



UNIVERSIDAD DE COSTA RICA

ESCUELA DE INGENIERÍA ELÉCTRICA

IE-0624 LABORATORIO DE MICROPROCESADORES

Proyecto Final: Sistema Hidropónico

José Antonio Ramos Pereira, B86485

Grupo 01

Prof. MSc. Marco Villalta Fallas.

https://github.com/JAR1224/proyecto_final

Índice de contenidos

1.	Introducción	1
1.1.	Resumen	1
1.2.	Conclusiones	1
2.	Justificación	1
3.	Objetivos	3
3.1.	Objetivo principal	3
3.2.	Objetivos específicos	3
3.3.	Alcances	3
4.	Metodología	3
4.1.	Software y bibliotecas	3
5.	Nota Teórica	5
5.1.	Microcontrolador	5
5.2.	Componentes Electrónicos Complementarios	10
5.3.	Precios	12
5.4.	Diseño del circuito	13
5.5.	Temas adicionales	14
6.	Desarrollo y análisis	16
6.1.	Programa	16
6.2.	Producto Final	19
7.	Conclusiones	20
7.1.	Recomendaciones	20

8. GIT	22
--------	----

1. Introducción

1.1. Resumen

La elaboración del proyecto consistió en plantear una solución utilizando un microcontrolador ArduinoUno y otros componentes electrónicos para crear un sistema hidropónico con control automático y monitoreo estilo IoT. La aplicación debía de poder controlar la temperatura del ambiente según las mediciones de temperatura y operar la bomba de agua según los niveles de agua al igual que mandar los datos por medio de la internet por medio del protocolo mqtt.

Los retos principales para la elaboración de la solución fueron los siguientes:

1. Manejo de la potencia
2. Uso del módulo WiFi Esp8266-01

Para poder superar los retos, se realizaron los siguientes pasos:

1. Usar múltiples fuentes de alimentación y relés.
2. Utilizar un módulo ftdi (convertidor UART a USB) para debuggear el módulo por medio de la PC directamente.

1.2. Conclusiones

Se concluyó que por medio de las metodologías mencionadas anteriormente, se pudo implementar una solución al problema planteado que cumple con todos los requisitos de diseño.

2. Justificación

El desarrollo de la hidroponía es importante por varias razones, entre las cuales se incluyen la sostenibilidad, la eficiencia, la capacidad de utilizar áreas urbanas y el control de la calidad y el ambiente de los cultivos.

En primer lugar, la hidroponía es un método más sostenible y eficiente de producir alimentos que los métodos tradicionales de cultivo en tierra. La hidroponía utiliza significativamente menos agua que los métodos de riego convencionales, ya que el agua se recircula constantemente y los nutrientes se proporcionan en la cantidad exacta necesaria para el crecimiento óptimo de las plantas. Además, la hidroponía no requiere el uso de pesticidas y herbicidas, lo que reduce la exposición a los químicos dañinos y la contaminación del suelo.

Adicionalmente, la hidroponía ocupa menos espacio que los métodos de cultivo en tierra. Esto es particularmente útil en áreas urbanas donde el espacio es limitado. Con la hidroponía, los cultivos pueden crecer en estructuras verticales o en espacios pequeños que no se utilizan para otros fines.

Por otra parte, la hidroponía puede desarrollarse en áreas urbanas, lo que es especialmente importante dado el aumento de la población mundial y la creciente urbanización. La capacidad de producir alimentos localmente en áreas urbanas reduce la necesidad de importar alimentos de otras áreas y, por lo tanto, reduce la huella de carbono asociada con el transporte de alimentos.

Por otro lado, la hidroponía utiliza de manera más eficiente los recursos hídricos y nutricionales en comparación con los métodos tradicionales de cultivo en tierra. Como resultado, se reduce la cantidad de agua y nutrientes necesarios para cultivar las plantas y se minimiza el desperdicio.

Por último, la hidroponía ofrece un mayor control sobre la calidad y el ambiente de los cultivos. Los cultivos hidropónicos pueden crecer en ambientes controlados, lo que significa que se puede ajustar la temperatura, la humedad y la iluminación para optimizar el crecimiento y el rendimiento de los cultivos. Además, la hidroponía reduce los problemas ambientales asociados con la contaminación por fertilizantes, ya que los nutrientes se proporcionan en cantidades precisas y no se pierden en el suelo o en el agua subterránea.

3. Objetivos

3.1. Objetivo principal

- Crear un invernadero hidropónico que se regule automáticamente el flujo de agua y la temperatura para crear un ambiente ideal para el desarrollo de plantas.

3.2. Objetivos específicos

- Desarrollar una aplicación en c++ para el Arduino Nano Ble 33 que sea capaz de controlar y monitorear un invernadero hidropónico utilizando sensores de agua y temperatura al igual que bomba de agua y ventiladores.
- Utilizar un módulo Wifi y la plataforma abierta de IoT Thingsboard para poder tener acceso remoto a la condición del invernadero.
- Lograr crear un ambiente adecuado dentro del invernadero hidropónico (cantidad correcta de nutrientes, temperatura ideal) para el crecimiento exitoso de plantas comestibles.

3.3. Alcances

- Reducir contaminación por fertilizantes
- Reducir desperdicio de agua y agroquímicos
- Brindar una forma alternativa de generación de alimentos en zonas urbanas

4. Metodología

4.1. Software y bibliotecas

- IDE: Arduino

- Lenguaje: C++
- Bibliotecas:
 - WiFiEsp
 - SoftwareSerial.h
 - PubSubClient.h

Estas bibliotecas se van a utilizar principalmente para simplificar el código relacionado al aspecto IoT del proyecto.

5. Nota Teórica

5.1. Microcontrolador

Características generales

Las características generales del Arduino Uno son las siguientes [1]:

- **ATMega328P Processor**
 - **Memory**
 - AVR CPU at up to 16 MHz
 - 32KB Flash
 - 2KB SRAM
 - 1KB EEPROM
 - **Security**
 - Power On Reset (POR)
 - Brown Out Detection (BOD)
 - **Peripherals**
 - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
 - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
 - 1x USART with fractional baud rate generator and start-of-frame detection
 - 1x controller/peripheral Serial Peripheral Interface (SPI)
 - 1x Dual mode controller/peripheral I2C
 - 1x Analog Comparator (AC) with a scalable reference input
 - Watchdog Timer with separate on-chip oscillator
 - Six PWM channels
 - Interrupt and wake-up on pin change
- **ATmega16U2 Processor**
 - 8-bit AVR® RISC-based microcontroller
- **Memory**
 - 16 KB ISP Flash
 - 512B EEPROM
 - 512B SRAM
 - debugWIRE interface for on-chip debugging and programming
- **Power**
 - 2.7-5.5 volts

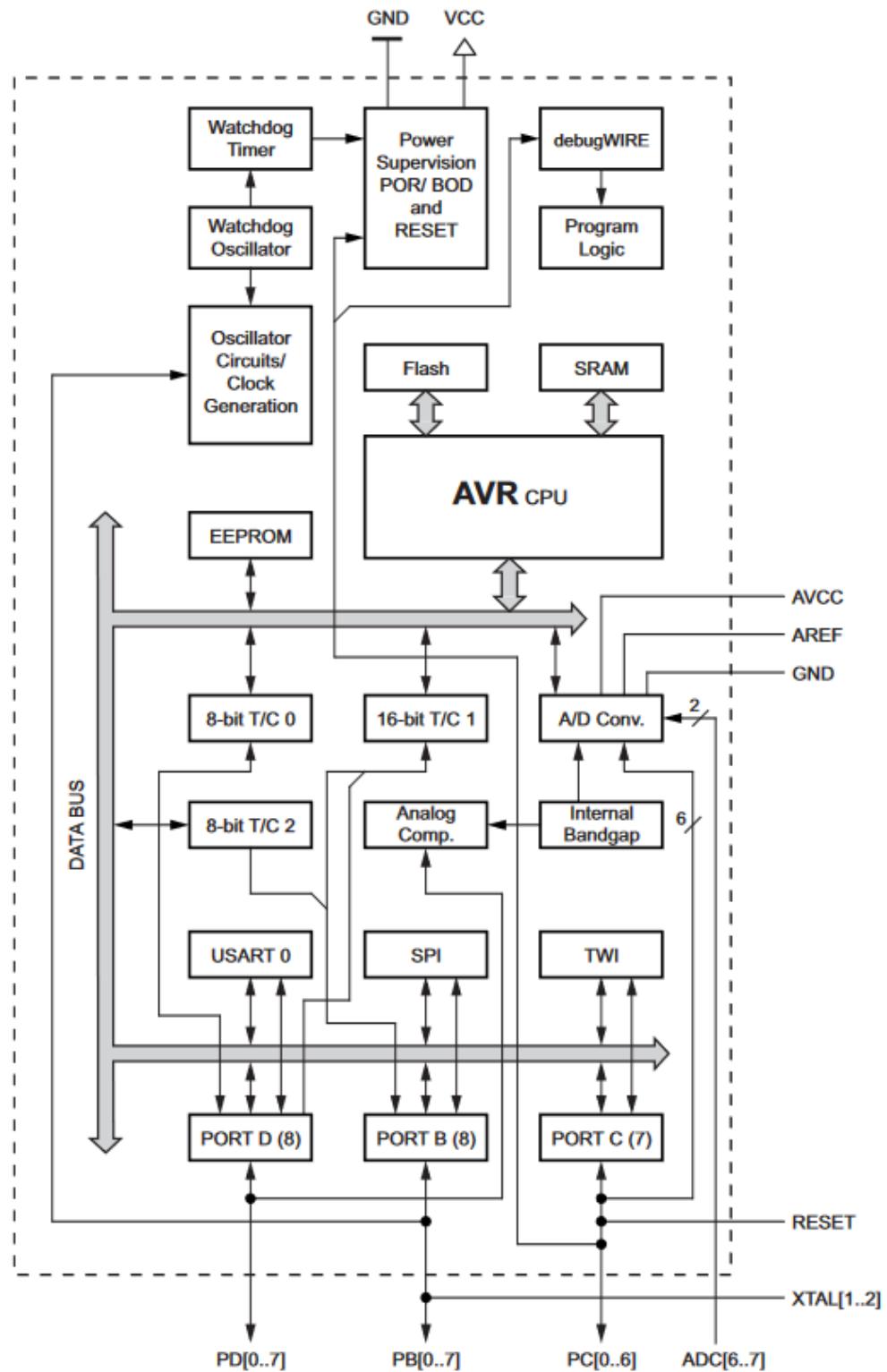
Características Generales 5.1: Obtenido de [1]

Las características generales del Atmega328P son las siguientes [2]:

- High performance, low power AVR® 8-bit microcontroller
- Advanced RISC architecture
 - 131 powerful instructions – most single clock cycle execution
 - 32×8 general purpose working registers
 - Fully static operation
 - Up to 16MIPS throughput at 16MHz
 - On-chip 2-cycle multiplier
- High endurance non-volatile memory segments
 - 32K bytes of in-system self-programmable flash program memory
 - 1Kbytes EEPROM
 - 2Kbytes internal SRAM
 - Write/erase cycles: 10,000 flash/100,000 EEPROM
 - Optional boot code section with independent lock bits
 - In-system programming by on-chip boot program
 - True read-while-write operation
 - Programming lock for software security
- Peripheral features
 - Two 8-bit Timer/Counters with separate prescaler and compare mode
 - One 16-bit Timer/Counter with separate prescaler, compare mode, and capture mode
 - Real time counter with separate oscillator
 - Six PWM channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature measurement
 - Programmable serial USART
 - Master/slave SPI serial interface
 - Byte-oriented 2-wire serial interface (Phillips I²C compatible)
 - Programmable watchdog timer with separate on-chip oscillator
 - On-chip analog comparator
 - Interrupt and wake-up on pin change
- Special microcontroller features
 - Power-on reset and programmable brown-out detection
 - Internal calibrated oscillator
 - External and internal interrupt sources
 - Six sleep modes: Idle, ADC noise reduction, power-save, power-down, standby, and extended standby

Características Generales 5.2: Obtenido de [2]

Diagrama de bloques Atmega328P



7
Diagrama de bloques 5.3: Obtenido de [2]

Diagrama de pines Arduino Uno

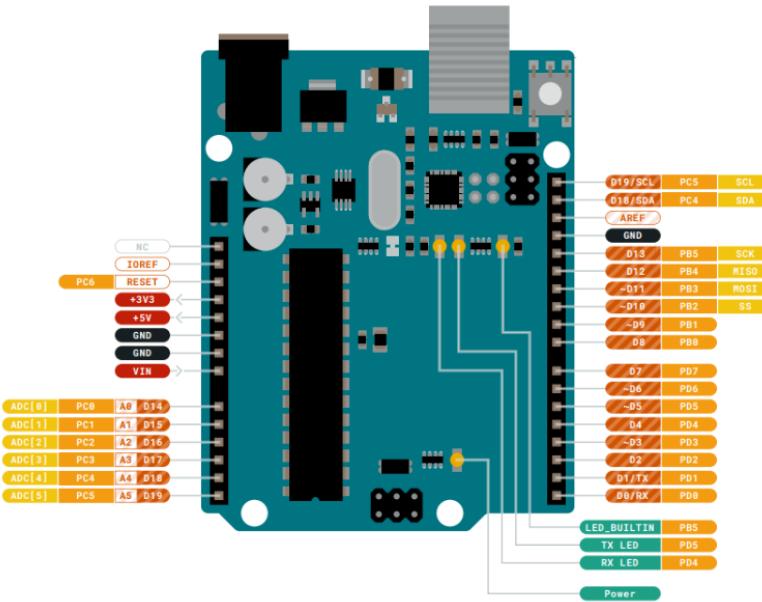


Diagrama de Pines Arduino Uno 5.4: Obtenido de [1]

Características eléctricas

Parameters	Min.	Typ.	Max.	Unit
Operating temperature	-55		+125	°C
Storage temperature	-65		+150	°C
Voltage on any pin except RESET with respect to ground	-0.5		V _{CC} + 0.5	V
Voltage on RESET with respect to ground	-0.5		+13.0	V
Maximum operating voltage		6.0		V
DC current per I/O pin		40.0		mA
DC current V _{CC} and GND pins		200.0		mA
Injection current at V _{CC} = 0V		±5.0 ⁽¹⁾		mA
Injection current at V _{CC} = 5V		±1.0		mA

Note: 1. Maximum current per port = ±30mA

Características eléctricas 5.5: Obtenido de [2]

Registros

Los registros que se utilizaron para la configuración de las interrupciones de conversión analógica digital fueron los siguientes:

ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Registro 5.6: Obtenido de [2]

MUX3..0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8 ⁽¹⁾

Bits MUX0...MUX3 5.7: Obtenido de [2]

Los bits que se configuraron para ADMUX fueron MUX0...MUX3. Estos 4 bits se utilizan para elegir cual canal se utiliza para la siguiente conversión.

ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Registro ADCSRA 5.8: Obtenido de [2]

Se configuraron los bits ADEN (para habilitar conversiones), ADSC (para iniciar una conversión) y ADIE (para habilitar la interrupción).

5.2. Componentes Electrónicos Complementarios

Módulo WiFi Esp8266-01

Características generales

1.1. Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- Wi-Fi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO

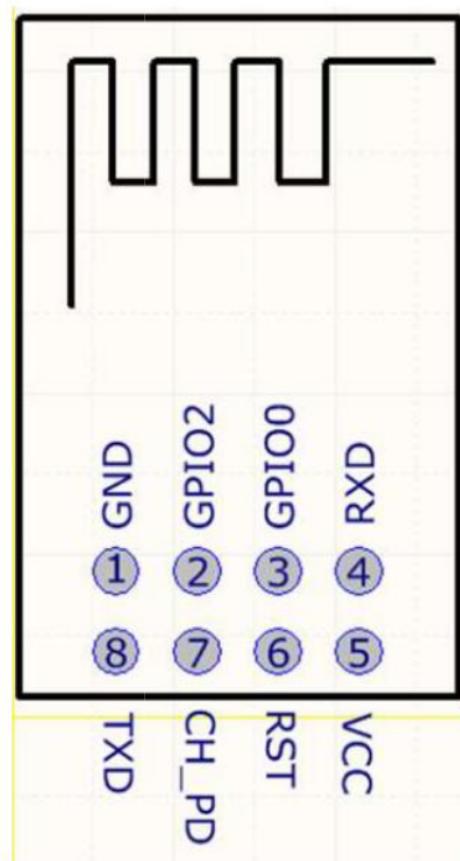
Características generales 5.9: Obtenido de [3]

Características eléctricas

Hardware Parameters	Peripheral Bus	UART/HSPI/I2C/I2S/Ir Remote Control
	GPIO/PWM	
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40~125°
	Ambient Temperature Range	Normal temperature
	Package Size	14.3mm*24.8mm*3mm
	External Interface	N/A

Características eléctricas 5.10: Obtenido de [3]

Pines



Pines 5.11: Obtenido de [3]

Otros componentes y materiales

- 3x resistencias 100 Ω
- 2x relé 5V
- 2x banco de batería AA
- 1x fuente de alimentación para protoboard 5/3,3V
- 1x sensor de agua
- 1x sensor de temperatura
- 1x Ventilador 5V
- 1x Bomba de agua 3-6V

5.3. Precios

- ArduionoUno: \$34.95
- Esp8266-01: \$8.95
- Resistencias: \$0.25
- Relés: \$3.9
- Bancos de batería: \$3
- Fuente alimentación protoboard: \$5.95
- Sensor de agua: \$1.95
- Sensor de temperatura: \$3.95
- Ventilador: \$5.95
- Bomba de agua: \$2.49

- Materiales de construcción: \$20
- Total: **\$87.39**

Obtenido de: [4]

5.4. Diseño del circuito

El circuito consta de 2 partes principales.

1) Comunicación con sensores

Los 2 sensores que se utilizaron tienen 3 pines. Uno para fuente de alimentación, tierra y un pin de señal. Los pines de señal fueron conectados a los pines de conversión analógico digital del Arduino Uno y los datos se obtienen por medio de polling.

Como los ADC del ArduinoUno son de 10 bits, entonces las conversiones son valores de 0 a 1023. Sin embargo, queremos estos valores de forma que sean fácil de leer.

Para el caso del sensor de agua se quiere normalizar estos datos del 0% al 100%. Esto es sencillo y solo se tiene que dividir el valor entre 1023 y multiplicar por 100.

$$NivelAgua = \frac{ValorADC}{1023} \times 100 \quad (1)$$

Para el caso del sensor de temperatura se quiere linealizar estos datos y convertirlos a Celsius. Este es el caso ya que el sensor utiliza un termistor no lineal. Para linealizar, se obtienen dos datos conocidos y se resuelve los valores de m y b de la ecuación $y = mx + b$. Los datos conocidos son: cuando el ADC reporta un valor de 500 esto equivale a una temperatura de 24 °C, y cuando reporta 420 esto equivale a una temperatura de 32 °C. Al resolver la ecuación obtenemos la siguiente ecuación de linealización:

$$Celsius = ValorADC \times \frac{-1}{10} + 74 \quad (2)$$

2) Manejo del ventilador y bomba de agua

Como el ventilador y la bomba da agua consumen mucha corriente, estos se alimentan por un banco de baterías por medio de un relé. El relé es controlado a su vez por los pines GPIO del ArduinoUno.

5.5. Temas adicionales

USART

La UART, conocida como Receptor/Transmisor Universal Asíncrono, desempeña un papel esencial en la comunicación de datos digitales entre dispositivos electrónicos. Funciona como una interfaz serial asíncrona que facilita la transferencia y recepción de información binaria mediante un par de cables, uno para transmitir (TX) y otro para recibir (RX).

El proceso de transmisión UART implica enviar bits secuencialmente, donde cada bit está precedido por un bit de inicio y seguido por uno o más bits de parada. La velocidad de transmisión se establece según una tasa de baudios, que representa la frecuencia de transmisión de bits por segundo.

ADC

En Arduino, se llevan a cabo las conversiones analógicas a digitales (ADC) mediante un convertidor analógico a digital de 10 bits. Este ADC mide el voltaje de entrada en una entrada analógica y lo convierte en un valor digital que abarca de 0 a 1023. En esta escala, 0 representa un voltaje de 0 voltios, mientras que 1023 representa un voltaje de referencia de 5 voltios. Para mejorar la precisión de las conversiones ADC, es posible utilizar un voltaje de referencia externo o ajustar la resolución del convertidor utilizando las funciones proporcionadas por la biblioteca de Arduino.

La biblioteca de Arduino ofrece funciones que simplifican el proceso de realizar conversiones ADC. Mediante la función `.analogRead()`, los usuarios pueden leer el valor de voltaje en una entrada analógica específica y convertirlo en un valor digital. Además, se pueden

configurar la resolución del convertidor y el voltaje de referencia utilizando las funciones `.analogReference()` y `.analogReadResolution()`. De esta manera, los usuarios tienen control sobre la precisión y configuración de las conversiones ADC en Arduino.

IoT

El Internet de las Cosas (IoT), es el término utilizado para describir la conexión en red de objetos físicos, dispositivos y sistemas a través de Internet, lo cual les permite recopilar y compartir datos.

El IoT abarca una amplia variedad de dispositivos, desde electrodomésticos inteligentes y dispositivos portátiles, hasta maquinaria industrial y vehículos. Estos dispositivos interconectados están equipados con sensores, actuadores y capacidades de comunicación, lo que les permite recopilar y transmitir datos a través de la nube u otros dispositivos. Esto posibilita el monitoreo en tiempo real, el control y la automatización de procesos.

El objetivo principal del IoT es establecer una red inteligente e interconectada en la que los objetos físicos puedan comunicarse y colaborar de manera fluida. Esto da lugar a nuevas aplicaciones y servicios que mejoran la eficiencia, la productividad y la comodidad en diversos campos, como la salud, el transporte, la agricultura y las ciudades inteligentes. En resumen, el IoT busca crear un entorno en el que los objetos cotidianos estén conectados y trabajen juntos para brindar beneficios significativos a las personas y a la sociedad en general.

Hidroponía

La hidroponía es un método de cultivo de plantas que se basa en el cultivo sin suelo tradicional. En lugar de utilizar tierra, las plantas se cultivan en soluciones acuosas que contienen nutrientes necesarios para su crecimiento y desarrollo.

La hidroponía permite a los agricultores tener un control mucho mayor sobre el entorno de crecimiento de las plantas, lo que les permite optimizar las condiciones para obtener un crecimiento y rendimiento óptimos. En lugar de depender de la calidad variable del suelo, los nutrientes se suministran directamente a las raíces de las plantas a través del agua.

Esto permite que las plantas absorban los nutrientes de manera más eficiente y les brinda la oportunidad de utilizar su energía en el crecimiento y desarrollo en lugar de buscar nutrientes en el suelo.

La hidroponía ofrece varias ventajas sobre la agricultura tradicional en suelo. En primer lugar, permite el crecimiento de plantas en áreas donde el suelo es de mala calidad o incluso inexistente, como en entornos urbanos o desérticos. Además, al controlar de cerca los nutrientes y el ambiente de crecimiento, se pueden optimizar las condiciones para maximizar la producción de alimentos y minimizar el uso de recursos, como agua y fertilizantes. También se reducen los problemas asociados con enfermedades del suelo y plagas, lo que reduce la necesidad de pesticidas.

6. Desarrollo y análisis

6.1. Programa

Descripción del programa

El programa principal utiliza la biblioteca SoftwareSerial además de WiFiesp y paho mqtt para manejar el módulo Esp8266-01 y poder conectarse con un servidor mqtt (en este caso thingsboard). Para lograr que el módulo Esp8266-01 funcione correctamente, este primero se tiene que configurar para que opere a 9600 bps en vez de 115200 (valor defecto), ya que esta frecuencia es incompatible con la biblioteca SoftwareSerial. Adicionalmente, se tuvo que modificar la biblioteca SoftwareSerial para que agregara un delay de $5\mu s$ entre el envío de cada byte.

La funcionalidad del programa consiste de lo siguiente:

Se leen los valores de los convertidores ADC para los sensores de temperatura y agua. Luego se normalizan los datos provenientes del sensor del agua y se linealizan los que provienen del sensor de temperatura. Luego, basado en estos valores, se decide si se tiene que activar/desactivar el ventilador/bomba de agua. Por último, se manda la telemetría a la plataforma thingsboard en paquetes json por medio del módulo WiFi.

La lógica que enciende y apaga el ventilador es la siguiente:

Esta sigue una curva de histéresis. Es decir, si el ventilador está apagado, este solo se enciende si la temperatura supera los 28 °C. Por otro lado, si el ventilador está prendido, este solo se va a apagar cuando la temperatura baje de los 26 °C. Esta funcionalidad se diseñó de esta forma para evitar que el ruido de la instrumentación causara en algunos casos que el relé que maneja el ventilador se prendiera y apagara a frecuencias altas en momentos cuando la temperatura es cercana a la temperatura de umbral.

La lógica que enciende y apaga la bomba de agua es la siguiente:

Cuando el sensor de agua detecta un nivel de agua por debajo del 40 %, se apaga la bomba para evitar que este corra en seco y se queme por operar sin carga.

Diagrama de bloques

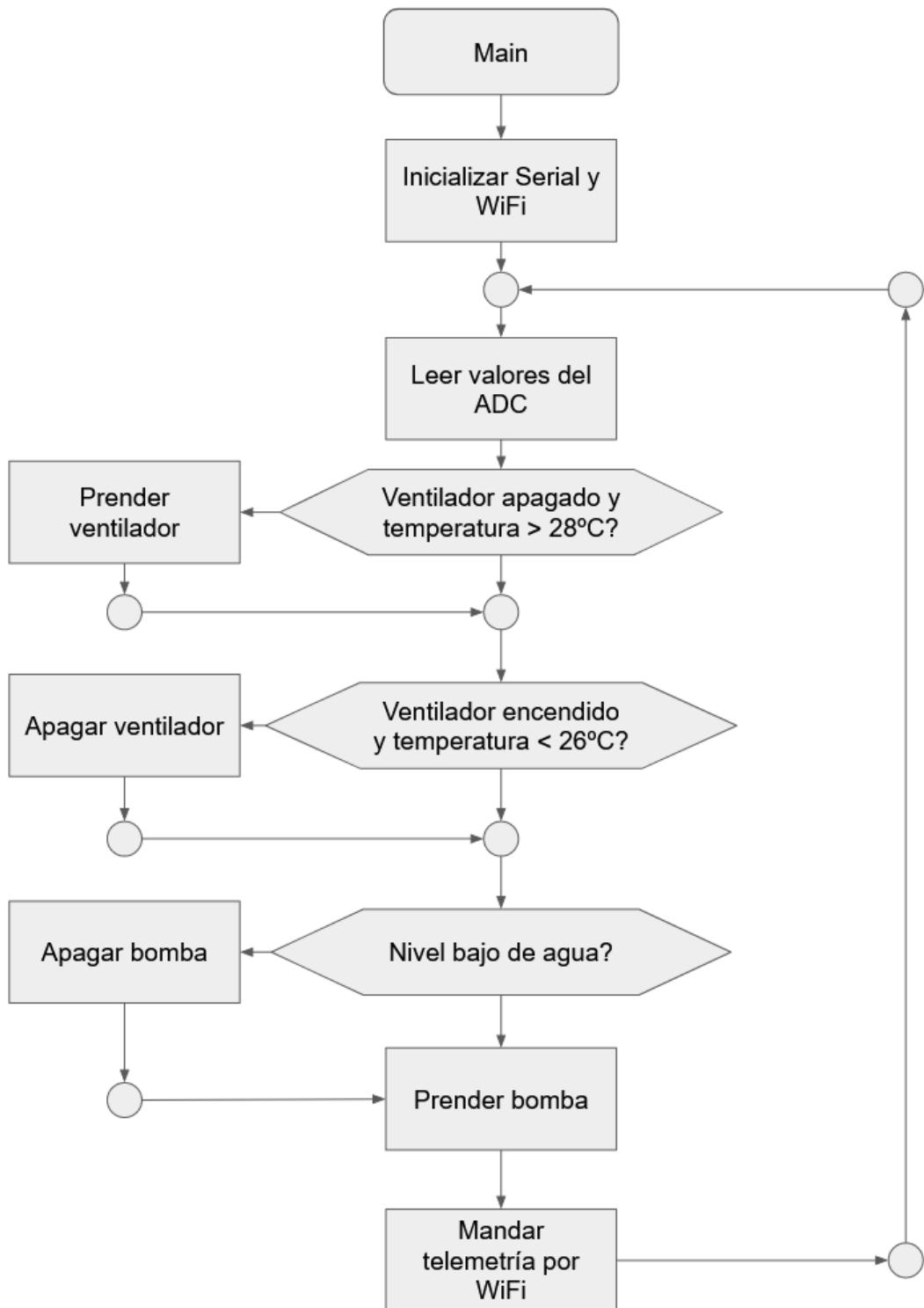
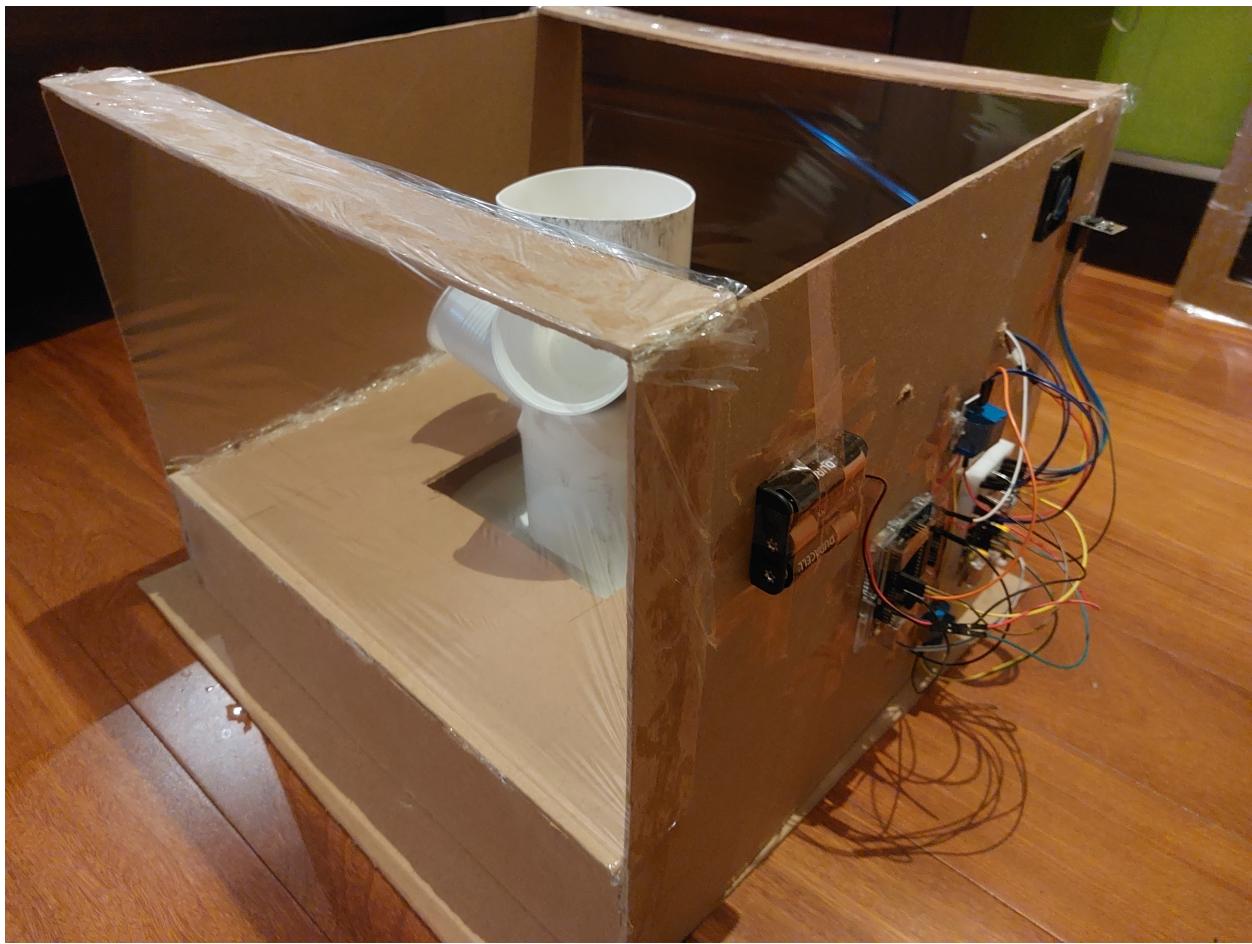
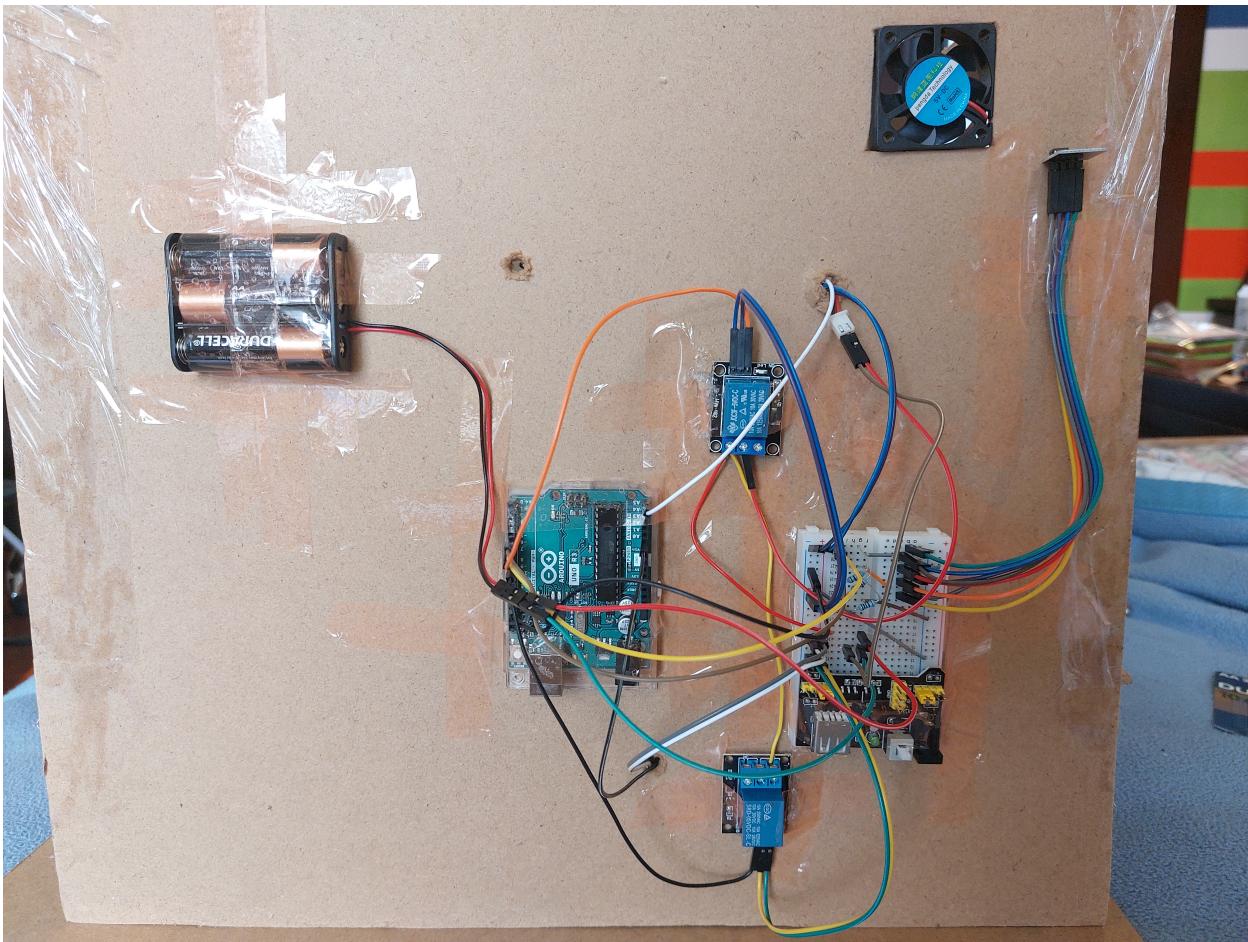


Diagrama de bloques program^a principal 6.1: Obtenido de [3]

6.2. Producto Final



Vista 1 6.2:



Vista 2 6.3:

7. Conclusiones

Se concluyé que la solución propuesta para este proyecto final cumple con los requerimientos de la propuesta y se comporta según el diseño teórico realizado.

7.1. Recomendaciones

Se tienen las siguientes posibles continuaciones que se le pueden dar al proyecto:

Agregar un teclado y una pantalla LCD para que el usuario pueda configurar a gusto propio los umbrales de encendido para el control del ventilador y bomba de agua

Agregar sensores adicionales como sensor pH y sensor NPK.

Agregar una cámara y por medio de machine learning, implementar la detección automática del estado de crecimiento de la planta al igual que la detección automática de posibles enfermedades en la planta.

Efectuar el proyecto en general a una escala más grande.

Bibliografía

- [1] “Arduino uno data sheet.” <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>, 2007. (Accessed on 21/04/2023).
- [2] “Atmega328p data sheet.” https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf, 2007. (Accessed on 21/04/2023).
- [3] “Eso8266-01 data sheet.” <https://pdf1.alldatasheet.com/datasheet-pdf/download/1179098/ETC2/ESP-01.html>, 2007. (Accessed on 21/04/2023).
- [4] “Componentes electrónicos.” <https://www.crcibernetica.com/>, 2003. (Accessed on 25/03/2023).

8. GIT

https://github.com/JAR1224/proyecto_final