

Proyecto Final

**DISEÑO Y FABRICACIÓN DE UN
INVERNADERO INTELIGENTE, CON
CONTROL Y CORRECCIÓN DE LAS
CONDICIONES CLIMÁTICAS DENTRO DEL
MISMO**

Realizado por
Mazzeo Carlos Alejandro

Institución
Instituto Manuel Savio

Invernaderos CONAUT

(conectados y automatizados)

"Invernadero inteligente, equipado con sensores y tecnología de toma de decisiones, que cuida y acompaña tus plantas de manera especial."

Resumen

Un invernadero es un tipo de construcción muy habitual en el mundo de la agricultura. Generalmente, se trata de una estructura metálica cerrada cubierta con una película translúcida de plástico que permite la entrada de los rayos de sol, pero no de la lluvia, del viento o de otros factores climatológicos que podrían ser adversos para la producción.

El objetivo de estas estructuras es reproducir de la forma más fiel posible las condiciones climáticas ideales para el cultivo de un determinado tipo de planta, las cuales no podrían darse en el lugar de manera natural. Poseen el tamaño suficiente como para permitir el trabajo humano en el interior.

Al añadir la etiqueta “inteligente”, a lo que hacemos referencia es a invernaderos capaces de recrear esta situación ambiental idónea para el cultivo de forma autónoma. Para ello, es necesario el uso de sensores, actuadores y sistemas de riego automatizados, así como de un software que se ocupe de gestionarlo todo.

Al reducir al mínimo la intervención humana y actuar con base en datos objetivos y medibles, las probabilidades de tomar en todo momento las decisiones más adecuadas crecen considerablemente. Además, toda la información se puede gestionar de forma remota, por lo que ni siquiera es necesario que el agricultor se encuentre en el mismo lugar que el invernadero inteligente.

Los invernaderos CONAUT llegaron para cumplir con esas condiciones. Adaptados a cada presupuesto y a los requerimientos específicos de cada cliente.

Índice General

Contenido

Invernaderos CONAUT	2
(conectados y automatizados)	2
Resumen.....	3
1.Introducción	7
1.1 Panorama General de los Invernaderos en Latinoamérica y especialmente en Argentina	7
1.1.1 Crecimiento del sector	7
1.1.2 Principales Cultivos Crecimiento del sector	7
1.1.3 Tamaño del mercado	7
1.2 Argentina	8
1.2.1 Principales cultivos	8
1.2.2 Estadísticas	8
1.3 Ventajas	9
1.3.1 Control del clima	9
1.3.2 Mayor productividad	11
1.3.3 Extensión de la temporada de cultivo	11
1.3.4 Protección contra plagas y enfermedades	11
1.3.5 Uso eficiente del espacio	11
1.3.6 Mejor calidad de los productos	11
1.3.7 Reducción del uso de agua y fertilizantes	12
1.3.8 Diversificación de cultivos	12
1.4 Desventajas de los invernaderos en Argentina	12
1.4.1 Alta inversión inicial	12
1.4.2 Costos de mantenimiento	13
1.4.3 Consumo energético	13
1.4.4 Dependencia tecnológica	13
1.4.5 Riesgo de fallas técnicas	13

1.4.6 Impacto ambiental	13
1.4.7 Limitaciones en algunos cultivos	14
1.4.8 Clima local y regulación	14
1.4.9 Polinización	14
1.5 Incidencia en la Mano de Obra	14
1.5.1 Reducción de mano de obra intensiva	14
1.5.2 Mayor eficiencia y productividad	15
1.5.3 Especialización laboral	15
1.5.4 Trabajo más seguro	15
1.5.5 Flexibilidad en horarios	15
1.5.6 Impacto en empleo rural	15
2. Objetivos del Proyecto	17
3. Desarrollo del proyecto	19
3.1 Elección de los sensores	19
3.2 Sensores	21
3.2.1 Sensor de temperatura DS18B203	21
.....	21
3.2.2 Sensor de Humedad Relativa DHT22	27
3.2.3 Sensor capacitivo-analógico de Humedad en el suelo "Soil Moisture V1.2"	29
3.3 Elección del Microcontrolador	31
.....	33
3.4 Componentes adicionales	34
3.4.1 Relay	34
3.4.2 Componentes de bajo nivel	34
4. Desarrollo del prototipo	36
Función y desarrollo	36
4.1 Fuente	37
4.2 Microcontrolador	39
4.3 Sensor DHT22	42
4.4 Sensor DS18B20	42
4.5 Sensor Soil Moisture	43
4.6 Diodos leds, indicadores de estado	43
4.7 Accionamiento de relés	44

4.8 Interconexión de display LCD	46
4.9 Descripción del funcionamiento	46
5. Ensamble del prototipo	48
5.1 Primeros Pasos	48
5.2 Funcionamiento de los relés	49
5.3 Conectando varios DS18B20 con solo 3 cables	50
5.4 Funcionamiento del Soil Moisture	51
5.5 Funcionamiento del modo Manual-Automático	51
6. Conclusión	53
7. Bibliografía.....	55
Anexo I- Costos.....	57
Anexo II- Programa.....	58
Anexo III- Direcciones DS18B20.....	78
Anexo IV- Fabricación de la placa	80
Anexo V- Armado Final	83

1.Introducción

Los invernaderos son estructuras que permiten controlar las condiciones climáticas para optimizar la producción agrícola. En Latinoamérica, su uso ha ido en aumento debido a la necesidad de mejorar la productividad, reducir riesgos climáticos y aprovechar recursos de manera eficiente.

1.1 Panorama General de los Invernaderos en Latinoamérica y especialmente en Argentina

1.1.1 Crecimiento del sector

La adopción de invernaderos ha crecido en países como Argentina, México, Brasil, Chile y Perú.

Inversión en infraestructura de invernaderos (millones de USD)		
País	Inversión último año	Crecimiento respecto al año anterior
México	USD 150,00	12,00%
Brasil	USD 120,00	10,00%
Chile	USD 80,00	8,00%
Perú	USD 50,00	5,00%
Argentina	USD 10,00	5,00%

1.1.2 Principales Cultivos Crecimiento del sector

Tomates, pimientos, lechugas, flores y frutas tropicales.

1.1.3 Tamaño del mercado

Se estima que el mercado de invernaderos en la región ha crecido un 10% en los últimos 5 años. La inversión en infraestructura agrícola, el interés en cultivos fuera de temporada y la adopción de tecnologías modernas han impulsado este aumento. En algunos informes del sector agrícola, se menciona que la expansión de los

invernaderos en Argentina ha tenido un crecimiento anual aproximado del 5% durante ese período, aunque esta cifra puede variar según la fuente y la región específica.

1.2 Argentina

1.2.1 Principales cultivos

Este desglose es una estimación basada en las tendencias del mercado y la importancia de estos cultivos en invernaderos argentinos. Los tomates, por ejemplo, son uno de los cultivos más comunes en invernaderos en Argentina, seguidos por pimientos y lechugas.

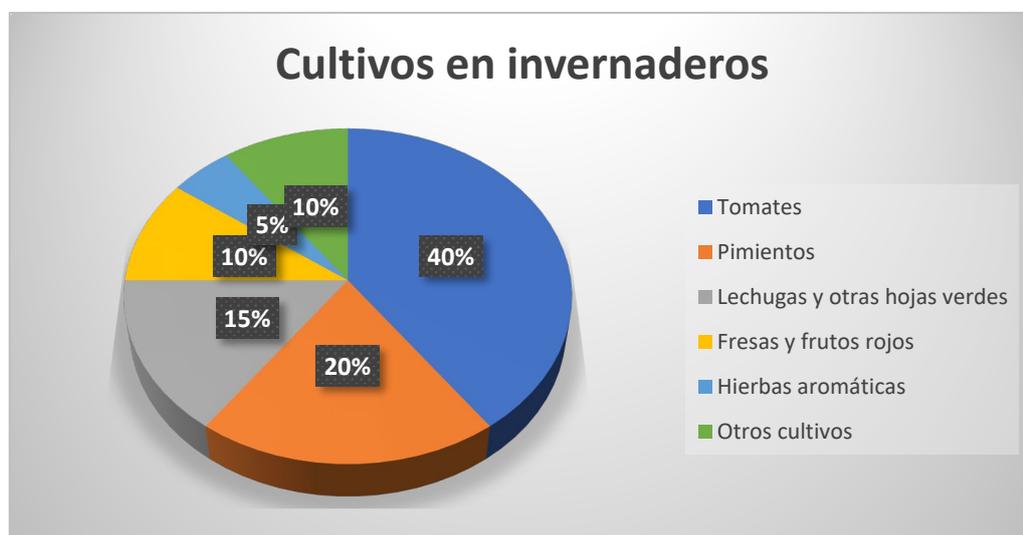


Figura 1: Producción Argentina en Invernaderos

1.2.2 Estadísticas

No hay datos específicos y actualizados sobre las estadísticas precisas de la producción argentina en invernaderos, incluyendo el valor en dólares anuales o el volumen de exportaciones derivadas de productos sembrados en invernaderos. Sin embargo, Argentina, ha estado incrementando su producción en invernaderos, especialmente en cultivos como tomates, pimientos y flores, ya que estas actividades contribuyen significativamente a la economía agrícola del país.

La razón por la que puede ser difícil encontrar estadísticas precisas sobre la producción en invernaderos en Argentina es que, en muchos casos, estos datos no se recopilan de manera centralizada o detallada por las instituciones oficiales. La producción en invernaderos a menudo se considera una parte específica de la agricultura y puede estar dispersa entre diferentes provincias, tipos de cultivos y pequeños productores, lo que dificulta obtener cifras globales precisas.

Además, algunos datos pueden estar en informes sectoriales o en estudios de mercado que no siempre están disponibles públicamente o que no se actualizan con frecuencia. También, la producción en invernaderos puede representar una proporción relativamente pequeña en comparación con la agricultura a campo abierto, lo que puede hacer que no sea prioritaria en los registros estadísticos nacionales.

Por estas razones, muchas veces las cifras disponibles son estimaciones o provienen de investigaciones específicas, en lugar de datos oficiales consolidados.

1.3 Ventajas

Los invernaderos ofrecen varias ventajas que los hacen muy útiles para la agricultura y la horticultura.

1.3.1 Control del clima

Las condiciones climáticas, las cuales los invernaderos inteligentes intentan recrear, son:

- **Temperatura:**
Mantener la temperatura adecuada es una de las funciones más básicas de los invernaderos, por muy tecnológicos o avanzados que sean.
Proporcionar temperaturas óptimas permite a las plantas absorber y descomponer los nutrientes de la forma más eficiente posible. Esto hace que el crecimiento y el desarrollo de los frutos sean más rápidos. Si las condiciones son demasiado frías o demasiado cálidas, las plantas sufren estrés, lo que afecta al crecimiento y a la calidad. En un entorno con temperatura controlada, no hay razón para alcanzar temperaturas fuera del rango ideal.

- Humedad relativa:

La humedad relativa es un parámetro climático crítico en el cultivo de cualquier cultivo. Por lo tanto, la mayoría de los cultivadores intentan mantener rangos de humedad óptimos, correspondientes a las plantas que cultivan.

El punto de ajuste de humedad relativa óptimo para la mayoría de las plantas es de alrededor del 80%. En este nivel, las tasas de crecimiento son más altas para las plantas comunes de invernadero. A niveles de humedad más altos o más bajos, los procesos fisiológicos de las plantas pueden ralentizarse, lo que lleva a un crecimiento más lento y una producción de menor calidad.

Los altos niveles de humedad relativa también aumentan drásticamente la susceptibilidad a enfermedades comunes de la humedad como la botritis o el mildiu polvoriento.

La humedad es un parámetro climático que los productores deben comprender, controlar y mantener para adaptarse a sus cultivos objetivo, en lugar de simplemente intentar reducirla al mínimo.

- Humedad del suelo:

La humedad del suelo y el riego son cruciales en los invernaderos para el desarrollo de las plantas. La humedad adecuada permite a las plantas mantener sus estomas abiertos para la fotosíntesis, absorbe nutrientes y regula la temperatura. El riego correcto, a su vez, asegura que las plantas tengan acceso al agua que necesitan para crecer y prosperar, mientras que un riego eficiente evita el desperdicio y reduce costos.

- CO₂:

El dióxido de carbono (CO₂) es esencial para la fotosíntesis, el proceso mediante el cual las plantas convierten la luz en energía. En el exterior los niveles de CO₂ se encuentran entre 300 y 400ppm, en un cultivo de interior es importante renovar el aire para reponer el CO₂ que consumen las plantas o se puede suplementar con CO₂ envasado, lo que aumenta hasta un 30% el crecimiento y el rendimiento de las plantas.

Aumentar el nivel de CO₂ puede mejorar la tasa de fotosíntesis, acelerando el crecimiento. Si el nivel de CO₂ es bajo, la fotosíntesis se ralentiza, lo que puede llevar a un crecimiento subóptimo y una menor producción. Aunque los niveles elevados de CO₂ pueden ser beneficiosos, concentraciones superiores pueden ser tóxicas para las plantas y peligrosas para los trabajadores.

Además de estas condiciones climáticas muy necesarias de ser controladas, también encontramos otras (circulación del aire, iluminación, pH, oxígeno disuelto, etc.) tan importantes como las anteriores, el control conjunto de todas, aran que nuestro cultivo tenga un rendimiento optimo.

1.3.2 Mayor productividad

Al ofrecer condiciones óptimas, los cultivos en invernadero suelen tener un crecimiento más rápido y una mayor producción en comparación con los cultivos al aire libre.

1.3.3 Extensión de la temporada de cultivo

Los invernaderos permiten cultivar durante todo el año, incluso en temporadas en las que normalmente no sería posible, aumentando la disponibilidad de productos.

1.3.4 Protección contra plagas y enfermedades

Al estar cerrados o semi cerrados, los invernaderos ayudan a reducir la incidencia de plagas y enfermedades, disminuyendo la necesidad de pesticidas.

1.3.5 Uso eficiente del espacio

Se pueden aprovechar mejor los recursos y el espacio, permitiendo cultivos en áreas más pequeñas y con mayor densidad.

1.3.6 Mejor calidad de los productos

Los cultivos en invernadero suelen tener una apariencia más uniforme y una calidad superior, lo que puede traducirse en mejores precios en el mercado.

1.3.7 Reducción del uso de agua y fertilizantes

La gestión controlada permite un uso más eficiente de estos recursos, promoviendo prácticas agrícolas más sostenibles.

1.3.8 Diversificación de cultivos

Facilitan el cultivo de especies que requieren condiciones específicas o que no son típicas de la región, ampliando la variedad de productos disponibles.

1.4 Desventajas de los invernaderos en Argentina

Aunque los invernaderos ofrecen muchas ventajas, también tienen sus contras y desafíos que es importante considerar antes de su instalación.

1.4.1 Alta inversión inicial

La construcción y equipamiento de un invernadero puede requerir una inversión significativa en infraestructura, tecnología y sistemas de control climático. Esto puede ser un obstáculo para pequeños productores.

El costo promedio de un invernadero en Argentina puede variar bastante dependiendo de varios factores, como el tamaño, el tipo de estructura, los materiales utilizados, la tecnología incorporada y la ubicación. Sin embargo, podemos dar una idea general para tener una referencia amigable y útil.

En términos generales, el costo de un invernadero simple y de tamaño mediano (por ejemplo, alrededor de 100 a 200 metros cuadrados) puede oscilar entre \$1.500.000 y \$4.000.000 de pesos argentinos aproximadamente (de Usd 1.000 a Usd 3.000). Este rango incluye la estructura básica, cubiertas, sistemas de ventilación y algunos sistemas de riego.

Para invernaderos más grandes, con tecnología avanzada, control climático automatizado y materiales de alta calidad, los costos pueden subir considerablemente, llegando a \$7.000.000 (aproximadamente unos Usd 5.500) o más dependiendo de la escala y las especificaciones.

Es importante tener en cuenta que estos valores son aproximados y que los precios pueden variar según la región, la disponibilidad de materiales y la mano de obra.

Además, no debemos olvidar considerar los costos adicionales como permisos, instalación, sistemas de control y posibles mejoras futuras.

1.4.2 Costos de mantenimiento

Los invernaderos necesitan mantenimiento constante, como reparación de estructuras, limpieza, control de plagas y actualización de sistemas tecnológicos, lo que implica gastos continuos.

1.4.3 Consumo energético

La utilización de sistemas de calefacción, iluminación artificial y ventilación puede incrementar el consumo de energía, afectando los costos y el medio ambiente.

1.4.4 Dependencia tecnológica

La gestión eficiente requiere conocimientos técnicos y tecnología avanzada. La falta de capacitación puede limitar el rendimiento y aumentar los riesgos.

1.4.5 Riesgo de fallas técnicas

Problemas en los sistemas de control climático, riego o iluminación pueden afectar la producción y causar pérdidas importantes.

1.4.6 Impacto ambiental

La construcción y operación de invernaderos puede tener efectos negativos si no se gestionan adecuadamente, como el uso excesivo de recursos o la generación de residuos.

1.4.7 Limitaciones en algunos cultivos

No todos los cultivos se adaptan bien a las condiciones de un invernadero, y algunos pueden requerir condiciones específicas difíciles de mantener.

1.4.8 Clima local y regulación

En algunas regiones, las condiciones climáticas extremas o las regulaciones ambientales pueden complicar la instalación y operación de invernaderos. A pesar de estos desafíos, muchas personas y empresas consideran que los beneficios superan los inconvenientes, especialmente si se planifica cuidadosamente y se invierte en tecnologías sostenibles.

1.4.9 Polinización

Algunos cultivos necesitan polinizadores naturales o técnicas específicas para una buena producción, lo cual puede ser complicado en ambientes cerrados.

1.5 Incidencia en la Mano de Obra

Los invernaderos han transformado significativamente la mano de obra en la agricultura, generando cambios en varias áreas clave.

1.5.1 Reducción de mano de obra intensiva

Los invernaderos permiten automatizar muchas tareas tradicionales como riego, fertilización, control climático y cosecha mediante sistemas mecánicos y tecnológicos. Esto reduce la necesidad de mano de obra manual intensiva, permitiendo que menos trabajadores realicen tareas más especializadas.

1.5.2 Mayor eficiencia y productividad

La automatización y el control preciso del ambiente facilitan una producción más rápida y consistente, lo que disminuye el tiempo y esfuerzo requeridos para cultivar y cosechar los cultivos.

1.5.3 Especialización laboral

La operación de invernaderos requiere personal capacitado en tecnología, sistemas de control ambiental, manejo de plagas y enfermedades, lo que ha llevado a una mayor especialización laboral en comparación con la agricultura tradicional.

1.5.4 Trabajo más seguro

Al reducir las tareas manuales peligrosas o repetitivas, los invernaderos pueden mejorar las condiciones laborales y disminuir riesgos asociados a trabajos agrícolas tradicionales.

1.5.5 Flexibilidad en horarios

La automatización permite realizar ciertas tareas en horarios no dependientes del clima o del ciclo natural del cultivo, facilitando una gestión más flexible del trabajo.

1.5.6 Impacto en empleo rural

En algunos casos, la introducción de invernaderos puede reducir empleos tradicionales en zonas rurales si no se complementa con nuevas oportunidades laborales relacionadas con la tecnología y el mantenimiento de los sistemas.

En resumen, los invernaderos han llevado a una transformación hacia una mano de obra más especializada, eficiente y segura, aunque también plantean desafíos

relacionados con la reducción de empleos tradicionales en ciertos contextos agrícolas.

2. Objetivos del Proyecto

En Invernaderos CONAUT buscamos la integración de las nuevas tecnologías (sensores, IA, IoT, microcontroladores, etc.) para lograr un óptimo desarrollo de las plantas en cada proceso. En CONAUT no tienes que adaptar tu invernadero a nuestro modelo, sino que nosotros nos adaptamos a tus necesidades.

Nuestros equipos se adaptan a todas tus necesidades y tamaños, desde invernaderos familiares (con equipos más económicos y sencillos) a invernaderos comerciales con requerimientos mucho más específicos, sofisticados, automatizados, con más calidad y precisión en la lectura de las condiciones climáticas y por tal motivo, mejor toma de decisiones.

Por ejemplo, podemos hablar de la temperatura. Mantener la temperatura adecuada difiere de un invernadero a otro, algunos están situados en climas más cálidos que otros. Así, aunque la mayoría de los invernaderos requiera calefacción, algunos necesitan refrigeración para alcanzar el rango ideal. Se pueden utilizar conductos, bombas de calor, calderas, enfriadoras o un sistema HVCA (heating, ventilation and air conditioning) o en castellano, calefacción, ventilación y aire acondicionado.

La implementación de nuestro invernadero inteligente se basa en conceptos de agricultura de precisión, que busca maximizar el rendimiento y la calidad de los cultivos mediante el uso de tecnologías digitales.

Según estudios recientes, estos sistemas permiten reducir el consumo de recursos como agua y energía, además de disminuir el uso de pesticidas, promoviendo una agricultura más sostenible y respetuosa con el medio ambiente.

Además, la integración de tecnologías IoT (Internet de las Cosas) en los invernaderos facilita la monitorización continua y la toma de decisiones automatizadas, lo que resulta en una gestión más eficiente y adaptada a las condiciones específicas de cada cultivo. La inteligencia artificial, por su parte, permite predecir patrones de crecimiento y detectar posibles problemas antes de que se vuelvan críticos.

En resumen, los invernaderos inteligentes representan una evolución significativa en la agricultura, combinando tecnología y sostenibilidad para ofrecer soluciones más eficientes, productivas y responsables con el medio ambiente

3. Desarrollo del proyecto

Para el desarrollo del invernadero se ha pasado por un constante análisis y estudio de los diferentes tipos de sensores que se encuentran en el mercado, buscando el balance ideal entre eficacia, precio y uso.

3.1 Elección de los sensores

La eficacia y el uso están directamente vinculado con su precio. A tal efecto, si necesitáramos desarrollar un invernadero para uso comercial deberíamos contar con sensores 4-20 mA

Los sensores de 4-20mA son dispositivos que convierten una medida física (como presión, temperatura, nivel) en una señal de corriente de 4 a 20 miliamperios (mA). Esta señal es ampliamente utilizada en sistemas de control y automatización industriales debido a su estabilidad y facilidad de transmisión a largas distancias.

Ventajas de estos sensores:

- Estabilidad de la señal:
La señal de corriente es menos susceptible a los efectos del ruido y las caídas de voltaje en el cableado, lo que la hace ideal para aplicaciones industriales.
- Longitud de cableado:
Pueden transmitir señales a largas distancias sin pérdida significativa de calidad.
- Fácil de conectar:
La mayoría de los sensores de 4-20mA utilizan una configuración de dos cables, lo que simplifica la instalación y el mantenimiento.
- Alimentación por bucle:
La corriente de 4-20mA también puede alimentar el sensor, lo que elimina la necesidad de una fuente de alimentación externa en algunos casos.

¿Cómo funciona un sensor de 4-20mA?

El sensor mide una magnitud física y la convierte en una señal de corriente de 4-20mA. El valor de la corriente varía linealmente con la magnitud medida. Por ejemplo, una corriente de 4mA puede representar el valor mínimo de la magnitud medida, y una corriente de 20mA puede representar el valor máximo.

De esta forma lo podemos conectar a un PLC (Programmable Logic Controller), utilizando la entrada analógica del PLC para leer la señal de corriente. La señal de 4-20mA se conecta a la entrada analógica, y el PLC interpreta el valor de la corriente como un valor de la magnitud medida.

En resumen, estos sensores son una herramienta fundamental en la automatización industrial, ofreciendo una señal robusta y precisa para la medición y el control de diversos parámetros físicos. Por tal motivo, el costo de estos, es considerablemente mucho más alto que el de los implementados en un invernadero familiar. El costo puede llegar a ser de 20 a 200 veces más caros:

- ❖ Ds18b20
 - Sensor de temperatura
 - Gama baja, ideal para uso familiar
 - Precio promedio \$2.500

- ❖ Modbus Rtu Rs485 Th-mb-03s-h
 - Sensor de temperatura
 - Gama media
 - Precio promedio \$69.000

- ❖ Pt1000 Salida 4-20ma Balluff Bft0006
 - Sensor temperatura
 - Gama alta
 - Precio promedio \$480.000

Los precios fueron obtenidos de Mercado Libre, ya que este ofrece precios de referencias al alcance de cualquier persona.

Explicada la diferencia notoria entre el costo de desarrollar, de modo experimental, un invernadero se ha decidido desarrollar y construir un modelo familiar. Ya que, solo

con pequeños ajustes de programación y construcción, se puede adaptar de una manera muy sencilla a las exigencias más altas.

3.2 Sensores

A continuación, se hará un detalle de los sensores utilizados en este proyecto

3.2.1 Sensor de temperatura DS18B20

El sensor DS18B20 es un sensor barato y, sin embargo, bastante avanzado. Dispone de un rango amplio de medición de -55°C a $+125^{\circ}\text{C}$ y una precisión superior a $\pm 0.5^{\circ}\text{C}$ en el rango -10°C de $+85^{\circ}\text{C}$.

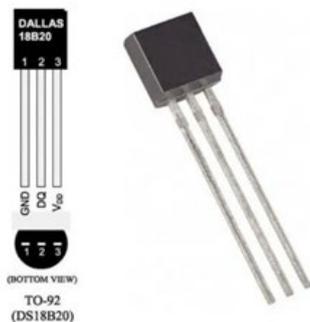


Figura 3: Componente DS18B20



Figura 2: Componente DS18B20 encapsulado

Una de las ventajas del DS18B20 es que se comercializa tanto en un integrado TO-92 (figura 2) como en forma de sonda impermeable (figura 3), lo que permite realizar mediciones de temperatura en líquidos y gases.

El DS18B20 emplea un bus de comunicación denominado 1-Wire propietario de la empresa Maxim Integrated, aunque podemos usarlo sin tener que pagar por ninguna tasa (es parte del precio del dispositivo).

La principal ventaja del bus 1-Wire es que necesita un único conductor para realizar la comunicación (sin contar el conductor de tierra). Los dispositivos pueden ser alimentados directamente por la línea de datos, o mediante una línea adicional con una tensión de 3.0 a 5.5V.

Dentro del mismo bus 1-Wire podemos instalar tantos sensores como deseemos. Además, el bus 1-Wire permite emplear cables más largos que otros sistemas antes de que se deteriore la comunicación.

El DS18B20 también dispone de un sistema de alarma que permite grabar en la memoria no volátil del DS18B20 los límites inferiores y superiores. El bus 1-Wire permite consultar si algún dispositivo conectado ha activado una alarma.

Es un gran sensor para la medición de temperatura, tanto en ambientes domésticos como industriales. Sus características permiten crear redes con gran número de sensores para controlar, por ejemplo, la climatización o el sistema HVAC de un edificio comercial.

Cada sensor tiene un código serial de 64 bits que permite conectar múltiples sensores.

Con un simple ajuste de resistencias podemos alcanzar una buena distancia de cobertura

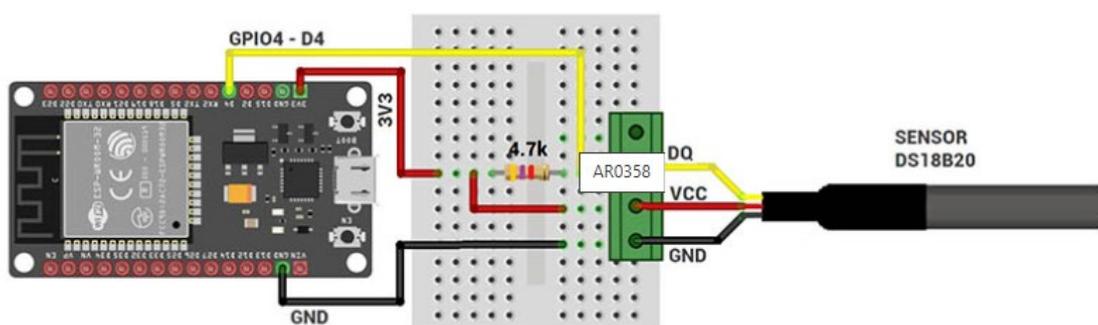


Figura 4: Conexión del DS18B20 con el ESP32

4.7 kΩ----- 0 a 5 m
 3.3 kΩ----- 5 a 10 m
 2.2 kΩ----- 10 a 20 m
 1.2 kΩ----- 20 a 50 m

Como hemos dicho, internamente el sensor DS18B20 es más complicado de lo que en principio podríamos creer. Está formado por un procesador con múltiples módulos, que se encargan de controlar la comunicación, medir la temperatura, y gestionar el sistema de alarmas.

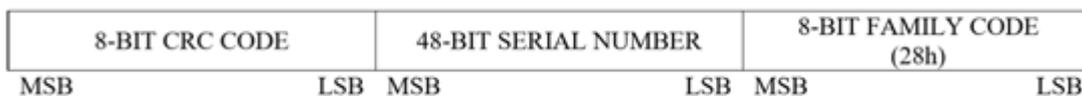
Una de las principales ventajas de DS18B20 es su bus de comunicación 1-Wire que **le permite realizar la transmisión empleando únicamente un cable de datos**. Para ello, 1-Wire está basado en un complejo sistema de timings en la señal, entre el dispositivo emisor y el receptor.

La mayor desventaja del sistema 1-Wire es que requiere un código complejo, lo que a su vez **supone una alta carga del procesador** para consultar el estado de los sensores. El tiempo de adquisición total de una medición de 750ms.

El dispositivo 1-Wire permite que todos los dispositivos conectados al bus se alimenten a través de la línea de datos. Para ello, disponen de un condensador que almacena energía mientras la línea de datos está en HIGH. Este modo se denomina «modo parásito». En caso de no usar el modo parásito, los dispositivos deberán ser alimentados a una tensión entre 3.0V y 5.5V.

En cualquier caso, **el bus 1-Wire requiere una resistencia de Pull-up de 4k7** entre Vcc y Vq para que funcione correctamente.

Para poder disponer de múltiples dispositivos en un bus 1-Wire cada sensor dispone de **una memoria ROM que es grabada de fábrica con un número de 64 bits**. Los 8 primeros bits corresponden a la familia (0x28 para el DS18B20). Los siguiente 48 bits son el número de serie único. Los últimos 8 bits son un código CRC.



Esto significa que potencialmente podríamos llegar a tener 2⁴⁸ (más de 218 billones) de DS18B20 conectados en una misma red 1-Wire, además de otros tantos dispositivos de otras familias (2²⁸, unos 268 millones de familias de dispositivos posibles). A efectos prácticos, **esto supone infinitos dispositivos**.

La resolución del DS18B20 es configurable a 9, 10, 11 o 12 bits, siendo 12 bits el modo por defecto. Esto equivale a una resolución en temperatura de 0.5°C, 0.25°C, 0.125°C, o 0.0625°C, respectivamente.

Que el DS18B20 disponga de una resolución por defecto de 0.0625°C, no significa que esa sea su precisión. Sin embargo, **el DS18B20 es considerablemente preciso en todo el rango -10°C de +85°C**. Pueden consultar las desviaciones medias a 3 sigma en la siguiente gráfica.

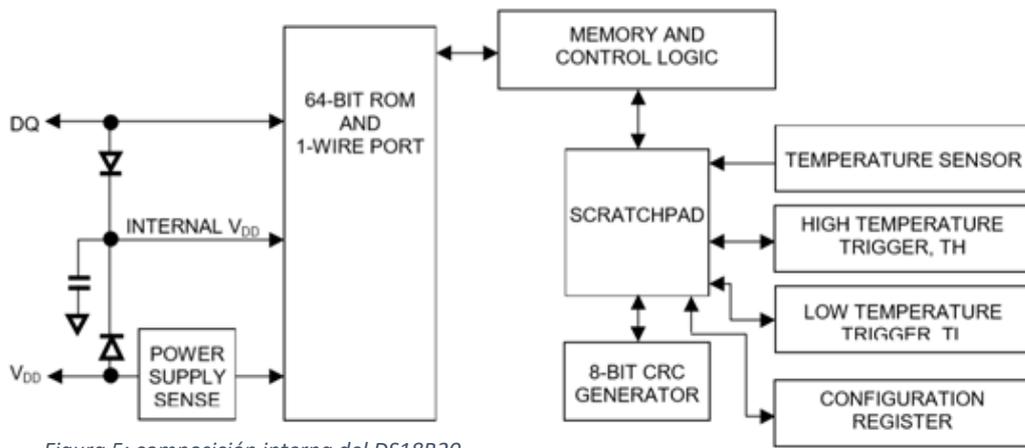


Figura 5: composición interna del DS18B20

Esquema de montaje

Los dispositivos 1-Wire disponen de tres terminales

- DQ la línea de datos
- VDD línea de alimentación
- GND línea de tierra

En la siguiente imagen figura la posición de estos pines en ambos formatos del sensor, como integrado y como sonda impermeable.



Figura 6: Pin-out DS18B20

Hemos comentado que el bus 1-Wire necesita una resistencia de pull-up de 4K7, y que podemos alimentar el sensor directamente a través del pin VDD o usar el modo «parásito» y alimentarlo con la propia línea de datos.

Por tanto, el esquema más simple de conexión se muestra en la siguiente imagen, donde podemos conectar el sensor a cualquier entrada digital de Arduino.

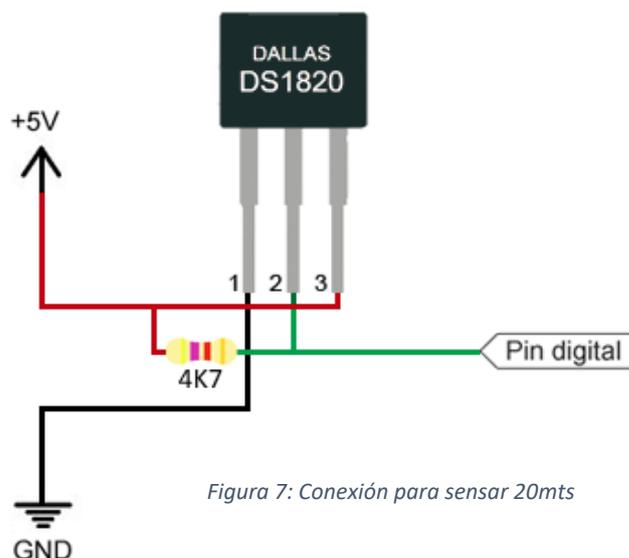


Figura 7: Conexión para sensor 20mts

Si representamos el mismo montaje, añadiendo la disposición que ocuparían los sensores adicionales (en caso de necesitarlos) observamos más claramente la estructura del bus 1-Wire.

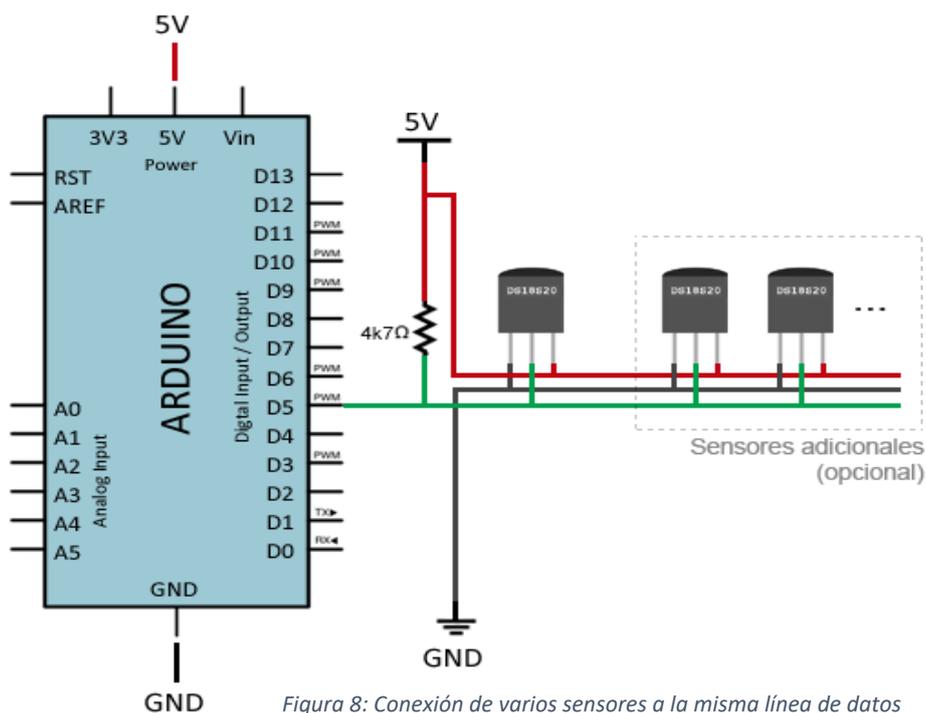


Figura 8: Conexión de varios sensores a la misma línea de datos

Finalmente, si quisiéramos emplear el modo parásito, simplemente tenemos que conectar VDD de todos los dispositivos a GND. De esta forma, los sensores tomarán su alimentación de la línea de datos DQ. El montaje con el modo parásito quedaría así.

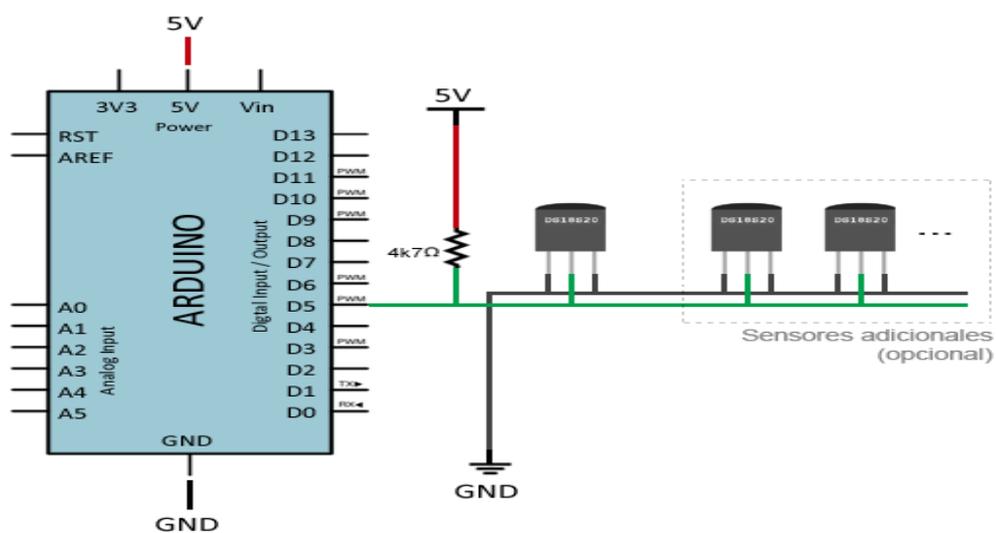


Figura 9: Conexión de varios sensores a la misma línea de datos en forma parasitaria

3.2.2 Sensor de Humedad Relativa DHT22

Para empezar, vamos a aclarar porque decidimos usar el DHT22 en lugar de su versión más conocida DHT11.

Por lo general, a simple vista la única diferencia entre el DHT11 y el DHT22 es que el primero viene en una carcasa azul y el segundo es blanco. De hecho, ambos son hermanos de una misma familia de sensores.

El **DHT11 es el hermano pequeño**, es decir, tiene algunas carencias o prestaciones de menos con respecto al DHT22, y por tanto un precio superior. El DHT11 se puede usar para proyectos donde no requieras una alta precisión en las mediciones, mientras que si quieres algo más preciso deberías elegir el DHT22. Realmente el 22 tampoco es de alta precisión, pero sí que tiene unas prestaciones más que aceptables para la mayoría de proyectos DIY de makers.

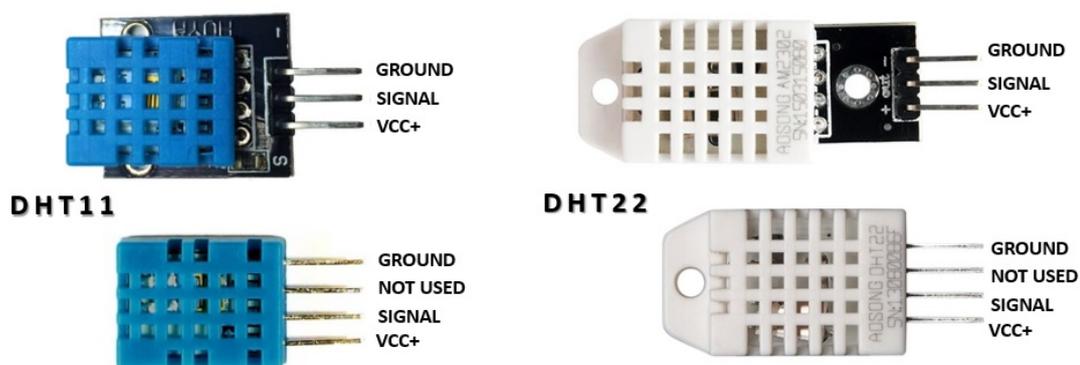


Figura 10: Sensor DHT11 y DHT22

El DHT22 es un sensor de temperatura y humedad con unas prestaciones que lo acercan mucho a los de alta precisión. Lo puedes encontrar fácilmente en tiendas especializadas o grandes superficies. Eso te permite no tener que depender de un sensor de temperatura y otro de humedad por separado, sino tenerlo todo integrado en un mismo dispositivo. Aunque para este proyecto solo lo usaremos para medir la humedad relativa ambiental.

Lo puedes encontrar suelto o en módulos especialmente diseñados para Arduino, es decir, el DHT22 montado sobre una placa PCB ya lista para usar, sin tener que agregar resistencias pull-up, etc. Hasta aquí todo se parece bastante al DHT11. Y también tendrás una alta fiabilidad y estabilidad en las mediciones debido a la señal digital calibrada que usa.

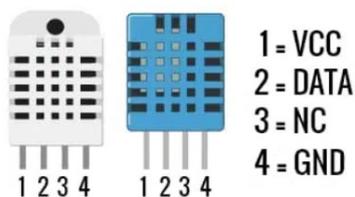


Figura 11: Pin-out DHT11 / DHT22

En la figura 11 podemos ver una comparación del pin-out del DHT22 y DHT11, y como puedes apreciar son idénticos en cuanto a pines. Por eso, su montaje sería exactamente igual, y lo mejor, podrías sustituir en cualquier momento el DHT11 por un DHT22, y viceversa, en tu proyecto sin hacer demasiados cambios.

Tiene 3 pines para usar: GND, VCC y Datos. El pin #3 no se usa y en los módulos viene anulado, es decir, solo se verán tres pines.

Aunque la mayor parte de valores pueda parecerse igual, podría haber alguna pequeña variación de uno a otro. Sus características técnicas más importantes son:

- Alimentación de 3,3v a 6v
- Consumo de corriente de 2,5mA
- Señal de salida digital
- Rango de temperatura de -40°C a 125°C
- Precisión para medir temperatura a 25°C de 0.5°C de variación
- La resolución para medir temperatura es de 8-bit, 0,1°C
- La humedad puede medir desde 0% RH hasta los 100% RH
- Con precisión para la humedad del 2-5% RH para temperaturas que se encuentren entre 0-50°C
- La resolución es de 0,1% RH, no puede captar variaciones por debajo de esa
- Frecuencia de muestreo de 2 muestras por segundo: 2Hz

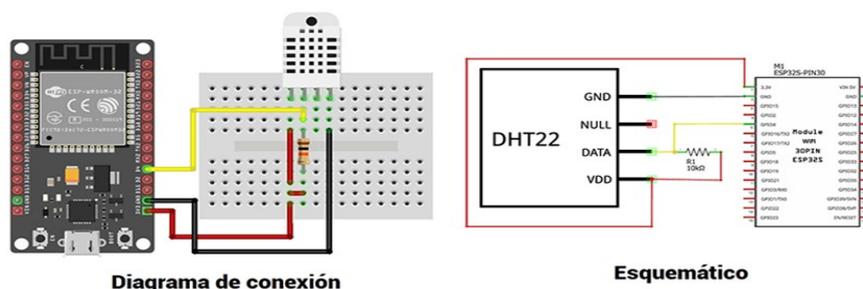


Figura 12: Diagrama conexión entre un DHT22 y un ESP32

3.2.3 Sensor capacitivo-analógico de Humedad en el suelo “Soil Moisture V1.2”

El Sensor de humedad de suelo capacitivo v1.2 permite medir la humedad en el suelo utilizando el principio de capacitancia entre electrodos en lugar de resistencia, lo que aumenta considerablemente la vida útil del sensor. El electrodo posee una capa de protección anticorrosión para una mayor duración (no se oxida como el sensor resistivo).

Compatible con Arduino, PIC, ESP8266/NodeMCU/NodeMCU-32. El sensor es ideal para monitorear el nivel de humedad de tus plantas y así recordar cuando necesitan ser regadas o incluso para realizar un sistema totalmente automatizado de riego añadiendo una válvula o una bomba de agua. Si el sistema se conecta a Internet podríamos controlar/monitorear nuestro jardín desde cualquier lugar del mundo

Hygrometer Modul V1.2 Datenblatt



Figura 13: Sensor Capacitivo de humedad del suelo

El funcionamiento del sensor se basa en medir la capacitancia entre 2 electrodos insertados dentro del suelo, la capacitancia entre los electrodos dependerá de la humedad del suelo, por lo que para un suelo muy húmedo tendremos una capacitancia muy baja y para un suelo muy seco la capacitancia será muy alta. El electrodo va conectado a una tarjeta de acondicionamiento que entrega una salida analógica. La salida analógica (AO) entrega un voltaje analógico desde 0V para un suelo muy húmedo hasta 5V para un suelo muy seco.

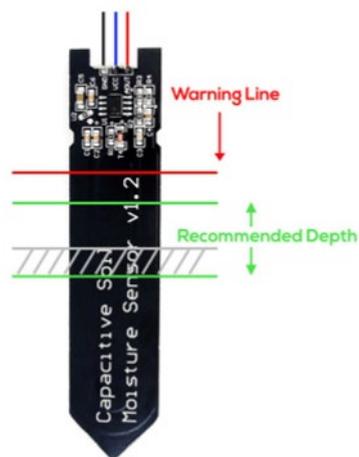


Figura 14: Como recomendación solo se debe enterrar la parte de los electrodos

Especificaciones Técnicas

- Voltaje de alimentación: 3.3V - 5V DC
- Corriente operación: 5mA
- Voltaje de la señal de salida: 0 a 5V (Analógico)
- Modelo: capacitive Soil Moisture sensor v1.2
- Vida útil: 3 años mín.
- Conector: PH2.0-3P
- Incluye: Electrodo y cable jumper hembra
- Dimensiones: 98*23 mm
- Peso: 15 gramos

Pines de conexión

- GND: Tierra (GND 0V)
- VCC: Voltaje de alimentación (3.3V - 5V DC)
- OUT: Salida analógica



Figura 15: Área donde se encuentran los electrodos

3.3 Elección del Microcontrolador

Además de hablar de los sensores, que se han elegido para el desarrollo, también debemos hablar del microcontrolador que se usará para este diseño. Se ha optado por usar el ESP32 por sus múltiples beneficios y prestaciones, las cuales desarrollaremos a continuación.

- El Esp32 es una familia de Microcontroladores, fabricados por la marca china Espressif Systems de bajo costo
- Poco consumo de energía
- Wi-fi y bluetooth integrados
- Es ideal para proyectos IoT (internet de las cosas)
- Microprocesador de 1 o 2 núcleos
- RAM y memoria flash, para almacenar código y datos

Sus 3 partes más importantes:

- Un chip, en el cual se encuentra el microprocesador, digamos que son el cerebro de nuestro ESP.



Figura 16: Chip ESP32

- Después tenemos los módulos (figura 17), que incluyen dentro, nuestro chip anterior. También incluyen una memoria flash, que suele ser de 4 MB o incluso una memoria PSRAM de 8 MB. Lo más significativo es que cuentan con una antena impresa en la placa que se usa, para las conexiones WI-FI y bluetooth. Los módulos también facilitan la integración del ESP en otras placas, ya que incluyen, los pines entrada-salida listos para soldaduras SMD.

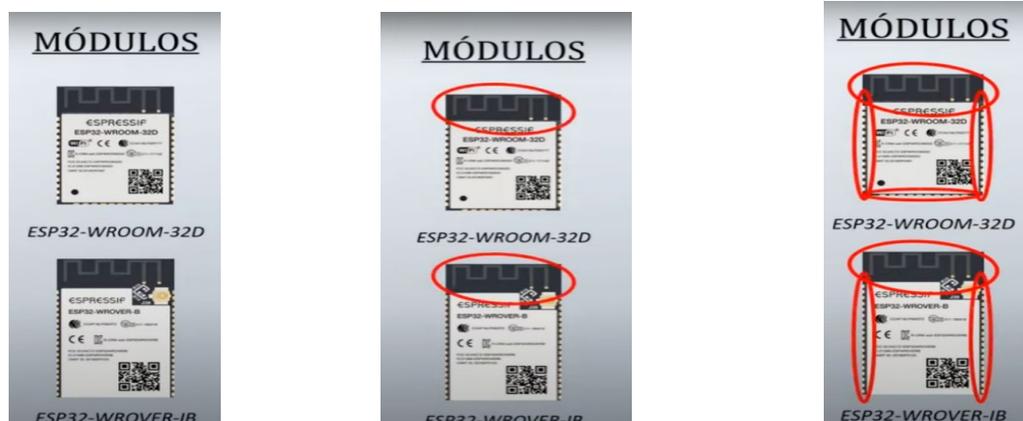


Figura 17: Módulos ESP32

- por último, tenemos a las placas de desarrollo, que son las más conocidas popularmente.

Estas ya traen soldado el módulo y facilita mucho su integración en prototipos, esto es gracias a los pines diseñados para encajar en una protoboard; también la podemos añadir en nuestros proyectos a través de agujeros pasantes.

Se destaca por su capacidad de realizar conexiones inalámbricas.



Figura 18: Placa de desarrollo para los ESP32

Para este proyecto se ha elegido usar un Esp32s Nodemcu Modulo Wifi Bluetooth 4.2 38 Pines Con Micro-usb Wroom.

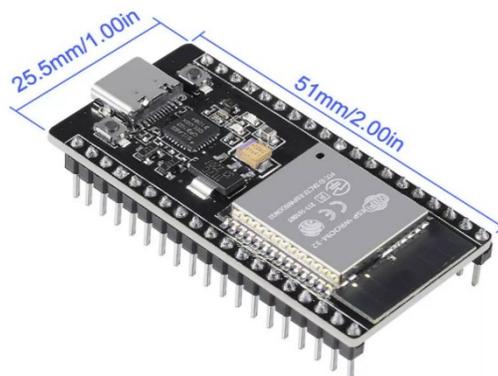


Figura 19: Placa desarrollo ESP32 elegida para

3.4 Componentes adicionales

Además de los componentes, anteriormente mencionados, debemos hacer hincapié en otros elementos que formaran parte y son igual de importantes.

Entre estos elementos encontramos:

3.4.1 Relay

Se ha seleccionado por múltiples cualidades (conectividad, capacidad de carga eléctrica, versatilidad y relación costo-calidad) un relay con zócalo para riel din, de 4 Contactos Inversores, con Bobina 24vdc.



Figura 20: Relay 24VCA con zócalo para riel din

3.4.2 Componentes de bajo nivel

- 6 resistencias 39K Metal film ¼ w 5%
- 6 resistencias 120K Metal film ¼ w 5%
- 6 resistencias 330Ω Metal film ¼ w 5%
- 6 resistencias 150Ω Metal film ¼ w 5%
- 2 resistencias 10K Metal film ¼ w 5%
- 1 resistencias 2K2 Metal film ¼ w 5%
- 6 transistores Bc548d
- 6 diodos led rojos
- 6 diodos led verdes
- 6 diodo rectificador 1N4007
- 6 transistores Npn Bipolares bc548b 100ma 30v 500mw



- 1 Puente Rectificador W10m 1.5a 1000v T-mesa (redondo)
- 1 placa Virgen Pcb 10x20cm Fenólica Pertinax Simple Faz
- 6 relés Bobina 12v 1a 120v Doble Inversor Hjr1-2c L-12v
- 1 capacitores electrolíticos 100uf
- 1 capacitores electrolíticos 470uf
- 1 capacitores electrolíticos 0.33uf
- 3 capacitores cerámicos 100nf
- 11 borneras para Pcb de 2 contactos
- 2 borneras para Pcb de 3 contactos
- 1 regulador de tensión 7812
- 1 regulador de tensión 7805



4. Desarrollo del prototipo

Función y desarrollo

Ya dejado atrás el detalle de los componentes principales de nuestro proyecto, y de una breve explicación de lo que es un invernadero, comenzaremos a desarrollar la idea y el funcionamiento de nuestro modelo de invernadero familiar.

Con la nueva tendencia mundial, donde las familias buscan bajar los costos de alimentación y, además, saber cuál es la calidad de los alimentos que ingieren, se ha desarrollado un invernadero de muy bajo costo, donde las familias no deberán estar pendientes de las condiciones ambientales óptimas para el desarrollo de su cultivo y podrán realizar sus tareas habituales. Como lo mencionamos anteriormente, con pequeños ajustes de programación y cambio de sensores se puede escalar a un uso comercial.

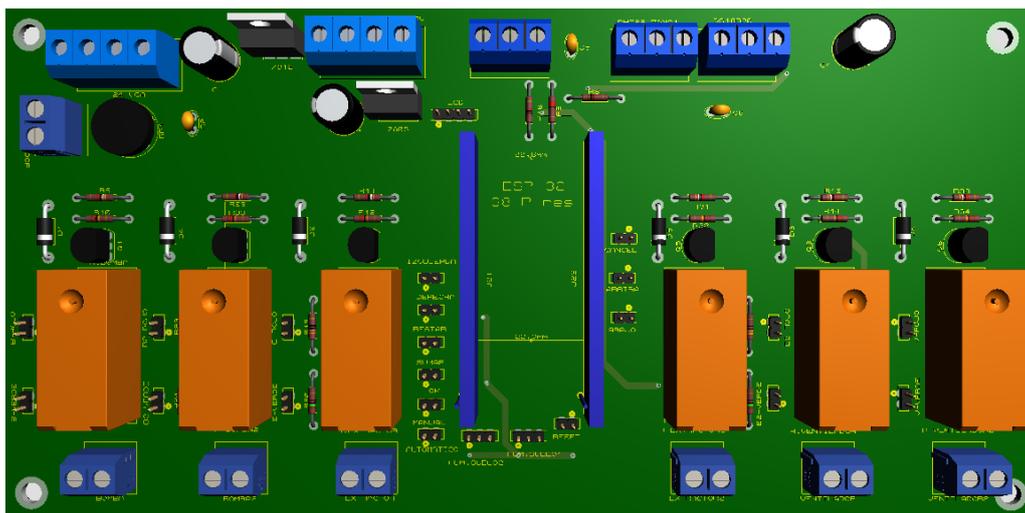


Figura 21: Pcb del invernadero

A continuación, seccionaremos el prototipo y pasaremos a explicar cada una de sus partes y su funcionamiento.

4.1 Fuente

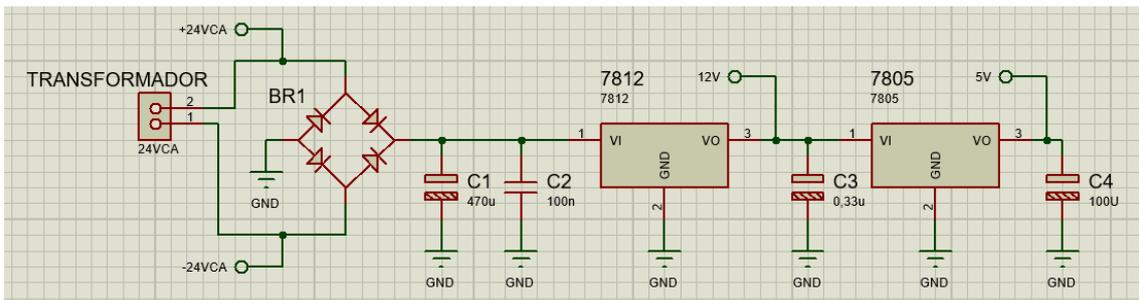


Figura 22: Diagrama de la fuente

En la figura 23, se observa la sección de la placa encargada de alimentación. En ella podemos observar la entrada de nuestro transformador (a través de una bornera de 2 vías).

A continuación de la entrada del transformador nos encontramos con el puente rectificador (BR1) encargado de convertir nuestra alterna en continua. Después de un puente rectificador, encontramos un capacitor que se utiliza principalmente para filtrar la señal de corriente continua (CC) pulsante y reducir la ondulación. El puente rectificador convierte la corriente alterna (CA) en CC, pero la salida no es una CC pura, sino que presenta una ondulación.

El capacitor actúa como un filtro (figura 24), almacenando carga cuando el voltaje es alto y descargándola cuando el voltaje disminuye, lo que ayuda a suavizar la señal y obtener una CC más limpia.

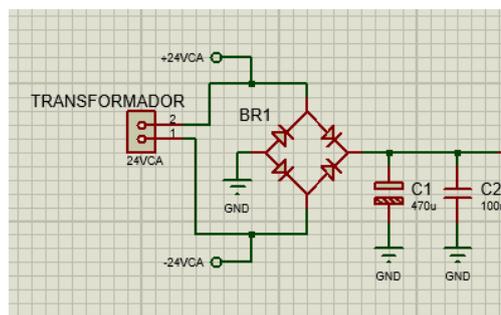


Figura 23: esquema de la rectificación de VCA a VCC

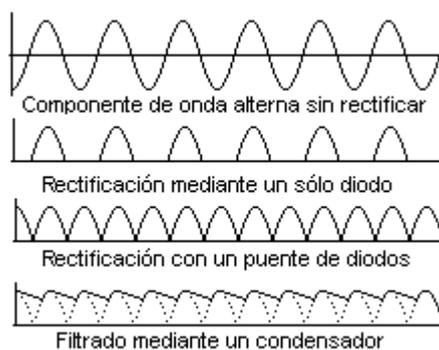


Figura 24: funcionamiento de los capacitores C1 y C2

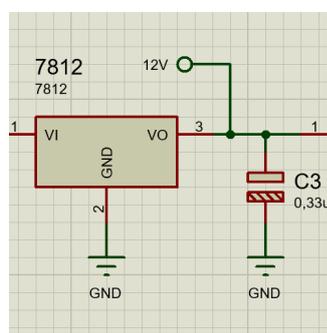


Figura 25: salida de 12V

En esta sección del diagrama (figura 25) se logra identificar un regulador de tensión. El 7812 reduce el voltaje de entrada a un voltaje de salida fijo de 12V, manteniendo la estabilidad incluso con variaciones en la tensión de entrada o cambios en la carga.

Incorpora protección interna contra cortocircuitos que desactiva el dispositivo para evitar daños en caso de un fallo.

También incluye protección contra sobrecalentamiento, apagando el regulador en caso de que la temperatura supere un cierto límite.

El 7812 limita la corriente de salida para proteger el circuito de sobrecargas.

Debido a su principio de funcionamiento, el regulador disipa el exceso de voltaje como calor, por lo que es recomendable usar un disipador de calor adecuado, especialmente con cargas elevadas, para evitar daños por temperatura.

El condensador (capacitor C3) está cumpliendo la función de filtrar la alta frecuencia en el regulador 7805 (figura 26)

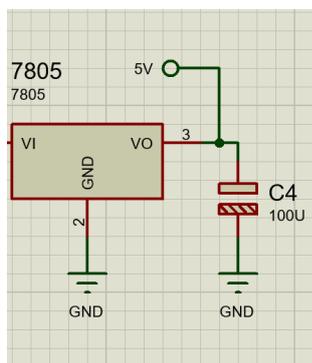


Figura 26: salida 5V

El regulador 7805 (figura 26) funciona de la misma forma que el regulador 7812. La única diferencia es que a su salida mantiene un voltaje fijo de 5v.

4.2 Microcontrolador

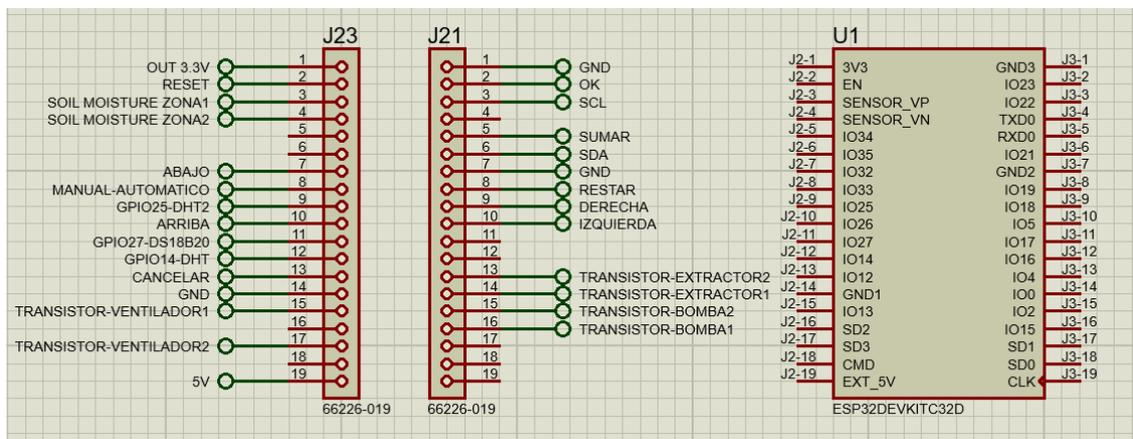


Figura 27: esquema de conexión ESP32

En esta imagen (figura 27) tenemos la representación de nuestro microcontrolador ESP32. En el podemos ver todas las conexiones a los gpios o puertos entrada-salida y que función cumple cada uno de ellos en nuestro proyecto. A continuación, haremos una lista breve de cada uno de ellos.

Gpio 00 = Funciona como pin de salida, enviándole (cuando sea necesario) una tensión al transistor (bc548) para que este active el relé del extractor1.

Gpio 01 = Libre

Gpio 02 = Funciona como pin de salida, enviándole (cuando sea necesario) una tensión al transistor (bc548) para que este active el relé de la bomba 2.

Gpio 03 = Funciona como pin de entrada para un pulsador

Gpio 04 = Funciona como pin de salida, enviándole (cuando sea necesario) una tensión al transistor (bc548) para que este active el relé del extractor 2.

Gpio 05 = Funciona como pin de entrada para un pulsador

Gpio 06 = Libre

Gpio 07 = Libre

Gpio 08 = Libre

Gpio 09 = Libre

Gpio 10 = Funciona como pin de salida, enviándole (cuando sea necesario) una tensión al transistor (bc548) para que este active el relé del ventilador 2.

Gpio 11 = Libre

Gpio 12 = Funciona como pin de entrada para un pulsador

Gpio 13 = Funciona como pin de salida, enviándole (cuando sea necesario) una tensión al transistor (bc548) para que este active el relé del ventilador 1.

Gpio 14 = Pin de entrada para el sensor de humedad DHT22 zona1

Gpio 15 = Funciona como pin de salida, enviándole (cuando sea necesario) una tensión al transistor (bc548) para que este active el relé de la bomba 1.

Gpio 16 = Libre

Gpio 17 = Libre

Gpio 18 = Funciona como pin de entrada para un pulsador

Gpio 19 = Funciona como pin de entrada para un pulsador

Gpio 21 = Pin destinado a la comunicación SDA de los LCD

Gpio 22 = Pin destinado a la comunicación SCL de los LCD

Gpio 23 = Funciona como pin de entrada para un pulsador

Gpio 25 = Pin de entrada para el sensor de humedad DHT22 zona2

Gpio 26 = Funciona como pin de entrada para un pulsador

Gpio 27 = Pin de entrada para los sensores de temperatura DS18B20

Gpio 32 = Funciona como pin de entrada para un pulsador

Gpio 33 = Funciona como pin de entrada para selección del modo manual/automático

Gpio 34 = Libre

Gpio 35 = Libre

Gpio 36 = Pin de entrada para el sensor de humedad del suelo Soil Moisture Zona1

Gpio 39 = Pin de entrada para el sensor de humedad del suelo Soil Moisture Zona1

4.3 Sensor DHT22

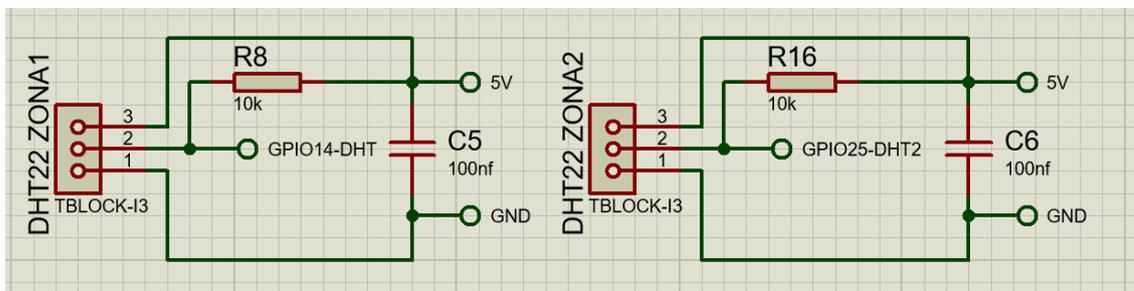


Figura 28: Esquema sensor DHT22

En la figura 28 se muestra el conexionado de los sensores de humedad relativa DHT22. Los datos de las resistencias y los condensadores fueron obtenidos por hoja de datos. Tal cual lo vimos anteriormente en la selección de los sensores, solo usaremos 3 patas de las 4 que trae el sensor. Se usan dos sensores para cubrir mayor rango de metros y por eso se dividen en dos zonas.

4.4 Sensor DS18B20

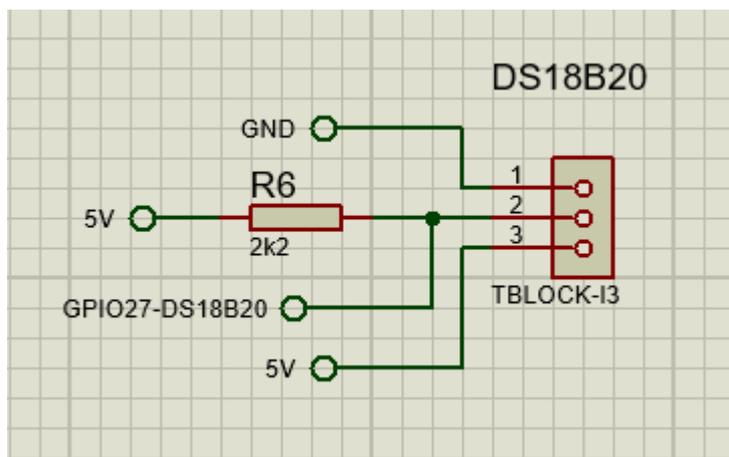


Figura 29: Sensor DS18B20

Se puede ver en la figura 29 la forma de conectar los sensores DS18B20, mediante una resistencia de 2k2 a 5v para tener una lectura óptima para la distancia (en este caso de 20 metros). A diferencia de los DHT22 los sensores DS18B20 se pueden conectar en el mismo pin de conexión ya que cada uno cuenta con su propia dirección de datos (anexo VI).

4.5 Sensor Soil Moisture

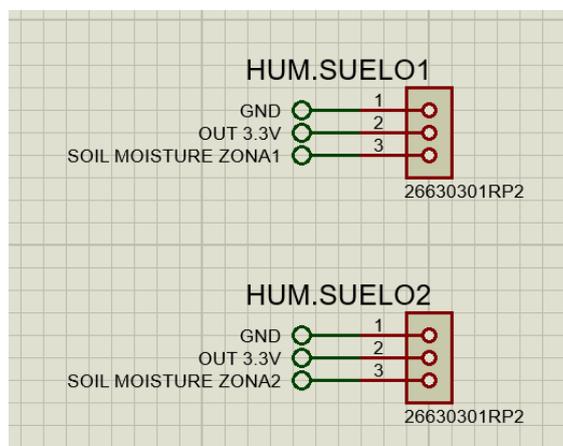


Figura 30: Sensor capacitivo Soil Moisture

En la figura 30 se muestra la forma en que debemos conectar a los sensores de humedad del suelo Soil Moisture, tanto el de la zona 1 como el de la zona 2. Tiene 1 pin de alimentación, uno de GND y otro de señal.

4.6 Diodos leds, indicadores de estado

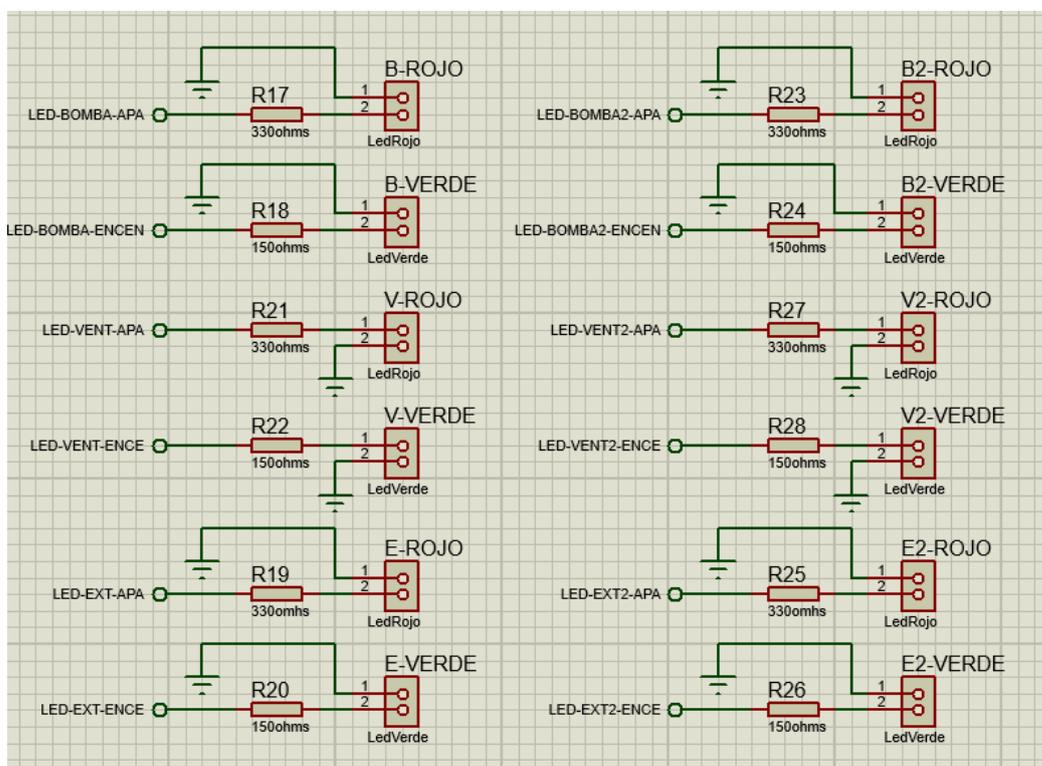


Figura 31: Diodos leds, indicadores de estado

La figura 31 vemos la conexión de los leds que nos indica en que estados se encuentran nuestros relés, un led rojo indica que la bobina de nuestro relé no está energizada y uno de color verde indica que si lo está. Estos leds cumplen la función de mostrarnos ante eventuales fallas que es lo que debería estar sucediendo en ese instante. Tenemos 1 led rojo y 1 led verde por cada relé en uso, cada uno de estos está protegido por su respectiva resistencia calculada según su color de led.

4.7 Accionamiento de relés

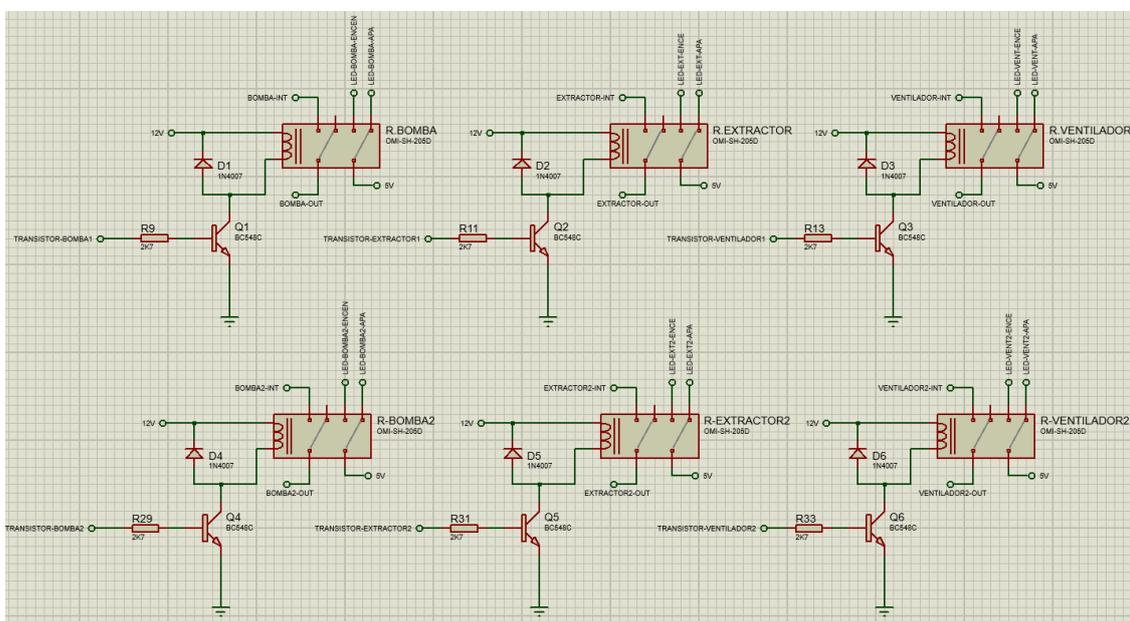


Figura 32: Relés

En la figura 32 se ve el grupo de relés y transistores encargados de encender las bombas, ventiladores y extractores, a continuación, se explicará el diagrama de conexión de uno de ellos, pero debemos saber, que todos se aplican de la misma manera.

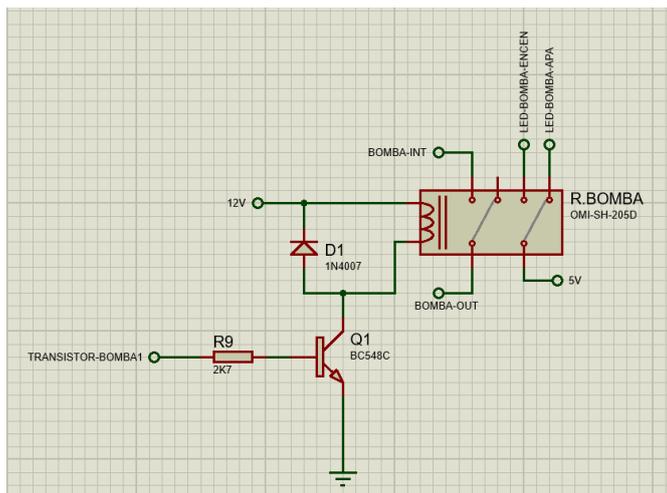


Figura 33: Accionamiento de un relé a través de un transistor

En la figura 33 se observa el accionamiento de un relé de bobina de 12v, a través de una señal de 3.3v del esp 32.

Cuando el pin del Esp 32 envía una señal de 3,3v (transistor-bomba1), este llega a la resistencia R9 y de ahí a la base del transistor. El transistor BC548 puede usarse como un interruptor electrónico que se activa o desactiva según la tensión aplicada a su base, funcionando como un circuito abierto (corte) o un circuito cerrado (saturación).

En el corte, el transistor se comporta como un circuito abierto, no permitiendo el paso de corriente entre colector y emisor. En la saturación, se comporta como un circuito cerrado, permitiendo el máximo flujo de corriente entre colector y emisor, con una caída de tensión mínima entre estos dos terminales. De esta forma circula una tensión de 12v entre colector y emisor energizando la bobina del relé y accionándolo.

En el esquema vemos un diodo, que se utiliza principalmente para proteger el circuito de la tensión inducida por la bobina del relé cuando se desactiva. Esta tensión, conocida como tensión inversa o pico de tensión, puede ser dañina para otros componentes del circuito, especialmente los transistores. El diodo, conectado en paralelo a la bobina, actúa como un camino de baja impedancia para la corriente inducida, permitiendo que se disipe de forma segura sin dañar los componentes circundantes.

4.8 Interconexión de display LCD

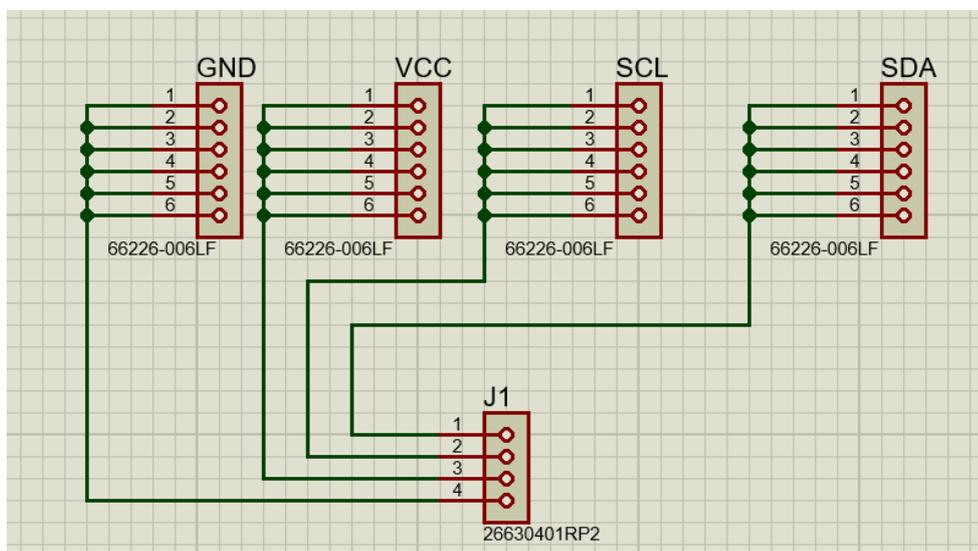


Figura 34: Interconexión de display

En la figura 34 lo que encontramos es un circuito de interconexión para varias pantallas Lcd. Compartiendo, todos, los mismos 4 pines en el esp32 (VCC, GND, SCL, SDA), de esta forma ahorraremos un montón de recursos.

4.9 Descripción del funcionamiento

El sistema cuenta con sensores de temperatura (DS18B20), sensores de humedad relativa (DHT22) y sensores capacitivos, para medir la humedad del suelo (SOIL MOISTURE). Todos divididas en 2 zonas diferentes, para poder cubrir una superficie mayor. Los sensores trabajan en forma independiente, los datos son tomados cada 2 segundos (según especificaciones del fabricante) y mostrados en un display Lcd 20X4. En caso, de que uno de los sensores se dañe, aparecerá (en el Lcd correspondiente) una leyenda indicándolo.

El sistema, en su modo automático, tomara datos cada cierto tiempo, este tiempo viene especificado por el fabricante del sensor. En algunos casos es de 2 segundos y en otros de 1,5 segundos. Nuestro programa comparara estos valores obtenidos con valores (óptimos según el cultivo) que ya hemos seteado en nuestro programa.

Luego de ser comparados tomara decisiones según los valores que recoja: por ejemplo, en caso de que la temperatura sea más alta o más baja, que los valores que haya seteado previamente en el programa, se mandara una señal a un relay para encenderlo, este comandara una o varias acciones específicas, decididas de forma particular por cada usuario (encender una alarma, ventiladores, sistemas de refrigeración, calefacción o cualquier otro medio que deseemos usar).

Como resumen, podemos decir, que lo que se ha querido construir y desarrollar con esto; es un sistema para ser utilizado en invernaderos a diferentes escalas (familiares y comerciales) donde la cantidad de tiempo que se le tenga que brindar a los cuidados sea lo menor posible. Sin descuidar y relegar la calidad de lo producido.

5. Ensamble del prototipo

5.1 Primeros Pasos

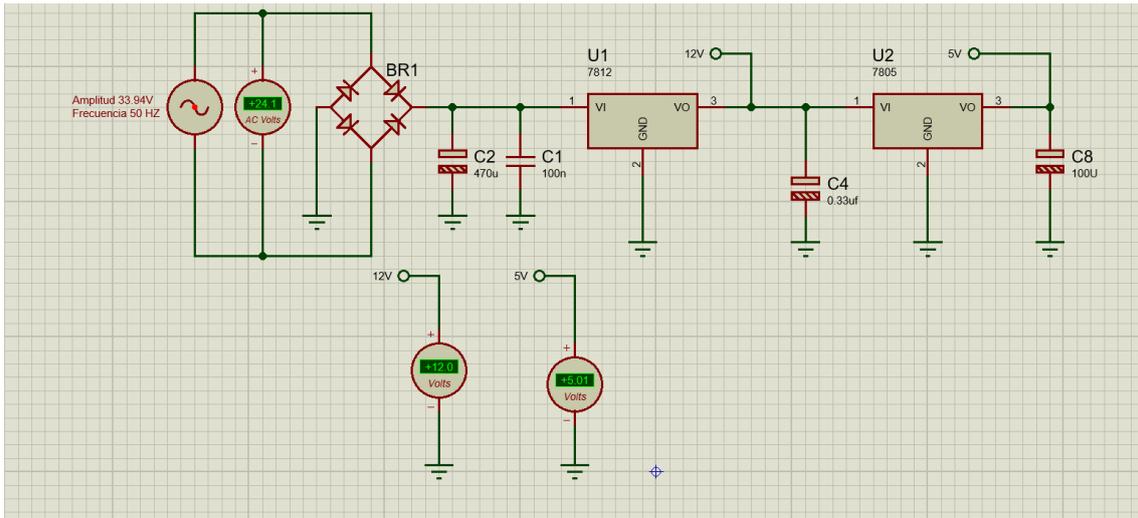


Figura 35: Funcionamiento de la fuente y sus reguladores

Podemos observar (figura 36) como logramos hacer una fuente lineal partiendo de un transformador 220VCA / 18VCA a dos salidas de 12VCC y de 5VCC.

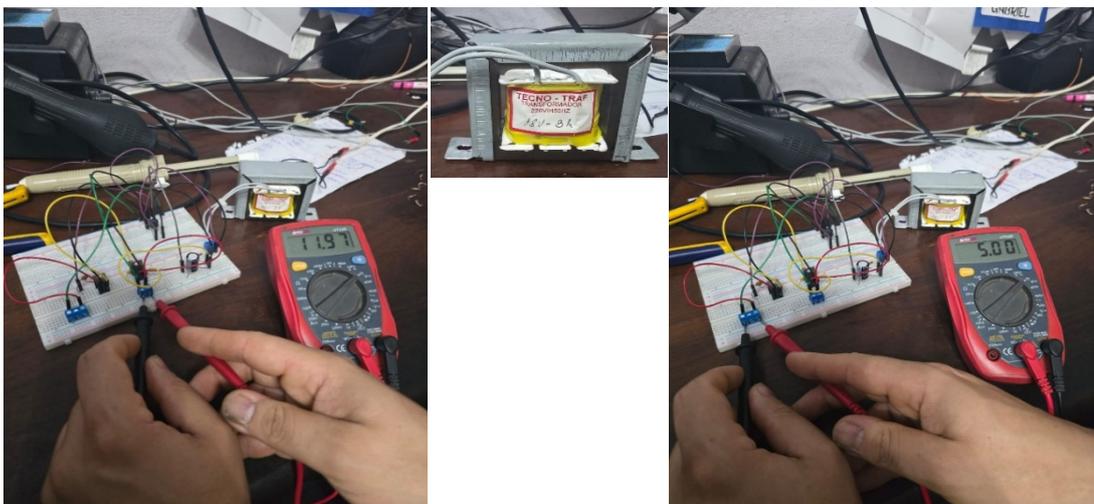


Figura 36: Comprobación de las salidas de tensión 12VCC y 5VCC

5.2 Funcionamiento de los relés

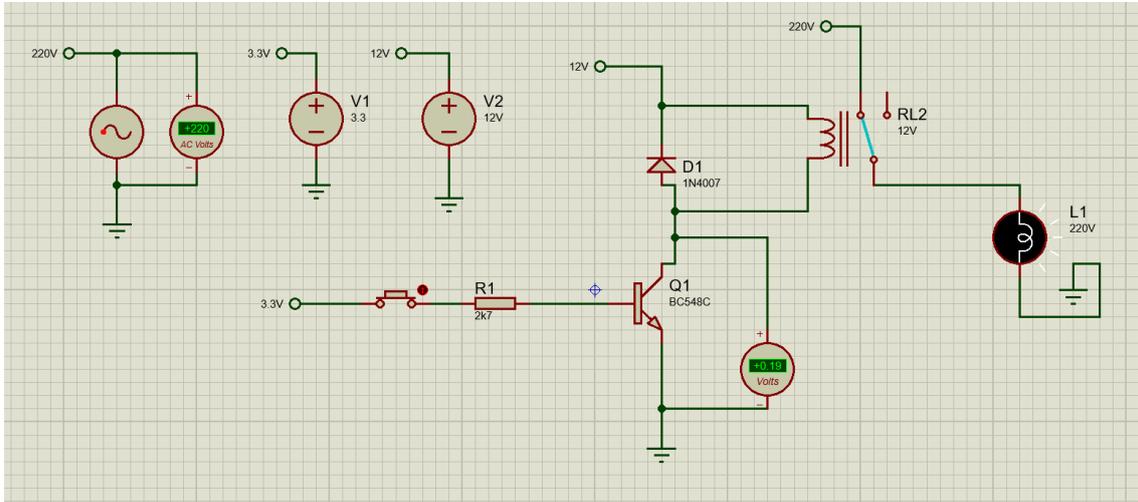


Figura 37: Ensayo en proteus sobre accionamiento de relé

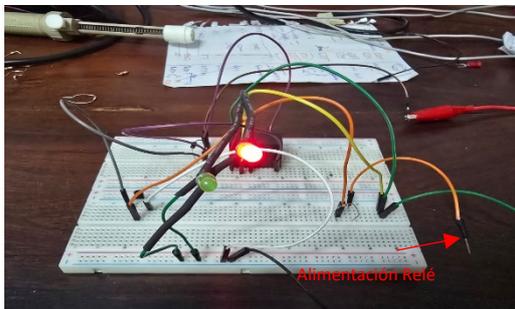


Figura 38: relé sin activar

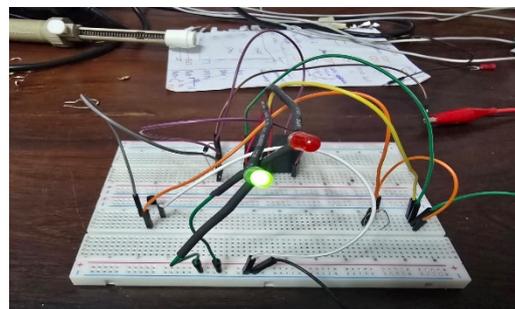


Figura 39: relé activado

Se puede observar, en la figura 38, un led rojo encendido, un led verde apagado y a su vez un cable desconectado (alimentación para el transistor bc548).

Esta imagen nos está mostrando que el relé no se encuentre activado; que el led rojo se encuentra conectado en un circuito NC (normalmente cerrado) y el led verde se encuentra conectado a un circuito NA (normalmente abierto) del relé.

En la siguiente imagen (figura 39) vemos que el led rojo se encuentra apagado, el led verde encendido y que el cable que antes estaba desconectado ahora se encuentra conectado y alimentando al transistor bc548 para que este encienda el relé.

5.3 Conectando varios DS18B20 con solo 3 cables

Es crucial comprender que la conexión de múltiples sensores de temperatura DS18B20 en solo tres cables se logra gracias a la arquitectura de bus 1-Wire de Dallas Semiconductor. Este protocolo permite la comunicación de múltiples dispositivos digitales utilizando un único hilo de datos, además de la alimentación.

Es un bus maestro-esclavo. Un único maestro (generalmente un microcontrolador, en nuestro caso un ESP32) se comunica con uno o varios esclavos (los sensores DS18B20).

Cada sensor DS18B20 posee un código de serie de 64 bits único grabado de fábrica. Este identificador es fundamental, ya que permite al maestro diferenciar y direccionar cada sensor individualmente, incluso cuando están conectados al mismo bus físico.

En el Anexo VI se deja el programa que se usó para obtener las direcciones de cada sensor usado en el prototipo.



Figura 40: obteniendo e imprimiendo direcciones

5.4 Funcionamiento del Soil Moisture

En las siguientes imágenes veremos los diferentes valores de referencia que fueron obtenidos en varias circunstancias (al aire libre, en agua, en tierra seca, en tierra con algún porcentaje de humedad, etc.) estos valores fueron de vital importancia para poder calibrar el sensor y la función map.

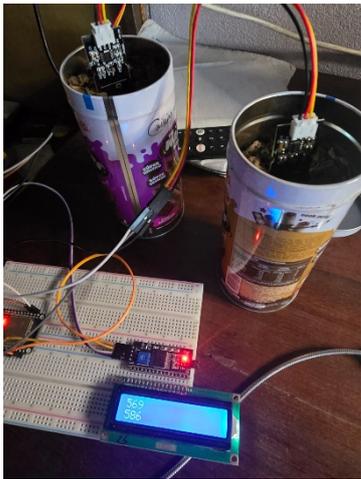


Figura 43: datos en tierra húmeda

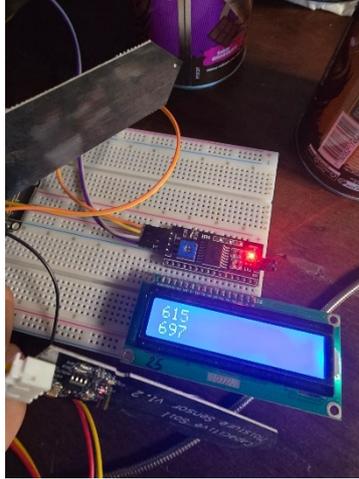


Figura 42: datos al aire libre

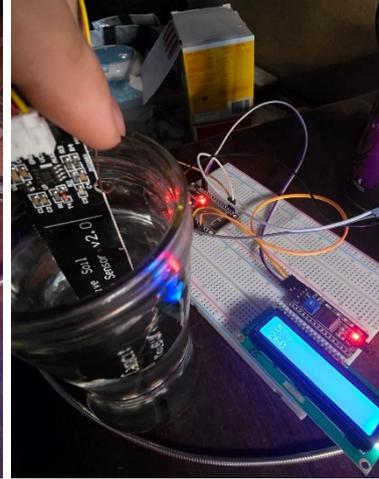


Figura 41: datos en agua

5.5 Funcionamiento del modo Manual-Automático

Con esta función de nuestro prototipo lo que logramos hacer es pasar del modo automático al modo manual y viceversa, pero para poder entender mejor la función de este selector debemos dar una breve explicación de lo que hace cada modo.

Modo Automático: estando en este modo nuestro sistema censara y tomara todas las decisiones de forma automática (cuando encender o apagar cada bomba, ventilador, extractor, etc.)

Modo Manual: estando aquí, nosotros podremos manejar las decisiones del sistema (encender o apagar cada bomba, ventilador, extractor, etc.). Pero además

podremos setear los datos con lo cual nuestro sistema en modo automático trabajara.

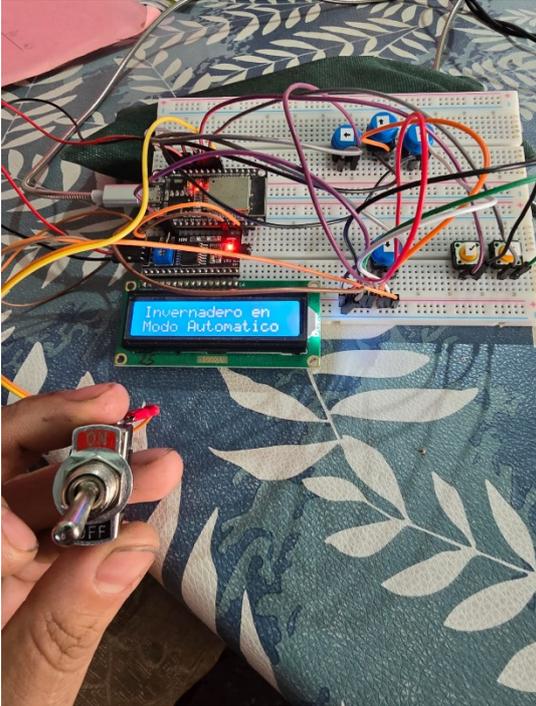


Figura 45: Invernadero en modo Automático

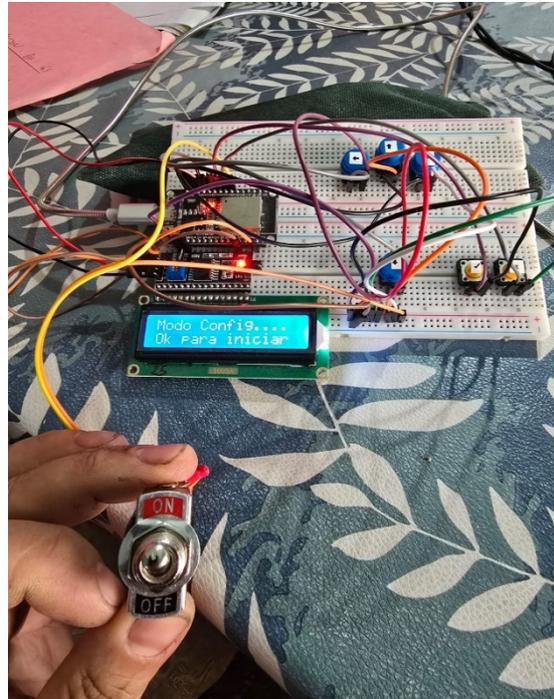


Figura 44: Invernadero en modo Manual

6. Conclusión

Se ha observado, durante la investigación y el desarrollo de este proyecto, muchos puntos altamente satisfactorios. Lo cuales, hacen pensar que es un proyecto con una muy buena proyección de desarrollo comercial.

Los invernaderos representan un paso fundamental hacia la optimización y modernización de la agricultura en Argentina (mucho mas presente en pequeña y mediana escala) Se ha podido demostrar como esta tecnología puede mitigar los riesgos asociados a factores externos, reducir el uso de insumos y potenciar la producción y calidad final.

Si bien, en la Argentina de hoy, los costos pueden ser de una consideración importante, se logra ver, que cada vez hay mas gente interesada en controlar lo que consume. En este sentido, la gente no ve a los invernaderos como un gasto sino como una inversión en el bienestar de su cuerpo, y por tal, de su salud. Esto esta impulsando y generando un gran interés en desarrollar tecnologías confiables y de bajos costos.

Por eso en invernaderos Conaut estamos seguros que hay un gran futuro por delante y si bien es un camino difícil, complejo y con mucha competencia en desarrollo, también hay un mercado muy amplio por satisfacer.

Lo experimentado durante el desarrollo de este proyecto nos hizo sentir muy emocionados. Los logros alcanzados, como así también los fallos, nos motivan a seguir encontrando soluciones a los problemas que se presentan a la hora de cuidar tus plantas.

Hoy, gracias a las tecnologías modernas, los problemas dejan de serlo y pasan a ser, solo simples obstáculos con soluciones al alcance de tus manos.

Pensábamos que el mercado era un mercado muy reducido y específico, pero el camino nos mostró otra cara totalmente diferente. Nos mostró un sinfín de gente deseosa de controlar el crecimiento de sus plantas, gente que quiere invertir en el

cuidado de ellas, pero que no cuentan con el tiempo para hacerlo; gente que no ingiere productos sin antes saber de su procedencia y es por ellos que nos encontramos muy entusiasmados en el desarrollo de este prototipo de invernadero inteligente.

En Invernaderos CONAUT hemos tomado la decisión de buscar maximizar los beneficios y acompañarte en el cuidado de tus plantas (tanto de uso alimenticio como medicinal).

- 📄 Calculadora de Capacitores <https://www.digikey.com/es/resources/conversion-calculators/conversion-calculator-series-and-parallel-capacitor>
- 📄 Calculadora para capacitor por filtro <https://www.profetolocka.com.ar/calculadora-de-capacitor-de-filtro/>
- 📄 ¿Por qué se pone un DIODO en paralelo con el RELAY?
<https://www.youtube.com/watch?v=tO8WhK4RI50>
- 📄 Como controlar un relé con un transistor <https://www.inventable.eu/controlar-rele-con-transistor/>

Anexo I- Costos

Se debe aclarar que a esta lista de costos se le suma una gran cantidad de otros componentes que no pueden ser detallados, pero si se los debe tener en cuenta (cables, estaño, placas de desarrollo, protoboard, etc.)

Detalle	Cantidad	Costo x unidad	Costo final
Sensor Soil Moisture	1	USD 3,10	USD 3,10
Sensor DHT22	2	USD 5,14	USD 10,28
Placa ESP32	1	USD 10,83	USD 10,83
Sensor DS18B20	2	USD 2,00	USD 4,00
Relé y zocalo	6	USD 19,55	USD 117,30
Placa PCb virgen 10x20	1	USD 4,00	USD 4,00
Cloruro Ferrico	1	USD 8,20	USD 8,20
Rele 12v para pcb doble inversor 8A	6	USD 1,80	USD 10,80
Bornera doble contacto para impreso	10	USD 0,35	USD 3,50
Capacitores varios	10	USD 0,06	USD 0,60
zocalo para esp32	1	USD 0,60	USD 0,60
Pulsador Metalico 12v	10	USD 2,30	USD 23,00
Regulador 7805	1	USD 0,70	USD 0,70
Regulador 7812	1	USD 0,80	USD 0,80
Pulsador Metalico con retencion	1	USD 12,00	USD 12,00
Acohol isopropilico	1	USD 16,21	USD 16,21
Espagueti termocontraible	1	USD 0,90	USD 0,90
Resistencias Varios valores	15	USD 0,10	USD 1,50
Gabinete metalico CCTV	1	USD 210,00	USD 210,00
Display LCD 20x4	4	USD 15,00	USD 60,00
Transformador 220v 24V	1	USD 30,20	USD 30,20
Transformador 220v 18V	1	USD 48,15	USD 48,15
Diodos led varios colores	20	USD 0,11	USD 2,20
Puente rectificador W1	1	USD 1,16	USD 1,16
		Total	USD 580,03

Anexo II- Programa

```

/*
Datos
Apellido: Mazzeo
Nombre: Carlos Alejandro
Invernadero Armado para la plantación de Tomates
La Humedad relativa Optima se encuentra entre 65 y 85%

Dispositivos utilizados:
1- Esp32 NodeMCU-32s 38pines
2- 3 Display Lcd 20x4
3- 2 Sensores de Temperatura DS18B20, Zona1 y Zona2, usando solo 1 pin
de datos
4- 2 Sensores de Humedad DHT22; Zon1 y Zona2
5- 2 Sensores de Humedad del suelo Soil Moisture
*/
/*----- Librerias -----
-----*/

#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include "DHT.h"
#include <Keypad.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Arduino.h>

/*-----Pulsadores -----
-----*/

#define automaticoManual 33
#define pulsadorArriba 26
#define pulsadorAbajo 32
#define pulsadorDerecha 18
#define pulsadorIzquierda 5
#define pulsadorSumar 3
#define pulsadorRestar 19
#define pulsadorOk 23
#define pulsadorCancel 12

const int umbralTiempo = 170;
Long tiempoDeInicio = 0;

```

```

const int umbralTiempo1 = 170;
Long tiempoDeInicio1 = 0;
const int umbralTiempo2 = 170;
Long tiempoDeInicio2 = 0;
const int umbralTiempo3 = 170;
Long tiempoDeInicio3 = 0;
// const int umbralTiempo4 = 50;
// long tiempoDeInicio4 = 0;

// int anterior = LOW;
// int estado = LOW;

volatile bool modoConfiguracion = false;
volatile int pantallaPrincipal = 0;

volatile bool menu = false;
volatile bool subMenu = false;

volatile bool encender = false;

void IRAM_ATTR entrarMenu(void);
void IRAM_ATTR salirMenu(void);
void IRAM_ATTR sumarSubMenu(void);
void IRAM_ATTR restarSubMenu(void);

/*-----Variables Millis -----
-----*/

unsigned Long Actual_millis = 0;
unsigned Long Previo_millis = 0;
const Long Intervalo = 2000;

unsigned Long Actual_millis1 = 0;
unsigned Long Previo_millis1 = 0;
const Long Intervalo1 = 3000;

unsigned Long actualMillis2 = 0;
unsigned Long previoMillis2 = 0;
const Long intervalo2 = 1500;

/*-----Símbolos-----
-----*/
a= Flecha Abajo
b= Flecha Arriba
c= Flecha Izquierda
d= Flecha Derecha
-----*/
byte a[8] = {
    B00000,
    B00100,

```

```

    B00100,
    B00100,
    B11111,
    B01110,
    B00100,
    B00000,
};
byte b[8] = {
    B00000,
    B00100,
    B01110,
    B11111,
    B00100,
    B00100,
    B00100,
    B00000,
};
byte c[8] = {
    B00000,
    B00010,
    B00110,
    B01110,
    B11111,
    B01110,
    B00110,
    B00010,
};
byte d[8] = {
    B00000,
    B01000,
    B01100,
    B01110,
    B11111,
    B01110,
    B01100,
    B01000,
};

/*-----Sensor DS18B20 -----*/
-----*/

OneWire ourWire(27);           // Se establece el pin 27 del ESP32
para la lectura del sensor
DallasTemperature DS18B20(&ourWire); // Se declara una variable u objeto
para el sensor
DeviceAddress address1 = {0x28, 0x33, 0xA9, 0xF2, 0x24, 0x23, 0xB, 0xA2};
DeviceAddress address2 = {0x28, 0x47, 0x7D, 0x80, 0xE3, 0xE1, 0x3C,
0x39};

```

```

#define pinExtractor 0
#define pinExtractor2 4

int tempMax1 = 25;
int tempMin1 = 22;

int tempMax2 = 22;
int tempMin2 = 21;
float temperatura = 0;

/*-----Sensor DHT22 -----
-----*/

#define DHTPIN1 14 // Pin de conexion DHT22 Zona1
#define DHTPIN2 25 // Pin de conexion DHT22 Zona2
#define DHTTYPE1 DHT22 // Modelo de sensor
#define DHTTYPE2 DHT22

#define pinVentilador 13
#define pinVentilador2 10

DHT dht1(DHTPIN1, DHTTYPE1);
DHT dht2(DHTPIN2, DHTTYPE2);

int humMax1 = 85;
int humMin1 = 75;

int humMax2 = 85;
int humMin2 = 75;

/*-----Sensor -humedad Suelo -----
-----*/

#define pinBomba 15
#define pinBomba2 2

#define pinHumedadSuelo1 36
#define pinHumedadSuelo2 39

int valorHumedad1;
int valorHumedad2;
const int valorAlAire = 610;
const int valorEnAgua = 240;

int humSueloMax1 = 33;
int humSueloMin1 = 12;

int humSueloMax2 = 25;
int humSueloMin2 = 12;

```

```

/*----- Display LCD -----
-----*/

LiquidCrystal_I2C lcd0(0x23, 20, 4);
LiquidCrystal_I2C lcd1(0x26, 20, 4);
LiquidCrystal_I2C lcd2(0x27, 20, 4);
LiquidCrystal_I2C lcd3(0x25, 20, 4);

/*-----Declaracion de Funciones -----
-----*/

void Sens_Humedad(void);
void Sens_Temperatura(void);
void sensHumedadSuelo(void);
void Inicio_conf_pines(void);
void Mensaje_Bienvenida(void);
void modoManual(void);
void datosTemperatura1(void);
void datosTemperatura2(void);
void datosHumedadSuelo1(void);
void datosHumedadSuelo2(void);
void datosHumedadRelativa1(void);
void datosHumedadRelativa2(void);
void bombaManual_1(void);
void bombaManual_2(void);
void ventiladorManual_1(void);
void ventiladorManual_2(void);
void extractorManual_1(void);
void extractorManual_2(void);

/*-----Void Setup -----
-----*/

void setup()
{
  attachInterrupt(digitalPinToInterrupt(pulsadorOk), entrarMenu, RISING);
  attachInterrupt(digitalPinToInterrupt(pulsadorCancel), salirMenu,
RISING);
  attachInterrupt(digitalPinToInterrupt(pulsadorArriba), sumarSubMenu,
RISING);
  attachInterrupt(digitalPinToInterrupt(pulsadorAbajo), restarSubMenu,
RISING);
  analogReadResolution(10);
  Inicio_conf_pines();
  Mensaje_Bienvenida();
}

/*-----Void Loop -----
-----*/

```

```
void loop()
{
  modoConfiguracion = digitalRead(automaticoManual);

  if (modoConfiguracion == true)
  {
    if (pantallaPrincipal == 0)
    {
      modoManual();
    }

    if (pantallaPrincipal == 1)
    {
      datosTemperatura1();
    }
    if (pantallaPrincipal == 2)
    {
      datosTemperatura2();
    }

    if (pantallaPrincipal == 3)
    {
      datosHumedadSuelo1();
    }
    if (pantallaPrincipal == 4)
    {
      datosHumedadSuelo2();
    }

    if (pantallaPrincipal == 5)
    {
      datosHumedadRelativa1();
    }
    if (pantallaPrincipal == 6)
    {
      datosHumedadRelativa2();
    }

    if (pantallaPrincipal == 7)
    {
      bombaManual_1();
      if (encender == true)
      {
        digitalWrite(pinBomba, HIGH);
      }
      if (encender == false)
      {
        digitalWrite(pinBomba, LOW);
      }
    }
  }
}
```

```
    }  
  }  
  
  if (pantallaPrincipal == 8)  
  {  
    bombaManual_2();  
    if (encender == true)  
    {  
      digitalWrite(pinBomba2, HIGH);  
    }  
    if (encender == false)  
    {  
      digitalWrite(pinBomba2, LOW);  
    }  
  }  
  
  if (pantallaPrincipal == 9)  
  {  
    ventiladorManual_1();  
    if (encender == true)  
    {  
      digitalWrite(pinVentilador, HIGH);  
    }  
    if (encender == false)  
    {  
      digitalWrite(pinVentilador, LOW);  
    }  
  }  
  if (pantallaPrincipal == 10)  
  {  
    ventiladorManual_2();  
    if (encender == true)  
    {  
      digitalWrite(pinVentilador2, HIGH);  
    }  
    if (encender == false)  
    {  
      digitalWrite(pinVentilador2, LOW);  
    }  
  }  
  if (pantallaPrincipal == 11)  
  {  
    extractorManual_1();  
    if (encender == true)  
    {  
      digitalWrite(pinExtractor, HIGH);  
    }  
    if (encender == false)  
    {  
      digitalWrite(pinExtractor, LOW);  
    }  
  }  
}
```

```

        digitalWrite(pinExtractor, LOW);
    }
}
if (pantallaPrincipal == 12)
{
    extractorManual_2();
    if (encender == true)
    {
        digitalWrite(pinExtractor2, HIGH);
    }
    if (encender == false)
    {
        digitalWrite(pinExtractor2, LOW);
    }
}
}
if (modoConfiguracion == false)
{
    lcd0.backlight();
    lcd1.backlight();
    lcd2.backlight();
    lcd3.backlight();
    lcd0.clear();
    lcd0.setCursor(5, 0);
    lcd0.print("Bienvenido");
    lcd0.setCursor(2, 1);
    lcd0.print("a tu invernadero");
    lcd0.setCursor(4, 2);
    lcd0.print("Inteligente");
    lcd0.setCursor(2, 3);
    lcd0.print("version 2025.00");

    Sens_Humedad();
    Sens_Temperatura();
    sensHumedadSuelo();
    encender = false;
    pantallaPrincipal = 0;
}
}

/*-----Funciones -----*/
-----*/

void Mensaje_Bienvenida()
{
    lcd0.clear();
    lcd1.clear();
    lcd2.clear();
    lcd3.clear();

```

```

lcd0.init();
lcd0.createChar(0, a);
lcd0.createChar(1, b);
lcd0.createChar(2, c);
lcd0.createChar(3, d);
lcd1.init();
lcd2.init();
lcd3.init();
lcd0.backlight();
lcd1.backlight();
lcd2.backlight();
lcd3.backlight();

lcd0.setCursor(0, 0);
lcd0.print("Iniciando....");
delay(5000);
lcd0.clear();
lcd0.setCursor(5, 0);
lcd0.print("Bienvenido");
lcd0.setCursor(2, 1);
lcd0.print("a tu invernadero");
lcd0.setCursor(4, 2);
lcd0.print("Inteligente");
lcd0.setCursor(2, 3);
lcd0.print("version 2025.00");

lcd1.setCursor(0, 0);
lcd1.print("Cargando...");
lcd2.setCursor(0, 0);
lcd2.print("Cargando...");
lcd3.setCursor(0, 0);
lcd3.print("Cargando...");
delay(2000);
}
void Inicio_conf_pines()
{
  DS18B20.begin(); // Se inicia el sensor
  dht1.begin();
  dht2.begin();
  pinMode(pinExtractor, OUTPUT);
  pinMode(pinExtractor2, OUTPUT);
  pinMode(pinVentilador, OUTPUT);
  pinMode(pinVentilador2, OUTPUT);
  pinMode(pinBomba, OUTPUT);
  pinMode(pinBomba2, OUTPUT);
  pinMode(pulsadorArriba, INPUT_PULLDOWN);
  pinMode(pulsadorAbajo, INPUT_PULLDOWN);
  pinMode(pulsadorDerecha, INPUT_PULLDOWN);
  pinMode(pulsadorIzquierda, INPUT_PULLDOWN);
}

```

```

pinMode(pulsadorOk, INPUT_PULLDOWN);
pinMode(pulsadorCancel, INPUT_PULLDOWN);
pinMode(automaticoManual, INPUT_PULLDOWN);
}
void Sens_Humedad()
{
  Actual_millis = millis();
  if (Actual_millis - Previo_millis > Intervalo) // hago que espere 2
segundos para que lo calcule otra vez
  {
    Previo_millis = Actual_millis;
    float humedad1 = dht1.readHumidity();
    float humedad2 = dht2.readHumidity();

    if (isnan(humedad1)) // si no es un numero (nan) me devuelve error
    {
      lcd1.clear();
      lcd1.setCursor(0, 0);
      lcd1.print("!!FALLO DE LECTURA!!");
      lcd1.setCursor(0, 1);
      lcd1.print("!!EN SENSOR ZONA1!!");
      if (isnan(humedad2))
      {
        lcd1.setCursor(0, 2);
        lcd1.print("!!FALLO DE LECTURA!!");
        lcd1.setCursor(0, 3);
        lcd1.print("!!EN SENSOR ZONA2!!");
        return;
      }
      lcd1.setCursor(0, 2);
      lcd1.print("Hum. Relativa Zona2");
      lcd1.setCursor(6, 3);
      lcd1.print(humedad2);
      lcd1.setCursor(12, 3);
      lcd1.print("%");
      return;
    }
    if (isnan(humedad2))
    {
      lcd1.clear();
      lcd1.setCursor(0, 0);
      lcd1.print("Hum. Relativa Zona1");
      lcd1.setCursor(6, 1);
      lcd1.print(humedad1);
      lcd1.setCursor(0, 2);
      lcd1.print("!!FALLO DE LECTURA!!");
      lcd1.setCursor(0, 3);
      lcd1.print("!!EN SENSOR ZONA2");
      return;
    }
  }
}

```

```

}
lcd1.clear();
lcd1.setCursor(0, 0);
lcd1.print("Hum. Relativa Zona1");
lcd1.setCursor(6, 1);
lcd1.print(humedad1);
lcd1.setCursor(12, 1);
lcd1.print("%");
lcd1.setCursor(0, 2);
lcd1.print("Hum. Relativa Zona2");
lcd1.setCursor(6, 3);
lcd1.print(humedad2);
lcd1.setCursor(12, 3);
lcd1.print("%");

if (humedad1 > humMax1)
{
    digitalWrite(pinVentilador, HIGH);
}
if (humedad1 <= humMin1)
{
    digitalWrite(pinVentilador, LOW);
}

if (humedad2 > humMax2)
{
    digitalWrite(pinVentilador2, HIGH);
}
if (humedad2 <= humMin2)
{
    digitalWrite(pinVentilador2, LOW);
}
}
}
void Sens_Temperatura()
{
    Actual_millis1 = millis();
    if (Actual_millis1 - Previo_millis1 > Intervalo1)
    {
        DS18B20.requestTemperatures(); // Se envía el comando para
        leer la temperatura
        float tem1 = DS18B20.getTempC(address1); // Se obtiene la temperatura
        en °C del sensor 1
        float tem2 = DS18B20.getTempC(address2); // Se obtiene la temperatura
        en °C del sensor 2

        if (tem1 <= -55 || tem1 >= 125) // Si algunos de los sensores no
        funciona correctamente devuelve error informando cual es el dañado
        {

```

```

lcd2.clear();
lcd2.setCursor(0, 0);
lcd2.print("!!FALLO DE LECTURA!!");
lcd2.setCursor(0, 1);
lcd2.print("!!EN SENSOR ZONA 1!!");
lcd2.setCursor(0, 2);
if (tem2 <= -55 || tem2 >= 125)
{
  lcd2.setCursor(0, 2);
  lcd2.print("!!FALLO DE LECTURA!!");
  lcd2.setCursor(0, 3);
  lcd2.print("!!EN SENSOR ZONA 2!!");
  return;
}
lcd2.print("Temperatura en Zona2");
lcd2.setCursor(7, 3);
lcd2.print(tem2);
lcd2.setCursor(13, 3);
lcd2.print("\xDF");
lcd2.setCursor(14, 3);
lcd2.print("C");
return;
}

if (tem2 <= -55 || tem2 >= 125)
{
  lcd2.clear();
  lcd2.setCursor(0, 0);
  lcd2.print("Temperatura en Zona1");
  lcd2.setCursor(7, 1);
  lcd2.print(tem1);
  lcd2.setCursor(13, 1);
  lcd2.print("\xDF");
  lcd2.setCursor(14, 1);
  lcd2.print("C");
  lcd2.setCursor(0, 2);
  lcd2.print("!!FALLO DE LECTURA!!");
  lcd2.setCursor(0, 3);
  lcd2.print("!!EN SENSOR ZONA 2!!");
  return;
}
lcd2.clear();
lcd2.setCursor(0, 0);
lcd2.print("Temperatura en Zona1");
lcd2.setCursor(7, 1);
lcd2.print(tem1);
lcd2.setCursor(13, 1);
lcd2.print("\xDF");
lcd2.setCursor(14, 1);

```

```

lcd2.print("C");

lcd2.setCursor(0, 2);
lcd2.print("Temperatura en Zona2");
lcd2.setCursor(7, 3);
lcd2.print(tem2);
lcd2.setCursor(13, 3);
lcd2.print("\xDF");
lcd2.setCursor(14, 3);
lcd2.print("C");

if (tem1 > tempMax1)
{
    digitalWrite(pinExtractor, HIGH);
}
if (tem1 <= tempMin1)
{
    digitalWrite(pinExtractor, LOW);
}

if (tem2 > tempMax2)
{
    digitalWrite(pinExtractor2, HIGH);
}
if (tem2 <= tempMin2)
{
    digitalWrite(pinExtractor2, LOW);
}
}
}
void sensHumedadSuelo()
{
    actualMillis2 = millis();
    if (actualMillis2 - previoMillis2 > intervalo2)
    {
        previoMillis2 = actualMillis2;

        int lecturaADC1 = analogRead(pinHumedadSuelo1);
        int lecturaADC2 = analogRead(pinHumedadSuelo2);

        valorHumedad1 = map(lecturaADC1, valorEnAgua, valorAlAire, 100, 0);
        valorHumedad2 = map(lecturaADC2, valorEnAgua, valorAlAire, 100, 0);

        if (valorHumedad1 < 0)
        {
            valorHumedad1 = 0;
        }
        if (valorHumedad2 < 0)
        {

```

```

    valorHumedad2 = 0;
}
if (valorHumedad1 > 100)
{
    valorHumedad1 = 100;
}
if (valorHumedad2 > 100)
{
    valorHumedad2 = 100;
}
lcd3.clear();
lcd3.setCursor(0, 0);
lcd3.print(" H. del Suelo Zona1");
lcd3.setCursor(7, 1);
lcd3.print(valorHumedad1);
lcd3.setCursor(10, 1);
lcd3.print("%");
lcd3.setCursor(0, 2);
lcd3.print(" H. del Suelo Zona2");
lcd3.setCursor(7, 3);
lcd3.print(valorHumedad2);
lcd3.setCursor(10, 3);
lcd3.print("%");
delay(1500);

if (valorHumedad1 < humSueloMin1)
{
    digitalWrite(pinBomba, HIGH);
}
if (valorHumedad1 >= humSueloMax1)
{
    digitalWrite(pinBomba, LOW);
}

if (valorHumedad2 < humSueloMin2)
{
    digitalWrite(pinBomba2, HIGH);
}
if (valorHumedad2 >= humSueloMax2)
{
    digitalWrite(pinBomba2, LOW);
}
}
}
void IRAM_ATTR entrarMenu() // ok
{
    if (millis() - tiempoDeInicio > umbralTiempo)
    {
        encender = true;
    }
}

```

```

    tiempoDeInicio = millis();
  }
}
void IRAM_ATTR salirMenu() // cancel
{
  if (millis() - tiempoDeInicio1 > umbralTiempo1)
  {
    encender = false;
    tiempoDeInicio1 = millis();
  }
}
void IRAM_ATTR sumarSubMenu() // Arriba
{
  if (millis() - tiempoDeInicio2 > umbralTiempo2)
  {
    pantallaPrincipal++;
    if (pantallaPrincipal == 13)
    {
      pantallaPrincipal = 0;
    }
    encender = false;
    tiempoDeInicio2 = millis();
  }
}
void IRAM_ATTR restarSubMenu() // Abajo
{
  if (millis() - tiempoDeInicio3 > umbralTiempo3)
  {
    pantallaPrincipal--;
    if (pantallaPrincipal < 0)
    {
      pantallaPrincipal = 12;
    }
    encender = false;
    tiempoDeInicio3 = millis();
  }
}
void modoManual()
{
  lcd1.noBacklight();
  lcd2.noBacklight();
  lcd3.noBacklight();
  lcd0.clear();
  lcd0.setCursor(0, 0);
  lcd0.print("-Riego    -Extractor");
  lcd0.setCursor(0, 1);
  lcd0.print("-Ventilador");
  lcd0.setCursor(0, 2);
  lcd0.print("-Parametros seteados");
}

```

```

lcd0.setCursor(0, 3);
lcd0.write(byte(1));
lcd0.setCursor(2, 3);
lcd0.write(byte(0));
lcd0.setCursor(4, 3);
lcd0.print("para desplazar");
lcd0.setCursor(0, 2);
lcd0.print("");
digitalWrite(pinBomba, LOW);
digitalWrite(pinBomba2, LOW);
digitalWrite(pinExtractor, LOW);
digitalWrite(pinExtractor2, LOW);
digitalWrite(pinVentilador, LOW);
digitalWrite(pinVentilador2, LOW);
delay(1500);
}
void datosTemperatura1()
{
  lcd0.clear();
  lcd0.setCursor(1, 0);
  lcd0.print("Temperatura elegida");
  lcd0.setCursor(7, 1);
  lcd0.print("Zona 1");
  lcd0.setCursor(1, 2);
  lcd0.print(tempMax1);
  lcd0.setCursor(3, 2);
  lcd0.print("\xDF");
  lcd0.setCursor(4, 2);
  lcd0.print("C Maxima");
  lcd0.setCursor(1, 3);
  lcd0.print(tempMin1);
  lcd0.setCursor(3, 3);
  lcd0.print("\xDF");
  lcd0.setCursor(4, 3);
  lcd0.print("C Minima");
  digitalWrite(pinBomba, LOW);
  digitalWrite(pinExtractor, LOW);
  digitalWrite(pinVentilador, LOW);
  delay(1000);
}
void datosTemperatura2()
{
  lcd0.clear();
  lcd0.setCursor(1, 0);
  lcd0.print("Temperatura elegida");
  lcd0.setCursor(7, 1);
  lcd0.print("Zona 2");
  lcd0.setCursor(1, 2);
  lcd0.print(tempMax2);

```

```

lcd0.setCursor(3, 2);
lcd0.print("\xDF");
lcd0.setCursor(4, 2);
lcd0.print("C  Maxima");
lcd0.setCursor(1, 3);
lcd0.print(tempMin2);
lcd0.setCursor(3, 3);
lcd0.print("\xDF");
lcd0.setCursor(4, 3);
lcd0.print("C  Minima");
digitalWrite(pinBomba, LOW);
digitalWrite(pinExtractor, LOW);
digitalWrite(pinVentilador, LOW);
delay(1000);
}
void datosHumedadSuelo1()
{
  lcd0.clear();
  lcd0.setCursor(3, 0);
  lcd0.print("Humedad Suelo");
  lcd0.setCursor(7, 1);
  lcd0.print("Zona 1");
  lcd0.setCursor(1, 2);
  lcd0.print(humSueloMax1);
  lcd0.setCursor(3, 2);
  lcd0.print("%  Maxima");
  lcd0.setCursor(1, 3);
  lcd0.print(humSueloMin1);
  lcd0.setCursor(3, 3);
  lcd0.print("%  Minima");
  digitalWrite(pinBomba, LOW);
  digitalWrite(pinExtractor, LOW);
  digitalWrite(pinVentilador, LOW);
  delay(1000);
}
void datosHumedadSuelo2()
{
  lcd0.clear();
  lcd0.setCursor(3, 0);
  lcd0.print("Humedad Suelo");
  lcd0.setCursor(7, 1);
  lcd0.print("Zona 2");
  lcd0.setCursor(1, 2);
  lcd0.print(humSueloMax2);
  lcd0.setCursor(3, 2);
  lcd0.print("%  Maxima");
  lcd0.setCursor(1, 3);
  lcd0.print(humSueloMin2);
  lcd0.setCursor(3, 3);

```

```

    lcd0.print("%  Minima");
    digitalWrite(pinBomba, LOW);
    digitalWrite(pinExtractor, LOW);
    digitalWrite(pinVentilador, LOW);
    delay(1000);
}
void datosHumedadRelativa1()
{
    lcd0.clear();
    lcd0.setCursor(2, 0);
    lcd0.print("Humedad Relativa");
    lcd0.setCursor(7, 1);
    lcd0.print("Zona 1");
    lcd0.setCursor(1, 2);
    lcd0.print(humMax1);
    lcd0.setCursor(3, 2);
    lcd0.print("% Maxima");
    lcd0.setCursor(1, 3);
    lcd0.print(humMin1);
    lcd0.setCursor(3, 3);
    lcd0.print("% Minima");
    digitalWrite(pinBomba, LOW);
    digitalWrite(pinExtractor, LOW);
    digitalWrite(pinVentilador, LOW);
    delay(1000);
}
void datosHumedadRelativa2()
{
    lcd0.clear();
    lcd0.setCursor(2, 0);
    lcd0.print("Humedad Relativa");
    lcd0.setCursor(7, 1);
    lcd0.print("Zona 2");
    lcd0.setCursor(1, 2);
    lcd0.print(humMax2);
    lcd0.setCursor(3, 2);
    lcd0.print("% Maxima");
    lcd0.setCursor(1, 3);
    lcd0.print(humMin2);
    lcd0.setCursor(3, 3);
    lcd0.print("% Minima");
    digitalWrite(pinBomba, LOW);
    digitalWrite(pinExtractor, LOW);
    digitalWrite(pinVentilador, LOW);
    delay(1000);
}
void bombaManual_1()
{
    lcd0.clear();

```

```

lcd0.setCursor(0, 0);
lcd0.print("***** Riego_1 *****");
lcd0.setCursor(1, 1);
lcd0.print("OK Encender");
lcd0.setCursor(1, 2);
lcd0.print("X Apagar");
digitalWrite(pinExtractor, LOW);
digitalWrite(pinExtractor2, LOW);
digitalWrite(pinVentilador, LOW);
digitalWrite(pinVentilador2, LOW);
digitalWrite(pinBomba2, LOW);
delay(1500);
}
void bombaManual_2()
{
  lcd0.clear();
  lcd0.setCursor(0, 0);
  lcd0.print("***** Riego_2 *****");
  lcd0.setCursor(1, 1);
  lcd0.print("OK Encender");
  lcd0.setCursor(1, 2);
  lcd0.print("X Apagar");
  digitalWrite(pinExtractor, LOW);
  digitalWrite(pinExtractor2, LOW);
  digitalWrite(pinVentilador, LOW);
  digitalWrite(pinVentilador2, LOW);
  digitalWrite(pinBomba, LOW);
  delay(1500);
}
void ventiladorManual_1()
{
  lcd0.clear();
  lcd0.setCursor(0, 0);
  lcd0.print("** Ventilador_1 **");
  lcd0.setCursor(1, 1);
  lcd0.print("OK Encender");
  lcd0.setCursor(1, 2);
  lcd0.print("X Apagar");
  digitalWrite(pinBomba, LOW);
  digitalWrite(pinExtractor, LOW);
  digitalWrite(pinBomba2, LOW);
  digitalWrite(pinExtractor2, LOW);
  digitalWrite(pinVentilador2, LOW);
  delay(1500);
}
void ventiladorManual_2()
{
  lcd0.clear();
  lcd0.setCursor(0, 0);

```

```

    lcd0.print("** Ventilador_2 ***");
    lcd0.setCursor(1, 1);
    lcd0.print("OK Encender");
    lcd0.setCursor(1, 2);
    lcd0.print("X Apagar");
    digitalWrite(pinBomba, LOW);
    digitalWrite(pinExtractor, LOW);
    digitalWrite(pinBomba2, LOW);
    digitalWrite(pinExtractor2, LOW);
    digitalWrite(pinVentilador, LOW);
    delay(1500);
}
void extractorManual_1()
{
    lcd0.clear();
    lcd0.setCursor(0, 0);
    lcd0.print("*** Extractor_1 ***");
    lcd0.setCursor(1, 1);
    lcd0.print("OK Encender");
    lcd0.setCursor(1, 2);
    lcd0.print("X Apagar");
    digitalWrite(pinBomba, LOW);
    digitalWrite(pinVentilador, LOW);
    digitalWrite(pinBomba2, LOW);
    digitalWrite(pinVentilador2, LOW);
    digitalWrite(pinExtractor2, LOW);
    delay(1500);
}
void extractorManual_2()
{
    lcd0.clear();
    lcd0.setCursor(0, 0);
    lcd0.print("*** Extractor_2 ***");
    lcd0.setCursor(1, 1);
    lcd0.print("OK Encender");
    lcd0.setCursor(1, 2);
    lcd0.print("X Apagar");
    digitalWrite(pinBomba, LOW);
    digitalWrite(pinVentilador, LOW);
    digitalWrite(pinBomba2, LOW);
    digitalWrite(pinVentilador2, LOW);
    digitalWrite(pinExtractor, LOW);
    delay(1500);
}
/*-----//--
-----*/

```

Anexo III- Direcciones DS18B20

```
#include <OneWire.h>

OneWire ourWire(32);           //Se establece el pin 32 como bus
OneWire

void setup(void) {
  Serial.begin(9600);
}

void loop(void) {
  byte addr[8];
  Serial.println("Obteniendo direcciones:");
  while (ourWire.search(addr))
  {
    Serial.print("Address = ");
    for( int i = 0; i < 8; i++) {
      Serial.print(" 0x");
      Serial.print(addr[i], HEX);
    }
    Serial.println();
  }

  Serial.println();
  ourWire.reset_search();
  delay(2000);
}
```

```

sketch_mar23a | Arduino IDE 2.3.4
Archivo  Editar  Sketch  Herramientas  Ayuda
LilyGo T-Display
sketch_mar23a.ino
1  #include <OneWire.h>
2
3  OneWire ourWire(32); //Se establece el pin 32 como bus OneWire
4
5  void setup(void) {
6    Serial.begin(9600);
7  }
8
9  void loop(void) {
10   byte addr[8];
11   Serial.println("Obteniendo direcciones:");
12   while (ourWire.search(addr))
13   {
14     Serial.print("Address = ");
15     for( int i = 0; i < 8; i++) {
16       Serial.print(" 0x");
17       Serial.print(addr[i], HEX);
18     }
19     Serial.println();
20   }
21
22   Serial.println();
23   ourWire.reset_search();
24   delay(2000);
25 }
Salida  Monitor Serie x
Mensaje (Intro para mandar el mensaje de 'LilyGo T-Display' a 'COM3')
Address = 0x28 0x47 0x7D 0x80 0xE3 0xE1 0x3C 0x39
Obteniendo direcciones:
Address = 0x28 0x33 0xA9 0xF2 0x24 0x23 0xB 0xA2
Address = 0x28 0x47 0x7D 0x80 0xE3 0xE1 0x3C 0x39

```

Figura 46: Obtención de direcciones DS18B20

Vemos que obtenemos 2 direcciones (ya que se conectaron 2 sensores)

0x28 0x33 0xA9 0xF2 0x24 0x23 0xB 0xA2

0x28 0x47 0x7D 0x80 0xE3 0xE1 0x3C 0x39

Anexo IV- Fabricación de la placa

1. Planchado de nuestro esquemático en la placa virgen



Figura 47: Planchado de 2 minutos en temperatura seda

2. Comprobación de la impresión y preparación para ingresar a la placa en el Cloruro férrico concentrado

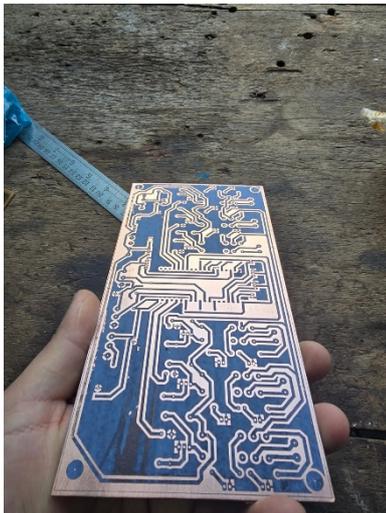


Figura 48: Placa con la transferencia realizada

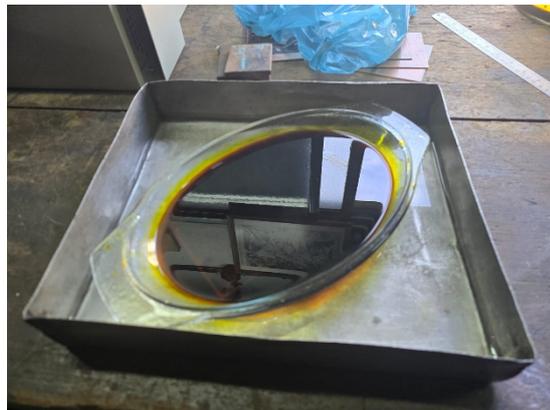


Figura 49: Cloruro férrico en baño maría (para obtener 40°C)

3. Placa luego de 8 minutos en cloruro férrico y limpieza de la tinta con virulana fina

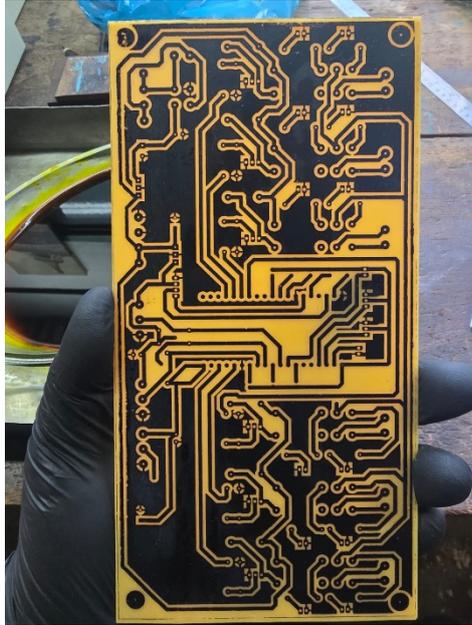


Figura 51: Placa con el cobre sobrante disuelto

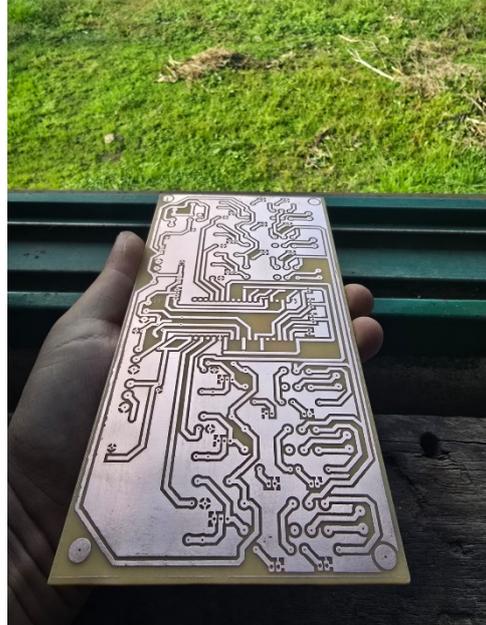


Figura 50: Placa luego de la virulana fina

4. Perforación, colocación y soldado de cada uno de los componentes



Figura 54: perforando placa

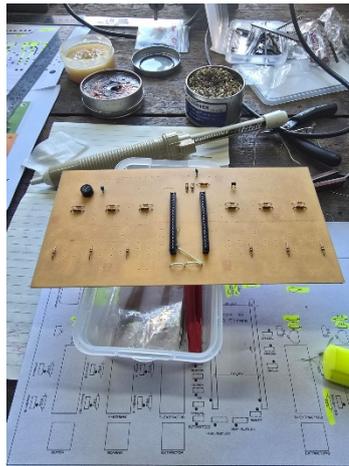


Figura 53: incorporando componentes

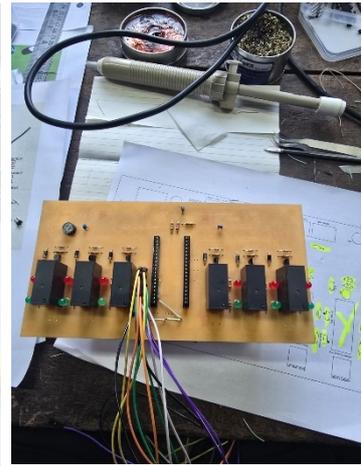


Figura 52: símil figura52

5. Ensayos del funcionamiento de la placa

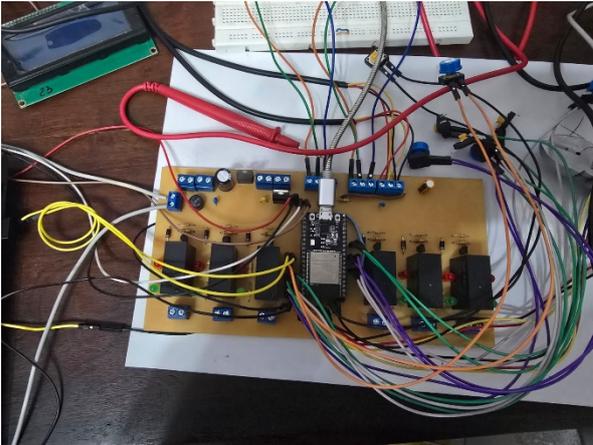


Figura 56: conectando los sensores

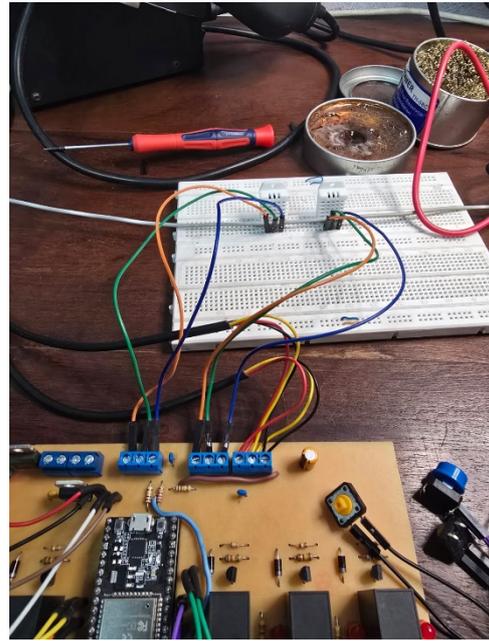


Figura 55: conectando los DHT22

Anexo V- Armado Final

1. Perforando gabinete para la colocación de los display



Figura 58: cortando con mini torno y colocando los display



Figura 57: Armado del Frente de controles

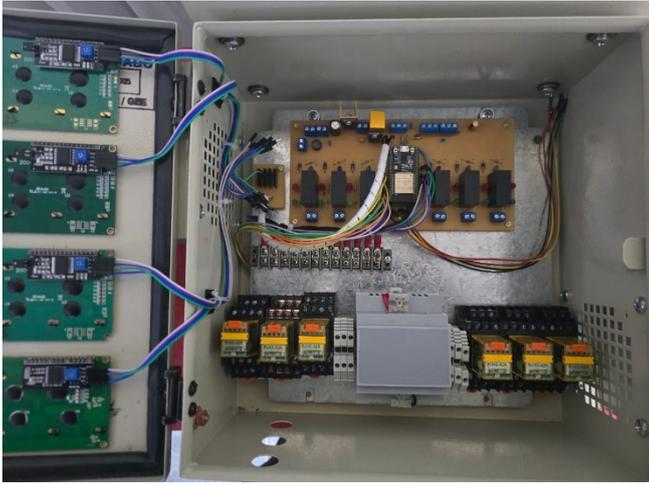


Figura 62: Empezando a cablear



Figura 61: Haciendo pruebas finales parte1

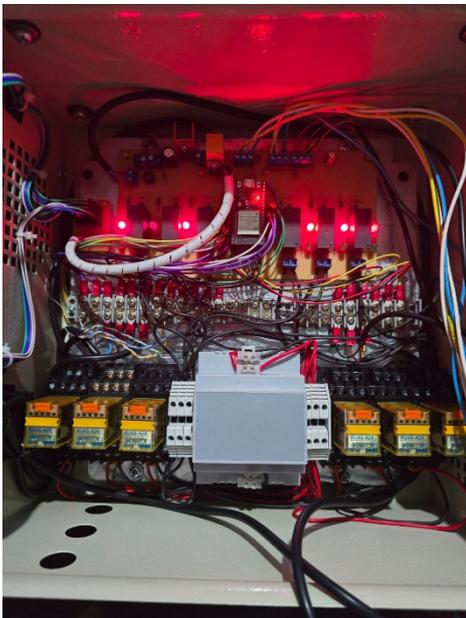


Figura 60: En pleno Funcionamiento



Figura 59: Vista final