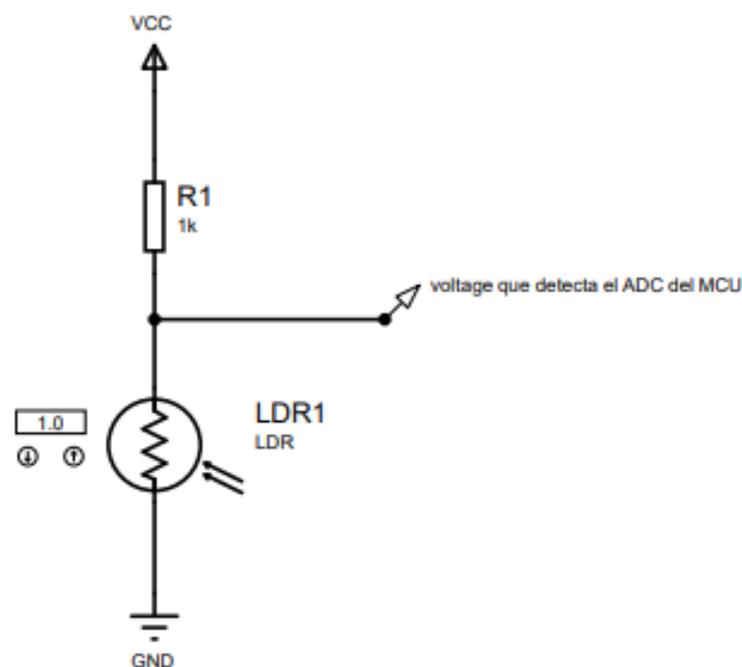
$$res_{LDR} = ((ad[1] * 1000)/i_{LDR}) \quad (9)$$

Finalmente para determinar la intensidad luminosa (LUX), aplicamos la ecuación (7) con $R_0 = 5000$, $y = 1.6$, $L_0 = 10$ nos queda:

$$lux = 10 / \left(\left(\frac{res_{LDR}}{5000} \right)^{1.6} \right) \quad (10)$$

Su implementación en C/C++ de nuestro proyecto se muestra en la figura 45, la variable `luxInt` es la que guardara y la que se enviara al exterior

Figura 54: Voltaje que transformado a digital por nuestro MCU



Elaboración propia.

Figura 55: Ecuaciones del LDR en C/C++.

```
EGIN 3 */
StatusADC1 = HAL_ADC_Start_DMA(&hadc1, ad, 2);
am2320_GetTemperatureAndHumidity(&Am2320_2, &temperature, &humidity);
tm=(int)temperature;
hu=(int)humidity;
adin HS=((4095-((int)ad[0]))*100)/2495;//humedad suelo
i_LDR=(4096-ad[1])/1;//hallando la corriente del LDR. 4095:=3.2V, 1k
res_LDR=((ad[1]*1000)/i_LDR);//hallando la resistencia del LDR
lux=pow((res_LDR/5000.0),1.6);//pow: potenciacion
lux=(10/lux);
luxInt=(int)(lux);
/*funciones para testeo de variables//
//my_printf("=====\r\n");
//my_printf("Temperature: %.1f°C\r\n", temperature);
//my_printf("Humidity: %.1f%%\r\n", humidity);
//HAL_Delay(100);
my_printf("corriente %u\r\n",i_LDR );
my_printf("resistencia %f\r\n",res_LDR );
```

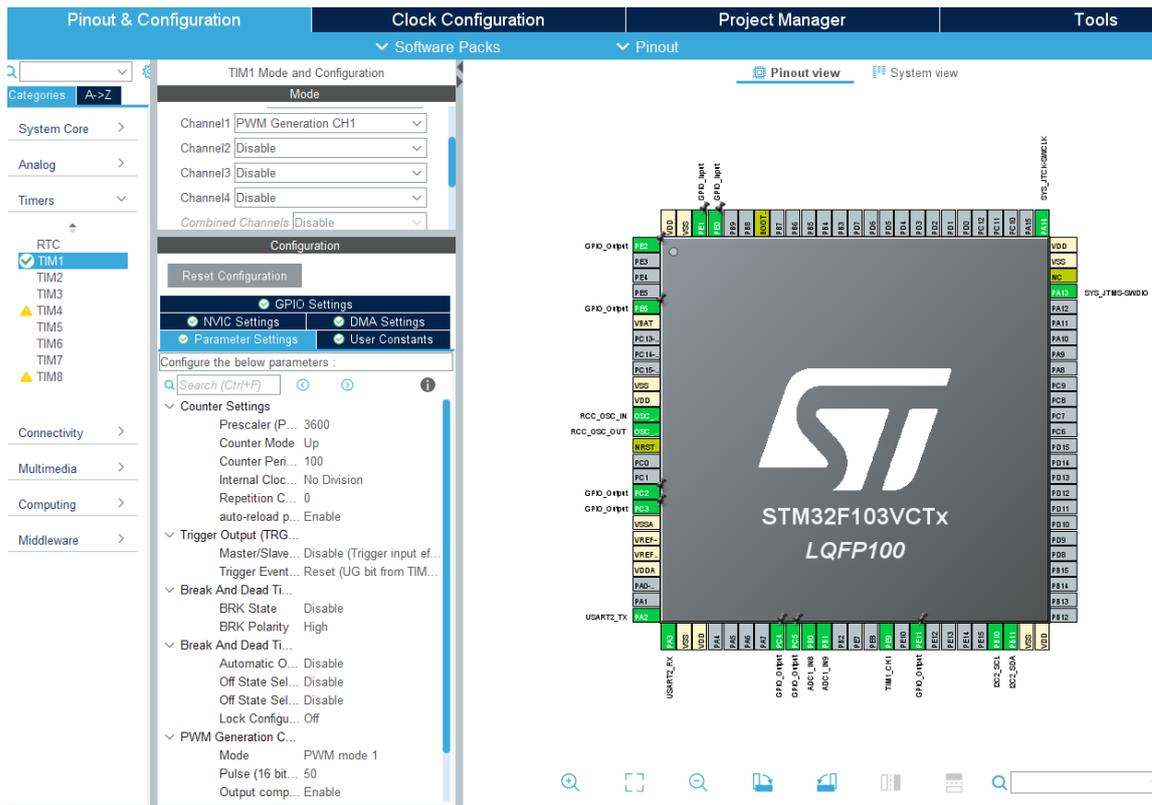
Elaboración Propia.

3.2.7.4. Código para actuadores

Para esta parte del código de control del invernadero, se utiliza la salida de control por Modulación por Ancho de Pulso (PWM) para los actuadores analógicos y el control ON/OFF, para las electroválvulas y relé que lleve el sistema.

Para crear el PWM, utilizamos el Periférico del TIM1 del MCU, que tiene como máximo de cuatro canales independientes para generar PWMs, en la figura 56 se muestra la habilitación y configuración del TIM1. Una vez inicializado el Periférico solo se requiere cargar el registro (TIM1->CCR1=45) de comparación para generar el pulso digital.

Figura 56: Habilitación del TIM1 y configuración del canal 1



Elaboración propia

De la figura 56 se configuran los parámetros principales como, el prescaler (que divide la frecuencia que ingresa al periférico en un factor establecido por el registro TIMx_PSC (STMicroelectronics, 2015, pág. 355)), el Periodo que tendrá nuestro PWM, es igual a 100 ciclos, y nuestro ancho de pulso pre establecido es 50, la frecuencia de nuestro PWM es:

$$f_{PWM} = (f_{mcu} / prescales) * 100 \quad (11)$$

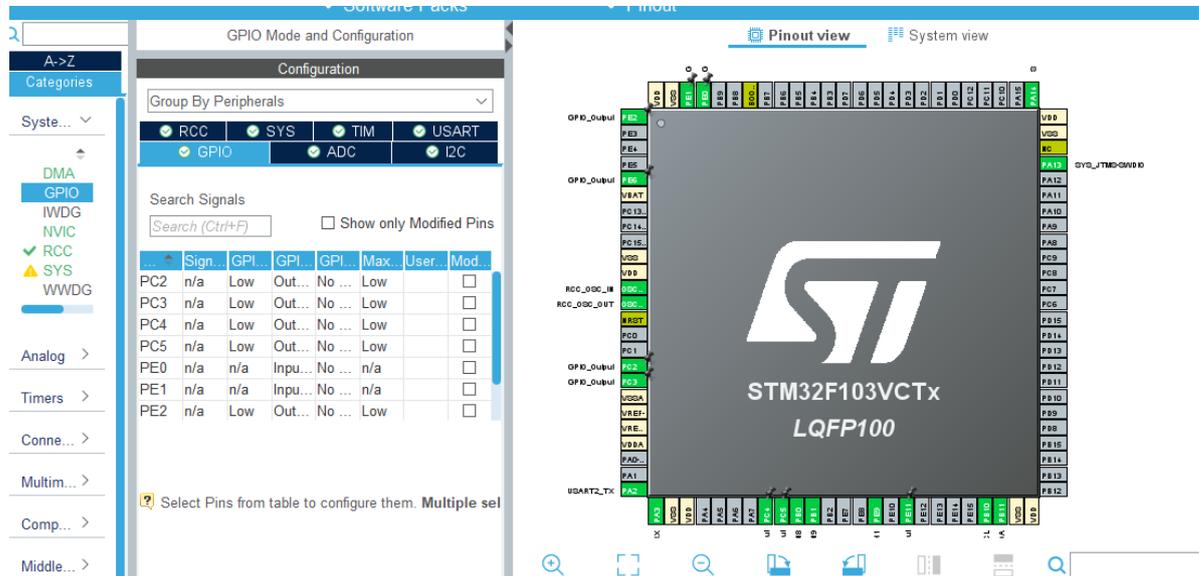
De la ecuación (11), con frecuencia de que ingresa al periférico de 72Mhz, prescaler igual a 65535 y con 100 ciclos nos da:

$$f_{PWM} = \left(\frac{72Mhz}{65535} \right) * 100 = 109864.9 Hz \quad (12)$$

Para el control de electroválvulas, configuramos algunos pines de nuestro MCU como salidas digitales y entradas digitales, en la figura 57 se muestra los puertos con sus

respectivos pines configurados como entrada o salida digital. PC2 significa Puerto C, Pin 2 configurado como salida.

Figura 57: Configuración de entradas y salidas digitales.



Elaboración Propia.

En el lengua C/C++, para activar y leer señales digitales, se utiliza la funciones implementadas en la capa HAL (void HAL_GPIO_WritePin (GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState (STMicroelectronic, 2016)).

3.2.7.5. Código IoT WEB

Todos los datos obtenidos por los sensores son procesados y enviados a internet, el módulo ESP8266 descrito en la anterior sección 3.2.2.5 es el encargado de establecer conexión con la pagina ThingSpeak que permite graficar el valor de las variables que le enviemos, en ThingSpeak creamos 8 graficas como se muestra en la figura 58, en donde las tres primeras graficas son de entrada, para establecer el valor de referencia de las condiciones climáticas que debe de tener el invernadero, y las gráficas 4 al 7 son las señales climáticas que tiene nuestro invernadero enviadas por los sensores, la gráfica 8 es de entrada para testear los actuadores instalados en el invernadero.

Figura 58: Distintas graficas de monitoreo.



Elaboración propia.

Su código implementado para el modulo es utilizar las funciones implementadas por ThingSpeak, en la figura 59 se muestras funciones de escritura y lectura (ThingSpeak.writeFields, ThingSpeak.readLongField (ThingSpeak, 2018)) de la base de

datos de ThingSpeak, todo este código es implementado en el entorno de desarrollo de Arduino(Software de código libre), que nos facilita la programación del ESP8266.

Figura 59: Parte del Código en C/C++.

```
actualizando_datos_thing$ secrets.h
ThingSpeak.setField(6,ET_value6);//humedad del suelo
ThingSpeak.setField(7,EE_value7);//intencidad luminosa
ThingSpeak.setField(8,SO_W_value8);//testSO_W_value8

statusWrite=ThingSpeak.writeFields(counterChannelNumber, myWriteApiKey);
if(statusWrite == 200){
    Serial.println("VM enviando datos");
}
else{
    Serial.println("Problema al escribir en los FIELDS. HTTP error code " + String(statusWrite));
}
//delay(15000); // tiempo de la actualizacion ThingSpeak
//GET POST PARA MODIFICAR POR LA PAGINA THINGSPEAK https://api.thingspeak.com/update?api\_key=HV43HJV1XPD

TS_value1S = ThingSpeak.readLongField(counterChannelNumber, TS_FiellNumber, myCounterReadAPIKey);
// chequeamos si la operacion se llevo correctamente////////
statusCode = ThingSpeak.getLastReadStatus();
if(statusCode == 200){
    TS_value1=TS_value1S;
    if(sepromm1!=(int)TS_value1){
        sepromm1=(int)TS_value1;
        EEPROM.put(0, sepromm1);
        EEPROM.commit();
        Serial.println("SETA" + String(TS_value1) + "guardado");
    }
    else{
        // mySerial2.println("SETA" + String(TS_value1));
    }
}
else{
    Serial.println("Problem reading FIELD 1. HTTP error code " + String(statusCode));
}
delay(1000);
// Leer estado del sistema TA_value2 de la grafica 2//////////
TA_value2S = ThingSpeak.readLongField(counterChannelNumber, TA_Fiel2Number, myCounterReadAPIKey);
// chequeamos si la operacion se llevo correctamente////////
statusCode = ThingSpeak.getLastReadStatus();
if(statusCode == 200){
    TA_value2=TA_value2S;
    if(sepromm2!=(int)TA_value2){
        sepromm2=(int)TA_value2;
        EEPROM.put(4, sepromm2);
        EEPROM.commit();
        Serial.println("SETB" + String(TA_value2) + "guardado" + String(sepromm2));
    }
    else{

```

Elaboración propia.

La comunicación del ESP8266 con el microcontrolador STM32f103ve es a través del UART, antes de enviar los datos de los sensores, primeramente nuestro sistema de control Microclimático es controlado por control proporcional, que detallamos a continuación.

3.2.7.6. Control proporcional

El tipo de control con retroalimentación más práctica, en donde el tiempo de respuesta de una variación de la señal del sensor no es crítico, no requiere complejas funciones de respuesta en el tiempo del sistema, para nuestro proyecto solo utiliza el control proporcional, para implementar este tipo de control, primeramente se requiere declarar una variable error, que guardara el valor de la diferencia de la señal de referencia y la del sensor, para luego ser multiplicada por un factor proporcional, también se agrega Histéresis a la señal para protección de interruptores electromecánicos, en la figura 60 se visualiza la implementación en código C/C++, en donde la variable HstA se comporta como Histéresis para el sensor de temperatura.

Figura 60: Implementando control proporcional en C/C++

```
// establecimiento por control proporcional temperatura
eA=SETA-tm;//extrayendo el error de la temperatura de sistema
//Agregando hestéresis al sistema para proteccion de relay//
eAZ=eA+HstA;
if(eAZ>0){
    HstA=2;
}
else{
    HstA=-2;
}

if(eAZ<0){ //sistema requiere menos temperatura
    if(100<(2*abs(eAZ))){
        eeA=100;
    }
    else{
        eeA=2*abs(eAZ);
    }
    if(eeA+eeB>100){
        //activar ventilador, calefactor apagado
        TIM1->CCR1=100; //CCR1: VENTILADOR
    }
    else{
        //activar ventilador, calefactor apagado
        TIM1->CCR1=eeA+eeB; //CCR1: VENTILADOR
    }
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_11, GPIO_PIN_SET); //CCR2: CALEFACTOR
}
else if(0<eAZ){ //Sistema requiere mas temperatura
    eeA=100;
    //ventilador al minimo, calefactor encendido
    TIM1->CCR1=eeB; //CCR1: VENTILADOR
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_11, GPIO_PIN_RESET); //CCR2: CALEFACTOR
}

// establecimiento por control proporcional humedad aire
// eB=SETB-tm;//extrayendo el error de la humedad de sistema
```

Elaboración propia.

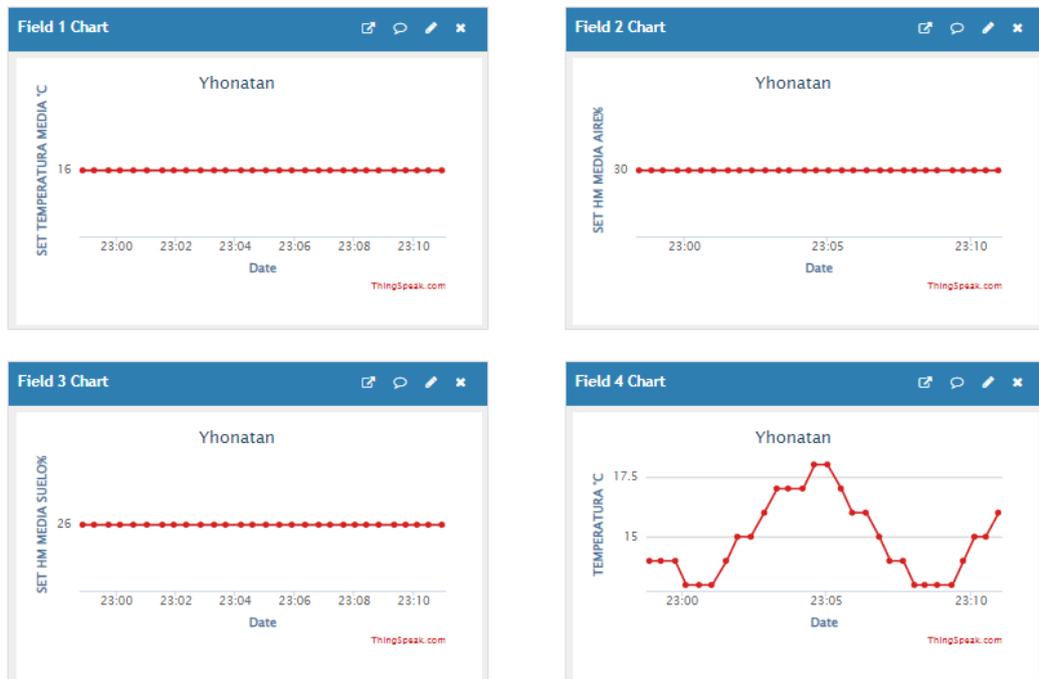


3.2.8. Prueba de funcionamiento

En esta etapa del proyecto se realizan dos tipos de prueba para el prototipo, la primera y la más importante es el comportamiento del sistema por la noche, por la presencia de caídas bruscas de temperatura y heladas, y la segunda es el comportamiento en el día. Nuestra planta fructífera de prueba es la fresa, según la tesis con “Evaluación de técnicas hidropónicas de producción en el cultivo de fresa (*Fragaria x ananassa*) bajo invernadero” (RAMÓN DONOVAN, 2013), que describe las condiciones climatológicas óptima que debe de tener para las distintas etapas de crecimiento de la fresa. Temperatura de crecimiento óptimo es de 20 °C durante el día y mayor de 12°C durante la noche (Zaragoza Nieto R. D., 2013, pág. 21), la humedad suelo debe de oscilar entre 50% y 80% (GERARDO MAXIMO, 2016), la humedad del relativa del aire por encima de 30% a 70%, la utilidad del sensor de intensidad luminosa es para encontrar la posición y el lugar con mayor incidencia de energía luminosa. La graficas 61 muestra los valores e referencia establecidos para nuestro prototipo Microclimático, la figura 62 y 63 son los valores enviados por los sensores partir de las 9:00 pm, cuando empieza el descenso de la temperatura.

Figura 61: Valores de referencia de nuestro microclima.

Last entry: [less than a minute ago](#)
Entries: 5100



Elaboración propia

Figura 62: Datos recibidos de 30 muestras dentro de las 9:00 pm



Elaboración propia.

Figura 63: Datos recibidos de 30 muestras dentro de las 10:00 pm



Elaboración propia.

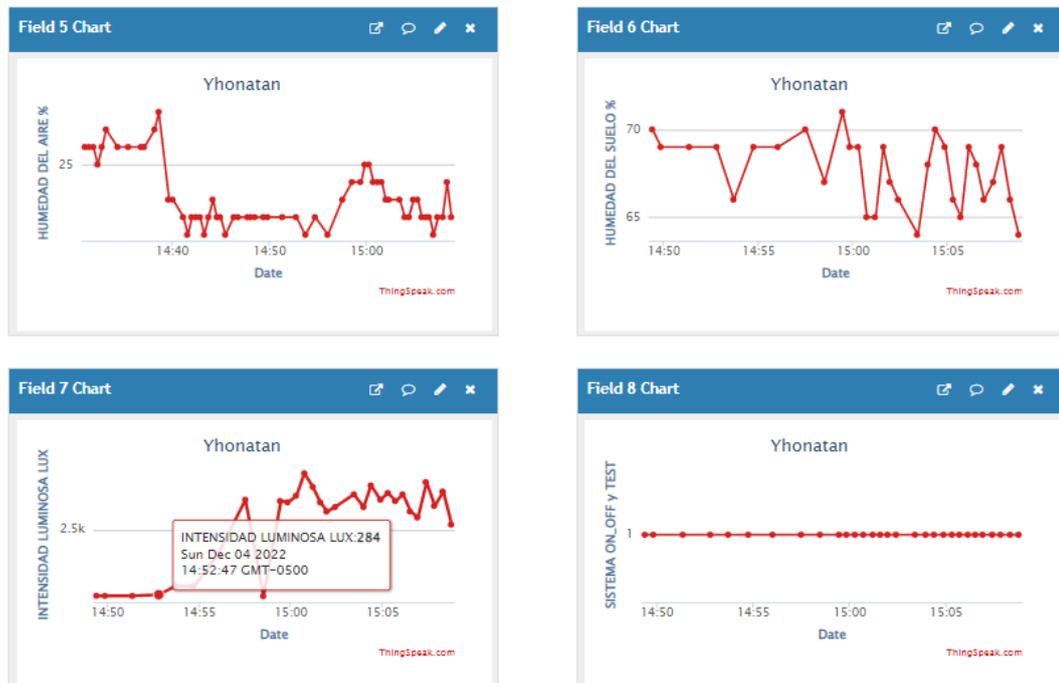
Las siguientes imágenes, figura 64 a 66 son obtenidas en el día, a partir de las 2:00 pm de la tarde, en donde se puede ver en como disminuye la intensidad luminosa en (LUX).

Figura 64: Datos recibidos de 30 muestras dentro de las 2:00 pm



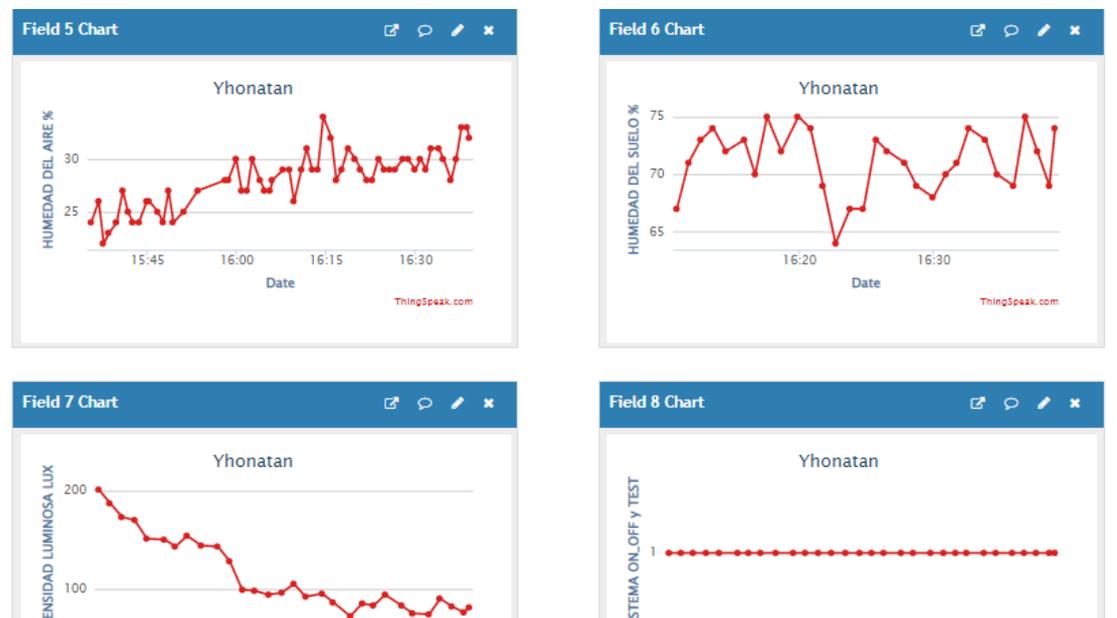
Elaboración propia.

Figura 65: Datos recibidos de 30 muestras dentro de las 3:00 pm.



Elaboración propia.

Figura 66: Datos recibidos de 30 muestras dentro de las 4:00 pm



Elaboración propia.



CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1. CONTRASTACIÓN DE HIPÓTESIS

Para esta etapa recolectamos los datos en un intervalo en el tiempo, que será la muestra obtenidos por los sensores, de las pruebas realizadas en el capítulo anterior para la planta fructífera fresa, todo estos datos serán tratados con fórmulas de ciencias de estadística, para luego realizar las comparaciones con las referencias teóricas para el clima a crear, la siguiente tabla muestra los datos recolectados por los sensores en la noche, en la figura 67 se muestra los datos obtenidos a partir de las 2:00 am.

Figura 67: Datos para análisis de contrastación.



Elaboración propia.

Tabla 4: Datos obtenidos en la madrugada.

HORA		TEMPERATURA	HUMEDAD DEL AIRE	HUMEDAD DEL SUELO
02:01:00	am	15.00	43.00	74.00
02:02:00	am	15.00	48.00	74.00
02:03:00	am	15.00	52.00	71.00
02:04:00	am	15.00	50.00	78.00
02:05:00	am	14.00	47.00	74.00
02:06:00	am	14.00	52.00	74.00



02:07:00	am	14.00	49.00	78.00
02:08:00	am	14.00	49.00	77.00
02:09:00	am	14.00	49.00	70.00
02:10:00	am	14.00	43.00	70.00
02:11:00	am	14.00	41.00	71.00
02:12:00	am	13.00	52.00	71.00
02:13:00	am	13.00	50.00	74.00
02:14:00	am	13.00	41.00	74.00
02:15:00	am	13.00	41.00	77.00
02:16:00	am	13.00	32.00	74.00
02:17:00	am	13.00	51.00	73.00
02:18:00	am	13.00	50.00	70.00
02:19:00	am	14.00	50.00	73.00
02:20:00	am	13.00	48.00	74.00
02:21:00	am	13.00	48.00	76.00
02:22:00	am	13.00	43.00	74.00
02:23:00	am	13.00	38.00	74.00
02:24:00	am	13.00	50.00	70.00
02:25:00	am	13.00	51.00	74.00
02:26:00	am	13.00	51.00	74.00
02:27:00	am	13.00	50.00	75.00
02:28:00	am	13.00	52.00	75.00
02:29:00	am	13.00	52.00	71.00
02:30:00	am	13.00	54.00	71.00
PROMEDIO:		13.53	47.57	73.50

Elaboración propia.

El promedio se obtiene aplicando la ecuación (2), el error definido por la ecuación (1), es la diferencia del promedio aritmético menos el valor de referencia, de la gráfica 67 te tiene las referencias 16 °C para la temperatura, 50% para la humedad del aire y 70% para la humedad del suelo, aplicando la ecuación (1) no da:

$$ERROR_{TEMPERATURA} = 16 - 13.53 = 2.47 \quad (13)$$

$$ERROR_{HUMEDADaire} = 50 - 47.57 = 2.43 \quad (14)$$

$$ERROR_{HUMEDADsuelo} = 70 - 73.5 = -3.5 \quad (15)$$

De los resultados anteriores, no olvidemos que agregamos Histéresis al sistema de control, figura 60 (variable HstA) para la protección de interruptores electromecánicos de +- 2, de los resultados de las ecuaciones (13), (14) y (15) el error final es; 0.47, 0.43 y -1.5.

4.2. RESULTADOS

La mayoría de cambios climáticos perjudiciales para el cultivo, se inicia al inicio de la noche, para poder mantener los niveles mínimos climatológicos optimas que debe de tener la fresa, en la figura 61 se muestra como se establece, monitorea y se carga al STM32F103vc las variables principales a través de ThingSpeak, una vez establecido los niveles mínimos, el microcontrolador trata de compensar cuando una variable cae por debajo del valor mínimo como se muestra en la figura 62 de la gráfica temperatura, que cae a 12 °C, el microcontrolador inicia el calefactor hasta alcanzar una temperatura de 17 °C, luego de apagar el calefactor, la temperatura cae lentamente en promedio de 5 minutos e iniciar nuevamente el ciclo para mantener una temperatura mayor a 12 °C, las demás graficas de los sensores se comportar de igual forma que la temperatura, se puede notar que a medida que avanza la noche, el intervalo de iniciar el ciclo de compensación de temperatura por el STM32F103vc es más corto, figura 63, resultado directamente por el aumento de caída de temperatura del exterior a medida que aumenta la noche. Durante el



día figura 64 la humedad del suelo y el aire son los más propensos que cambien, debido a la energía solar natural, mientras que la temperatura se mantiene en un promedio 18 °C en un día normal. Las gráficas del ThingSeak, y el microcontrolador STM32f103vc satisface la demanda de todos los sensores y actuadores para la corrección de las variaciones climáticas dentro del invernadero, como lo muestra las pruebas realizadas en horas de la madrugada a horas 2:01 am, figura 67 nos da un error menor a 1°C del aire y de -1.5 de la humedad del suelo.



V. CONCLUSIONES

Primero. La variación de las magnitudes respecto al tiempo de alguna variable climática dentro del invernadero es lenta, esto es debido a la cubierta protectora del invernadero que impide intercambios físicos climáticos bruscos, esto facilita implementar sistemas de control con baja velocidad de retroalimentación de respuesta al sistema, el control proporcional fue suficiente para mantener un microclima predeterminado de prueba.

Segundo. Los distintos sensores instalados en el invernadero, utilizan distintos protocolos de comunicación, para una rápida implementación e integración con los distintos actuadores con los periféricos embebidos como el: I2C, ADC, TIM, UART... se facilita la implementación y configuración con el entorno de programación STM32cube, de la misma empresa

Tercero. Las distintas graficas que monitorean el invernadero se actualizan cada 20 segundos, esto nos ayuda en mucho saber el estado actual en que se encuentra el invernadero, como también en establecer cambios rápidos de nuevo puntos de referencia, también nos proporciona rápidas respuestas de compensación de las variables climáticas y el monitoreo de las señales provenientes de los sensores, con todo esto nos ayuda en mucho simular un ambiente estable.



VI. RECOMENDACIONES

- Para poder crear un microclima más real, se requiere que un actuador; como por ejemplo, el ventilador dependa de varias variables climáticas, como la temperatura y la humedad esto dificulta implementar con sistemas de control automáticos tradicionales. La más recomendable es utilizar control difuso.
- Algunos actuadores de nuestro proyecto eran electromecánicos, esto limita a un control on/off de muy baja velocidad como por ejemplo las electroválvulas. Este delimita el potencial de los periféricos disponibles del microcontrolador, es recomendable utilizar otros métodos de control de flujo de agua para invernaderos.
- Al mejorar un microclima estable es necesario tener varios sensores de un mismo tipo, pero instalados en distintos lugares, para así obtener un como resultado un valor promedio, para que luego sea procesado por el microcontrolador, de esta manera se reducida los cambios abruptos causados por algún factor exterior.



VII. REFERENCIAS BIBLIOGRÁFICAS

- AgroSpray. (2021, 04 11). *Producción Agrícola: Factores que Influyen En Los Resultados*. Obtenido de <https://agrospray.com.ar/blog/produccion-agricola/>
- Alberto Orenca, M., & Enrique Manonellas, G. (2000). *Programacion en Ensamblador(x86_64)*. Catalunya: Universitat Oberta de Catalunya.
- Aliaga Mendoza, G., & Quispe Bolaños, P. R. (2015). *Sistema De Control De La Humedad Relativa Para Un Invernadero Utilizando El Controlador Lógico Programable*. HUANCAVELICA: UNIVERSIDAD NACIONAL DE HUANCAVELICA.
- Anthura. (2022, 07 13). *La Influencia Del pH En El Cultivo*. Obtenido de https://www.anthura.nl/growing-advise/la-influencia-del-ph-en-el-cultivo/?lang=es&cookies=not_ok
- AOSONG MICROELECTRONICS. (2020, Abril 03). *Digital Temperature and HumiditySensor*. Obtenido de <https://cdn-shop.adafruit.com/product-files/3721/AM2320.pdf>
- Apaza Mamani, D. F., & La torre Javier, I. J. (2016). *Diseño e Implementación De Un Sistema Automatizado Para Riego Tecnificado Basado En El Balance De Humedad De Suelo Con Tecnologia Arduino En El Laboratorio De Control Y Automatización EPIME 2016*. PUNO: UNIVERCIDAD NACIONAL DEL ALTIPLANO.
- ARM Developer. (2022, 07 12). *CORTEX-R4*. Obtenido de <https://developer.arm.com/Processors/Cortex-R4>



- ARMDeveloper. (2022, 07 13). *ARM CORTEX*. Obtenido de <https://developer.arm.com/Architectures/Helium>
- Barrera Martin, E., Herrero Martin, R. V., & Meraz Garía, A. (2014). *Invernadero Inteligente*. México.
- Bianchi, P. G. (1986). *Guia Completa De Cultivo De Fresa*. España: Editorial De Vecchi.
- BRICEÑO JEJÉN, M., & CUBIDES PORRAS, W. F. (2020). *Diseño De Una Solución IoT Para Monitoreo De Las Condiciones Optimas De Un Cultivo Hidropónico De Flores En Invernadero*. BUCARAMANGA: UNIVERCIDAD AUTÓNOMA DE BUCARAMANGA.
- Carrillo Reveles, D. A., & Vázquez Minjares, J. L. (2008). *Automarización De Un Invernadero Con El PLC S7-200*. MÉXICO.
- Cuji Cáceres, C. A., & Jara Barreto, P. E. (2022). *Control de Temperatura De Un Proceso Térmico Utilizando el Microcontrolador STM32F104 Bajo Programación e Interacción con Simulink de Matlab*. ECUADOR: UNIVERCIDAD POLITÉCNICA SALESIANA.
- ESPRESSIF. (2022, 06 20). *ESP8266 A Cost-Efective And Highly Integrated*. Obtenido de <https://www.espressif.com/en/products/socs/esp8266>
- Folquer, F. (1986). *La Frutilla o Fresa De La Planta y Su Produccion Comercial*. Argentina: Hemisferio Sur.
- FreeRTOS. (2021, 08 15). *Real-Time Operating System For Microcontrollers*. Obtenido de <https://www.freertos.org/RTOS-task-states.html>
- GERARDO MAXIMO, R. S. (2016). Efecto De Aplicación De Tres Frecuencias De Riego Y Fertilización En El Rendimiento De Fresa (Fragaria x ananassa), en el



C.I.E. DE CAÑASBAMBA YUNGAY 2017. *UNIVERSIDAD NACIONAL SANTIAGO ANTÚNEZ DE MAYOLO*, 153.

Gutierrez Carbajal, M. G., & Magaña Magaña, M. Á. (2017). Migración e Influencia Urbana en el Consumo de Alimentos En Dos Comunidades. *CIAD*.

Huanca Chuquicallata, J. E. (2022). *Diseño e Implementación de un prototipo para el Proceso de Transferencia de Imagen a la Placa de Circuito Impreso con un Diodo Laser de Longitud de Onda de 405 Nanómetros, utilizando la Interfaz Serial de Periféricos(SPI)*. puno.

INTAGRI. (2022, 09 22). *Importancia De La Radiación Solar En La Produccion Bajo Invernadero*. Obtenido de [https://www.intagri.com/articulos/horticultura-protegida/importancia-de-la-radiacion-solar-en-la-produccion-bajo-invernadero#:~:text=La%20Radiaci%C3%B3n%20Fotosint%C3%A9ticamente%20Activa%20\(RFA,de%20calor%20en%20el%20invernadero](https://www.intagri.com/articulos/horticultura-protegida/importancia-de-la-radiacion-solar-en-la-produccion-bajo-invernadero#:~:text=La%20Radiaci%C3%B3n%20Fotosint%C3%A9ticamente%20Activa%20(RFA,de%20calor%20en%20el%20invernadero).

Intagri S.C. (2021, 10 10). *La Conductividad Electrica del Suelo en el Desarrollo de los Cultivos*. Obtenido de <https://www.intagri.com/articulos/suelos/la-conductividad-electrica-del-suelo-en-el-desarrollo-de-los-cultivos#>

LUNA OPTOELECTRONICS. (2016). *Plastic Package Cds Photocells*. Avenida acaso: Luna Optoelectronics.

Maroto, J. V., Pascual, B., Alargada, J., & López, G. (1986). *Mejora De La Precocidad Del Cultivo De Fresón (Fragaria x Ananassa Duch. Cv Pájaro) Mediante Aplicaciones Invernales de Ácido Giberelico*. ITEA 63: 36-38.

MICROCHIP. (2022, 07 12). *MPLAB X IDE*. Obtenido de <https://www.microchip.com/en-us/tools-resources/develop/mplab-x-ide>



- Mora López , D., Abarca Reyes, P., Olgúin Rubio, J., Carrasco Jiménez, J., Aguirre Aguilera, C., & Silva Rubio, L. (2015, Septiembre 11). *Construcción de Invernaderos*. Obtenido de <https://biblioteca.inia.cl/bitstream/handle/20.500.14001/6475/NR40392.pdf?sequence=11&isAllowed=y>
- ON Semiconductor. (2017, 11 20). *ESD Protection Diode Array*. Obtenido de <https://www.onsemi.com/pdf/datasheet/srv05-4-d.pdf>
- Ortega Álvarez, E. D. (2017). *Sistema De Control De Temperatura Y Humedad Para La Producción De Cultivos Hidropónicos En Invernaderos Automatizados*. MÉXICO.
- Oxford University Press. (2022). *OxfordLanguages*. Reino Unido: Oxford University Press.
- PEDROLLO the spring ok life. (2021, 09 3). *Electrobombas con rodete periférico*. Obtenido de https://www.pedrollo.com/public/allegati/PK_ES_60Hz.pdf
- PROAIN Tecnología agrícola. (2022, 07 22). *La Humedad Del Suelo En La Agricultura*. Obtenido de <https://proain.com/blogs/notas-tecnicas/la-humedad-del-suelo-en-la-agricultura>
- PROMIX. (2022, 08 20). *¿Cómo Influye la Humedad En La Calidad De Los Cultivos?* Obtenido de <https://www.pthorticulture.com/es/centro-de-formacion/como-influye-la-humedad-en-la-calidad-de-los-cultivos/#:~:text=Mientras%20m%C3%A1s%20seca%20o%20caliente,transpiraci%C3%B3n%20no%20est%C3%A1%20directamente%20relacionada>.
- Pumasupa Paro, N. (2014). *Implementación de un Sistema de Gestión Inalámbrico para Sistemas Scada Basados en Microcontrolador Aplicado a Invernaderos*. Cusco.



- Pumasupa Paro, N. E. (2014). *Implementación De Un Sistema De Gestión Inalámbrico Para Sistemas Scada Basado En Microcontrolador Aplicado A Invernaderos*. CUSCO.
- RAMÓN DONOVAN , Z. N. (2013). Evaluación De Técnicas Hidropónicas De Producción En El Cultivo De Fresa (Fragaria x Ananassa) Bajo Invernadero. *CENTRO DE INVESTIGACIÓN EN QUÍMICA APLICADA*, 85.
- STMicroelectronic. (2007, 07 11). *ST485EB For E-Meter Applications*. Obtenido de https://www.st.com/resource/en/application_note/an1702-st485eb-for-emeter-applications-stmicroelectronics.pdf
- STMicroelectronic. (2015). *RM0008*.
- STMicroelectronic. (2016). *stm32f1xx_hal_gpio.c*. MCD Application Team.
- STULZ. (2022, 09 2). *Principio De Funcionamiento Del Humificador Ultrasónico*. Obtenido de <https://www.stulz.com.mx/es/ultrasonic/principio-de-funcionamiento/#:~:text=Los%20humidificadores%20ultras%C3%B3nicos%20funcionan%20con,alta%20frecuencia%20de%201.65%20MHz>.
- Taleng Oliag, S. (1995). *Curso de programación en C++*. Valencia.
- ThingSpeak. (2018). *Communication Library For Arduino, ESP8266 and ESP32*. ThingSpeak.
- TRU TECHNOLOGIES (SINGAPORE) PTE LTD. (2022, 09 3). *F2E-260B*. Obtenido de <https://www.tru-technologies.com/leipole/enclosure-climate-control/f2e-260b-260-x-227-x-80-mm/>
- WIKIPEDIA La enciclopedia libre. (2021, 06 25). *Modbus*. Obtenido de <https://es.wikipedia.org/wiki/Modbus>



WIKIPEDIA La enciclopedia Libre. (2022, 07 12). RS-232. Obtenido de
<https://es.wikipedia.org/wiki/RS-232>

Wikipedia la encicloperia . (2022, 08 28). *ARQUITECTURA ARM*. Obtenido de
https://es.wikipedia.org/wiki/Arquitectura_ARM

Yui, J. (2010). *THE DEFINITIVE GUIDE TO THE ARM CORTEX-M3*. UU.EE: Elsevier.

Zaragoza Nieto, R. D. (2013). *Evaluación de Técnicas Hidropónicas de Producción en el Cultivo de Fresa (Fragaria x ananassa) Bajo Invernadero*. Mexico: CENTRO DE INVESTIGACIÓN EN QUÍMICA APLICADA.

Zaragoza Nieto, R. D. (2013). *Evaluación de técnicas hidropónicas de prucción en el cultivo de fresa (Fragaria x ananassa) bajo invernadero*. Saltillo.



ANEXOS

Anexo 01.

Código main() de programación principal para el STM32F103VC

```
int main(void)
{
  /* USER CODE BEGIN 1 */
    HAL_StatusTypeDef StatusUart, StatusADC1;
    uint8_t rxBuffer[]={"NTA015SETB036SETC035SETD01\0xxx"};

    uint8_t sub_rxBuffer[]={"\0\0\0\0\0\0\0\0\0\0"};
    uint32_t ad[2];
    int adin_HS, i_LDR; //HS: humedad suelo, LX: lux, corriente del LDR
                        //resistencia del LDR

    ad[0]=23;
    ad[1]=111;
    int SETA=15;
    int SETB=36;
    int SETC=35;
    int SETD=1;
  /* USER CODE END 1 */

  /* MCU Configuration-----*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DMA_Init();
```



```
MX_USART2_UART_Init();
MX_I2C2_Init();
MX_ADC1_Init();
MX_TIM1_Init();
/* USER CODE BEGIN 2 */
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5, GPIO_PIN_SET);
HAL_Delay(1000);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5, GPIO_PIN_RESET);
    Am2320_HandleTypeDef Am2320_2;
    Am2320_2 = am2320_Init(&hi2c2, AM2320_ADDRESS);
    float temperature, humidity;
    double lux, res_LDR;
    int luxInt;
    int tm, hu;
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_11, GPIO_PIN_SET);
    HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
    TIM1->CCR1=1;
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
StatusADC1 = HAL_ADC_Start_DMA(&hadc1, ad, 2);
am2320_GetTemperatureAndHumidity(&Am2320_2, &temperature, &humidity);
    tm=(int)temperature;
    hu=(int)humidity;
    adin_HS=((4095-((int)ad[0]))*100)/2495;//humedad suelo
    i_LDR=(4096-ad[1])/1;//hallando la corriente del LDR. 4095:=3.2V, 1k
    res_LDR=((ad[1]*1000)/i_LDR);//hallando la resistencia del LDR
    lux=pow((res_LDR/5000.0),1.6);
    lux=(10/lux);
    luxInt=(int)(lux);
    /*funciones para testeo de variables*/
        //my_printf("=====\r\n");
        //my_printf("Temperature: %.1f°C\r\n", temperature);
        //my_printf("Humidity: %.1f%%\r\n", humidity);
        //HAL_Delay(100);
        my_printf("corriente %u\r\n",i_LDR );
        my_printf("resistencia %f\r\n",res_LDR );
        my_printf("VA%f\r\n",lux );

```



```
my_printf("adc1: %d %d %d\r\n", ad[0], ad[1], StatusADC1);

*///
if(timY>100){
    my_printf("TA%03uHA%03uHS%03uL%06uSO%02u\r\n",tm,hu,adin_HS, luxInt, SETD);
    // my_printf("SETA%03uSETB%03uSETC%03uSETD%02u\r\n",SETA,SETB,SETC,SETD);
    HAL_Delay(100);

    timY=0;
}

//NTA000SETB021SETC000SETD00//formato que debe ser recibido uart2//26 datos
StatusUart = HAL_UART_Receive_IT(&huart2, rxBuffer, 28);
if(0x00U==StatusUart){
    strncpy(sub_rxBuffer, rxBuffer+3,3);
    sub_rxBuffer[3]='\0';
    SETA = atoi(sub_rxBuffer);
    //my_printf("uartA: %d\r\n",SETA );
    strncpy(sub_rxBuffer, rxBuffer+10,3);
    sub_rxBuffer[3]='\0';

    SETB = atoi(sub_rxBuffer);
    //my_printf("uartB: %d\r\n",SETB);
    strncpy(sub_rxBuffer, rxBuffer+17,3);
    sub_rxBuffer[3]='\0';
    SETC = atoi(sub_rxBuffer);
    //my_printf("uartC: %d\r\n",SETC );
    strncpy(sub_rxBuffer, rxBuffer+24,2);
    sub_rxBuffer[2]='\0';
    SETD = atoi(sub_rxBuffer);

if(SETD==2){ //calefactor prueba
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_11, GPIO_PIN_RESET); //CCR2: CALEFACTOR
    HAL_Delay(3000);
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_11, GPIO_PIN_SET);
}
else if(SETD==3){ //ventilador prueba
    TIM1->CCR1=100; //CCR1: VENTILADOR
    HAL_Delay(3000);
}
else if(SETD==4){ //humidificador prueba
    while((HAL_GPIO_ReadPin(GPIOE,GPIO_PIN_0)==GPIO_PIN_SET)||((HAL_GPIO_ReadPin(GPIOE,
GPIO_PIN_1)==GPIO_PIN_SET)){
        HAL_GPIO_TogglePin(GPIOE, GPIO_PIN_2);//Humificador Ultrasonico
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_3,GPIO_PIN_SET);
        HAL_Delay(200);
    }
    HAL_Delay(3000);
}
```



```
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_3,GPIO_PIN_RESET);
    }
    else if(SETD==5){    //electrovalvula prueba
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_6, GPIO_PIN_SET);//encender electrovalvulas
    }
    else{

    }

    //my_printf("uartD: %d\r\n",SETD );
    SETD=1;

    // establecimiento por control proporcional
    }
    else{

    }

    // establecimiento por control proporcional temperatura
    eA=SETA-tm;//extrayendo el error de la temperatura de sistema
        //Agregando hestérisis al sistema para proteccion de relay///
    eAZ=eA+HstA;
    if(eAZ>0){
        HstA=2;
    }
    else{
        HstA=-2;
    }
    if(eAZ<0){ //sistema requiere menos temperatura
        if(100<(2*abs(eAZ))){
            eeA=100;
        }
        else{
            eeA=2*abs(eAZ);
        }
        if(eeA+eeB>100){
            //activar ventilador, calefactor apagado
            TIM1->CCR1=100;    //CCR1: VENTILADOR
        }
        else{
            //activar ventilador, calefactor apagado
            TIM1->CCR1=eeA+eeB;    //CCR1: VENTILADOR
        }
        HAL_GPIO_WritePin(GPIOE, GPIO_PIN_11, GPIO_PIN_SET);    //CCR2: CALEFACTOR
    }
}
```



```
else if(0<eAZ){ //Sistema requiere mas temperatura
    eeA=100;
    //ventilador al minimo, calefactor encendido
    TIM1->CCR1=eeB; //CCR1: VENTILADOR
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_11, GPIO_PIN_RESET); //CCR2: CALEFACTOR
}

// establecimiento por control proporcional humedad aire
eB=SETB-hu;//extrayendo el error de la humedad de sistema
eBZ=eB+HstB;
if(eBZ>0){
    HstB=2;
}
else{
    HstB=-2;
}
if(eBZ<0){ //sistema requiere menos humedad
    if(100<(2*abs(eBZ))){
        eeB=100;
    }
    else{
        eeB=2*abs(eBZ);
    }
    if(eeA+eeB>100){
        //activar ventilador, hunificador apagado
        TIM1->CCR1=100; //CCR1: VENTILADOR
    }
    else{
        TIM1->CCR1=eeA+eeB; //CCR1: VENTILADOR
    }
    //activar ventilador, hunificador apagado
    while((HAL_GPIO_ReadPin(GPIOE,GPIO_PIN_0)==GPIO_PIN_RESET)|| (HAL_GPIO_ReadPin(GPIOE,
GPIO_PIN_1)==GPIO_PIN_RESET)){
        HAL_GPIO_TogglePin(GPIOE, GPIO_PIN_2);//Hunificador Ultrasonico
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_2,GPIO_PIN_SET);
        HAL_Delay(200);
    }
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_2,GPIO_PIN_RESET);
}
else if(0<eBZ){ //Sistema requiere mas humedad
    //ventilador al minimo, hunificador encendido
    eeB=0;
    TIM1->CCR1=eeA; //CCR1: VENTILADOR al minimo
    while((HAL_GPIO_ReadPin(GPIOE, GPIO_PIN_0)==GPIO_PIN_SET)|| (HAL_GPIO_ReadPin(GPIOE,
GPIO_PIN_1)==GPIO_PIN_SET)){
```



```
        HAL_GPIO_TogglePin(GPIOE, GPIO_PIN_2);//Hunificador Ultrasonico
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_3,GPIO_PIN_SET);
        HAL_Delay(200);
    }
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_3,GPIO_PIN_RESET);
}

// establecimiento, por control proporcional humedad suelo
eC=SETC-adin_HS;//extrayendo el error de la humedad de sistema
//Agregando hestérisis al sistema para proteccion de relay///
eCZ=eC+HstC;
if(eCZ>0){
    HstC=2;
}
else{
    HstC=-2;
}
if(eCZ<0){ //sistema requiere menos humedad

    //apagar, aspersor, electrovalvula
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_6, GPIO_PIN_RESET);//apagar electrovalvulas
}
else if(0<eCZ){ //Sistema requiere mas humedad

    //activar, aspersor, electrovalvula
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_6, GPIO_PIN_SET);//encender electrovalvulas
}

if(timY1==10){
    HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_4);
    timY1=0;
}

    HAL_Delay(170);
    timY++;
    timY1++;
}
/* USER CODE END 3 */
}
```



Anexo 02.

Código para a IoT ESP8266.

```
#include <dummy.h>
/*
  CODIGO PARA ESTABLESER LA CONECCION CON LA BASE DE THINGSPEAK
*/
#include "ThingSpeak.h"
#include "secrets.h"
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
#include <EEPROM.h>
SoftwareSerial mySerial2;
#define D5 (14) //tx2
#define D6 (12) //rx2
#define EEPROM_TAMAÑO 20
char ssid[] = SECRET_SSID; // SSID (name)
char pass[] = SECRET_PASS; // password
int keyIndex = 0; // )
String SensorData="TAXXXHAXXXHSXXXILXXXXXSOXX"; //Formato de datos recibidos
// 1 5 9 13 17 21 25 29
/////"TA034HA020HS015IL000550SO01"//PRUEBAS
///"TA020HA025HS020IL001550SO03"
int Ndata=0;
WiFiClient client;
unsigned long counterChannelNumber = SECRET_CH_ID_COUNTER;
// Weather station channel details de lectura
unsigned long weatherStationChannelNumber = SECRET_CH_ID_WEATHER_STATION;
unsigned int temperatureFieldNumber = 4;
// Counting channel details de lectura
const char * myCounterReadAPIKey = SECRET_READ_APIKEY_COUNTER;
unsigned int counterFieldNumber = 8;
// declaracion de variables de escritura
const char * myWriteApiKey = SECRET_WRITE_APIKEY_YHONATAN;
unsigned int TS_Fiel1Number = 1;
unsigned int TA_Fiel2Number = 2;
unsigned int HS_Fiel3Number = 3;
unsigned int HA_Fiel4Number = 4;
unsigned int IL_Fiel5Number = 5;
unsigned int ET_Fiel6Number = 6;
unsigned int EE_Fiel7Number = 7;
unsigned int SO_Fiel8Number = 8;
int statusWrite=0;
```



```
long TS_value1 = 20;//°c
long TS_value1S;
long TA_value2 = 40;//%
long TA_value2S;
long HS_value3 = 45;//%
long HS_value3S;
long HA_value4 = 10;//°c
long HA_value4S;
long IL_value5 = 20;//%
long IL_value5S;
long ET_value6 = 25;//%
long ET_value6S;
long EE_value7 = 50000;//lux
long EE_value7S;
//0: off
//1: on
//2: ventilador max power, 2s
//3: calefactor max power, 2s
//4: humificador max power, 2s
//5: valvulas max power, 2s
long SO_W_value8 = 1;//lest/on
long SO_R_value8 = 30;
long SO_R_value8S;
int sepromm1,sepromm2,sepromm3;
void setup() {
  Serial.begin(115200); // Initialize serial
  mySerial2.begin(9600, SWSERIAL_8N1, D6, D5, false, 256);
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client); // Initialize ThingSpeak
  EEPROM.begin(EEPROM_TAMAÑO);
  EEPROM.get(0,sepromm1);
  TS_value1=(long)sepromm1;
  EEPROM.get(4,sepromm2);
  TA_value2=(long)sepromm2;
  EEPROM.get(8,sepromm3);
  HS_value3=(long)sepromm3;
}
void loop() {
  int statusCode = 0;
  //mySerial2.println("eeprom" + String(sepromm1)+String(sepromm2)+String(sepromm3));
  /////////////// Conectando o reconectando a la red por WIFI////////////////////
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
```



```
Serial.println(SECRET_SSID);
while(WiFi.status() != WL_CONNECTED){
  WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line if using
  Serial.print(".");
  delay(5000);
}
Serial.println("\nConnected");
}

ThingSpeak.setField(1,TS_value1);
ThingSpeak.setField(2,TA_value2);
ThingSpeak.setField(3,HS_value3);
ThingSpeak.setField(4,HA_value4);//temperatura
ThingSpeak.setField(5,IL_value5);//humedad del aire
ThingSpeak.setField(6,ET_value6);//humedad del suelo
ThingSpeak.setField(7,EE_value7);//intencidad luminosa
ThingSpeak.setField(8,SO_W_value8);//testSO_W_value8

statusWrite=ThingSpeak.writeFields(counterChannelNumber, myWriteApiKey);
  if(statusWrite == 200){
    Serial.println("VM enviando datos");
  }
  else{
    Serial.println("Problema al escribir en los FIELDS. HTTP error code " + String(statusWrite));
  }

  //delay(15000); // tiempo de la actualizacion ThingSpeak

//GET      POST      PARA      MODIFICAR      POR      LA      PAGINA      THINGSPEAK
https://api.thingSpeak.com/update?api_key=HV43HJV1XPDNKKX6&field8=0

// Leer estado del TS_value1 de la grafica 1////////////////////
TS_value1S = ThingSpeak.readLongField(counterChannelNumber, TS_Fiel1Number, myCounterReadAPIKey);
// chequeamos si la operacion se llevo correctamente////////
statusCode = ThingSpeak.getLastReadStatus();
if(statusCode == 200){
  TS_value1=TS_value1S;
  if(sepromm1!=(int)TS_value1){
    sepromm1=(int)TS_value1;
    EEPROM.put(0, sepromm1);
    EEPROM.commit();
    Serial.println("SETA" + String(TS_value1) + "guardado");
  }
  else{
    // mySerial2.println("SETA" + String(TS_value1));
  }
}
```



```
else{
    Serial.println("Problem reading FIELD 1. HTTP error code " + String(statusCode));
}
delay(1000);
// Leer estado del sistema TA_value2 de la grafica 2////////////////////
TA_value2S = ThingSpeak.readLongField(counterChannelNumber, TA_Fiel2Number, myCounterReadAPIKey);
// chequeamos si la operacion se llevo correctamente////////
statusCode = ThingSpeak.getLastReadStatus();
if(statusCode == 200){
    TA_value2=TA_value2S;
    if(sepromm2!=(int)TA_value2){
        sepromm2=(int)TA_value2;
        EEPROM.put(4, sepromm2);
        EEPROM.commit();
        Serial.println("SETB" + String(TA_value2) + "guardado" + String(sepromm2));
    }
    else{
        //mySerial2.println("SETB" + String(TA_value2));
    }
}
else{
    Serial.println("Problem reading FIELD 2. HTTP error code " + String(statusCode));
}
delay(1000);
// Leer estado del sistema HS_value3 de la grafica 3////////////////////
HS_value3S = ThingSpeak.readLongField(counterChannelNumber, HS_Fiel3Number, myCounterReadAPIKey);
// chequeamos si la operacion se llevo correctamente////////
statusCode = ThingSpeak.getLastReadStatus();
if(statusCode == 200){
    HS_value3=HS_value3S;
    if(sepromm3!=(int)HS_value3){
        sepromm3=(int)HS_value3;
        EEPROM.put(8, sepromm3);
        EEPROM.commit();
        Serial.println("SETC" + String(HS_value3) + "guardado");
    }
    else{
        //Serial.println("SETC" + String(HS_value3));
    }
}
else{
    Serial.println("Problem reading FIELD 3. HTTP error code " + String(statusCode));
}
}
```



```
delay(1000);
// Leer estado del sistema ON/OFF de la grafica 8////////////////////
SO_R_value8S = ThingSpeak.readLongField(counterChannelNumber, SO_Fiel8Number, myCounterReadAPIKey);
// chequeamos si la operacion se llevo correctamente////////
statusCode = ThingSpeak.getLastReadStatus();
if(statusCode == 200){
    SO_R_value8=SO_R_value8S;
    SO_W_value8=SO_R_value8;
    //mySerial2.println("SETD" + String(SO_R_value8));
}
else{
    Serial.println("Problem reading FIELD 8. HTTP error code " + String(statusCode));
}
delay(3000);
SO_W_value8=1;
//enviando nuevas actualizaciones  NTAxxxSETBxxxSETCxxxSETDxx
mySerial2.printf("NTA%03dSETB%03dSETC%03dSETD%02d\r\n",TS_value1,TA_value2,HS_value3,SO_R_value8);
Serial.printf("NTA%03dSETB%03dSETC%03dSETD%02d\r\n",TS_value1,TA_value2,HS_value3,SO_R_value8);
delay(12000);
////////////////////////extraccion de nuevos datos de el invernadero////////////////////////////////////
//FORMATO DE EL PAQUETE RECIBIDO// TAxxxHAxxxHSxxxILxxxxxxSOxx

if(mySerial2.available() != 0){
    SensorData = mySerial2.readString(); //extraendo todo el dato del buffer
    String TH_sub = SensorData.substring(2, 5);
    HA_value4 = TH_sub.toInt();
    TH_sub = SensorData.substring(7, 10);
    IL_value5 = TH_sub.toInt();
    TH_sub = SensorData.substring(12, 15);
    ET_value6 = TH_sub.toInt();
    TH_sub = SensorData.substring(17, 23);
    EE_value7 = TH_sub.toInt();
    TH_sub = SensorData.substring(25, 27);
    SO_W_value8 = TH_sub.toInt();
    //mySerial2.println(SensorData);
    Ndata = 0;
}
else{
    // mySerial2.println(SensorData);
    // mySerial2.println("No hay datos en el buffer de entrada");
}
Serial.println(SensorData);
delay(1000);
}
```



DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, Yo Yhonatan Quispe Condori
identificado con DNI 46829110 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado
Ingeniería Electrónica

informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:
" DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMÁTICO DE
CONTROL MICROCLIMATOLÓGICO PARA INVERNADEROS UTILIZANDO
MICROCONTROLADOR ARM CORTEX "

Es un tema original.

Declaro que el presente trabajo de tesis es elaborado por mi persona y **no existe plagio/copia** de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

Dejo constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumiré como tuyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

Asimismo, ratifico que soy plenamente consciente de todo el contenido de la tesis y asumo la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, me someto a las disposiciones legales vigentes y a las sanciones correspondientes de igual forma me someto a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 5 de junio del 2023

FIRMA (obligatoria)



Huella



AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, Yo Yhonatan Quispe Condori,
identificado con DNI 46829110 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

Ingeniería E Electrónica

informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:

“ DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMÁTICO DE CONTROL MICROCLIMATOLÓGICO PARA INVERNADEROS UTILIZANDO MICROCONTROLADOR ARM CORTEX ”

para la obtención de Grado, Título Profesional o Segunda Especialidad.

Por medio del presente documento, afirmo y garantizo ser el legítimo, único y exclusivo titular de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los “Contenidos”) que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, doy seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizo a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

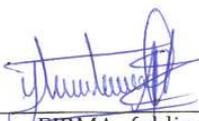
En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizo que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

En señal de conformidad, suscribo el presente documento.

Puno 5 de junio del 2023


FIRMA (obligatoria)



Huella