



Final Technical Report

Smart liquid level Controller for Industrial Internet of Things.

BY

Bara'a Eyad Al Masha'leh
Manar Mohammad Qrenawi
Mohammad Amjad AlQuraan
Osama Adnan Azaizah

ACKNOWLEDGEMENTS

First, we would like to say thanks to Dr. Mahmoud Zarini, the CEO of ISRAR ENGINEERING LLC for giving us this opportunity and make us challenge ourselves in this training starting from choosing the project and through the hard work to complete it, because of you we learned things that we wouldn't ever learn, and we have the chance to be in real-life work experience. We would also thank the engineer Ahmad Zarara and Ahmad Al-joghaimi for their efforts with us all the road till this moment we learned a lot from them in the hardware and software parts of the project and without them it would not be possible to complete it with high quality result.

ABSTRACT

Industrial Storage tanks are used to store liquids or fluids. Overflow of the liquids may threaten the safety of the factory or affect the production process or even create environmental problems, on other hand, low level of liquids can also cause problems and damage to the equipment or may stop the process line and waste power.

A lot of modern Technology depends on automation and control systems. We will describe our work of implementing liquid level measurement solution for industrial tanks and control the process of filling and draining them.

We will walk through our implementation from building the circuit schema and choosing the right component to design the PCB of this board.

Also, we will discuss the different methods of measuring the level and what we chose for our solution and write software that test the filling and draining operation using tanks and pumps that connected to our device.

We implemented a controller can be connected to several components to control the amount of liquid in tanks in automated way, and it can be configured remotely or in the embedded code.

TABLE OF CONTENTS

	PAGE
DEDICATIONS.....	ii
DECLARATIONS.....	iii
ACKNOWLEDGEMENTS.....	iv
ABSTRACT	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER	
1. INTRODUCTION.....	1
1.1. Problem	1
1.2. Aim of the project	2
1.3. Applications	2
2. REVIEW OF RELATED LITERATURE	2
2.1. Introduction	2
2.2. Related work	2
2.3. Why our solution is different	3
3. ANALYSIS AND DESIGN	4
3.1. Design Requirements	4
3.2. Engineering Standards	4
3.3. Realistic Constrains.....	5
3.4. Alternative/Different Designs Approaches/Choices	5
3.4.1. Methods of measuring	5
3.4.2. Types of controllers	7
3.5. Developed Design.....	8
3.5.1. Block diagram	9
3.5.2. ESP32 Wi-Fi modes	9
3.5.3. System workflow	10
3.5.4. Ultrasonic	11
3.5.5. Prototype design	12
3.6. Development tools	14
4. IMPLEMENTATION	17
4.1. Circuit implementation	17
4.2. PCB implementation	23
4.2.1. Signal integrity	23
4.2.2. Placing components	25
4.2.3. Routing	26
4.3. Prototype implementation	29
5. TESTING AND EVALUATION	31
5.1. Power supply testing	31
5.2. ESP32 testing	33
5.3. LCD testing	35
5.4. Analog output testing	35
5.5. Analog input testing	38
6. CONCLUSION	41
REFERENCES	42

LIST OF TABLES

TABLE	PAGE
5.1 Reverse Polarity and ESD Protection voltage test.	26
5.2 DC-DC Converter voltage test.	27
5.3 LDO voltage test.	28
5.4 Boost converter voltage test.....	28

LIST OF FIGURES

FIGURE	PAGE
1.1 Industrial tanks	1
2.1 Water buoy.....	2
2.2 Seven sensors measurement method.	3
3.1 Differential pressure measurement.	6
3.2 Radar measurement.....	7
3.3 SAMD21 microchip.....	8
3.4 ESP32 microcontroller.....	8
3.5 Product design process.....	9
3.6 Device block diagram.....	10
3.7 ESP32 Wi-Fi modes.....	11
3.8 System workflow.	11
3.9 Controller flowchart.	12
3.10 Ultrasonic measurement.	13
3.11 Prototype design.	13
3.12 Edges and cylinder in prototype.	14
3.13 EasyEDA schematic design.	15
3.14 EasyEDA PCB design.	15
3.15 VS Code IDE running Platform IO.	16
4.1 Circuit components.	17
4.2 Power connector circuit.	18
4.3 Protection circuit.	18
4.4 DC-DC converter circuit.	19
4.5 Human protection circuit.	19
4.6 LDO circuit.	19
4.7 EEPROM Circuit and pinout configuration.	20
4.8 Booster circuit.....	20
4.9 ESP32 circuit.	21
4.10 RS485 circuit.....	22
4.11 Ethernet module and W5500 Circuit.	22
4.12 Ethernet module connections with ESP32.	23
4.13 Crosstalk phenomenon.	24
4.14 Return path.....	24
4.15 PCB component placement.	26
4.16 PCB Nets.	27
4.17 PCB Nets without GND.	27
4.18 PCB After routing.	28
4.19 Prototype pseudocode.	29
4.20 Liquid level Controller.....	29
4.21 Prototype tanks and pumps.....	30
5.1 ESP32 blink test.....	34
5.2 LCD test.	35
5.3 Analog output module.....	36
5.4 Analog output module connections.....	36
5.5 Power supply connected to pin 5 and 3.	37
5.6 Power supply connected to pin 4 and 3.	37
5.7 Analog output module multimeter test.....	37

5.8 Analog input module.....	38
5.9 Analog input module connections.....	39
5.10 Power supply connected to pin 7 and 5.	39
5.11 Power supply connected to pin 1 and 2.	40
5.12 Analog output module multimeter test.....	40

CHAPTER 1

INTRODUCTION

Industrial Internet of Things (IIoT) refers to interconnected sensors, instruments, and other devices networked together with computers' industrial applications, including manufacturing and energy management. In our case, we deal with a problem in factories that have tanks to store different kinds of chemical liquids or fluids, getting accurate level measurements in the tank is very important.

1.1. Problem

Industrial Storage tanks, as shown in Figure 1.1, are extensively used to store chemical liquids or fluids. High levels can cause overflow that may threaten the safety of the factory or affect the production process or even create environmental problems. Low levels, on other hand, can also cause problems and damage the equipment such as pumps or may stop the process line and waste power. Connected sensors and actuators enabled factories or companies to pick up on inefficiencies and problems sooner. This connectivity allows for data collection, exchange, and analysis, potentially facilitating improvements in productivity and efficiency as well as other economic benefits.



Figure 1.1: Industrial tanks

1.2. Aim of the project

There is a need for a device that can sense, control, and actuate the tank level by opening/closing pumps and valves based on the pre-determined threshold of level measured using a measurement approach like ultrasonic sensor. This would save time, money and increase the productivity success in the factory.

Smart Liquid Level Controller is a module used to detect the liquid level in a tank and display the liquid level on a digital display screen. Also, based on the information received by devices, sensors or preprogrammed settings, the device performs complex applications with specific functionality to control huge pumps and valves responsible for making filling and draining operations actuated with little power.

The device can be actuated with different power supply options, and it can be configured easily and work for a long time. In addition, it can withstand and perform well under harsh industrial conditions.

1.3. Applications

Level control is often performed in industrial and scientific organizations to occasionally monitor the contents of tanks covering a wide range of operations.

There are several industrial applications for liquid level measurement Food and beverage industry for milk and ensure the acid, oil, and solvent contents of vessel in chemical plants are within range, monitor the water level in the tanks.

CHAPTER 2

REVIEW OF RELATED LITERATURE

2.1 Introduction

Monitoring tank levels was a hard issue in the past, especially in the industrial field. Liquid level controller system is very useful to reduce the waste of liquid from the tanks while filling them. A tank could be in a hard place that workers cannot reach due to the tank temperature or height.

The first solution was a water buoy as shown in Figure 2.1 Its function is to maintain the water level in tanks and basins, also allow water to enter them in the event of a drop in the level.



Figure 2.1: Water buoy

2.2 Related work

Many controllers were developed in the recent years to ensure that tanks in industrial fields have the right level that they need using sensors to read the levels.

One of the first solutions in water level measurement is to use 7 sensors, and put all of them inside the tank, as shown in the Figure 2.2, each sensor represents certain level of water this approach is not very accurate because it has only seven readings.

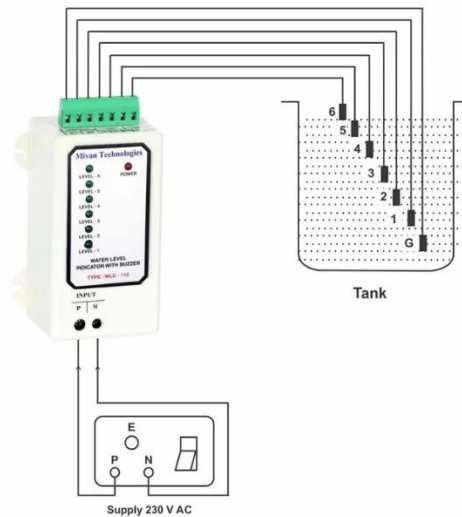


Figure 2.2: Seven sensors measurement method.

The researcher showed how to build a liquid tank level controller using Arduino microcontroller with LabVIEW program and ultrasonic sensor.

The ultrasonic sensor that they used is placed at the top of the tank to measure the filling level of the tank, so they can gain information about the depth of the water so it can be used to set the minimum and maximum threshold levels.

The LabVIEW program sends commands to the Arduino to execute a coded program. This operation is done by a USB bus to send back data to the PC from the digital input of the Arduino; they used the same USB bus. This system uses digital input and output, as mentioned before. The ultrasonic sensor is used to calculate the level of the tank. When running the program and the level of water is low, the controller will send a digital signal to the pump and the pump will start filling the tank, when the water reaches the highest level of the tank, the pump is switched off.

2.3 Why our solution is different.

In our solution, we chose to build a device that can control different equipment such as pumps and valves or both; it could be completely automated based on thresholds and pre-configured settings. Also, the device can be monitored and controlled remotely using Wi-Fi and can work under harsh industrial environments. In addition, the connected equipment can be controlled by sending analog output signals so we can accurately fill and drain the tank.

CHAPTER 3

ANALYSIS AND DESIGN

Design the solution of any problem is critical for any project a lot of details should be considered through all the design stages, so we must analysis the problem carefully by setting up the design requirements and consider any possible constraints or challenges while implementing our design from the first phase of building the board circuit to the testing phase of its functionality.

3.1. Design Requirements

We had to design a device that works in an automated system for controlling filling and draining operation, to protect pumps and valves from being damaged, and prevent any fluid overflow that can cause environmental problems. This will increase the quality of the production process. So, the design requirements to make such a device are: -

- Design circuit diagram of the device and get familiar with its component
- Knowledge of using EasyEDA to design PCB board of the device
- Practice on Multisim Live Online Circuit Simulator
- Understanding of different types of sensors (ultrasonic, pressure ... etc.)
- Understand different signal integrity topics in PCB Design as crosstalk, differential pairs, return path, EMI, EMC, coupling and decoupling
- Knowledge of C++ programming language for embedded systems
- VS Code IDE + platform IO or Arduino IDE
- ESP32 micro controller or SAMD21
- Web Server and knowledge of HTML, CSS, JS

3.2. Engineering Standards

There are several protocols and standards we will obey in our design of the device. In the communication, the data will be sent from the web server to the controller over Wi-Fi Protocol, other standards will be used in designing the PCB, protection of the device, or for security. Here is a list of a standards that will be followed:

- Ethernet standard
- IEEE 802.11
- IPC-2221A
- IPC-2220
- Ingress protection standard
- ROHS standard
- SSL security standard

3.3. Realistic Constrains

The ability to design a device that meets the desired needs within realistic constraints such as (economic, environmental, manufacturability and safety) is very important. For our final design, we need to be resourceful enough to predict any potential difficulty in manufacturing the device and determine the weaknesses that can threaten and damage the product on site based on ingress protection standards. The realistic constraints are:

- Lack of electronic components manufacturing in the world during the corona pandemic
- Protected only from limited dust ingress.
- The device is not protected against liquids.
- Vapor in liquids causes wrong readings for the ultrasonic sensor and it builds pressure in tanks. Also, it causes reflection of the waves
- Variable density can vary quite a bit with temperature variety. Reading the level of water in the morning it's not the same as the night.

3.4. Alternative/Different Designs Approaches/Choices

In the process of designing the device and archive its proper functionality like measuring the level and communicate with the connected components we found that there are different choices we can take regarding to how we want the device to work and for whom the device is directed to, so we will talk about different types of measuring and different type of controllers we can use.

3.4.1 Methods of measuring

The most important feature in the device is measuring the level of the liquid inside industrial tanks as accurate as possible while searching in this topic we found that there are different approaches can achieve this purpose.

differential pressure method (DP method), this approach necessitates the use of a DP transmitter and a port, which are linked to the external tank at opposite ends. The differential pressure in the tank is monitored between the DP transmitter at the bottom and the port at the top; the output of the differential pressure determined by the DP transmitter is proportionate to the liquid level. The more liquid in the tank, the greater pressure at the transmitter; the less liquid in the tank, the more pressure at the port, Figure 3.1 bellow shows the DP measurement approach.

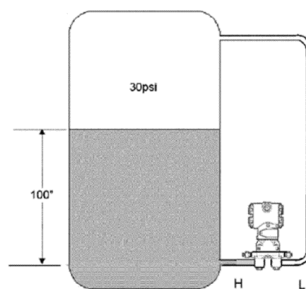


Figure 3.1: differential pressure measurement

The ultrasonic approach is the non-contact approach, a transmitter is positioned on top of the tank, and ultrasonic sound waves are delivered from the transmitter to the surface of the measured fluid. An echo of the wave is computed, and the time it takes for the wave to reach its destination from the transmitter becomes its distance. The time of the length of the distance is then calibrated in terms of the degree of process material.

The radar approach is a non-contact method that employs electromagnetic radiation. A transmitter sends electromagnetic waves to the surface of the measured material. A receiver at the tank's bottom captures and reflects some of the wave's energy back toward the medium's surface. The reflected energy is then used to calibrate the level measurement, Figure 3.2 below shows the radar measurement approach.

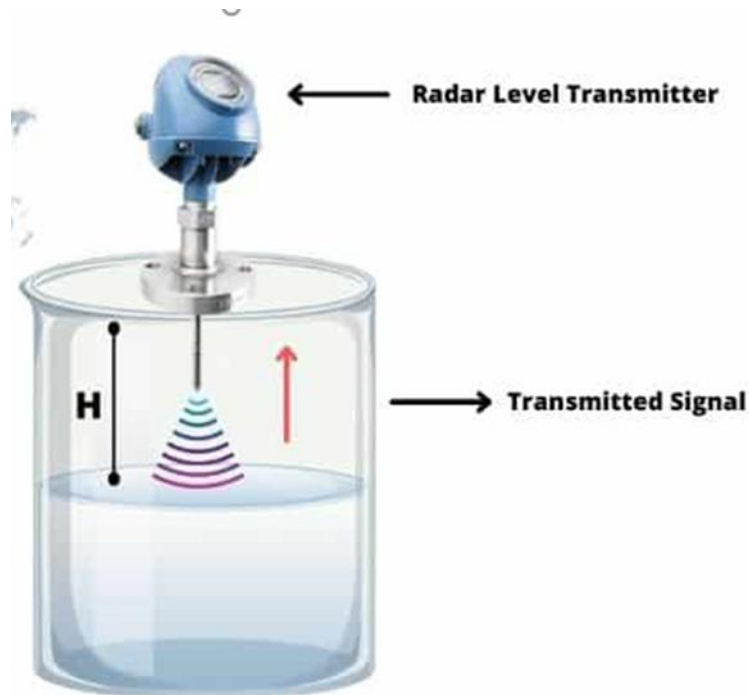


Figure 3.2: Radar measurement.

Magneto strictive level measurement approach, this approach can simultaneously measure multiple parameters such as liquid level, interface level, and temperature. The measurement range is large, and the maximum measurement range can be 20m, high measurement precision of up to 1mm is possible with measurement grade items, pure electronic technology is used to measure resulting in no wear, a safe and reliable measurement, and a long service life. Factors as container design and steam content have no effect on measurement accuracy.

3.4.2 Types of controllers

For the controllers there are different types of controllers depends on in which environment the device is directed to, if we want the device to have Wi-Fi connectivity, Bluetooth connectivity, and serial connectivity we will choose ESP32 as our microcontroller, and if we want to choose Modbus protocol as a communication protocol only, we may choose SAMD21 microchip for the controller.

SAMD21 is a low-power, high-performance microcontroller it has a large memory and fast operating speed with 256KB of flash and 32KB of SRAM and max CPU speed of 48MHz the Figure 3.3 bellow shows SAMD21 microchip.

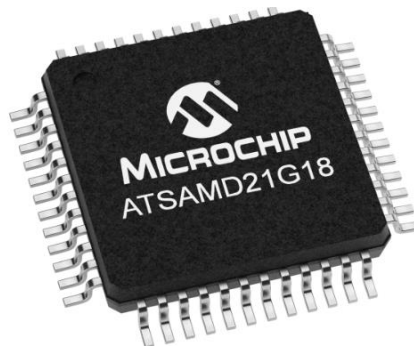


Figure 3.3: SAMD21 microchip.

ESP32 also is a low-power and low-cost, high-performance microcontroller it has a large memory 520 KB of SRAM, 448 KB of ROM and max clock frequency up to 240 MHz, and the most important feature it comes with integrated Wi-Fi and Bluetooth and integrated RF components like power amplifier, low-noise receive amplifier and antenna switch, the Figure 3.4 bellow show ESP32 microcontroller.

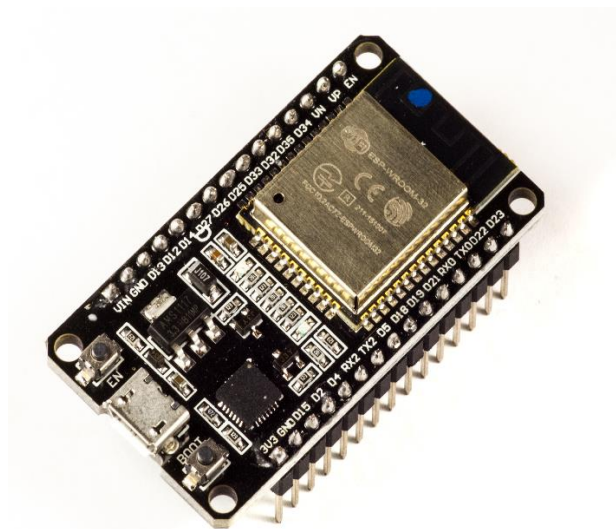


Figure 3.4: ESP32 microcontroller.

3.5. Developed Design

The design process of the device will be as Figure 3.5 bellow. First, we will design the circuit as the initial and the basic stage in the design workflow, we will select the needed components for the purpose of device function considering various things like cost, size, specification, availability ... etc. After the components selection we will use an electronic design automation tool in our case we will work on EasyEDA to design the schematic and the PCB board. To design the PCB, we will place the circuit component in an accurate way to make the PCB perform its functionality effectively, then we will do the routing and add proper wires to connect the components while obeying the IPC standard of designing PCB, then we will develop the software of the device by building a library containing the firmware of the product which will make it perform its expected function.

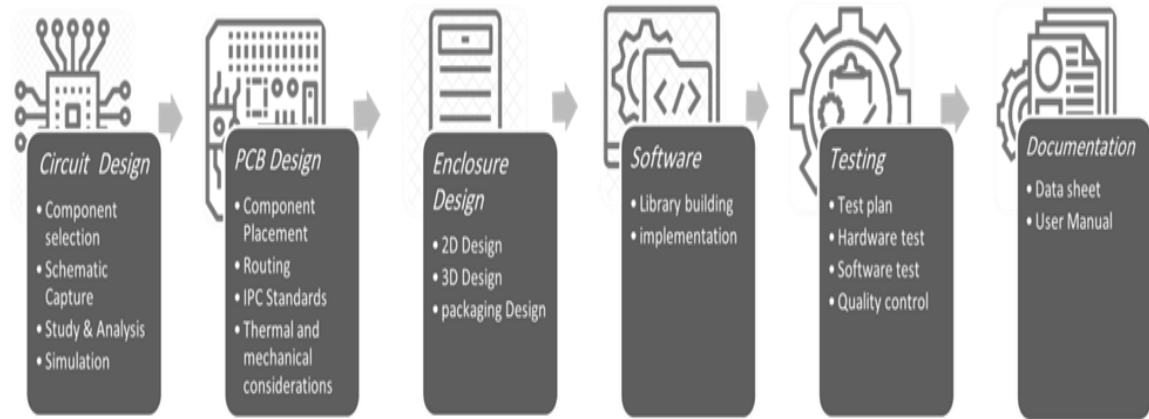


Figure 3.5: Product design process.

3.5.1 Block diagram

The Figure 3.8 presents the block diagram of our device. We have different modules performs different tasks and each module circuit have different hardware components picked carefully to make the device achieve its proper functionality.

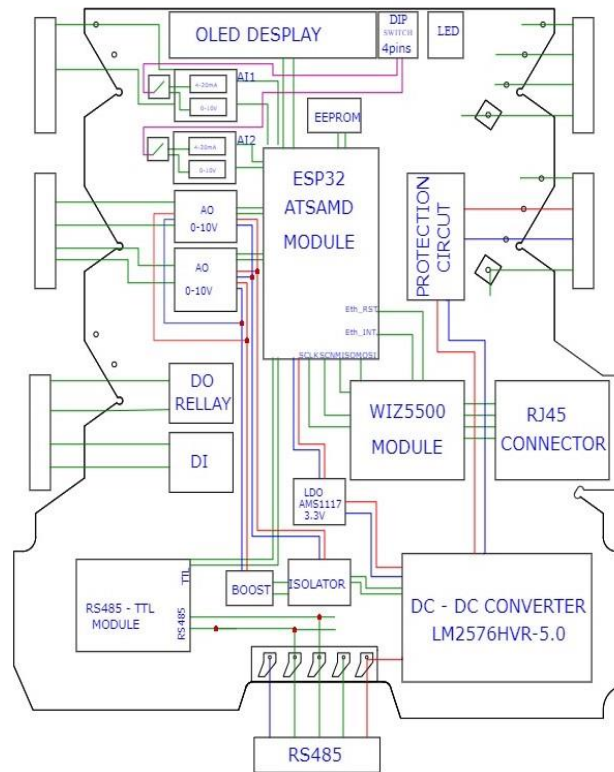


Figure 3.6: Device block diagram.

3.5.2 ESP32 Wi-Fi modes

We chose ESP32 microcontroller for the design because it has a built in Wi-Fi module, which is a huge advantage because many microcontrollers don't have it, and you should buy external Wi-Fi module to add Wi-Fi connectivity in their projects.

To take advantage and use the Wi-Fi functionality in ESP32 we used the built in Wi-Fi library. The ESP32 comes with two different Wi-Fi modes, AP-mode, station-mode, and it can operate both together.

Access Point mode makes the ESP32 act as AP and create its local network, stations like mobile and laptop can connect to it and control it directly without a need for a router.

Station mode makes the ESP32 act as station and connect to external router, and be part of network, this way other stations in the same network can control it by connection to its IP Address, Figure 3.7 shows ESP32 different Wi-Fi modes.

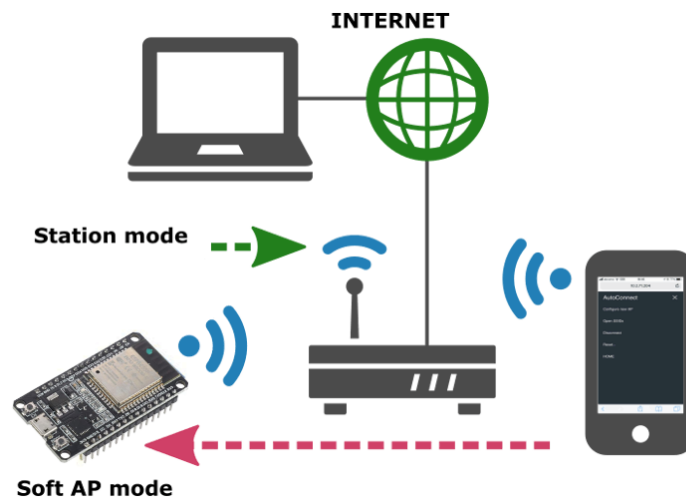


Figure 3.7: ESP32 Wi-Fi modes

3.5.3 System workflow

Figure 3.8 shows the system workflow, initially the controller will run a webserver and wait for clients to connect to it, after connection the client will setup configuration of the tank by determine the high and the low thresholds of the tank. The data transfer between the client and the controller will be over HTTP protocol.

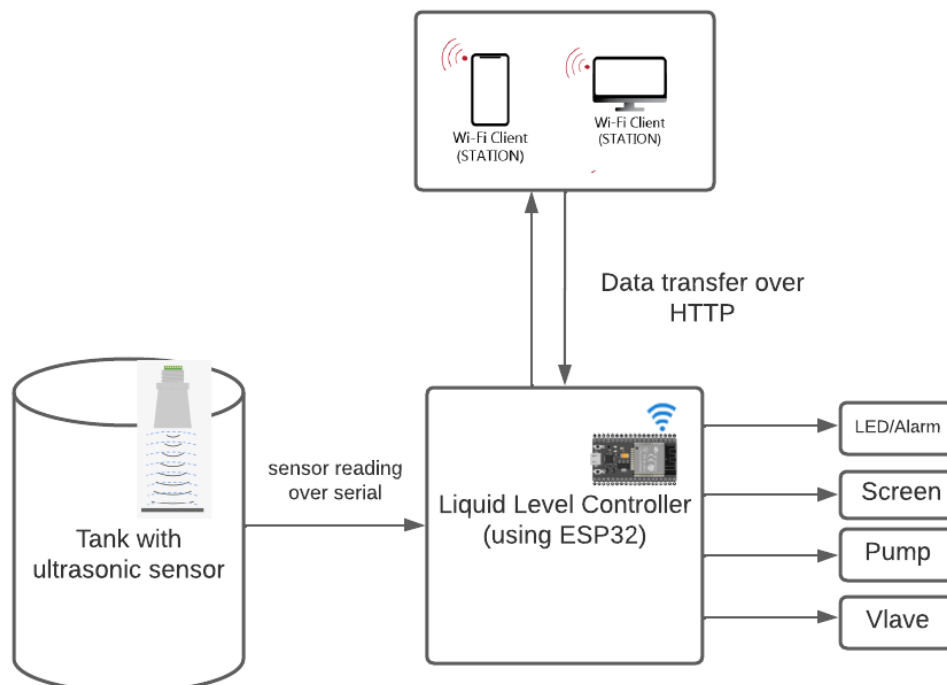


Figure 3.8: System workflow

Figure 3.4 presents the flow chart of the controller, after the device setup the configuration either by a webserver or pre-programmed settings in the embedded code, it will read data

from the ultra-sonic sensor to measure the level of the liquid and based on the thresholds it will drain or fill the tank in automated way.

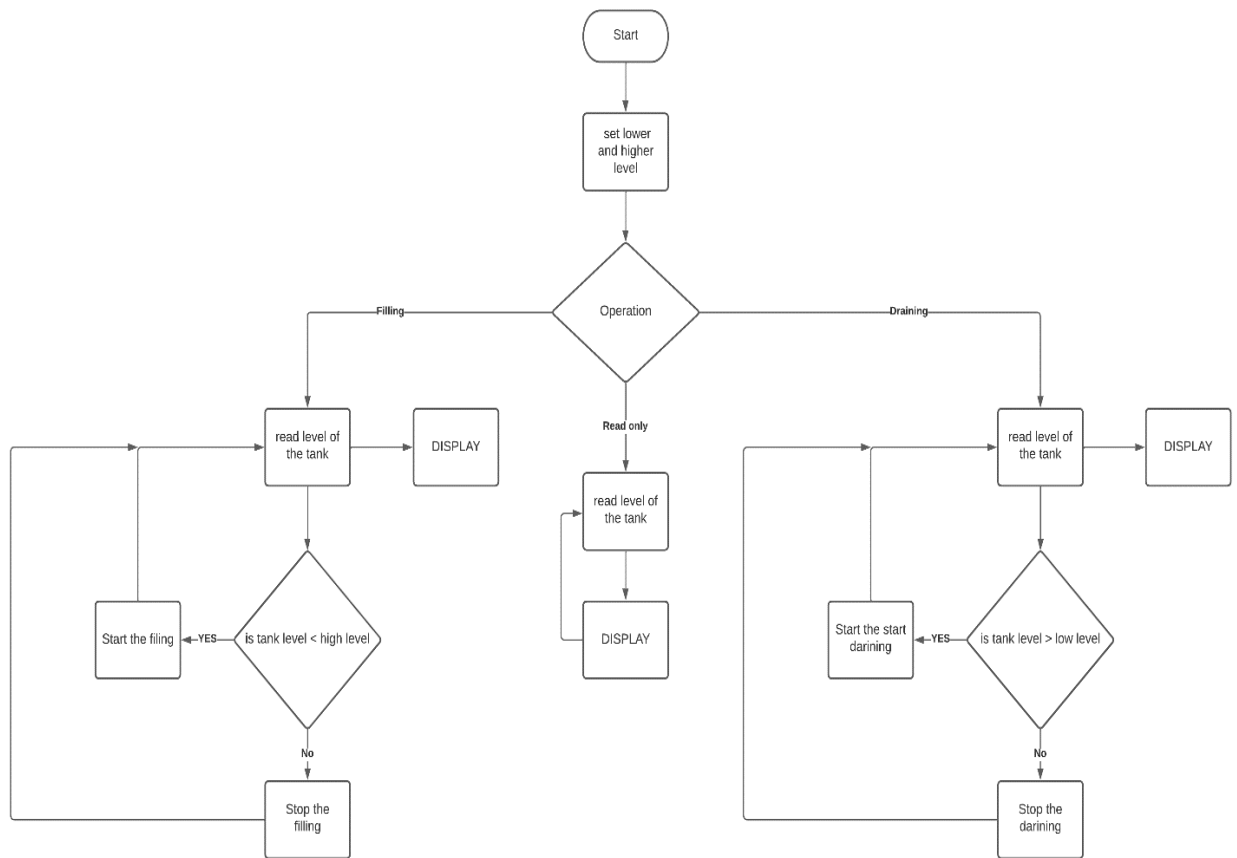
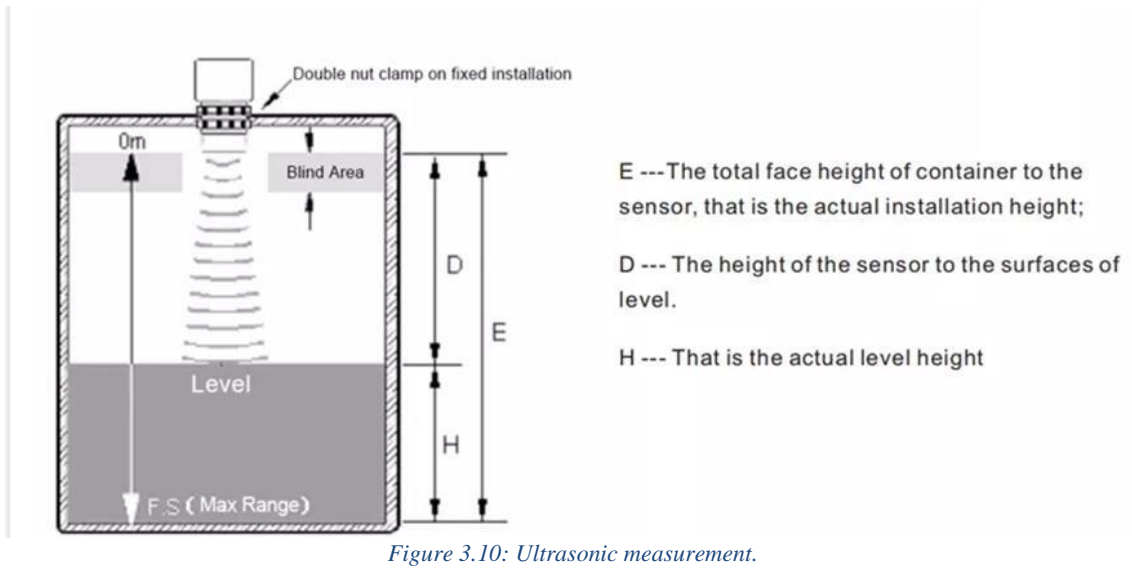


Figure 3.9: Controller flow chart.

3.5.4 Ultrasonic

The method of measuring we used is the ultrasonic sensor it will send sound waves to the surface of the liquid and calculate the travelling time for the echo waves to turn back and use it to calculate the distance, based on that we can get the level measurement. Figure 3.10 shows the ultrasonic sensor measurement.



3.5.5 Prototype Design

Prototype design consists of two tanks. Main tank is the one we used to control and measure the level of liquids inside it. The other one we used it to store and export the liquid from it. Basically, its closed loop for prototyping. Figure 3.11 show the prototype design.

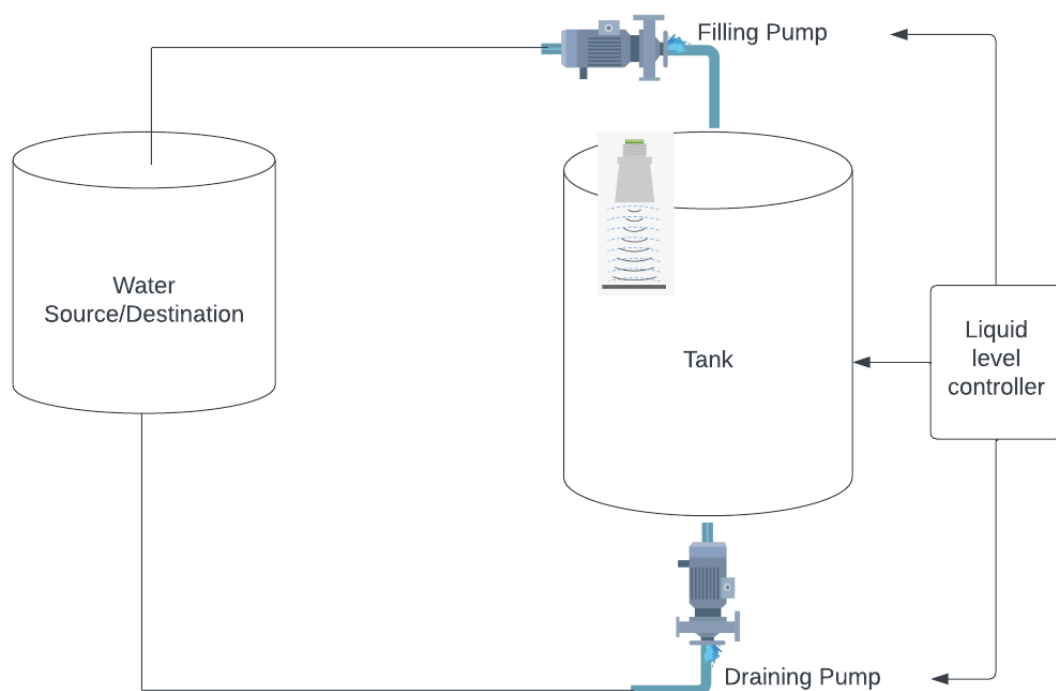


Figure 3.11: Prototype design

The main tank consists of four edges at the top, cylinder, slanting edge at the bottom and two pump used for the draining and filling process of the liquid between the two tanks.

Four edges are located at the top of the tank, so before the liquid goes down to the bottom it should go through them first one by one. The point of these edges is to filter the liquid from any impurities in three stages. Figure 3.12 shows the four edges in the main tank.

Slanting edge and the cylinder are used to remove the waves and keep the liquid level stable as much as possible. Slanting edge diffuse the liquid when it hits it, this would decrease the movement of the liquid, and give more accurate measurements for the ultra-sonic sensor. Figure 3.12 shows the slanting edge and the cylinder.



Figure 3.12: Edges and cylinder in prototype

3.6. Development tools

For designing the device, we used different tools for software and hardware building and testing such as:

EasyEDA: is web application EDA tool used by engineers to design circuit schematics as shown in Figure 3.9, also used to design PCB as shown in Figure 3.10 also it allows multiple sharing of design the same schematic or PCB board through different users.

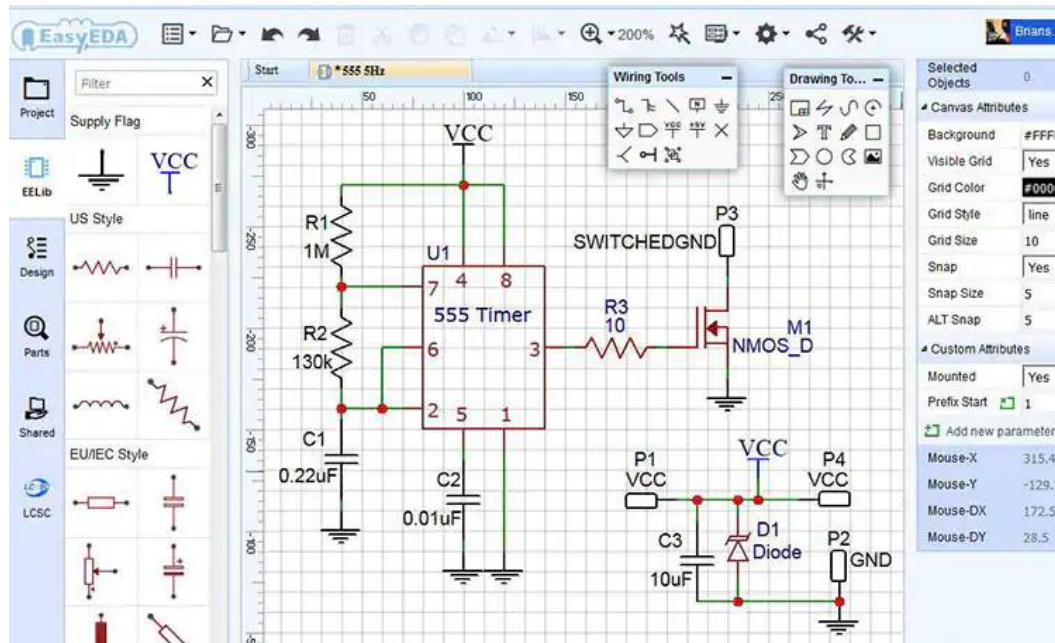


Figure 3.11: EasyEDA schematic design.

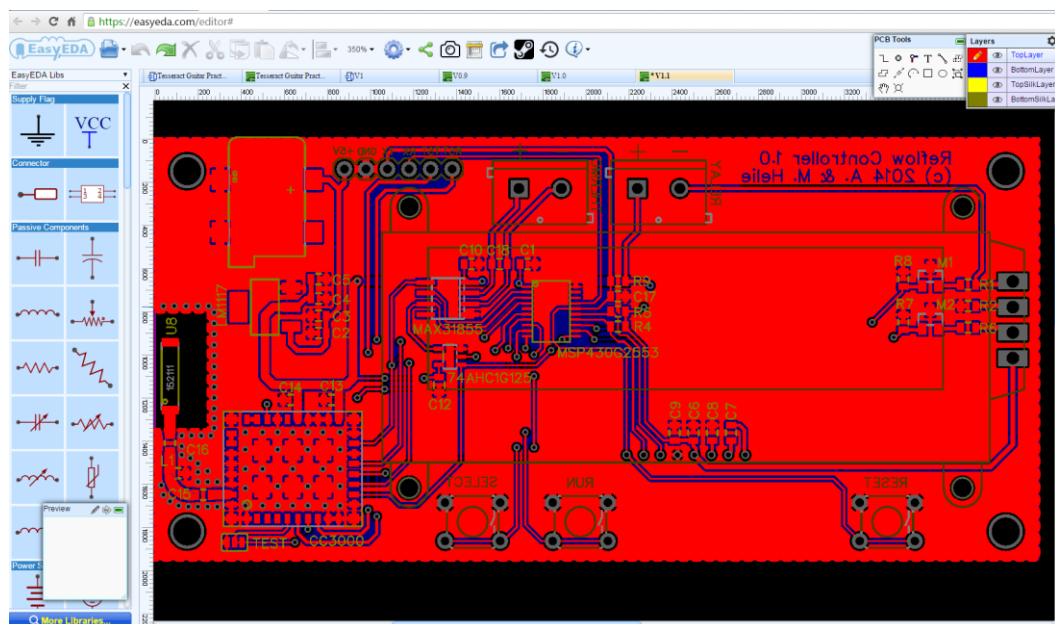


Figure 3.12: EasyEDA PCB design.

VS Code and Platform IO:

For testing the hardware and the controller we will use VS Code IDE as development environment, VS Code is a code editor used to build modern applications and it provides large functionality and extensions used to increase the productivity of building applications.

Platform IO is an extension can be installed on VS Code its professional extension for embedded systems engineers and software developers who wants to build and write code for embedded products the Figure 3.11 shows VS Code running Platform IO, also it has a lot of features such as:

- supports various Embedded Boards
- C/C++ Intelligent Code Completion
- Smart Code Linter for the super-fast coding
- Multi-projects workflow with Multiple Panes.
- Themes Support with dark and light colors
- Built-in Terminal with Platform IO CLI tool



Figure 3.11: VS Code IDE running Platform IO.

CHAPTER 4

IMPLEMENTATION

The device implementation started from the very low-level of picking the circuit components then design the schematic of the device after that building the design the PCB board and test it by uploading software test snippets of code and in this chapter, we will walk through these steps of implementation in details.

4.1 Circuit implementation

The first process of the implementation is designing the circuit of the device, this is the basic stage of any device design workflow, the circuit design process will be done through two steps, component selection and schematic capture.

Component selection is very critical step in any design, there are a lot of things to be considered as components cost, specifications, size, availability ...etc.

For our circuit component selection, the Figure 4.1 bellow presents all the components we used and selected from LCSC website which is a website offers a wide selection of electronics components:

ID	Name	Designator	Footprint	Quantity	Manufacturer Part	Price
2	NCP1117LPST33T3G	LDO	SOT-223_L6.5-W3.5-P2.30-LS7.0-BR	1	NCP1117LPST33T3G	1.038
3	U254-051N-4BH806	USB1	MICRO-USB-SMD_U254-051N-4BH806	1	U254-051N-4BH806	0.065
4	GREEN	D16	LED-TH_LED-DP4-2LED	1	A694B/2SYG/5530-E2	0.163
5	OLED'	U12	OLED'	1		
6	100uF	C4	CAP-SMD_BD10.0-L10.3-W10.3-FD	1	VZH101M1K1010-TRO	0.201
7	PMM_VERTICAL MODULE v1.1	U20	RS485-232FOOTPRINT	1		
8	Ethernet_Module	U16	ETHERNET CONNECTOR	1		
9	100uH	L3	IND-SMD_L13.5-W12.6_125CDMCC-DS	1	TMPC1206HP-101MG-D	0.444
10	PMM ANALOG INPUT V/I	U3,U5	PMM ANALOG INPUT V/I V2	2		
11	PMM ANALOG OUTPUT	U21,U23	PMM ANALOG OUTPUT V1	2		
12	V5-30BQ015-M3/9AT	D1,D17	SMC_L6.9-W5.9-LS7.9-RD	2	V5-30BQ015-M3/9AT	0.395
13	100kΩ	R7	R1206	1	ESR18EZPF1003	0.018
14	1K	R8,R9,R3,R4	R0603	4	WR06X114JTL	0.003
15	PMM ATSAM/D/ESP32 v2	U7	PMM ATSAM/D/ESP32 2	1		
16	MB6S-TP	UR1	DIO-BG-SMD_MBS	1	MB6S-TP	0.07
17	ME2159AM6G	U24	SOT-23-6_L2.9-W1.6-P0.95-LS2.8-BR	1	ME2159AM6G	0.189
18	LM2576HVR-5.0	U15	TO-263-5_L10.6-W9.6-P1.70-LS15.9-TL	1	LM2576HVR-5.0	1.411
19	AT24C01C-SSHM-T	U8	SOIC-8_L4.9-W3.9-P1.27-LS6.0-BL	1	AT24C01C-SSHM-T	0.354
20	SH806-22.5 CON	RS45_CONNECTOR	SH806-22.5 CON 2	1		
21	5.76K	R26	R0603	1	0603WAF576T5E	0.002
22	110K	R25	R0603	1	WR06X114JTL	0.003
23	4.7K	R10	R0603	1	0603WAJ0472T5E	0.001
24	4.7uH	L4	L0806	1	LQM2MPN4R7NGOL	0.084
25	HR911105A_C12074	J1	RJ45-TH_HR911105A	1	HR911105A	1.282
26	JK-NSMD050-13.2V	F6,F2	F1206	2	JK-NSMD050-13.2V	0.029
27	MBRS2040LT3G	D4	SMB_L4.6-W3.6-LS5.3-R-RD	1	MBRS2040LT3G	0.168
28	SS36	D3	SMA_L4.3-W2.6-LS5.2-RD	1	SS36	0.048
29	SMTPA62	D2	SMB_L4.6-W3.6-LS5.3-BI	1	SMTPA62	0.516
30	22uF	C24,C21	C0603	2	ZRB18AR61A226ME01L	0.193
31	100nF	C23,C22,C16	C0603	3	FN18X104K250P5G	0.033
32	10uF	C15	C0603	1	TMK107BBJ106MA-T	0.032
33	10uF	C14	C0603	1	ZRB18AR61C106ME01L	0.211
34	1nF	C28,C27,C13	C1808	3	302R29W102KV4E	0.087
35	1000uF	C12	CAP-SMD_BD10.0-L10.3-W10.3-FD	1	EEF1AA102UAP	0.361
36	100nF	C11	C0603	1	TCC0603X7R104K500CT	0.002
37	100nF	C2,C3,C5,C6	C0603	4	CC0603KRX7R9BB104	0.002
38	MAX4544EUT+T	U2,U4,U6,U9	SOT-23-6_L2.9-W1.6-P0.95-LS2.8-TL	4	MAX4544EUT+T	1.734
39	DSHP04TS-S	SW1	SW-SMD_DSHP04TS-S	1	DSHP04TS-S	0.446
40		120 R1	R0603	1	0603WAF1200T5E	0.001
41	681Ω	R2,R5	R0805	2	RCT05681RFLF	0.004
42	10K	R6	R0603	1	0603WAF1002T5E	0.001

Figure 4.1: Circuit components.

For the schematic capture, electronic design automation (EDA) provides the ability of designing electronic schematics of the electronic device circuits and we will use EasyEDA online tool to design our circuits and PCB as well. And now we will show the circuits we implemented for the device.

Power connector circuit shown in Figure 4.2 Contain 4 pins:

- 2 pins connect to the earth
- 1 pin connect to vin-
- 1 pin connect to vin+

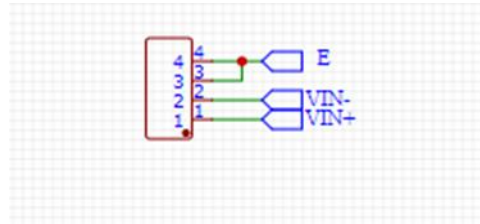


Figure 4.2: Power connector circuit.

Protection circuit shown in Figure 4.3 bellow is connected to the power connector it is used to:

- reverse polarity protection
- overvoltage protection

For protection against high voltage, we used Transient Voltage Suppressors (TVS) diode, with a rated voltage of 62, The diode becomes short-circuited and discharges into GND when exposed to high voltage.

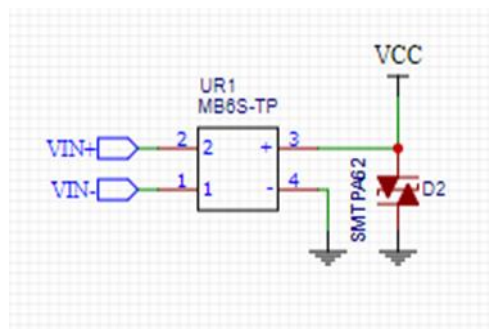


Figure 4.3: Protection circuit.

DC-DC Converter - 5V circuit shown in Figure 4.4 bellow we implement it by Plugging 40V-60V input from the protection circuit into LM2576HVR-5.0 Step-down Switching Regulator, The LM2576HVR-5.0 provides a fixed output voltage of 5 volts. DC-DC converter use fuse to protect against overcurrent.

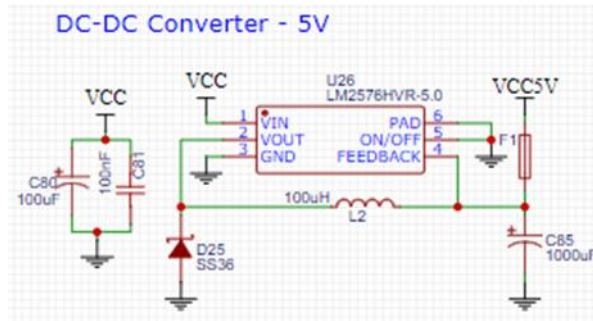


Figure 4.4: DC-DC Converter circuit.

Human protection circuit shown in Figure 4.5 below responsible of Electrical discharge from the human body.

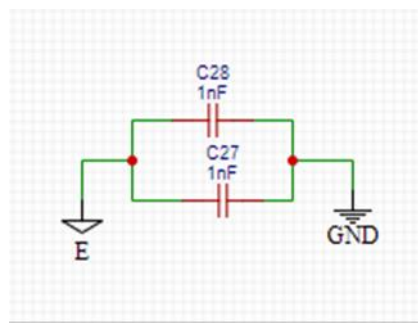


Figure 4.5: Human protection circuit.

Linear Voltage Regulators (LDO) circuit shown in Figure 4.6 below which stands for Low Dropout, can operate at a low potential difference between input and output. It is sometimes referred to as a low-loss or saturation type linear regulator the figure below presents its circuit.

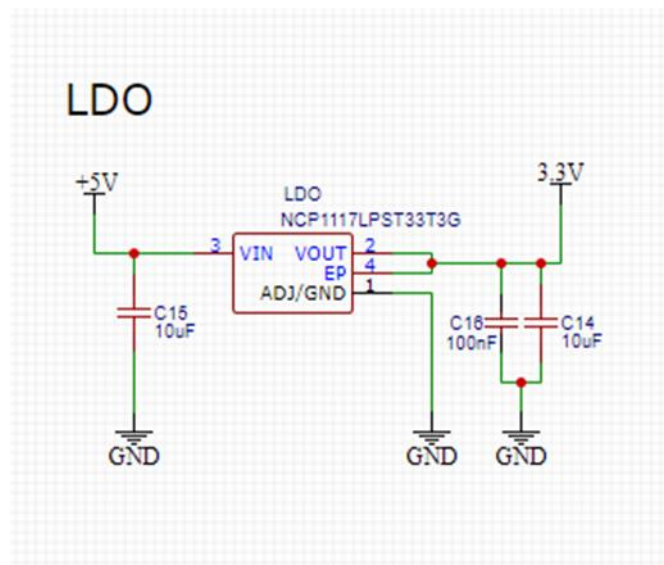


Figure 4.6: LDO Circuit.

EPROM (Microchip Tech AT24C01C-SSHM-T)

SAMD CPU doesn't contain any internal memory. In this case, we use EEPROM. When the power is turned off or removed, EEPROMs are used to store calibration or similar data that may be needed in the future, Figure 4.7 illustrates how the EEPROM connects to the CPU and the lists the pinout configuration.

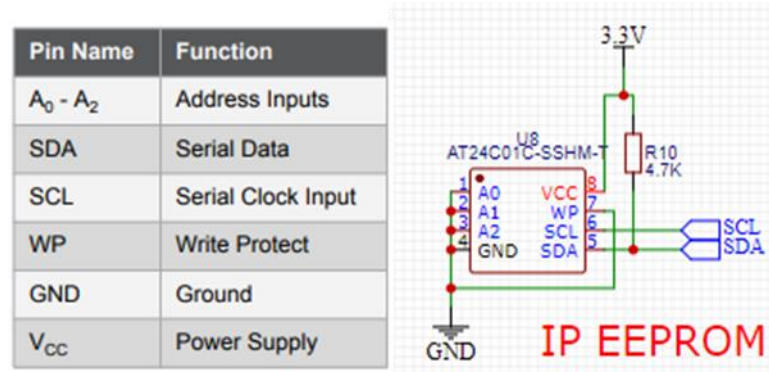


Figure 4.7: EEPROM Circuit and pinout configuration.

Booster circuit shown in Figure 4.8 bellow, booster is a low voltage high current series-wound DC generator inserted into a DC circuit to add or inject a certain voltage proportional to the circuit current.

The main purpose of providing a booster in the power system is to compensate for the voltage drop of the feeder, boosters add voltage to the feeder and compensate for the voltage drop. Hence it increases the efficiency and reliability of the system, that is why it is one of the important devices of power systems.

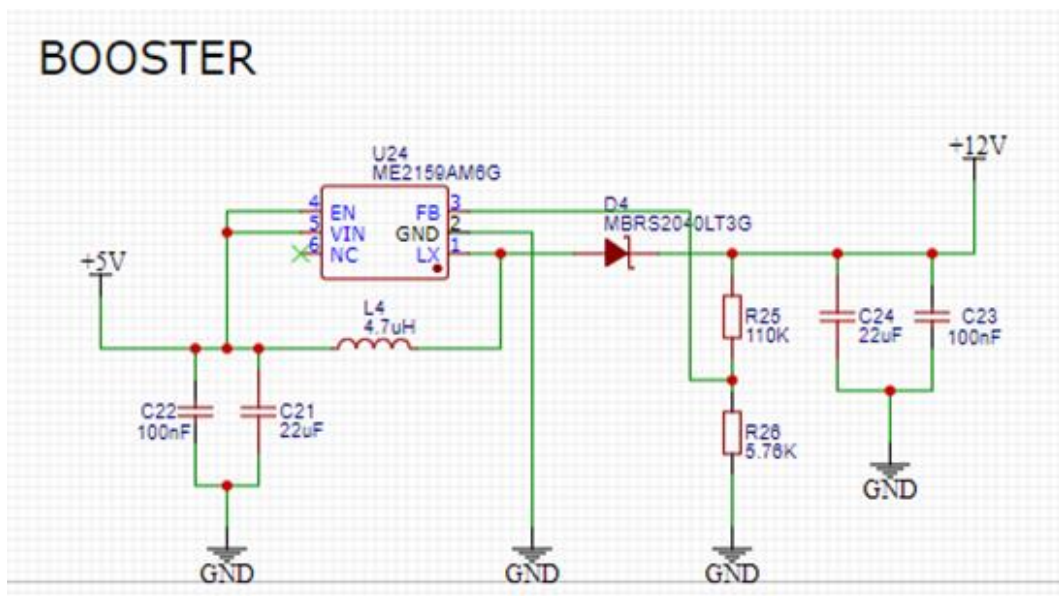


Figure 4.8: Booster circuit.

ESP32 CPU circuit show in the Figure 4.9 bellow, ESP32 can function as a complete stand-alone system or as a slave device to a host microcontroller. The ESP32 is also capable of providing wireless and Bluetooth capabilities through its SDIO/SPI and I2C/UART interfaces.

ESP32 Functions:

- Its wireless antenna and dual core allow embedded devices to communicate with other devices and transmit data.
- Data Processing: This includes anything from simple inputs from analog and digital sensors to far more complicated calculations.
- Web Server: Provides access to HTML or development language pages.
- P2P Networking: This allows direct connection between multiple ESPs and other devices via IoT P2P connectivity.
- Has different Wi-Fi modes it can act as AP or station mode or both together

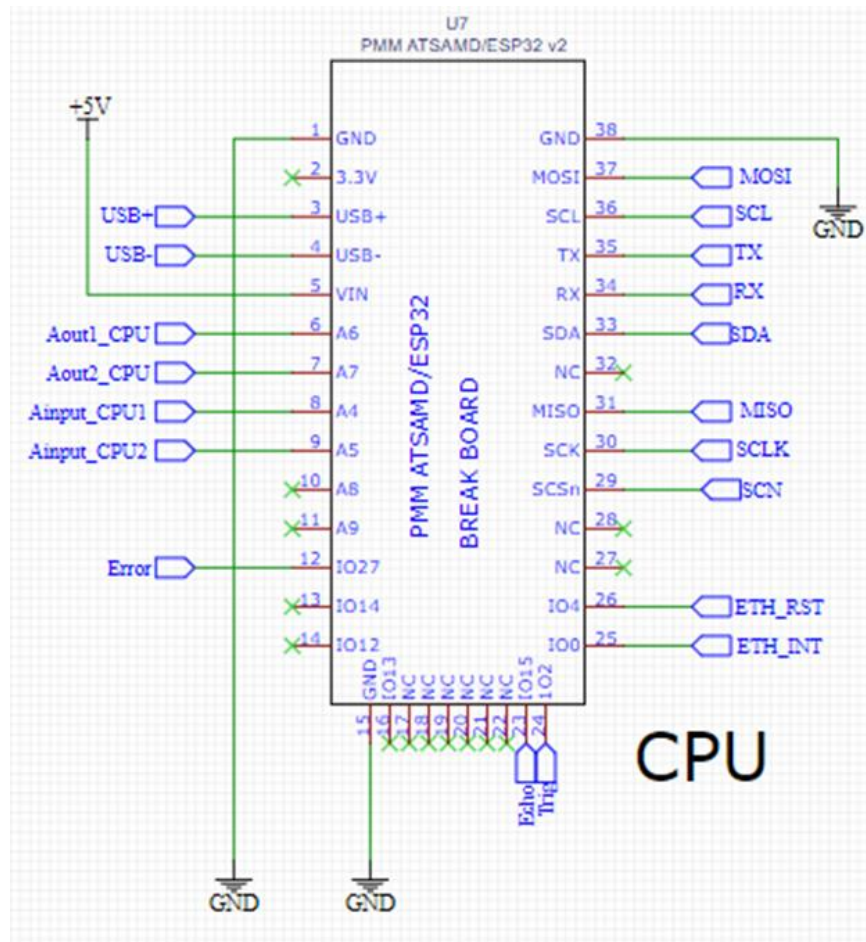


Figure 4.9: ESP32 Circuit.

RS-485 is an industrial specification that defines the electrical interface and physical layer for point-to-point communication of electrical devices. The RS-485 standard allows for long cabling distances in electrically noisy environments and can support multiple devices on the same bus, Figure 4.10 bellow shows RS-485 circuit.

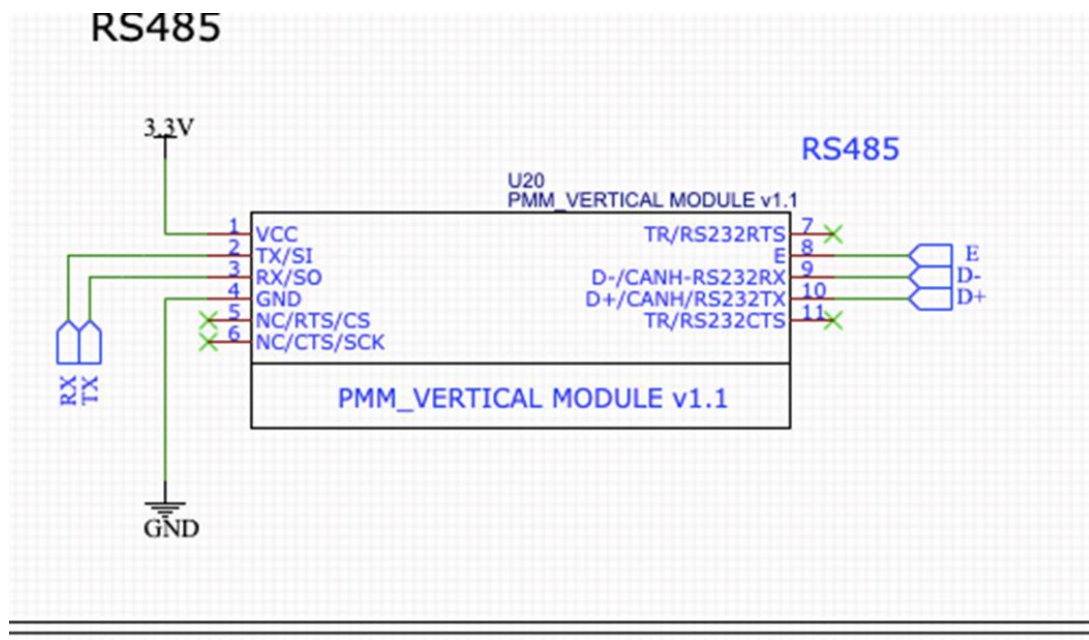


Figure 4.10: RS485 Circuit.

The WIZ850io is a compact-sized network module (Ethernet module) for Arduino and compatible boards based on the WIZNET W5500 chip, faster than the previous W5100. It provides an easy way to connect your module to the WIZNET Ethernet-based local network and internet with a TCP/IP interface. Compatible with libraries with the same Ethernet API. With this module with an RJ-45 port, you can build a web server. It can be used as a server or as a client, Figure 4.11 below shows the Ethernet module and W5500 circuit, and Figure 4.12 below shows the ethernet module connections with ESP32.

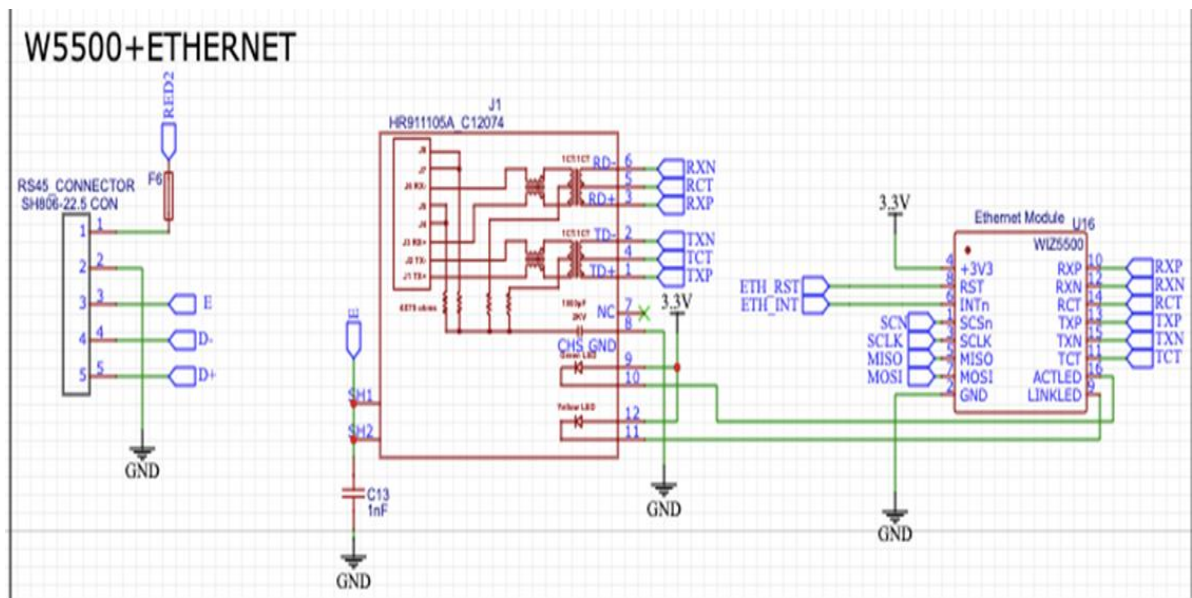


Figure 4.11: Ethernet module and W5500 Circuit.

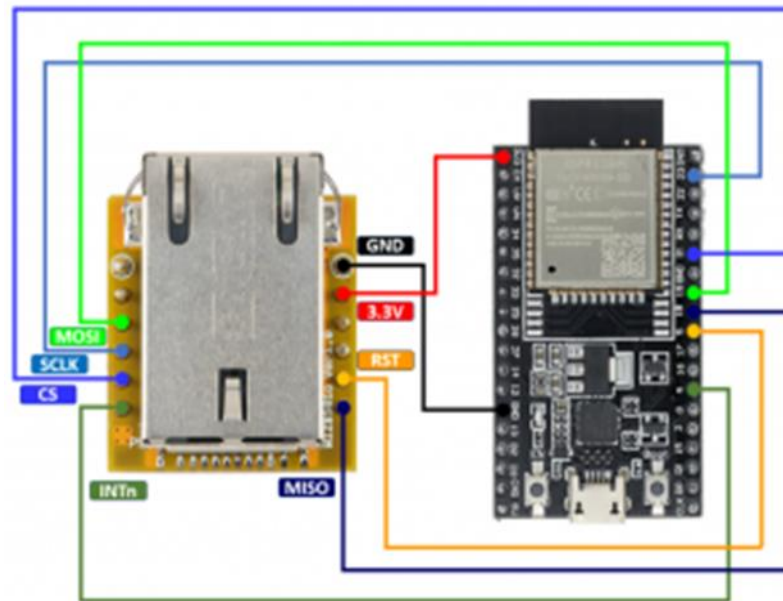


Figure 4.12: Ethernet module connections with ESP32.

4.2 PCB implementation

The second process of implementation after completing the circuit design is to design the PCB board on which the circuit components will be fitted in. This process requires an extreme accuracy with placing the components to get a proper functional product eventually.

In the PCB design we will take care of components placements and preserve the signal integrity by obeying some rules while designing also we should take care of defining trace routes and trace widths.

4.2.1 Signal integrity

Signal integrity is the ability to reduce the distortion of signals and preserve its quality in transmission lines, in this section we will talk about factors that prevents and achieves signal integrity and how we consider them in our implementation.

Differential pair PCB routing is a design approach used to build a balanced transmission system capable of carrying differential (equal and opposite) signals across a printed circuit board, we have applied the differential pair strategy by following the below guideline:

- The same trace lengths should be used for all the pairs and the same trace widths should be used throughout the trace
- Maintain symmetrical routing between the two Traces

Crosstalk phenomenon shown in Figure 4.13 bellow is unintentional electromagnetic coupling between traces on a printed circuit board, crosstalk is the noise that is coupled between adjacent conductors through the near electric field (capacitive coupling) and magnetic field (inductive coupling), this coupling can cause the signal pulses of one trace to overpower the signal pulses of the other trace even though they are not physically touching, we avoid crosstalk by obeying PCB design practices such as :

- separating traces as far apart as possible
- reduce the length that two lines are allowed to run in parallel
- keep traces in adjacent layers perpendicular to each other.

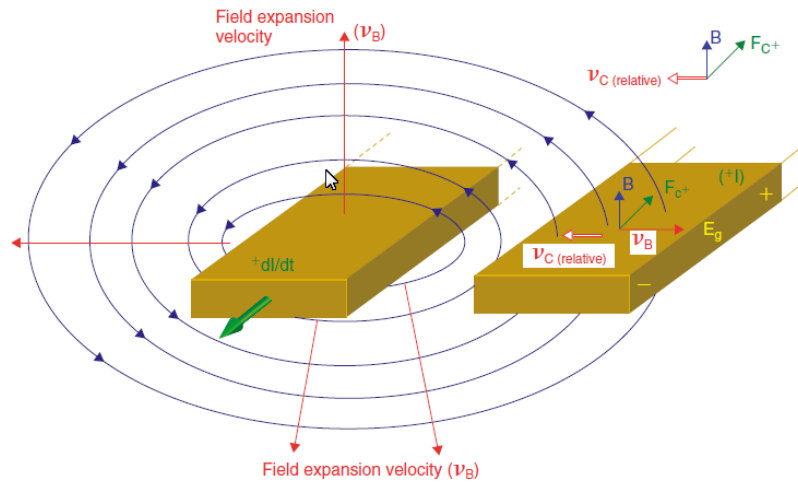


Figure 4.13: Crosstalk phenomenon.

Return path is a closed loop system is required for any electrical circuit. When using direct current, the return current travels backward. The return current flows along the lowest impedance route, which is generally the reference plane closest to the signal, the Figure 4.14 bellow shows return path.

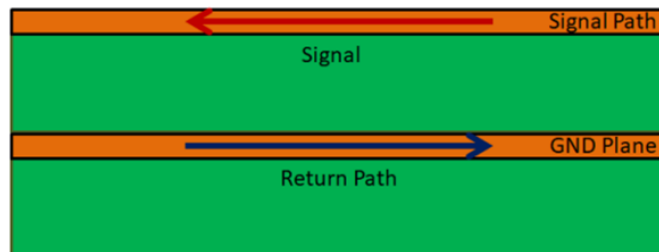


Figure 4.14: Return path.

4.2.2 Placing components

It is important to put components in the right places and be accurate as possible so that the board can be routed smoothly and reduces the cost of production and reduces project production lead time while maximizing the operational efficiency of the PCB assembly. So, to ensure proper placement of components we will obey these steps:

- ensure all components are given enough space, by doing the following:
 - Placing EEPROM under ESP32 for spacing efficiency
 - We choose to place analog in-out circuits on modules to maximize space efficiency and make sure that all components get adequate room
 - Place SW DSHP04TS-S on the PCB bottom for space and routing efficiency
- Place large processor and ICs first by doing the following:
 - Placing CPU first because of routing reasons to place the essential components first
 - We've placed the connectors, analog input modules, analog output modules, Ethernet and RS485 in the first position.
- Grouping Components and keeping things separate, it is best to group certain components together when designing the layout of the PCB Such as:
 - grouping booster LDO dc-dc converter reverse polarity circuit and power connector help us to minimize the number of crossing nets
 - Place the RS485 circuit near the RS485, place the analog in-out module near the analog connector
 - It's okay to rotate components to try and eliminate net intersections.

Note that we will keep the power away from the analog input-output connector as much as possible to avoid signal distortion, and Figure 4.15 below shows the final placement.

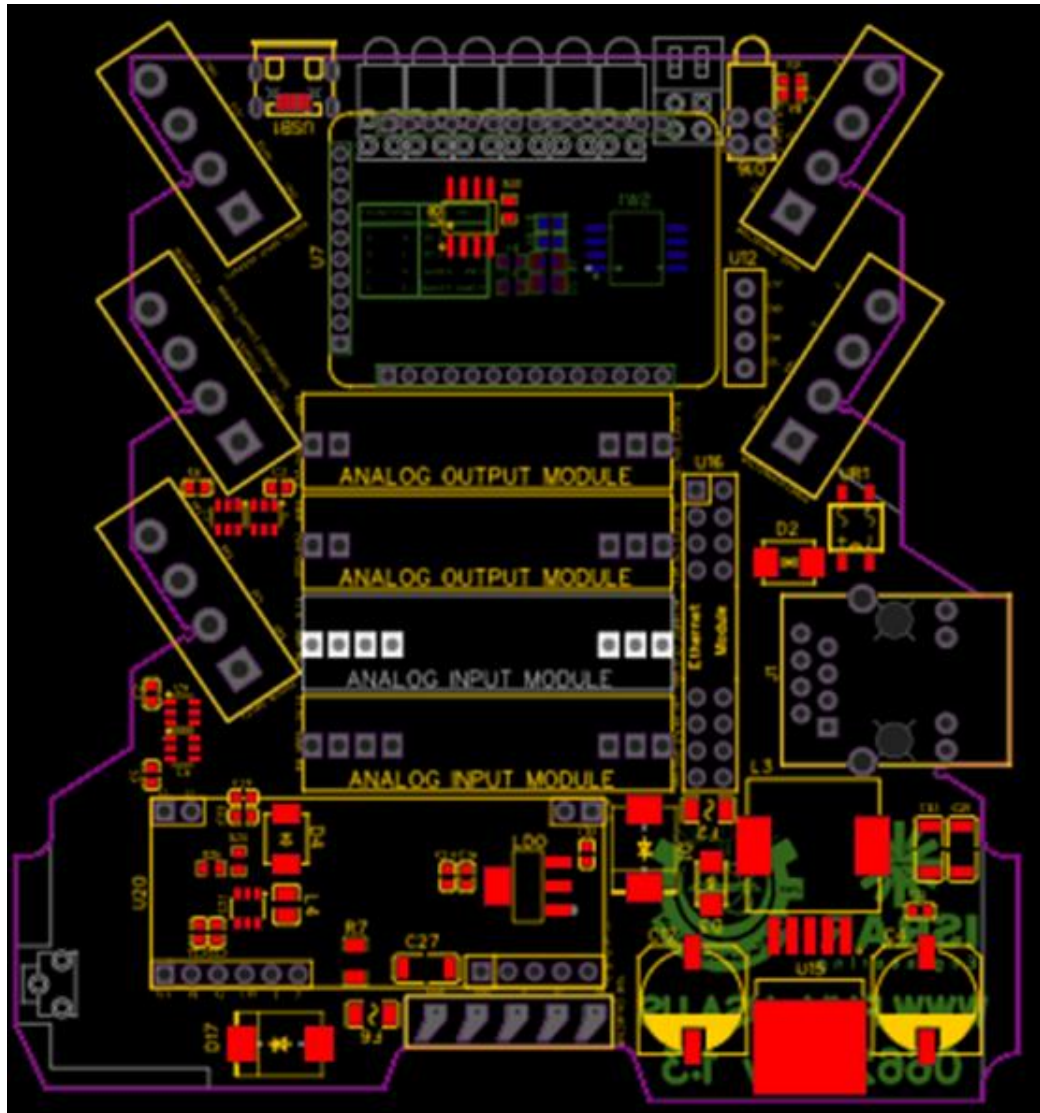


Figure 4.15: PCB Component placement.

4.2.3 Routing

After component placement we will add all the wires needed to properly connect the placed components by obeying all design rules for the IC, Figure 4.16 shows all the nets in the schematic.

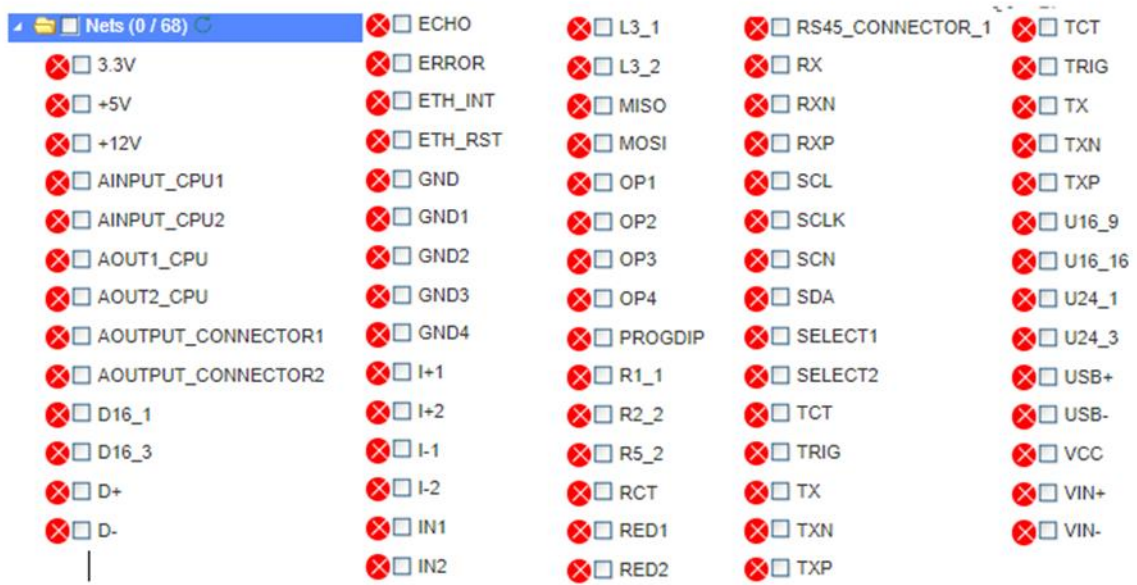


Figure 4.16: PCB Nets.

Since we are going to use copper areas for connecting the grounds for each circuit, it is a good idea to hide the ground nets GND, GND1, GND2, GND3, GND4 and E to avoid crossover nets Figure x shows the schematic after hiding the ground nets

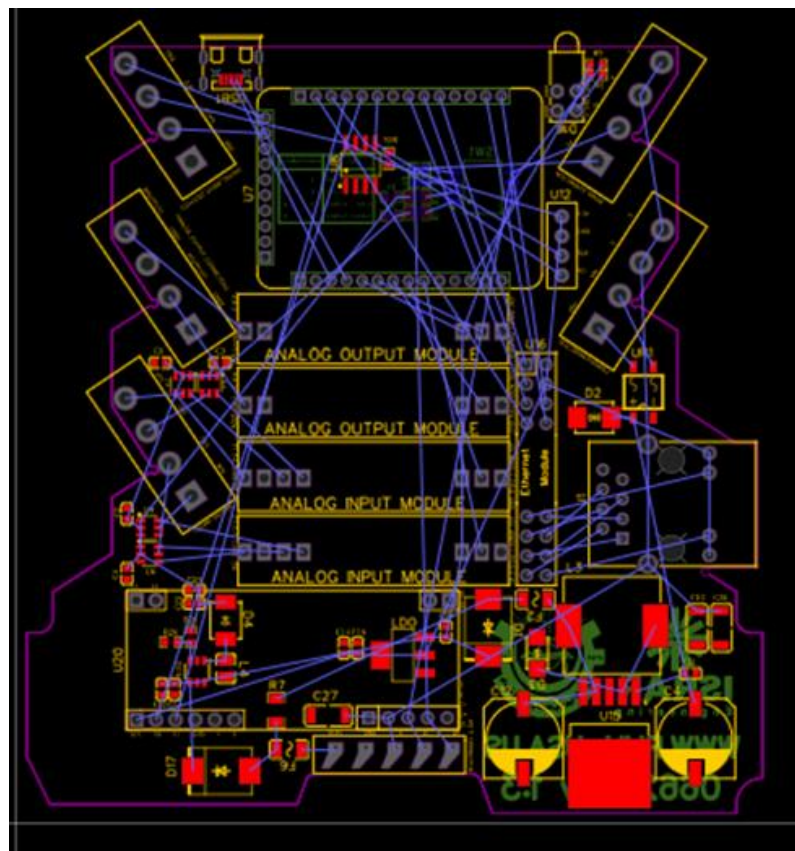


Figure 4.17: PCB Nets without GND.

Finally, we will route all the nets in the schematic by following these guidelines, the Figure 4.18 bellow shows the final PCB after routing:

- The needed trace width for various nets is determined by three criteria Manufacturability, Impedance, and Current. The width will be calculated to be greater or equal the current * 0.3.
- Trace width choices will be 1.2 mm for power traces (LDO, booster, DC-DC converter, and reverse polarity), 0.5mm for connector and VCC pins on each component and 0.3mm for rest traces
- Using VIA to connect difficult nets, the VIA is a hole on a PCB made up of two pads on separate levels of the board that are electrically linked via a hole through the board.
- Never use 90-degree trace angles. Always use angles greater than 90 degrees
- Increase the spacing between traces
- From tools use teardrop, The addition of teardrops during fabrication protects against drill breakout by increasing the amount of copper for the connection.
- From place add copper area.

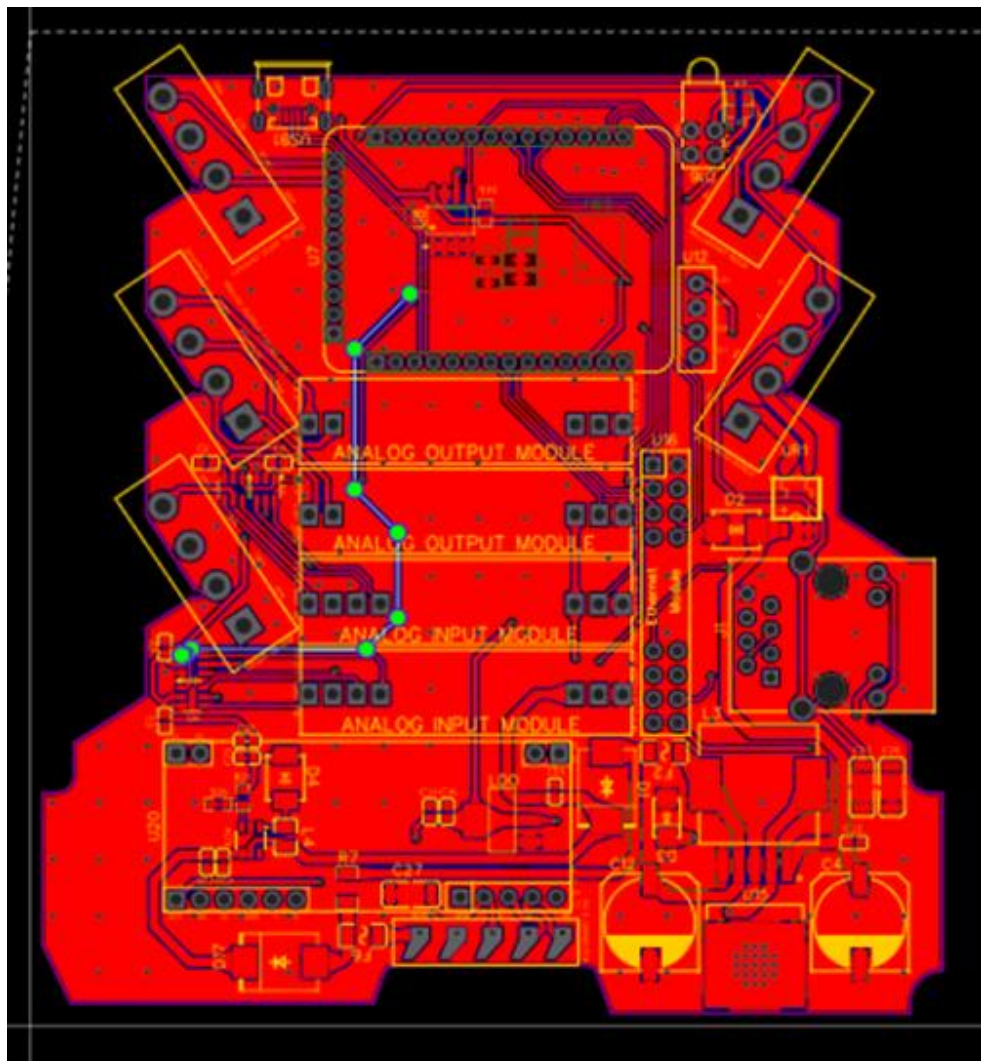


Figure 4.18: PCB after routing.

4.3 Prototype implementation

We implement our prototype by connecting the controller shown in the Figure 4.19 with two pumps each pump will move the liquid to a different tank as shown in Figure 4.20, and the pseudocode of the prototype shown in Figure 4.19.

```
// Sets the ultrasonic trigPin HIGH (ACTIVE) for 10 microseconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the ultrasonic echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance = duration * 0.148 / 2; // Speed of sound wave divided by 2 (go and back)
// Displays the distance on the Serial Monitor

if(distance > threshold){
    digitalWrite(drainPumbPin,HIGH);
    digitalWrite(fillPumbPin,LOW);
}
else {
    digitalWrite(drainPumbPin,LOW);
    digitalWrite(fillPumbPin,HIGH);
}
```

Figure 4.19: Prototype pseudocode



Figure 5.20: Liquid level controller



Figure 4.21: Prototype tanks and pumps.

CHAPTER 5

TESTING AND EVALUATION

After completing both the hardware and software designs and development, testing is an essential part in the production of electronic devices. To make sure that the design is working as expected and fix any possible errors soon before production if certain tests fail.

5.1 Power supply testing

The power supply circuitry is the part which provides the other electronics parts of the device with their necessary power to operate.

The power supply block is composed of the following circuitry parts:

1. Reverse Polarity and ESD Protection
2. DC-DC Converter 5V
3. 3V3 LDO
4. BOOST CONVERTER

Each part of the mentioned circuits shall be tested according to operation standards specified in a validation plan as to be explained in the following tabulated procedures.

Reverse Polarity and ESD Protection voltage test represented in the Table 5.1 bellow.

Voltage Input Range	Test Type: Load Expected Output drop .3v	Real Output	Remarks
10	9.7v	9.85	
20	19.7v	19.5	
30	29.7v	29.44	
40	39.7v	39.45	
50	49.7v	49.43	
52	51.7v	51.41	
54	53.7v	53.38	
56	55.7v	55.43	
58	57.7v	57.52	Burn→ short circuit tvs
60	59.7v	59.5	After 2sec burned

Table 5.1: Reverse Polarity and ESD Protection voltage test.

DC-DC Converter 5V

Obviously, it's necessary to measure the voltage and current as input (V_{in} and I_{in} respectively) and the voltage and current across the output (V_{out} and I_{out} respectively), these couples of values are the input power (P_{in}) and the output power (P_{out}), the expected output of voltage is 5v.

The load regulation test is the capability to sustain specified voltage output when load current (I_{load}) varies at constant input voltage, Table 5.2 shows output of voltage test.

Voltage Input Range	Test Type: Load Expected Output	Real Output	Remarks
10	5v	5.07	
20	5v	5.07	
30	5v	5.07	
40	5v	5.08	
50	5v	5.08	
52	5v	5.08	
54	5v	5.08	
56	5v	5.08	
58	5v	5.08	
60	5v	5.08	

Table 5.2: DC-DC Converter voltage test.

LDO (NCP1117LPST33T3G)

- Output Configuration Positive
- Output Type Fixed
- Number of Regulators 1
- Voltage - Input (Max) 18V
- Voltage - Output (Min/Fixed) 3.3V
- Voltage - Output (Max)
- Voltage Dropout (Max) 1.4V @ 1A
- Current – Output 1A
- Current - Quiescent (I_q) 700 μ A
- PSRR 60dB (120Hz)

Voltage Input Range	Test Type: Load Expected Output	Real Output	Remarks
10	3.3v	3.348v	
20	3.3v	3.348v	
30	3.3v	3.348v	
40	3.3v	3.35v	
50	3.3v	3.35v	
52	3.3v	3.35v	
54	3.3v	3.35v	
56	3.3v	3.35v	
58	3.3v	3.35v	
60	3.3v	3.35v	

Table 5.3: LDO voltage test.

Boost converter (ME2159AM6G)

Input voltage range VIN 2.5v min - 5.5 V max

Adjustable Output up to 12V

Current – Output 1 A

Frequency - Switching 1.2 MHz

Voltage Input Range	Test Type: Load Expected Output	Real Output	Remarks
10	12v	11.98	
20	12v	11.98	
30	12v	11.99	
40	12v	11.99	
50	12v	12v	
52	12v	12v	
54	12v	12v	
56	12v	12v	
58	12v	12v	
60	12v	12v	

Table 5.4: Boost converter voltage test

5.2 ESP32 testing

We will test ESP32 microcontroller used in the board by making sure that all pins of the controller are working properly using blink circuit and code test.

Items required to perform the test are:

- ESP32
- Connecting cable
- LED
- A resistor (220 ohm)
- Breadboard

Figure 5.1 bellow shows the ESP32 blink test, to perform the test we will do these steps:

- Attach the LED to the ESP32.
- To one end of the resistor, connect the analog pin.
- Attach the other end of the resistor to the positive end of the led (the longer leg anode).
- Attach the negative terminal of the LED to the GND pin of the ESP32
- Using the USB cable, connect the ESP32 to your laptop's port.
- Upload the following code:

```
void setup() {  
  pinMode(11, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(11, HIGH); // turn the LED on  
  delay(500);             // wait for 500 milliseconds  
  digitalWrite(11, LOW);  // turn the LED off  
  delay(500);             // wait for 500 milliseconds  
}
```

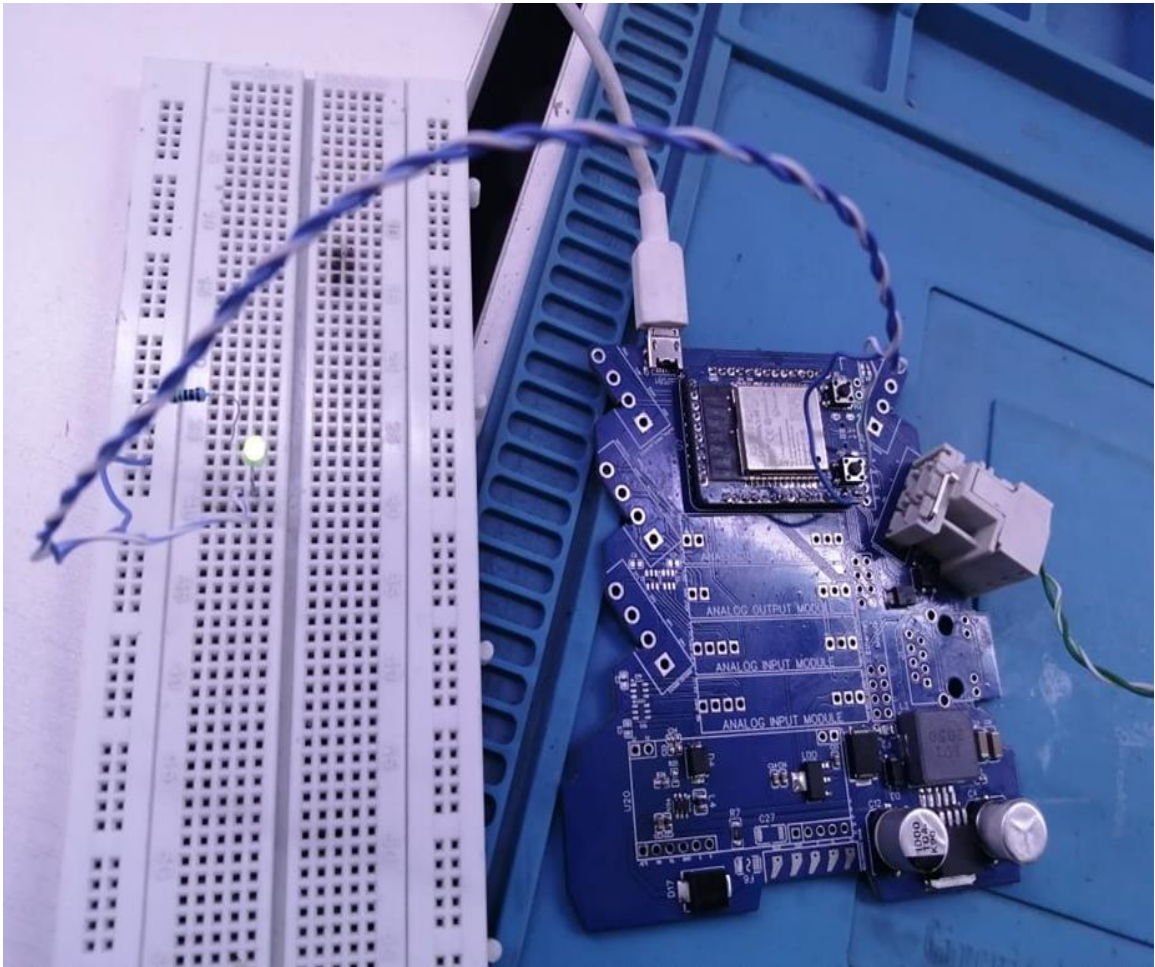


Figure 5.1: ESP32 blink test.

5.3 LCD testing

Figure 5.2 show the LCD test, the items required to perform the test are:

- ESP32
- LCD
- Connecting cable
- Breadboard

Circuit Connections:

- VCC pin connected to VCC(5V) on ESP32
- GND pin connect this pin to GND(0V) on ESP32
- 12C clock signal pin (SCL) to SCL pin on ESP32
- 12C data signal pin (SDA) to SDA pin on ESP32

Follow the steps to upload the code used for testing the LCD on Arduino:

File -> Examples -> upload ssd1306_128x64_i2c code

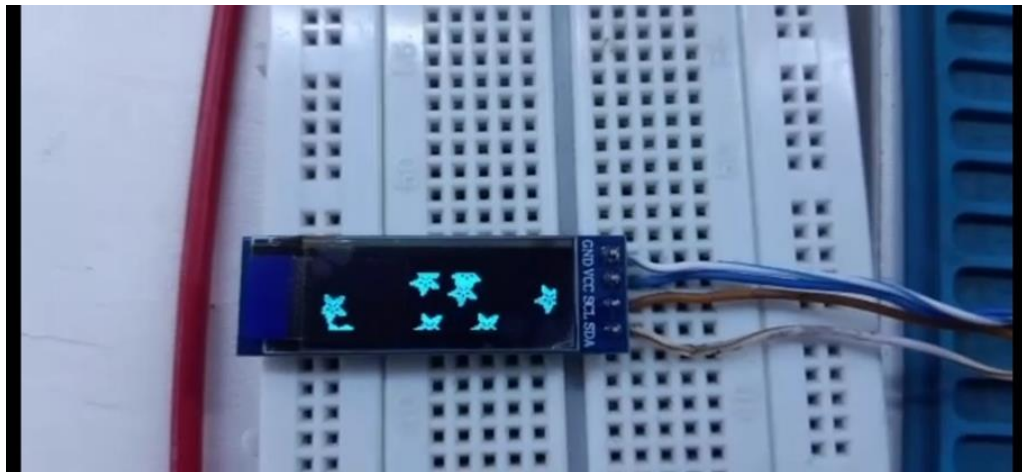


Figure 5.2: LCD test.

5.4 Analog output testing

Figure 5.3 bellow shows the pinout of the analog output module, and the items required to test it are:

- Multimeter
- Power supply
- Analog output module
- Bread board
- Wires

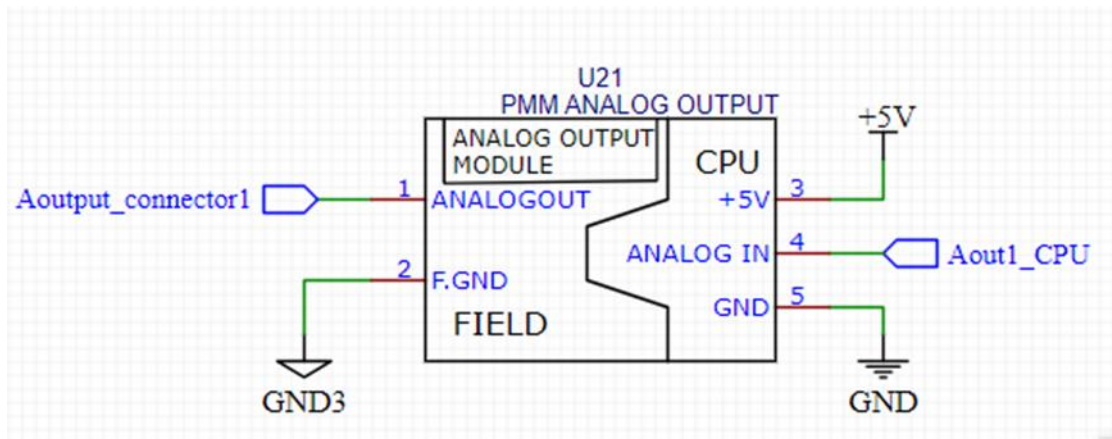


Figure 5.3: Analog output module.

The test is going to be as follow:

- Plug Aoutput_connector1 (pin 1) into multimeter Vo pin
- F.GND (pin2) should be connected to the multimeter com pin
- Connect pin (4) with 5v power source
- Connect Aout1_CPU (pin 4) with power source the voltage can be in range of 0-5v
- Plug pin 5 of the module into the board ground

Figure 5.4 bellow shows the connection of the analog output module test.

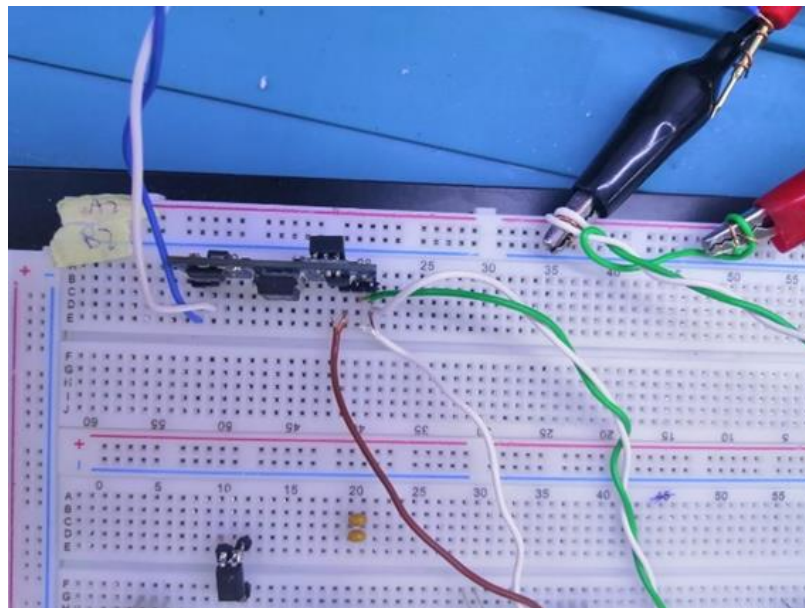


Figure 5.4: Analog output module connection.

Figure 5.5 bellow shows the power supply while it's connected to pin 5 and 3.



Figure 5.5: Power supply connected to pin 5 and 3.

Rather than connect the analog output module to the CPU we have connected it to the power supply to act as the CPU which provides the analog output module with 5v.

Figure 5.6 bellow shows the power supply while it's connected to pins 4 and 3.



Figure 5.6: Power supply connected to pin 4 and 3.

The output voltage on the multimeter is expected to be in the range of 0-10v. The output voltage is 1.9 multiplied factor by the input voltage.

And that's what we have achieved on the multimeter shown in Figure 5.7 bellow.



Figure 5.7: Analog output module multimeter test.

5.5 Analog input testing

Figure 5.8 below shows the pinout of the analog input module, and the items required to test it are:

- Multimeter
- Power supply
- Analog input module
- Bread board
- wires

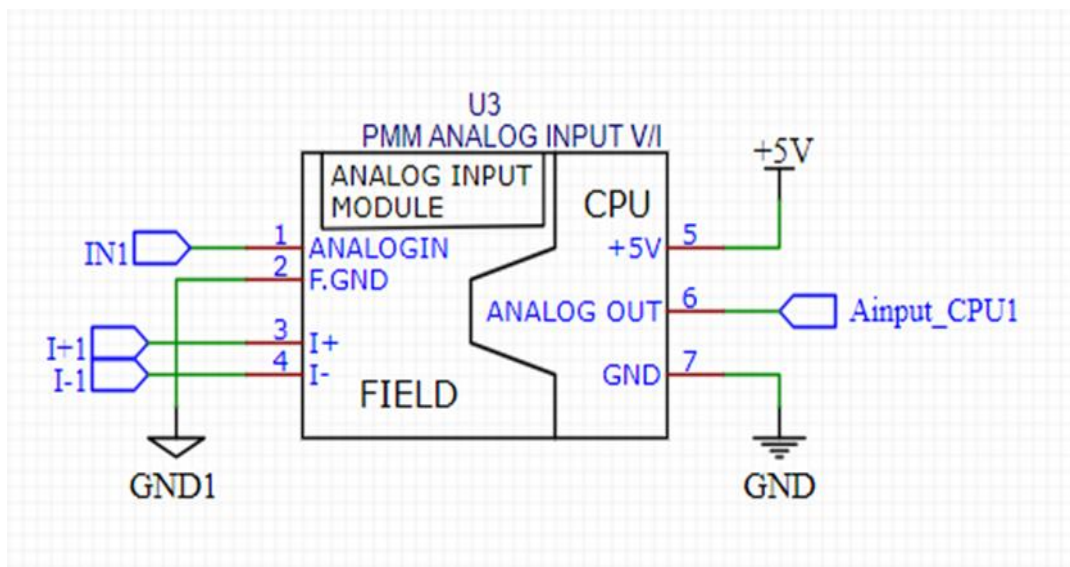


Figure 5.8: Analog input module.

The test is going to be as follows:

- Connect (pin 1) with a power source the voltage can be in range of 0-10v
- F.GND (pin2) should be connected to the board ground com
- Connect (pin 5) with a 5v power source
- Plug pin 7 of the module into the board ground and with multimeter com pin
- Plug Ainput_CPU1(pin 6) into multimeter Vo pin
- Plug Ainput_CPU1(pin 7) into multimeter com pin

Figure 5.9 bellow shows the connection of the analog output module test.

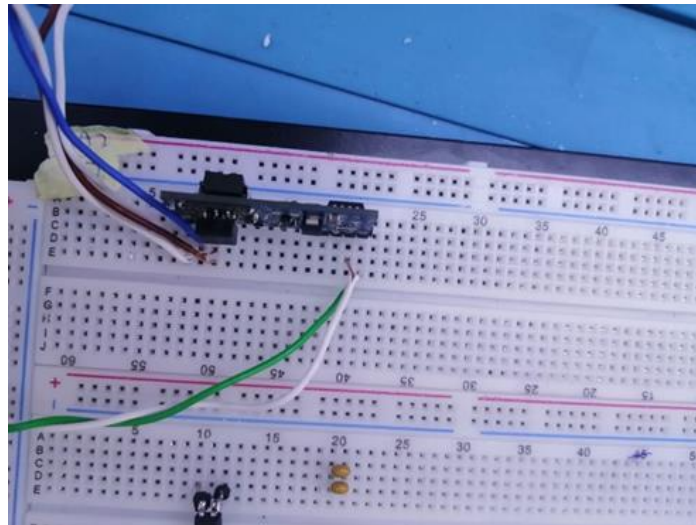


Figure 5.9: Analog input module connection.

Figure 5.10 shows the power supply while it's connected to pin 5 and 7



Figure 5.10: Power supply connected to pin 5 and 7

Figure 5.11 shows the power supply came from field connected to pin 1 and 2



Figure 5.11: Power supply connected to pin 1 and 2.

The output voltage on the multimeter is expected to be in range of 0 - 3.3 v, and that's what we have achieved on the multimeter shown in the Figure 5.12 bellow. Voltage on CPU equal voltage on filed divided by factor 6.5.



Figure 5.12: Analog input module multimeter test.

CHAPTER 6

CONCLUSION

Tank level measurement problem is popular in industry and our project intends to give a solution in an automated way, to save money, effort, and increase the production in factories.

The device implementation starts from the very low level of creating circuits and design PCB, to choose the appropriate controller and sensor and test it.

The implementation considered remote configuration and controlling for the controller using Wi-Fi for the device, also it can be done in the embedded code.

A major achievement of the device is to keep equipment from being destroyed to the greatest extent possible. Also, we were able to design a device well suited for industrial settings, such as pharmaceutical and food factories as well as water stations.

Future enhancement:

- 1- control, monitor and regulate liquid level through the cloud.
2. Voice alarm system used when liquid level reaches a certain level.

REFERENCES

[1] SST Sensing Ltd , “Main Types of Level Sensing Methods - How do they differ” , Jul -2017.

<https://sstensing.com/7-main-types-of-level-sensors/>

[2] Espressif Inc ESP32, “ESP32-WROOM-32 (ESP-WROOM-32) Datasheet”, 2018

https://www.mouser.com/datasheet/2/891/esp-wroom-32_datasheet_en-1223836.pdf

[3] Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, “Low-Voltage, Single-Supply Dual SPST/SPDT Analog Switches Datasheet”

https://datasheet.lcsc.com/lcsc/1809131549_Maxim-Integrated-MAX4544EUT-T_C239212.pdf

[4] Texas Instruments Incorporated, “LM2576xx Series SIMPLE SWITCHER® 3-A Step-Down Voltage Regulator”, SNVS107F datasheet, JUNE-1999 – REVISED MAY-2021.

<https://www.ti.com/lit/ds/symlink/lm2576.pdf?ts=1642399649827>

[5] Modbus Organization, “MODBUS APPLICATION PROTOCOL SPECIFICATION”, April-2012.

https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

[6] CUI DEVICES, “RS-485 Serial Interface Explained”,2022.

<https://www.cuidevices.com/blog/rs-485-serial-interface-explained>

[7] Random Nerd Tutorials, “ESP32 Useful Wi-Fi Library Functions (Arduino IDE)“, Feb-2021.

<https://randomnerdtutorials.com/esp32-useful-wi-fi-functions-arduino/>

[8] PlatormiO Labs, Visited in Jun-2022.

<https://platformio.org/>

[9] PCB software – PCB Design tool.

<https://easyeda.com/>

[10] IPC, “Generic Standard on Printed Board Design”, May-2003.

<https://www.ipc.org/TOC/IPC-2221A.pdf>

