Flying Shear Control System for Industrial Internet-of-Things

By

Hussain Mohammad AlOmari

Shahed Ali Maabreh

Batool Walid Aldalki

Raghad Emad Alsoboh

Spring 2022

DEDICATIONS

We dedicate this project first to Allah the Almighty, the Most Gracious, and the Most Merciful for His blessing given to us during our study and in completing this project.

May Allah's blessing goes to his final Prophet Muhammad (peace be up on him), his family and his companions.

To the memory of my father who never saw this adventure, you will always be loved and remembered
To my careful mother, my inspiration and strength source
To my supportive and beloved brothers and sisters
To my beloved nephews
-Eng. Hussain AlOmari

To my dear father and mother, and my beloved brothers and sisters
All the words of thanks cannot describe my feelings, I will not forget what you did for me and thank you for everything you have given me.
-Eng. Batool Walid

To my dear father and mother,
And my beloved brothers and sisters
-Eng. Shahed Maabreh

To my dear father
To my careful mother, my sweet family
-Eng. Raghad Emad

ACKNOWLEDGEMENTS

First and foremost, Thanks to Allah for his blessings throughout our project work. As well as we would like to express our appreciation toward ISRAR ENGINEERING LLC. for the efforts they did to provide us with all useful information, equipment, and training. and for making the path clear for the us to implement all the education periods in real-time project design and analysis.

We would like to express our deepest gratitude to the CEO of ISRAR ENGINEERING LLC. Dr. Mahmoud Zarini, and the technical staff; Eng. Ahmad Zarara and Eng. Ahmad AlJoghaimi for building our technical skills and their support of the make of the prototype, and their patience and guidance along the journey.

At last, we would like to thank all the people who helped, supported and encouraged us to successfully finish the graduation project, whether they were in the university or in the industry.

ABSTRACT

Driven by recent development of mission-critical Industrial Internet of Things (IIoT) applications [1] and significant advances in wireless communications, networking, computing, sensing and control, wireless networked control systems (WNCSs) have recently emerged as a promising technology to enable reliable and remote control of industrial control systems [2].

The Flying Shear (also known as Flying Knife) is a common industrial application for cyclic cutting an input stream of material at the exact speed of the material and at a predefined length. The production process is continuous and not interrupted thus maximizing machine productivity. In order to achieve high accuracy and quality, it is essential to control and monitor the flying shear. The significant economic effects of its operation are due to high reliability, high quality, accuracy of cutting (precise cut-to-length), energy savings, and scrap reduction [3-4].

The system is oriented toward helping the manufacturers of any type application where a carriage must be synchronized at line speed, and these can also be accommodated using the control techniques of a Flying Shear Control System.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

TABLE **PAGE**

LIST OF FIGURES

FIGURE **PAGE**

CHAPTER 1

INTRODUCTION

There are countless industries that make use of Industrial Internet of Things (IIoT). One example is the automotive industry, which uses IIoT devices in the manufacturing process. The automotive industry extensively uses industrial robots, and IIoT can help proactively maintain these systems and spot potential problems before they can disrupt production. Our proposed system insures the process of cutting any manufactured material using Flying Shear Control System, integrated with Internet of Things.

1.1. An Overview of Flying Shear Control System.

1.1.1.. What is Flying Shear in motion control?

The term "motion control" spans a wide range of applications that involve controlling the movement of a linear or rotary axis to achieve a goal such as moving to a specified position or precisely following a defined path. And although there are numerous variations, at their core, many of these applications can be grouped into one of a few general categories: pick-and-place, positioning (linear or rotary), path following (such as dispensing), winding, and flying shear.

1.1.2.. The Functionality of Flying Shear

Flying Shears are used periodically for cutting a continuously moving material, where the endless material to be cut to length cannot be stopped during the cutting process and the cut must be very accurate "on the fly". The mechanical construction provides a saw or shear system mounted on a carriage that follows the material with synchronous speed, while cutting is in progress, and then returns to a home position to wait for the next cut.

1.1.3.. Cutting mechanisms of Flying Shear

There are two possible cutting mechanisms depending on the application requirements:

1.1.3.1. Parallel Flying Shear

In Figure 1.1 an example of parallel flying shear in which the cutting tool moves parallel to the material flow. This is used when the entire cut is done at once, as with a blade or punching tool that spans the width of the material to be cut.

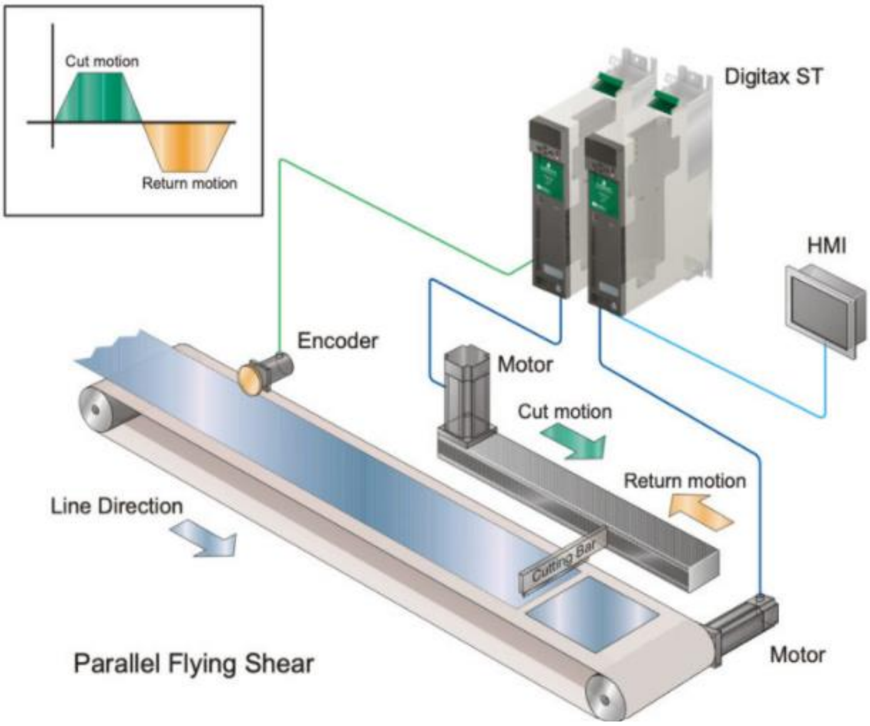

Figure 1.1 Parallel Flying Shear

1.1.3.2. Angular Flying Shear

In Figure 1.2 an example of angular flying shear in which the cutting tool (such as saws or plasma cutters) moves at an angle to the material flow and it is used when the cutters must move across the material in order to make a cut.

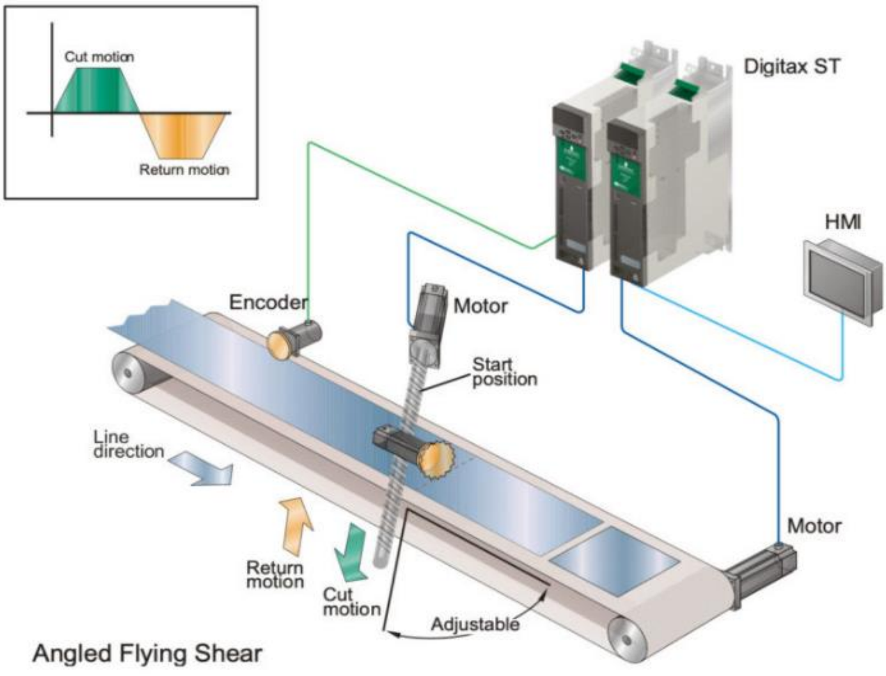

Figure 1.2 Angular Flying Shear

1.2. An Overview of Industrial Internet of Things - IIoT

1.2.1.. What is Industrial Internet of Things - IIoT?

The industrial internet of things (IIoT) is the use of smart sensors and actuators to enhance manufacturing and industrial processes. IIoT uses the power of smart machines and real-time analytics to take advantage of the data that "dumb machines" have produced in industrial settings for years. Connected sensors and actuators enable companies to pick up on inefficiencies and problems sooner and save time and money, while supporting business intelligence efforts. In an industrial setting, IIoT is key to processes such as predictive maintenance, enhanced field service, asset tracking.

1.2.2.. What are the benefits of Industrial Internet of Things - IIoT?

One of the top touted benefits of IIoT devices used in the manufacturing industry is that they enable predictive maintenance. Organizations can use real-time data generated from IIoT systems to predict when a machine will need to be serviced. That way, the necessary maintenance can be performed before a failure occurs. This can be especially beneficial on a production line, where the failure of a machine might result in a work stoppage and huge costs. By proactively addressing maintenance issues, an organization can achieve better operational efficiency.

CHAPTER 2

REVIEW OF RELATED LITERATURE

This chapter presents the related literature and studies after the thorough and in-depth search. This will also present the synthesis of the art, theoretical and conceptual framework to fully understand the research to be done and lastly the definition of terms for better comprehension of the study

Flying shear is the visual plant of producing steel, bar and the isometric material of wire rod, it is at metallurgy industry, particularly steel industry, machine industry and various excellent wire rod industry all have a wide range of applications, and are requisite key equipment on the production lines such as contemporary band steel, bar and wire rod. But still at the shearing system that uses the first generation or the second generation, these system's ubiquity speed are slow, efficient is low, precision is low, fault rate is high and the problem of dimension maintenance difficulty etc. on many production lines, had a strong impact on the benefit of enterprise. Most shearing system is badly in need of updating, but does not at present domestically also have producer to produce to possess high accuracy and high efficiency shearing system.

The requirement of producing in order to satisfy Modern High-Speed, improve lumber recovery and scale rate, must reduce the flying shear fault rate, reduce cost of equipment maintenance, improve simultaneously equipment precision and the control level of flying shearing machine, reduce velocity perturbation, realize accurate positioning, high accuracy to shear according to length enacted, be necessary the more advanced reliable flying shearing machine of development, with the flying shearing machine of replace imported costliness, not only reduce investment outlay and facilitate for domestic numerous potential users, and will produce huge economic and social benefit.

In the past few decades, a fourth industrial revolution has emerged, known as Industry 4.0. Industry 4.0 takes the emphasis on digital technology from recent decades to a whole new level with the help of interconnectivity through the Internet of Things (IoT), access to real-time data.

Industry 4.0 is a new phase in the Industrial Revolution that focuses heavily on interconnectivity, automation, machine learning, and real-time data. Industry 4.0, which encompasses IIoT and smart manufacturing, marries physical production and operations with smart digital technology, machine learning, and big data to create a more holistic and better connected ecosystem for companies that focus on manufacturing and supply chain management. While every company and organization operating today is different, they all face a common challenge the need for connectedness and access to real-time insights across processes, partners, products, and people.

The vision and the idea of our project has been created due to this upcoming industrial revolution, and we believe that IIoT will grow exponentially in this decade.

CHAPTER 3

ANALYSIS AND DESIGN

In Figure 3.1 an example of a flying shear implementation in a material production line. As indicated in Figure 3.1, the line is moving continuously in the given direction by some kind of motor depending on the belt length and weight, and the weight of the carried material. In order to properly apply continuous cutting of a material at a given length, one of the cutting mechanisms (Parallel, Angular) would apply, depending on the application requirements. The actuator needs to move from the home position when triggered, synchronize with the product, and perform the desired action depending on the required application.

Upon executing the action, the actuator needs to return quickly to the home position, and wait for the next trigger. The same operating principle is used for printing, packaging, depositors, punches, product inspection, filling and sealing bottles, topping the cupcakes [13], or any other process, where synchronization at line speed is required, and these can also be accommodated using flying shear application.

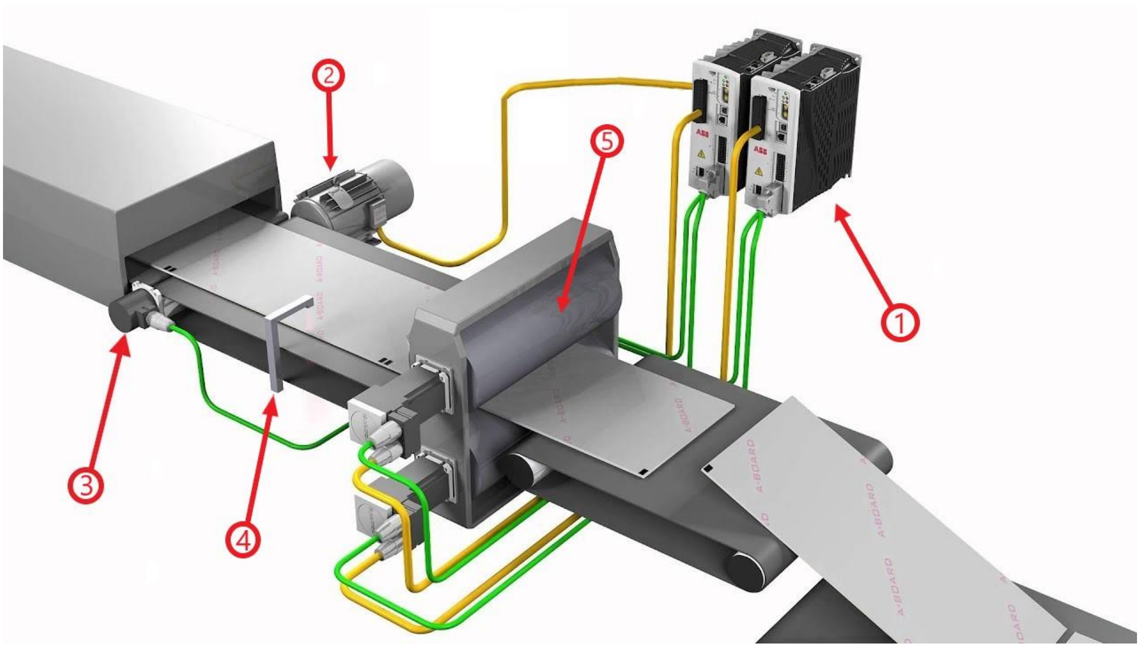

Figure 3.1 Example of a Parallel Flying Shear Application. *Image credit: ABB* [14].
1-Controllers; 2- Motor; 3- Master Encoder; 4- Trigger; 5- Actuator.

3.1. Design Requirements

The system is oriented toward helping the manufacturers of any type application where a carriage must be synchronized at line speed, and these can also be accommodated using the control techniques of a Flying Shear Control System.

The system has been designed for the special requirements of flying shears under consideration of maximum efficiency and accuracy at minimum stress for all mechanical parts.

The workflow shown in Figure 3.2 illustrates the processes which the controller goes through from the basic design to releasing the final product. The processes do not have to be followed consequently as in the proposed product, it was preferred to work in parallel.

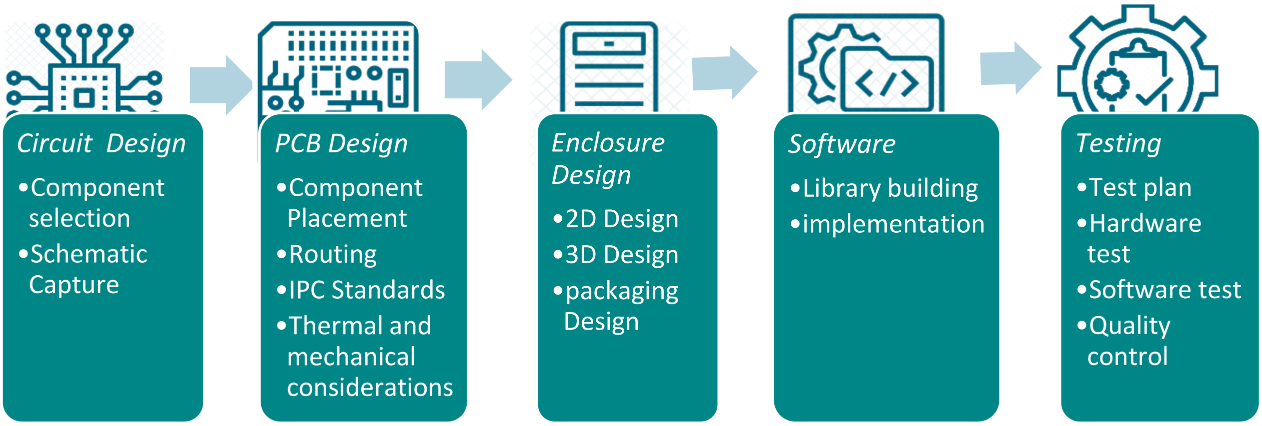

Figure 3.2. The production process of Flying Shear Controller

3.1.1 Circuit Design

The first process is initiating the circuit design. It is the basic stage in the entire workflow; on which the product's function is being visualized. Circuit designing process consists of two main activities: components selection, schematic capture.

The Table 3.1 explains each activity in the process of designing the circuit.

Table 3.1. Circuit Design Main Activities.

Component Selection	Component selection is a critical activity in designing any electronic product; there are many considerations to be thoroughly considered such as: components cost, specifications, size, suppliers and the required manufacturing techniques.
Schematic Capture	Electronic Design Automation (EDA) at which the electronic diagram or electronic schematic of the designed electronic circuit is created. This activity is done interactively with the help of a schematic capture tool also known as a schematic editor called EasyEDA [5].

3.1.2 Printed Circuit Board (PCB) Design

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

The Table 3.2 explains each activity in the process of designing the PCB.

Table 3.2. Printed Circuit Board (PCB) Main Activities.

Component Placement	Proper placement of components reduces the cost of production and reduces project production lead time while maximizing the operational efficiency of the PCB assembly.
Routing	After component placement, the routing step adds wires needed to properly connect the placed components while obeying all design rules for the IC.
IPC Standard	IPC standards are the electronics-industry-adopted standards for design, PCB manufacturing, and electronic assembly. There's an IPC standard associated with every step of PCB design, production, and assembly.
Thermal and Mechanical Considerations	Heat dissipation is very crucial for electrical products as the overheating may damage the entire product. In addition to considerations regarding the mechanical design and placing the board inside the enclosures.

3.1.3 Enclosure Design.

This process designs the housing that includes the internal components of the product. The enclosure must be tough and able to withstand any harsh environmental conditions in addition to be easily fitted on the targeted applications.

The Table 3.3 explains each activity in the process of designing the enclosure.

Table 3.3. Enclosure Design Main Activities.

2D Design	2D Design is the basic input for manufacturing the enclosure as it is required to set the dimensions and the overall look.
3D Design	This stage visualizes the product as 3D and displays the interfaces. If the design meets the expectations, then it is approved.
Packaging Design	The packaging must be designed to let the enclosure perfectly fitted in, with all the needed accessories. In addition, the final enclosure must abide by IP65 protection degree [12].

3.1.4. Software

After completing the hardware design, the software is developed and tested frequently to enable the product to function properly. The software is built to meet the product functions as well as any customizable functions the client requested based on the field that the product will be used in.

The Table 3.4 explains each activity in the process of designing the software.

Table 3.4. Software Design Main Activities.

Library Building	This step contains building a library that contains the basic firmware for the product to implement the basic functions that are expected from. Then some libraries are going to be custom made for specific functions.
Implementation	This step contains implementing the built library on the product to make sure it works properly. At this stage any further developments or modifications are done to the primary firmware; in order to develop the product.

3.1.5. Testing

After completing both the hardware and software designs, testing is an essential part in the production of electronic devices. The testing plan have been established to test both the product's hardware and software, to eliminate any possible defects and ensure that the final product version is meeting or exceeding the client's requirements.

The Table 3.5 explains each activity in the process of testing the product.

Table 3.5. Testing Main Activities.

Testing Plan	Establishment of the testing plan to be comprehensive in both hardware and software testing. The plan includes the testing procedures, checklists and corrective actions.
Hardware Testing	The hardware is tested thoroughly which includes all the mechanical and electrical aspects such as: the product's enclosure, heat dissipation, components selection, placement and packaging.
Software Testing	The software is tested thoroughly as well including that the basic web interface is functioning properly, the software functions are giving the true values and all the connections and networks are successfully connected with no disturbances.

3.2. Engineering Standards

The Engineering Standards that our system followed and abided by are:

- **MODBUS TCP/IP Standard [7].**
  MODBUS TCP/IP is a variant of the MODBUS family of simple, vendor-neutral communication protocols intended for supervision and control of automation equipment. Specifically, it covers the use of MODBUS messaging in an 'Intranet' or 'Internet' environment using the TCP/IP protocols.

- **RS-485 Standard [8].**
  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.

- **IEEE 802.11 b/g/n/e/i (Wi-Fi Standard) [9].**
  IEEE 802.11 is part of the IEEE 802 set of local area network (LAN) technical standards, and specifies the set of media access control (MAC) and physical layer (PHY) protocols for implementing wireless local area network (WLAN) computer communication.

- **IEEE 802.3 (Ethernet Standard) [10].**
  Ethernet is a family of wired computer networking technologies commonly used in local area networks (LAN), metropolitan area networks (MAN) and wide area networks (WAN).

- **USB Standard (USB2.0 Type micro-B) [11].**
  Micro USB is a miniaturized version of the Universal Serial Bus (USB) interface developed for connecting compact and mobile devices such as smartphones. in our proposed project, it is used to connect the controller to a PC to install the code into it, and for testing purposes.

- **IPC-2221A (PCB Standard) [6].**
  The IPC-2221 is the generic standard that covers almost every aspect of PCB design. The standard details how electrical considerations such as PDN bus layouts, conductor clearance, and impedance control should be implemented on a PCB.

- **IP65 protection degree [12].**
  An IP65 Rating means the product has the highest level of dust protection, and is able to withstand low-pressure water jets from all directions.

- **Hypertext Transfer Protocol (HTTP) RFC 2616** [17].
  HTTP is a protocol for fetching resources such as HTML documents.

- **DNS Protocol RFC 1035** [18].
  The Domain Name System (DNS) is the phone book of the Internet, Web browsers interact through IP addresses, DNS translates domain names to IP addresses

3.3. Realistic Constrains

The realistic constraints that will impact our design:

- Manufacturability and Sustainability: The controller dimensions should not exceed 150×150×40 mm and should abide with IP65 protection degree [12].
- Economic:
  - The design cost should not exceed $300 for the hardware components that are used to build the controller. The cost of firmware depends on the required application to match the needs of the corresponding manufacturer (i.e.; the customer).
  - The number of inputs and outputs may vary depending on the required application to match the needs of the corresponding manufacturer (i.e.; the customer.

3.4. Alternative/Different Designs Approaches/Choices

Typical applications include various types of cut-to-length machines, printing, packaging, depositors, punches, product inspection, or any other process where synchronization at line speed is required.

Linear flying shears are used in a variety of applications ranging from cutting material on the fly, filling bottles as they are fed on a conveyor, to forming soft material in a mold while being transferred along the process. Linear flying shear applications can be used to solve both random in-feed and constant feed applications. This application controls the linear axis of the saw to ensure accurate cutting as well as a digital output for providing management of the cutting mechanism.

In Figure 3.3. an example of a flying shear implementation in a cupcake production line is presented. A more detailed description of the specific application is given at the ABB products website [14]. As indicated in the figure, the line is moving continuously in the given direction. In order to properly apply topping on the cupcakes, the tool (applier) needs to move from the home position when triggered, synchronize with the product, and perform the desired action (in this case, a squirt). Upon executing the action, the tool needs to return quickly to home position, and wait for the next trigger. The same operating principle is used for continuous cutting of a material at a given length, filling and sealing bottles, etc.

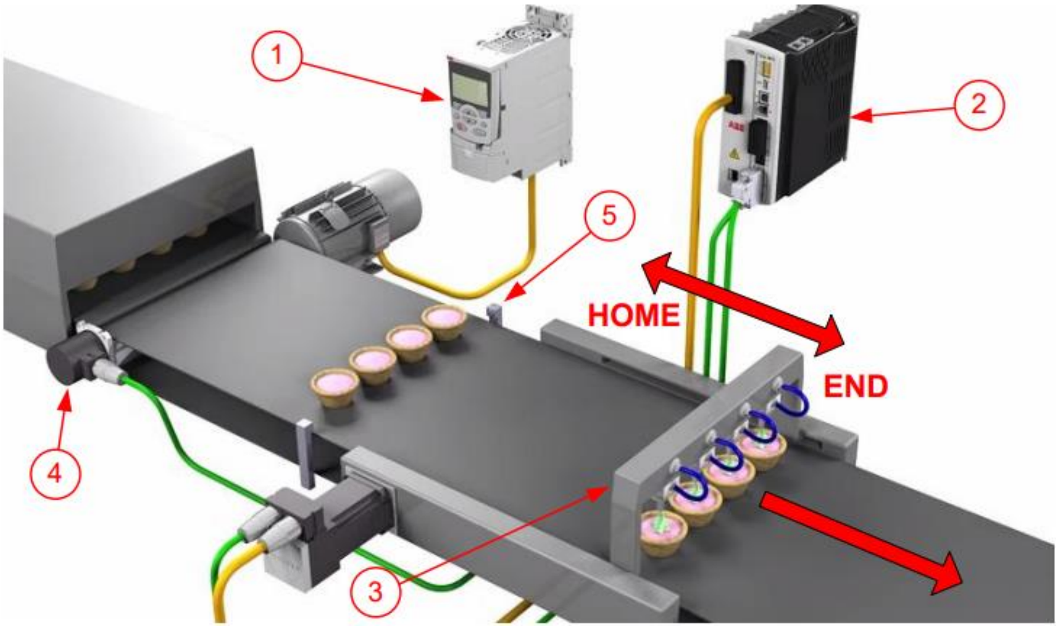

Figure 3.3. Flying Shear implementation in a cupcake production line.
1-Converyor Drive; 2- Controller; 3- Actuator; 4- Master Encoder; 5- Trigger.
Image credit: ABB [14].

3.5. Developed Design

The flowchart shown in Figure 3.4. illustrates the process that flying shear control system goes through. Flying Shears are used periodically for cutting a continuously moving material, where the endless material to be cut to length cannot be stopped during the cutting process and the cut must be very accurate "on the fly". The mechanical construction provides a saw or shear system mounted on a carriage that moves either parallel to the product flow or at an angle across the product flow. The Flying Shear drive accelerates the carriage to synchronize with the line speed, at this point the cut tool can be activated. When the cycle of cutting ends, the carriage then decelerates and returns to its original position ready to cut again.

The length of pieces is configured using engineering units of mm, and the number of pieces are configured using decimal digits. This means that configuration of the system is easy, through an operator interface or by entering configuration parameters directly on the controller.

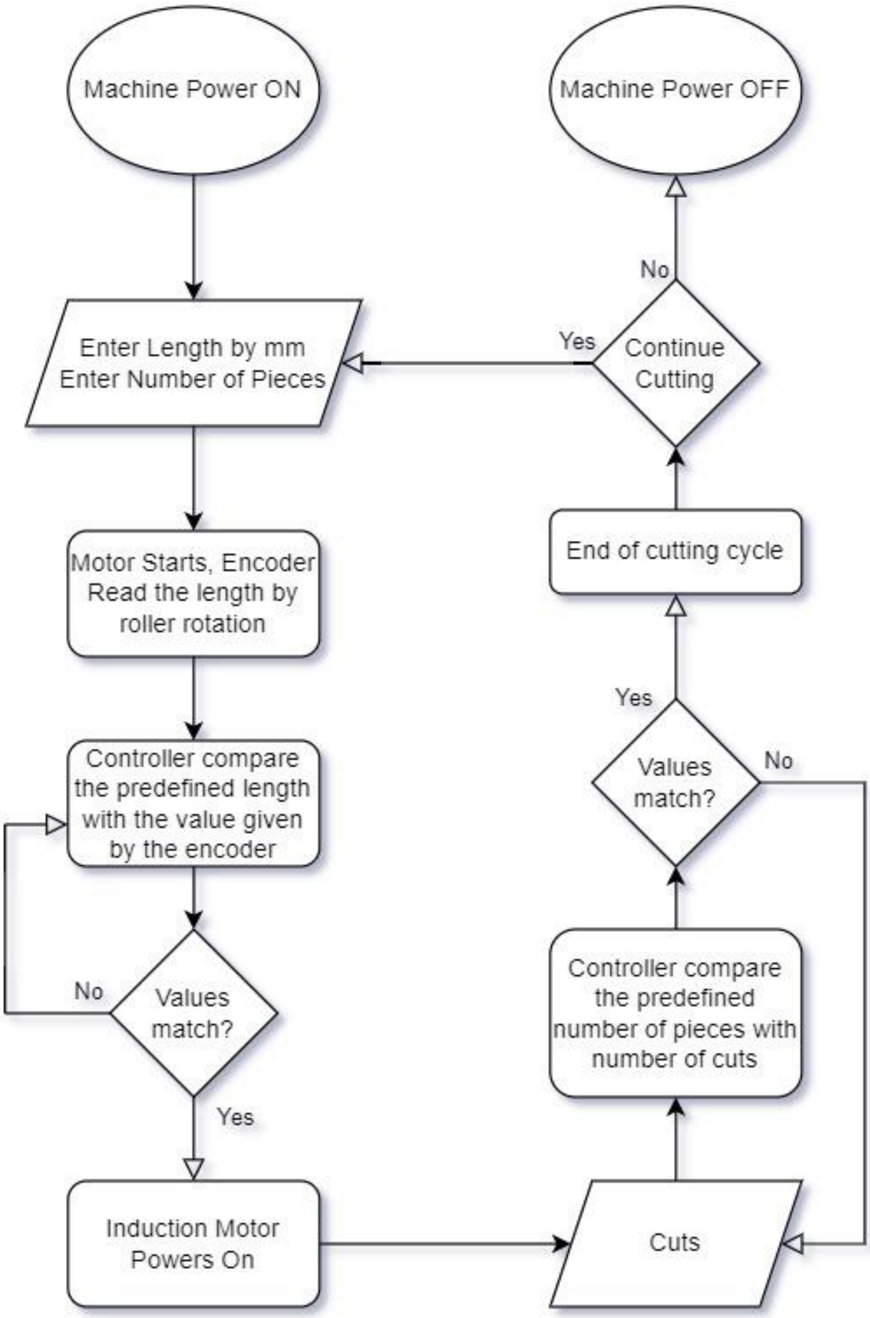

Figure 3.4. The process that flying shear control system goes through.

3.5.1. Flying Shear Operational Modes

The flying shear controller operates under three operational modes:

- Modular Operation Mode.
- Fail Safe Mode.
- Stand-Alone Mode.

The Figure 3.5. illustrates the **Modular Operation Mode** which the module is connected to a Programmable Logic Controller (PLC) by RS-485 serial communication protocol and implement a specific function assigned by the PLC.

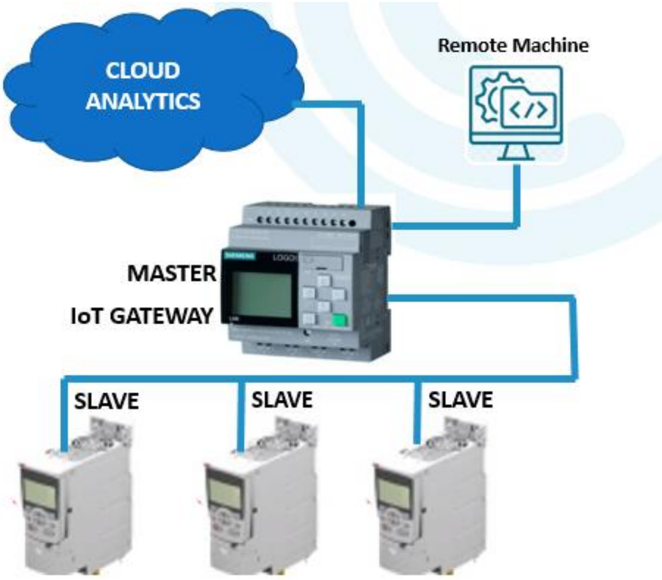

Figure 3.5. Modular Operation Mode

The Figure 3.6. illustrates the **Fail Safe Mode** which the module should be pre-programmed in case of a connection loss with the PLC to carry on its function effectively.

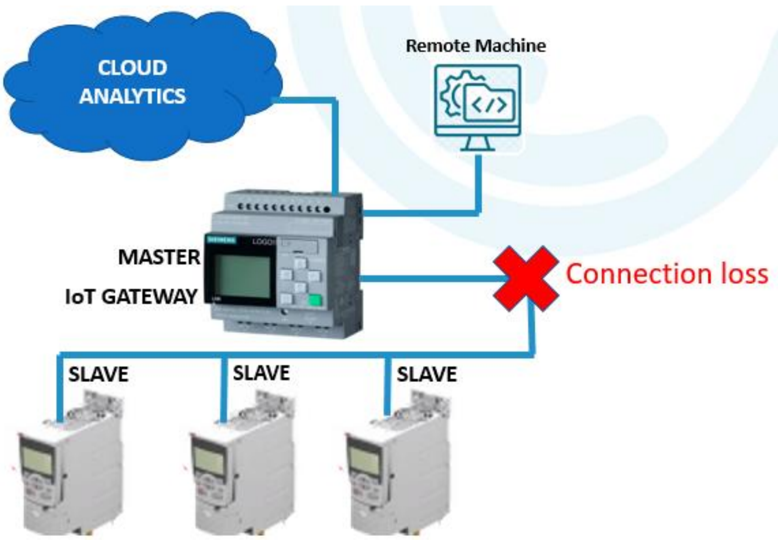

Figure 3.6. Fail Safe Mode

The Figure 3.7. illustrates the **Stand-alone** which the module can be programmed to work as a PLC and control the field devices.

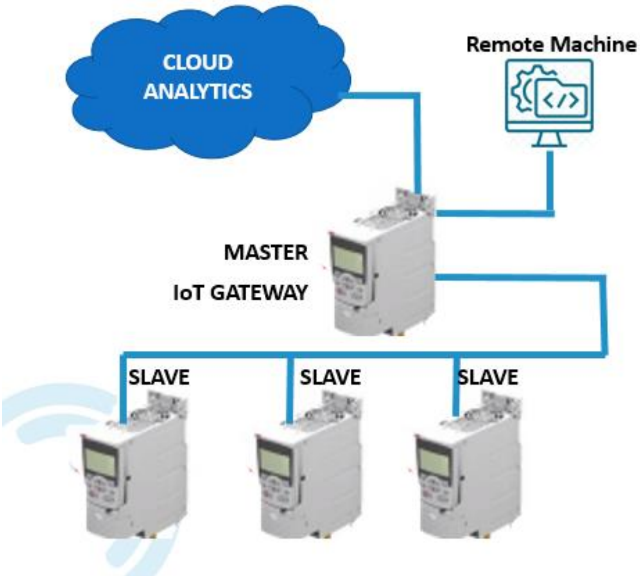

Figure 3.7 Stand-Alone Mode

3.5.2. Schematics

The below figures shows the product schematic, but separated into pieces to make it easy to display. Each label connected to a pin, refers to the connected circuit that matches the label's name.

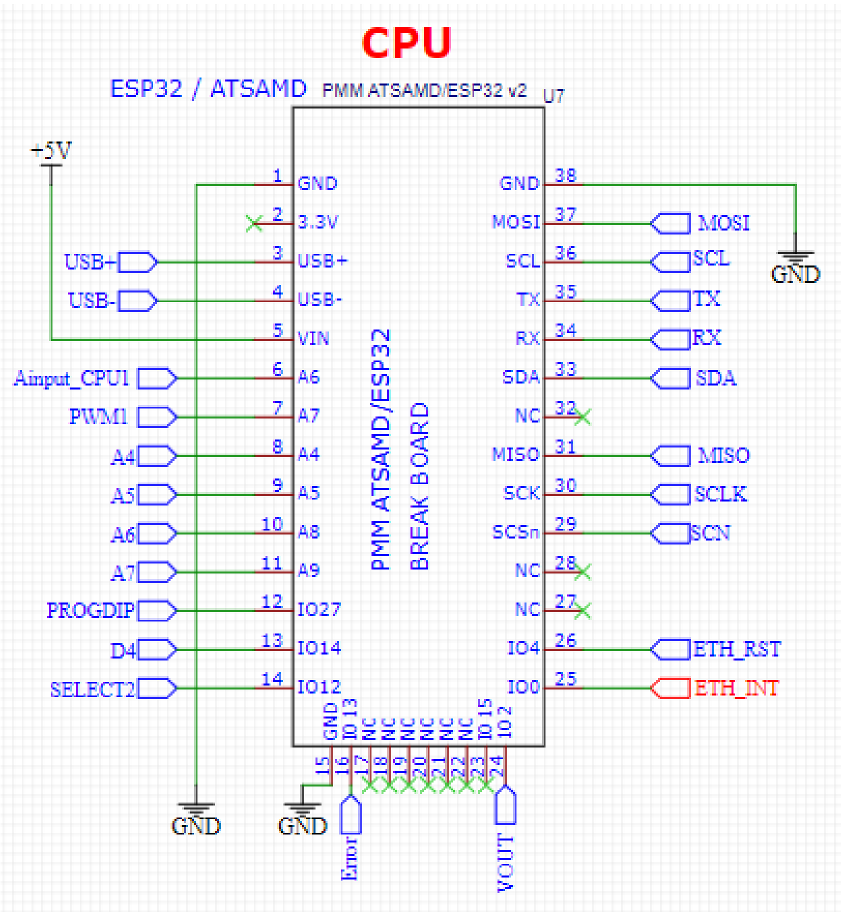

Figure 3.8 CPU Schematic

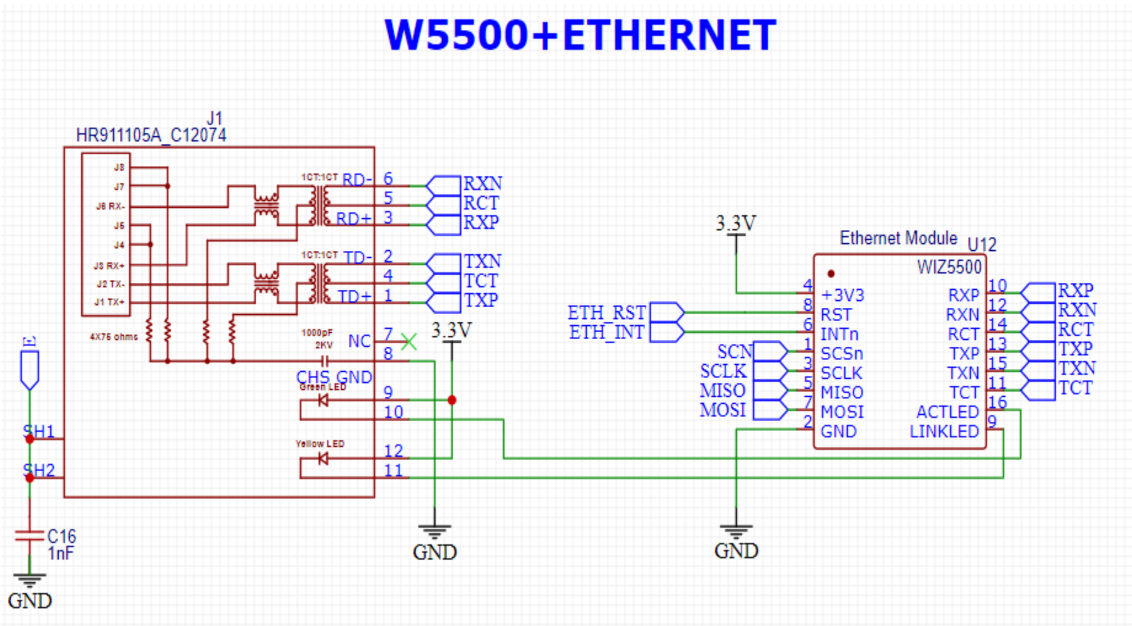

Figure 3.9 Ethernet Module Schematic

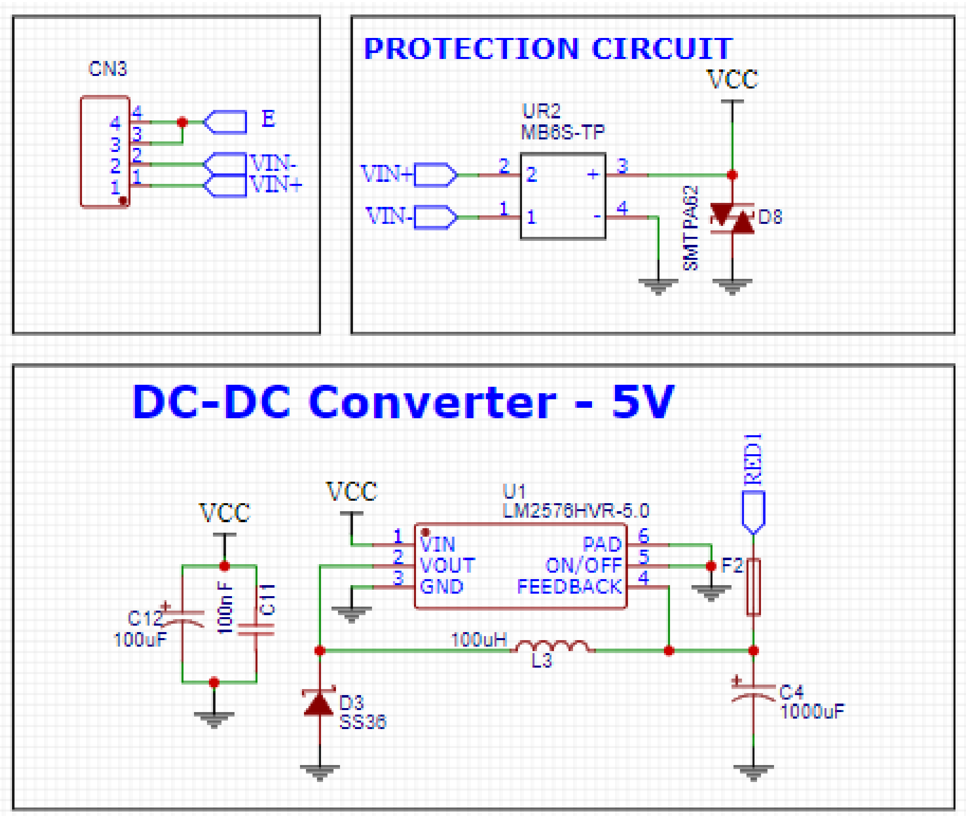

Figure 3.10 Protection Circuit & DC-DC Converter Schematics

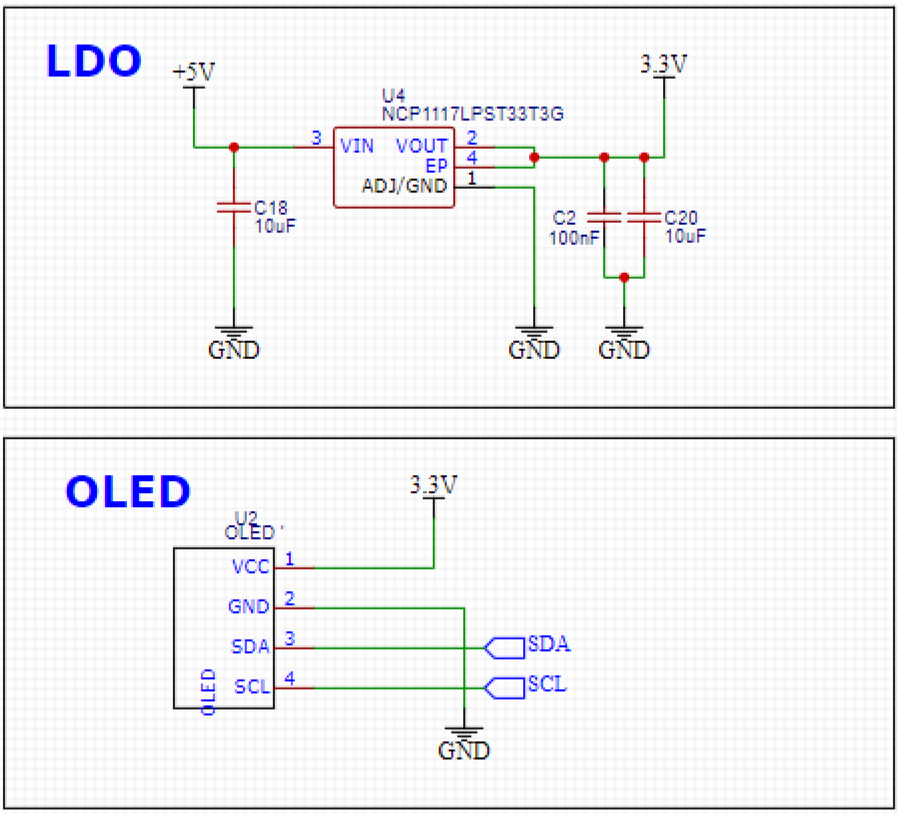

Figure 3.11 Low Dropout & Organic Light-Emitting Diode Schematics

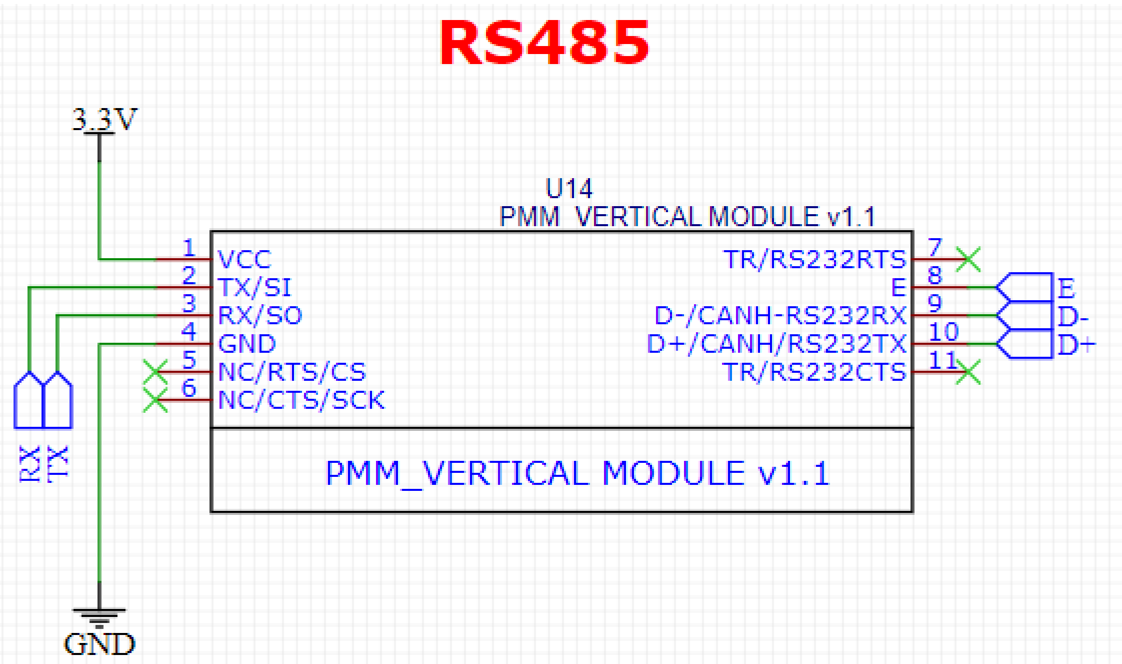

Figure 3.12 RS485 Schematic

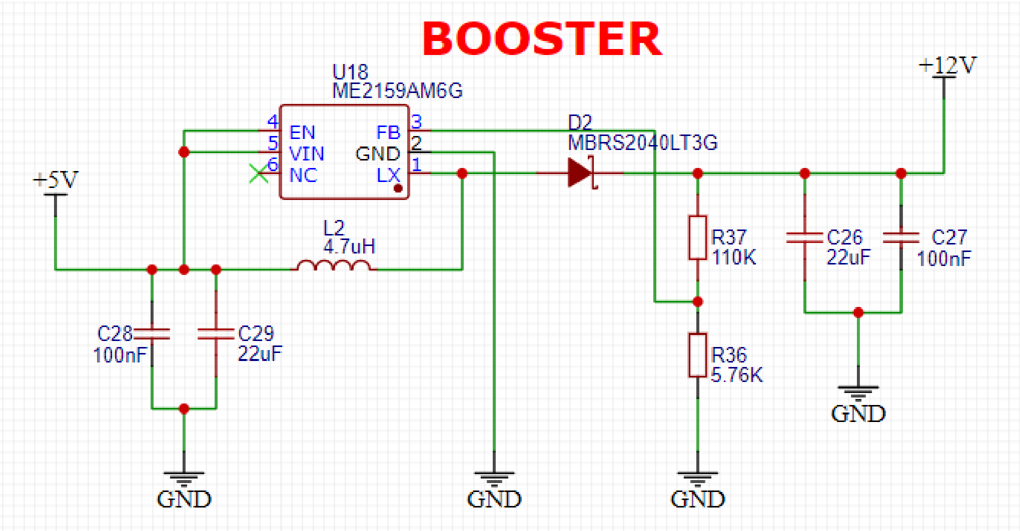

Figure 3.13 Booster Schematic

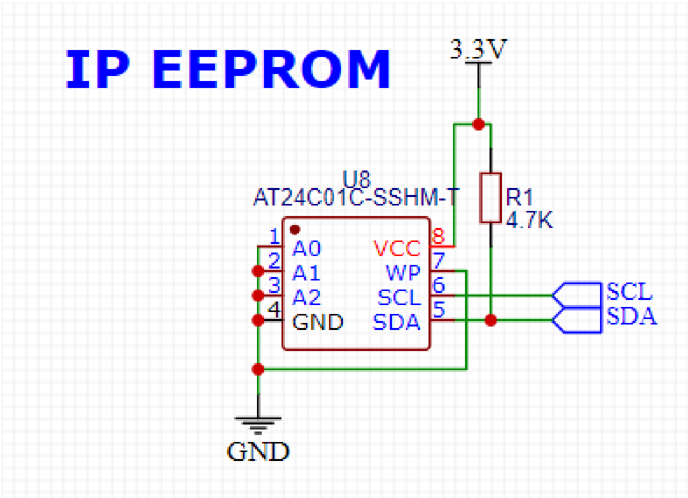

Figure 3.14 EEPROM Schematic

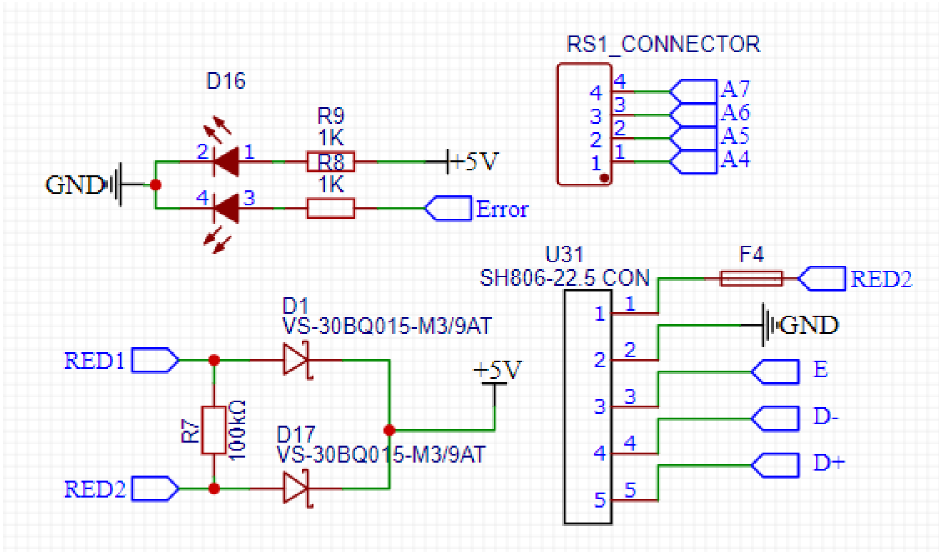

Figure 3.15 Supplementary Schematics

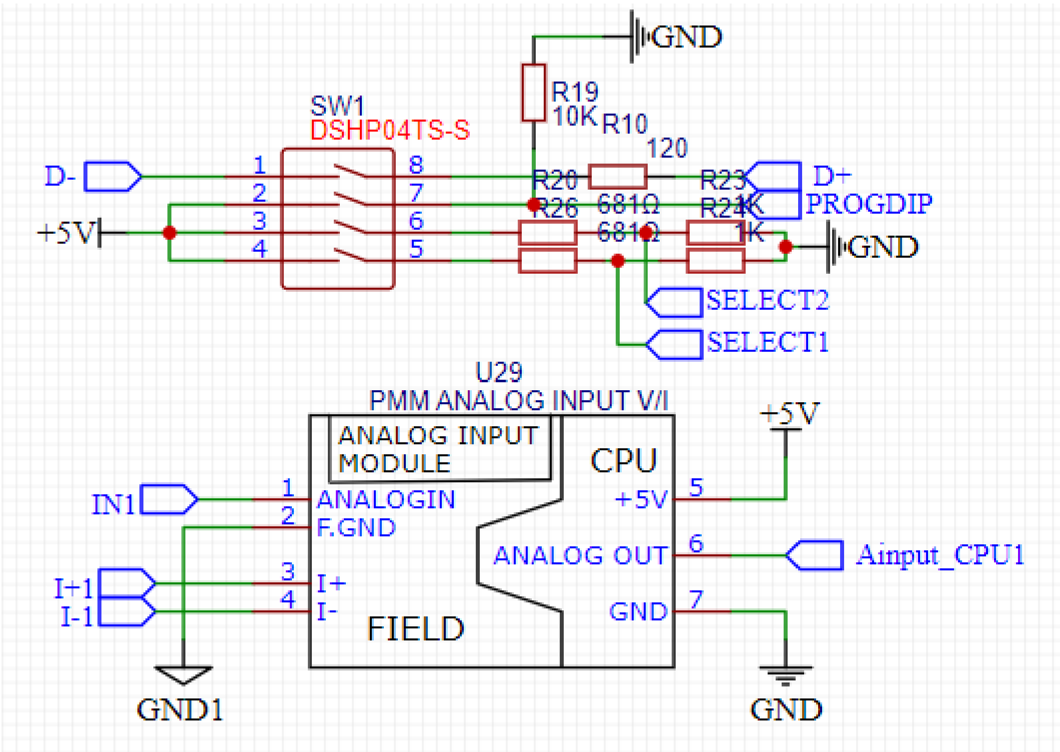

Figure 3.16 Analog Input Schematic

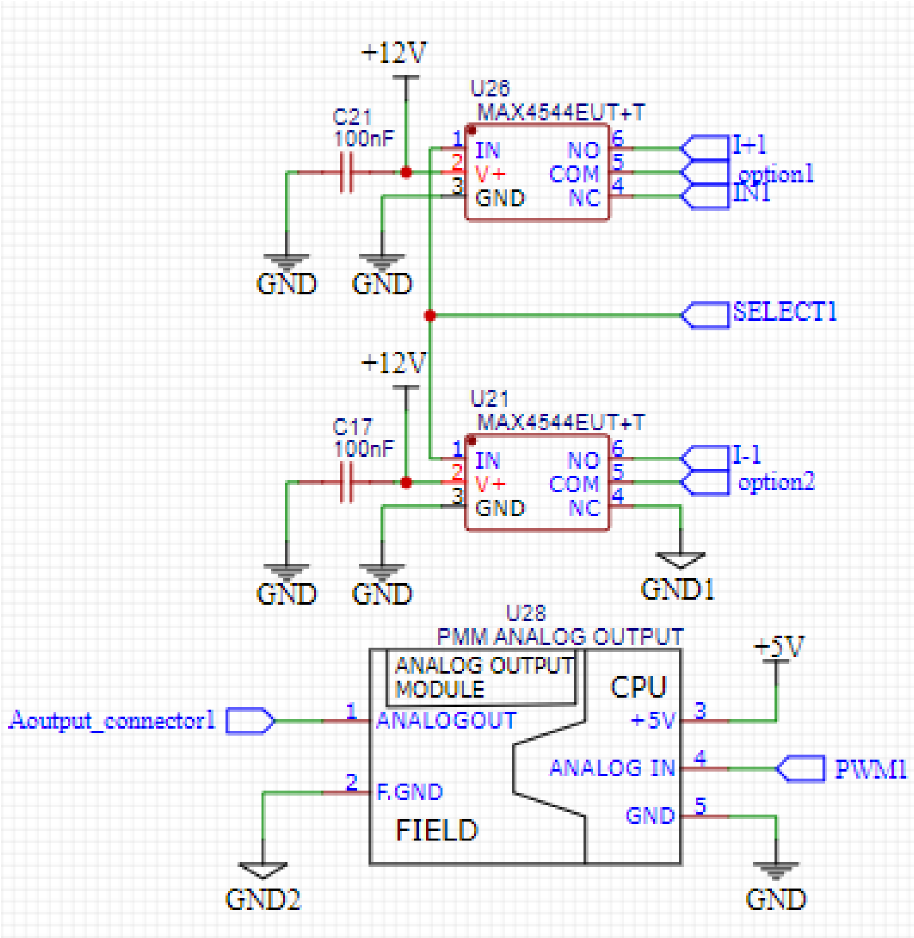

Figure 3.17 Analog Output Schematic

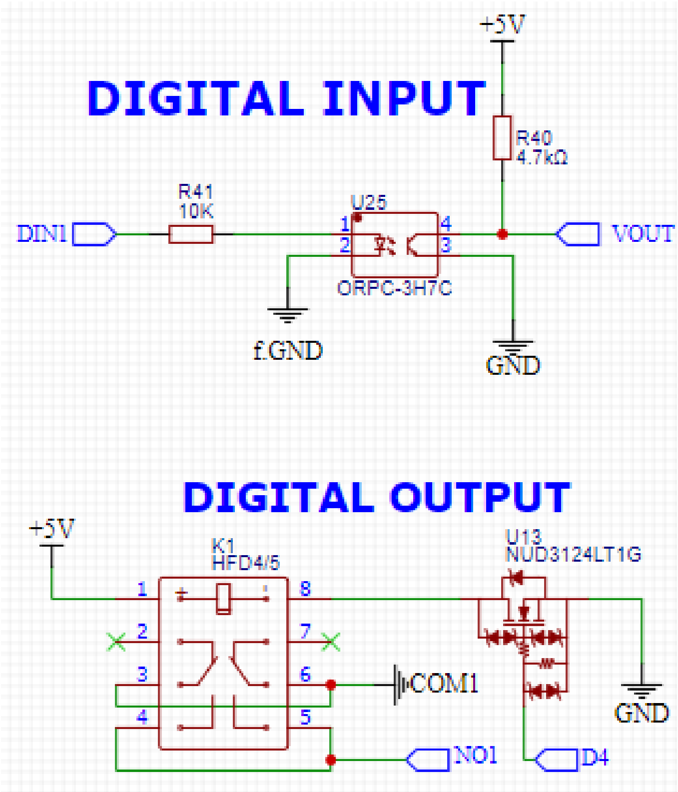

Figure 3.18 Digital I/O Schematic

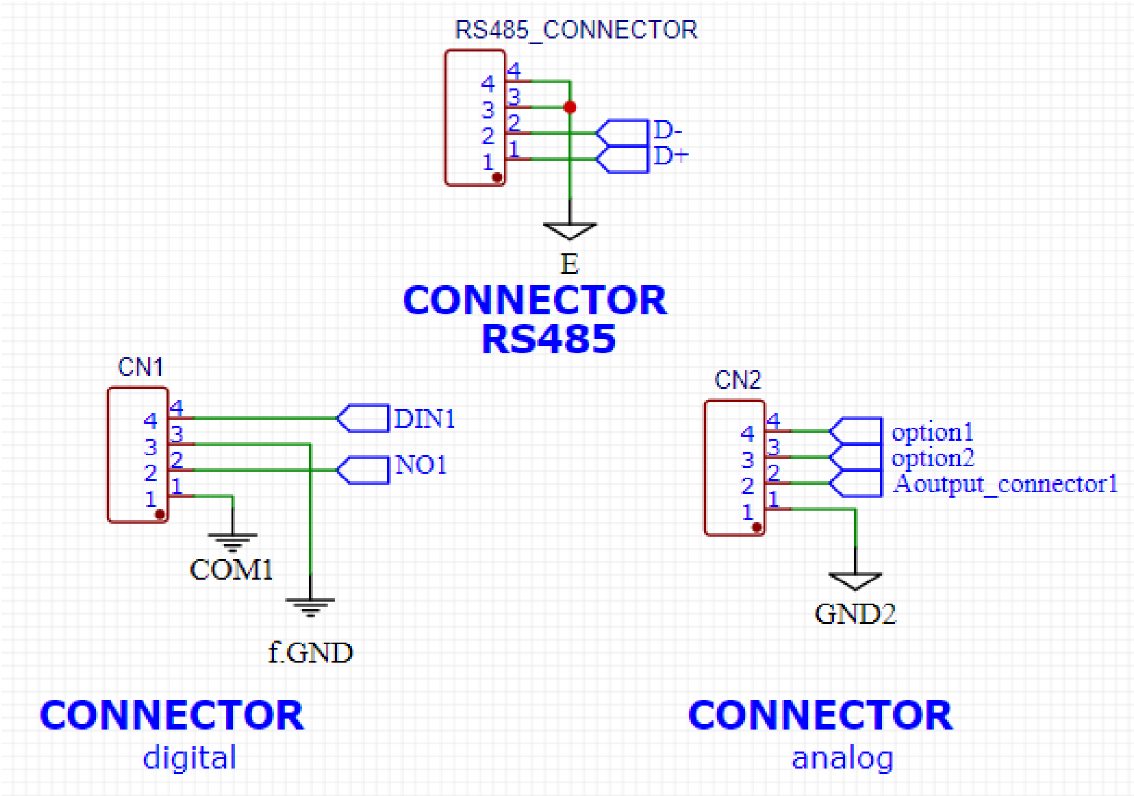

Figure 3.19 RS485, Digital, Analog Connectors Schematic

3.5.3. Block Diagram

The Figure 3.20 shows the block diagram of the Flying Shear Controller, which also, shows the routes and the wires that properly connects the above schematics and its ICs, which builds the entire controller.

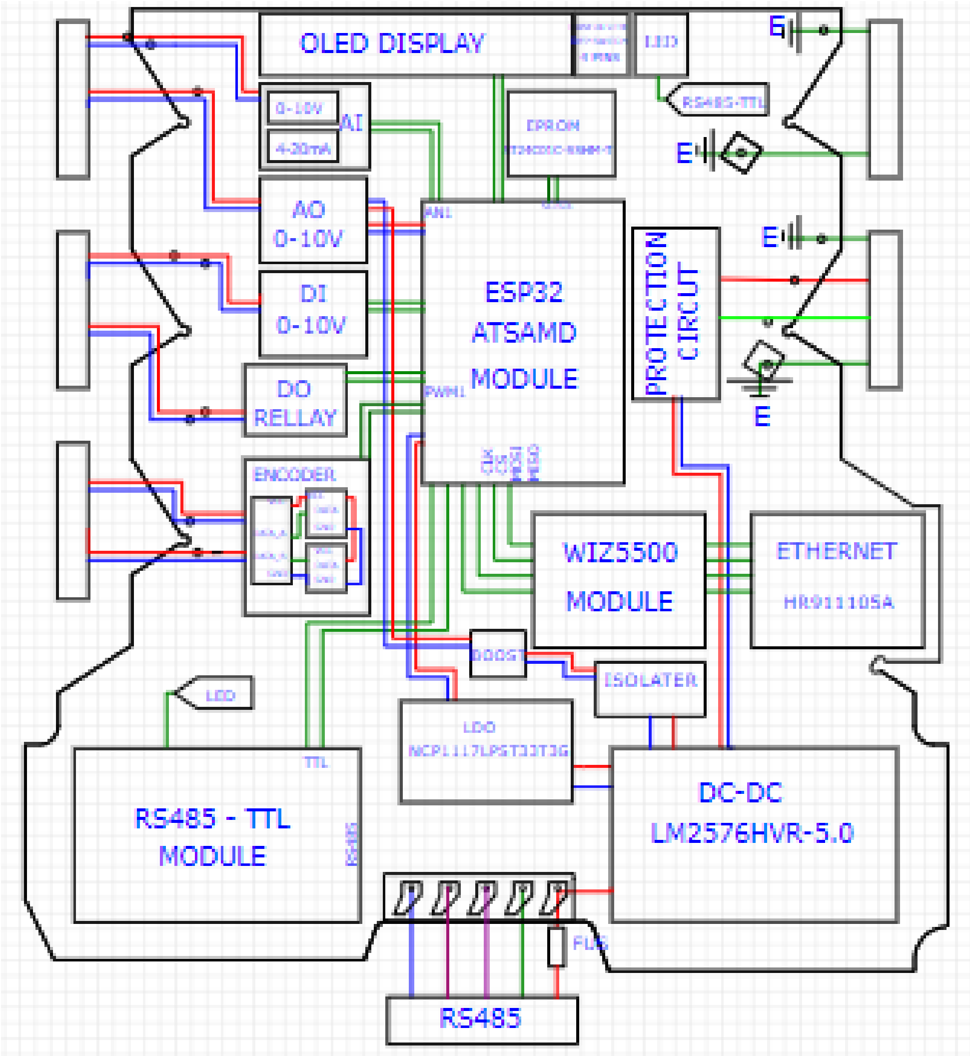

Figure 3.20 The Block Diagram of Flying Shear Controller

3.5.4. Printed Circuit Board (PCB)

The Figures 3.21 and 3.22 illustrate the Printed Circuit Board of the Flying Shear Controller, with the respect of IPC-2221 standard, it is the generic standard that covers almost every aspect of PCB design. The standard details how electrical considerations such as conductor clearance, and impedance control should be implemented on a PCB.

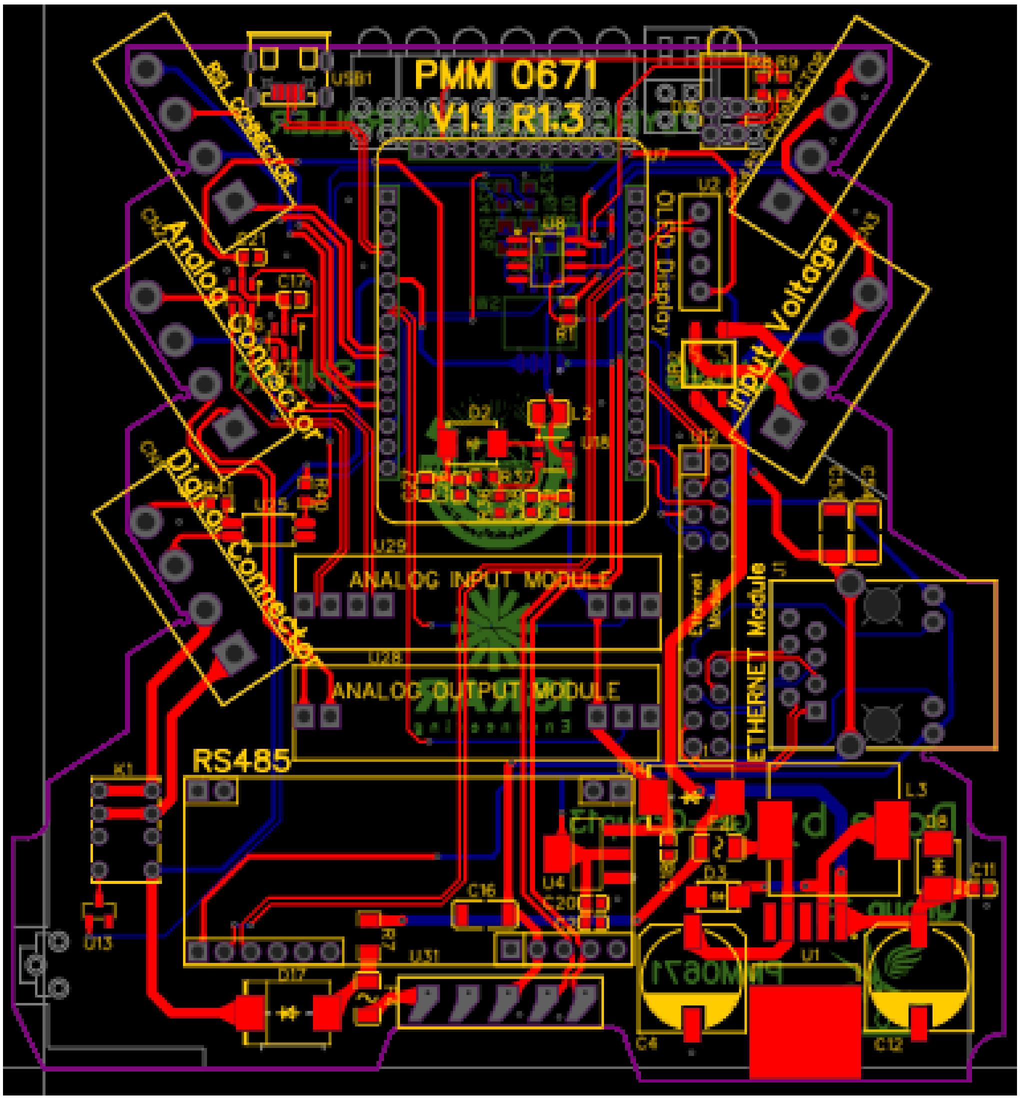

Figure 3.21 The PCB of Flying Shear Controller

The Figure 3.22 illustrates the copper area in the PCB. It is an area of the PCB top layer that is filled with copper, to isolate specific components or circuitry from the rest of the elements of the layer, and it is used as a ground.

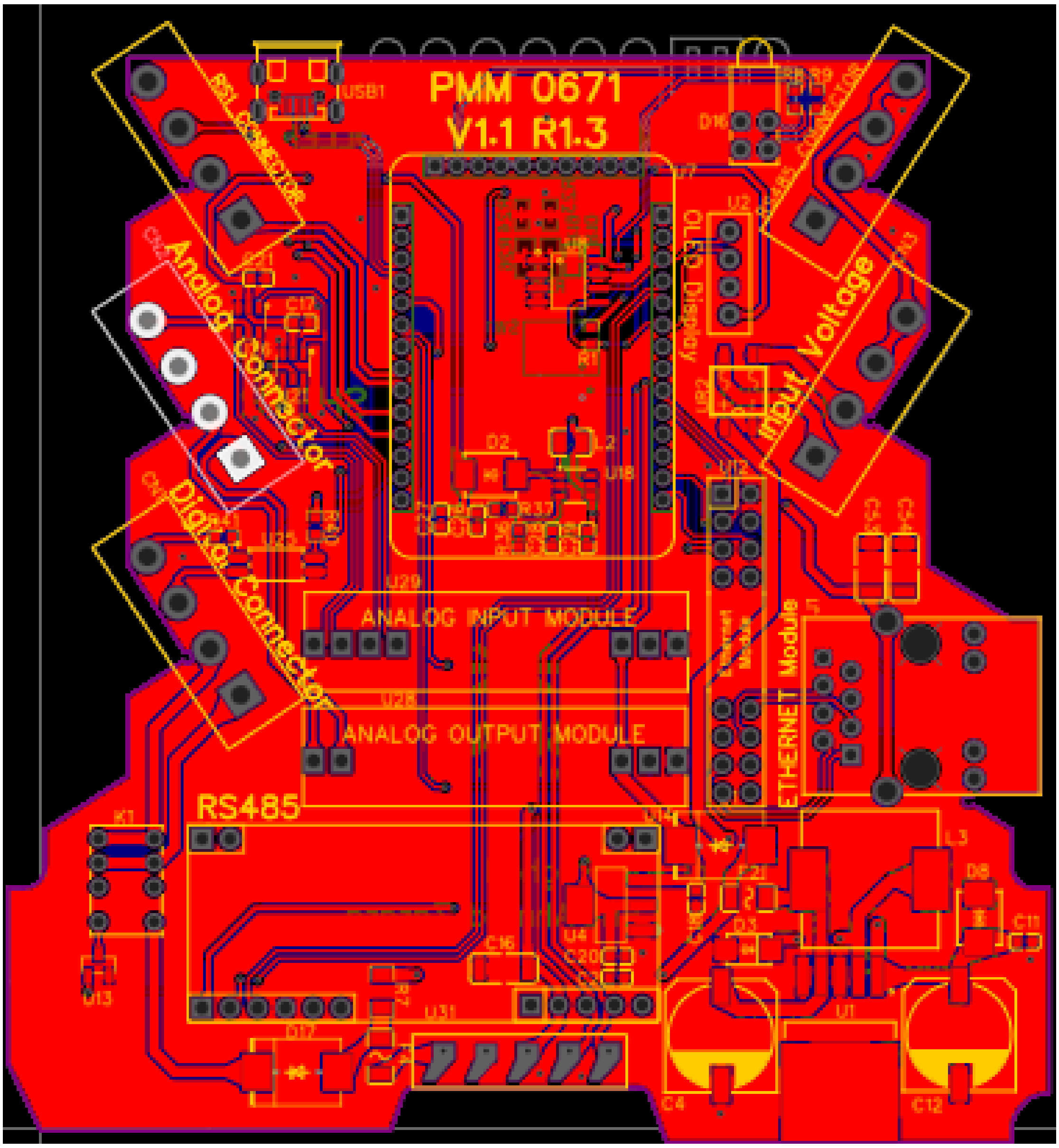

Figure 3.22 The copper area in the PCB of the Flying Shear Controller.

3.5.3. 3D View of Flying Shear Controller.

The Figures 3.23 and 3.24 shows the 3D view of the PCB after the components placement,

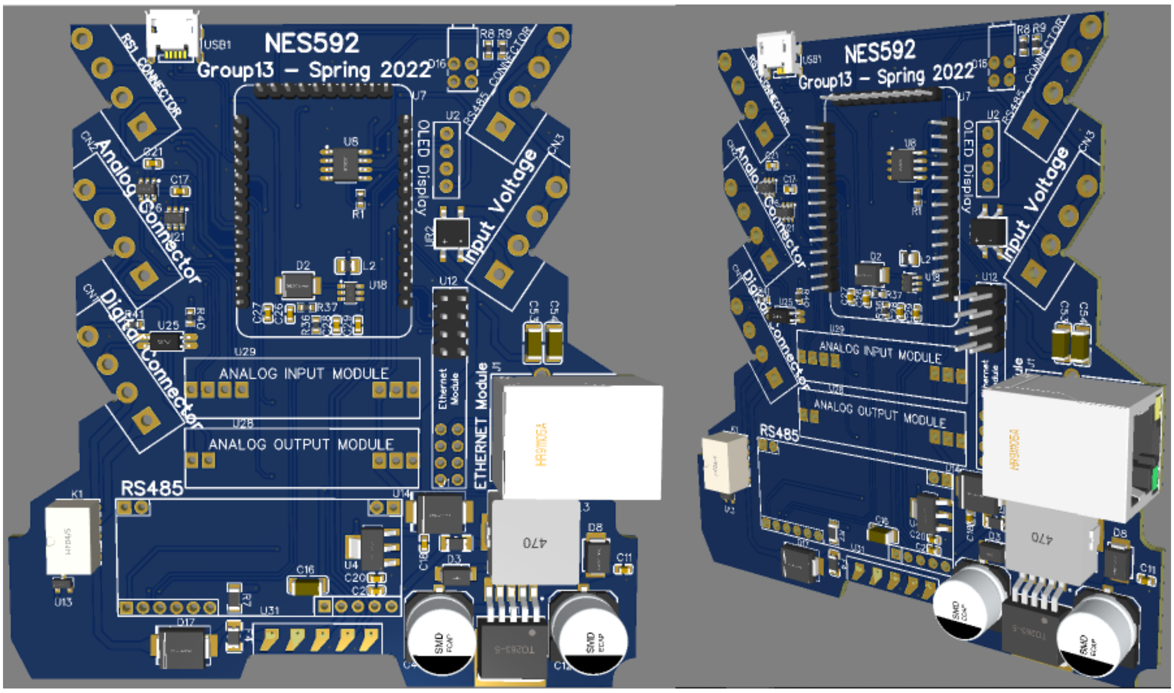

Figure 3.23. Top and Side View of the PCB of Flying Shear Controller

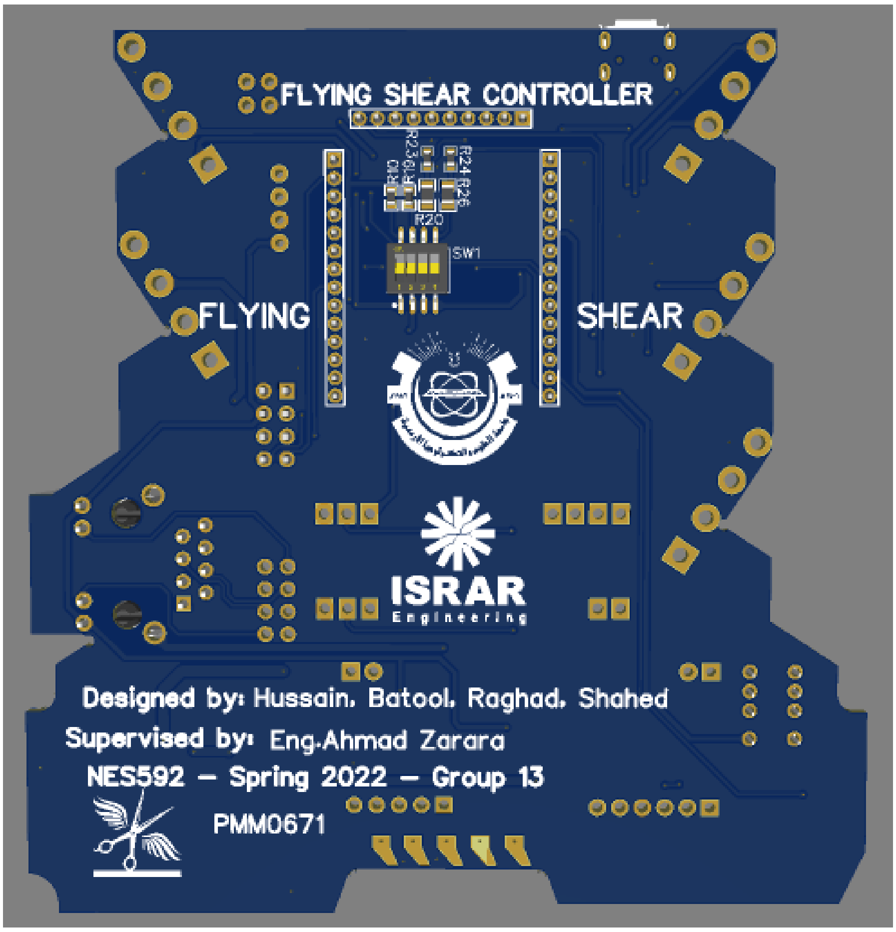

Figure 3.24. Bottom View of the PCB of Flying Shear Controller

CHAPTER 4

IMPLEMENTATION

ESP32 is the microcontroller that we developed, to control the Flying Shear system, with the ability to control it under the consideration of Internet of Things concepts and features, by connecting physical devices to each other as part of a computer network and leveraging the power of smart devices and real-time analytics, the IIoT enhances and improves production processes, enables new levels of factory automation, and allows for greater data visibility, collection, exchange, and analysis.

4.1 Development Environment

We used Visual Studio Code as an environment to write and test the firmware, before uploading it to the ESP32. Visual Studio Code uses extensions to add new languages, themes, debuggers, and to connect to additional services. The Figure 4.1 shows the two installed extensions;

- C/C++ to add the programming language that we will use to write the firmware.
- PlatformIO to provide a universal IDE interface to be able to program the hardware in a more developing-friendly way.

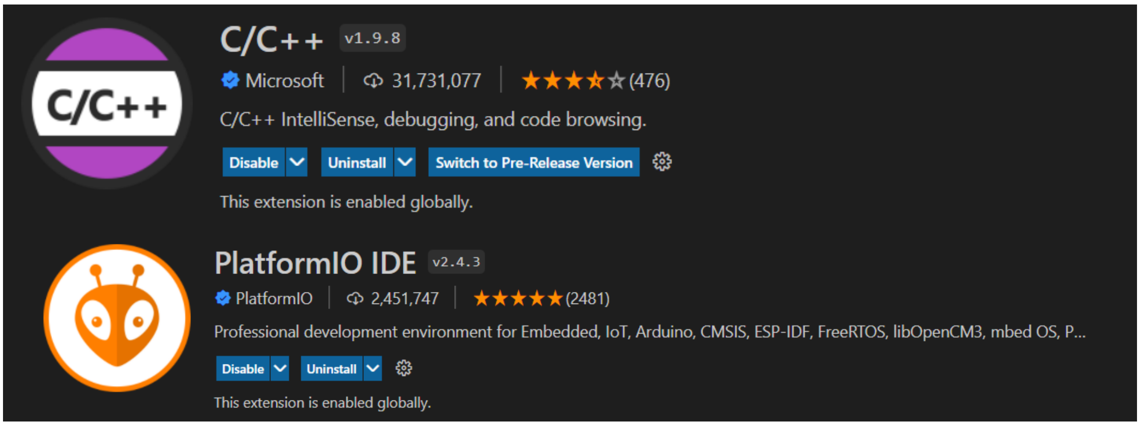

Figure 4.1 Visual Studio Code Extensions

4.2 IoT Gateway with Embedded Web Server

An IoT Gateway is a solution for enabling IoT communication, usually device -to-device communications or device-to-cloud communications. The first step in building the IoT Gateway was building a standalone web server, then connecting it to an existing Wi-Fi network and act as a Web Server, or by making the ESP32 to work as a standalone access point (AP), in case of internet disconnection, to serve the connected devices within the range of its own signal.

4.2.1 ESP32 Station Mode, Soft Access Mode

The ESP32 that creates its own Wi-Fi network and acts as a hub (Just like Wi-Fi router) for one or more stations is called Access Point (AP). Unlike Wi-Fi router, it does not have interface to a wired network. So, such mode of operation is called Soft Access Point (soft-AP). Also the maximum number of stations that can connect to it is limited to five.

The Figure 4.2 shows a code snippet that is responsible for setting up the operating modes of ESP32, to act as an IoT Gateway.

```
void PMMWifiSetup (){
 const char* ssid = "Group13_Existing_WiFi"; // REPLACE_WITH_YOUR_SSID //STATION MODE
 const char* password = "Group_13"; //REPLACE_WITH_YOUR_PASSWORD       //STATION MODE
 WiFi.begin(ssid, password);

 const char* ssid_AP = "Flying Shear Controller";  // Enter SSID here //SOFT ACCESS POINT
 const char* password_AP = "12345678";  //Enter Password here         //SOFT ACCESS POINT
 IPAddress local_ip(192,168,1,1);                                     //SOFT ACCESS POINT
 IPAddress gateway(192,168,1,1);                                      //SOFT ACCESS POINT
 IPAddress subnet(255,255,255,0);                                     //SOFT ACCESS POINT

  //IF NOT STATION MODE AFTER 2.5s -> MAKE IT SOFT ACCESS POINT
  for (int i=0;WiFi.status() != WL_CONNECTED&& i<5;i++) {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }
  //MAKE IT SOFT ACCESS POINT
 if (WiFi.status() != WL_CONNECTED){
    WiFi.softAP(ssid_AP, password_AP);
    WiFi.softAPConfig(local_ip, gateway, subnet);
    delay(1000);
    Serial.println("connect the esp as access point");
 }
 if (WiFi.status() == WL_CONNECTED){
Serial.println(WiFi.localIP()); // Print ESP Local IP Address
 }
///DNS
  if (MDNS.begin("Flying Shear Controller")) { //Flying Shear Controller.local/
    Serial.println("MDNS responder started");
  }
}
```

Figure 4.2 Station Mode, Soft Access Point Set Up

4.2.2 Web Pages

The Web Server provide the connectivity to handle the required settings to initiate the cutting process, by initializing the configurations of the required length, and number of pieces to be cut.

The Figure 4.3 shows the login page when searching for the URL: <http://FlyingShearController.local/> on a web browser, and the resolution from the domain name to an IP address is done by a mDNS-server that is embedded in the firmware.

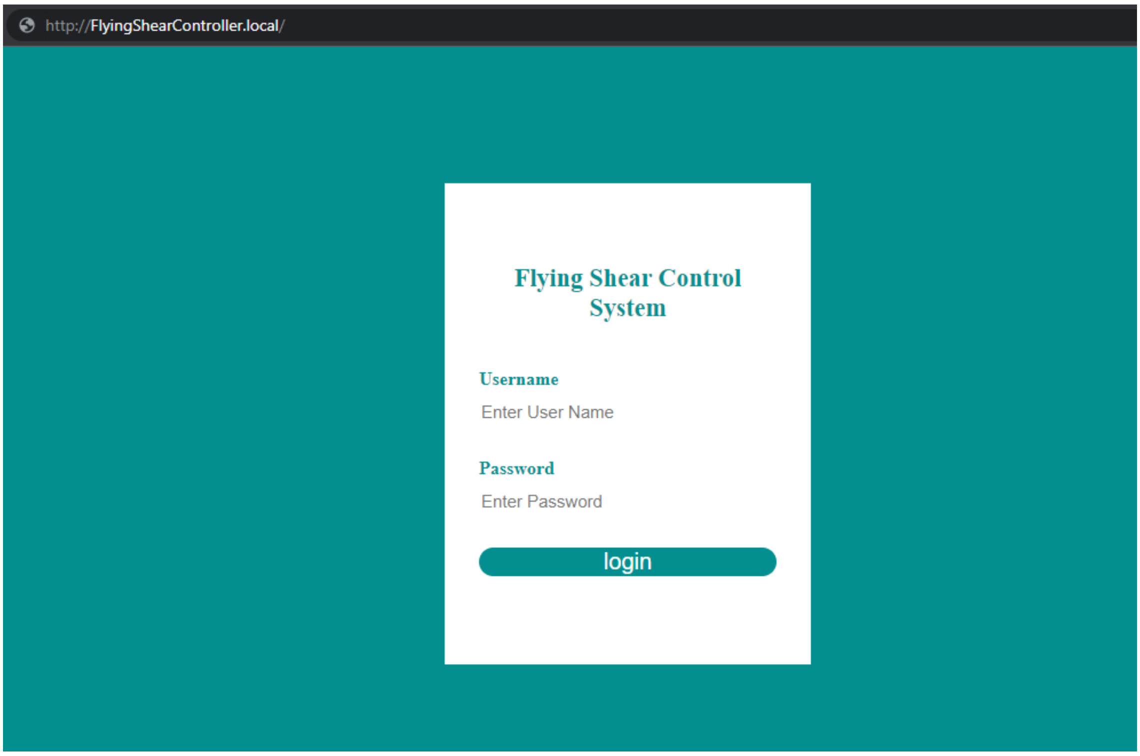

Figure 4.3 Flying Shear Control System Login Page

After filling the credentials and hitting the login button, the html page will send AJAX request to the web server to handle it.

The Figure 4.4 is a snippet of the function that sends the AJAX request from the login web page to the web server behind the scene.

```
function login() {
        console.log('test');
        let u = document.getElementById('Username').value;
        let p = document.getElementById('Password').value;
        console.log('run', p, u); var xhr = new XMLHttpRequest();
        xhr.onreadystatechange = function () {
            if (xhr.readyState == 4 && xhr.status == 200) {
                console.log('done');
                console.log(xhr.responseText);
            }
        };
        xhr.open('GET', 'login?Username=' + u + '&password=' + p, true);
        xhr.send();
        // IF LOGIN PASS -> OPEN THE SETTINGS WEB PAGE
        if (xhr.responseText=="pass login"){
             setTimeout(function(){ window.open("/main","_self"); }, 1000);}
}
```

Figure 4.4 AJAX request to exchange the Credentials with the Web Server

After verifying that the credentials are correct, the login web page will open the Cutting Settings web page to let the user choose his required configurations to start the cutting process.

The Figure 4.5 shows the Settings web page.

Figure 4.5 Cutting Settings Page

After filling the settings and hitting the send button, the html page will send AJAX request to the web server to handle it.

The Figure 4.6 is a snippet of the function that sends the AJAX request from the Settings web page to the web server behind the scene.

```
<div class='col-4-sm'>
  <input type='submit' name='' value='Send' onclick='setVlue()'>
</div>
<script>
function setVlue() {
      let lengthofmaterial = document.getElementById('length').value;
      let number = document.getElementById('number').value;
      console.log('run', lengthofmaterial ,number);
      var xhr = new XMLHttpRequest();
      xhr.onreadystatechange = function () { if (xhr.readyState == 4 && xhr.status == 200)
       {console.log(xhr.responseText);}
      };
        xhr.open('GET','setVlue?number='+number+"&length="+lengthofmaterial,true);
        xhr.send();
}
</script>
```

Figure 4.6 AJAX request to exchange the Settings with the Web Server.

4.3 Firmware Building

This section mainly explains the handling process that is required from the firmware to authenticate and authorize the client. As well as the basic firmware for the product to implement the basic functions that are expected from.

4.3.1 Handling Client

The Figure 4.7 shows the used libraries that are included to use the web server, mDNS, Wi-Fi functions.

The Figure 4.8 shows the code snippet that handles the AJAX requests that have been sent by the web pages, by checking the validity of the credentials, and assigning the length and number of pieces to variables, to make further actions by the firmware.

```
#include <WiFi.h>
#include <WebServer.h>
#include<ESPmDNS.h>
```

Figure 4.7 Required Libraries.

```
  server.on("/", GETLoginPage);
  server.on("/login", checklogin);
  server.on("/main", handle_SettingsPage);
  server.on("/setVlue",setValue );
  Serial.println("HTTP server started");
}
void GETLoginPage() {
  server.send(200, "text/html", loginpage);
}
void checklogin(){
 String username = server.arg("Username");
 String password = server.arg("password");
  if (username==http_username && password==http_password )
  {islogedin=true;
   Serial.println("CORRECT PASSWORD AND USERNAME");
   server.send(200, "text/plane", "pass login");
  }
  else
  {
    Serial.println("incorrect password or username");
    server.send(200, "text/plane", "fail login");
    islogedin=false;
  }
}
void handle_SettingsPage() {
  server.send(200, "text/html", mainPage);
}
void setValue(){
 String NumberofPieces = server.arg("number");
 String Length = server.arg("length");
 Serial.println(length);
  server.send(200, "text/html", mainPage);
}
```

Figure 4.8 AJAX Requests Handling.

4.3.2 Main Firmware

This section explains the firmware for the product to implement the basic functions that are expected from.

We have designed a prototype with four stepper motors, each motor is controlled by a driver called HY_DIV268N, to move the material web, and to hold it while the cutting is in progress, and to cut the material web (Flying Shear).

Figure 4.9 shows the set up portion of the firmware, which is including the needed libraries, and the declaration of the needed variables, stepper motors set up, Wi-Fi setup.

```
/*Coded by: Hussain, Shahed, Batool, Raghad*/

/*We have created the below header file and we have included
the needed libraries to make use of its functions */
#include <generalEEPROMFunction.h>

#include <AccelStepper.h>

//To Count Number of Cut Pieces
int counter = 0;
//OUTPUT VARIABLES
byte benchRollPin = 1;
byte benchSelectPin = 5;
byte TellBreakerOnePin = 6;
byte TellBreakerOTwoPin = 7;

//INPUT VARIABLES
byte flyingShearPositionPin = 9;
byte machineStart = 10;
byte PiecePosition = 11;

//Stepper Motors to move the material web to be cut
AccelStepper stepper(AccelStepper::FULL2WIRE, 2, 11);
AccelStepper stepper2(AccelStepper::FULL2WIRE, 3, 11);
AccelStepper stepper3(AccelStepper::FULL2WIRE, 4, 11);
AccelStepper stepper4(AccelStepper::FULL2WIRE, 8, 11);

//BOOL IF FAILURE HAPPENS AND THE WEB MATERIAL IS STUCK WITHOUT CUTTING
bool place = false;

void setup()
{
  Serial.begin(9600);
  PMMWifiSetup(); // Figure 4.2 IN THE REPORT
                   //Station Mode, Soft Access Point Set Up

  // Get Saved Value from EEPROM (Length and Num Of Pieces)
  PMMGetSavedParameter(); //Figure 4.10 IN THE REPORT
  // Digital Output Setup
  pinMode(benchRollPin, OUTPUT);       // benchRoll
  pinMode(benchSelectPin, OUTPUT);     // benchSelect
  pinMode(TellBreakerOnePin, OUTPUT);  // TellBreaker One
  pinMode(TellBreakerOTwoPin, OUTPUT); // TellBreaker Two
  // Digital Input Setup
  pinMode(flyingShearPositionPin, INPUT);
  pinMode(machineStart, INPUT);
  pinMode(PiecePosition, INPUT);           // PiecePosition Two
  // Stepper Motor Settings
  stepper.setMaxSpeed(200.0);
  stepper.setAcceleration(100.0);
  stepper.moveTo(24);
  stepper2.setMaxSpeed(200.0);
  stepper2.setAcceleration(100.0);
  stepper2.moveTo(24);
  stepper3.setMaxSpeed(200.0);
  stepper3.setAcceleration(100.0);
  stepper3.moveTo(24);
  stepper4.setMaxSpeed(200.0);
  stepper4.setAcceleration(100.0);
  stepper4.moveTo(24);
}
```

Figure 4.9 Firmware Setups.

The Figure 4.10 shows the function that extracts the cutting settings from the EEPROM that represents the length of the cut, number of pieces.

```
void PMMGetSavedParameter()
{
    /**
     * @brief Get All Saved Parameter on the EEPROM
     */
    SerialUSB.begin(9600); // Initalize USB For Debug
    Debugprintln("Start Project Init ...");
    projectParamConfigtaion(); // get default Project parameter
    initEEPROM();              // Initalize EEPROM
    Debugprintln("...");
    // Check if its saved before+
    SerialUSB.println(isItFirstRun());
    if (isItFirstRun())
    {
        cleanEEPROM(); // Clean EEPROM
        markAsRead();
        PMMSetNetworkInfoInfo();         // Set Default Network Info
        PMMSetLengthAndNumberOfPieces(); // Set Flying Shear Settings
    }
    else
    {
        PMMGetNetworkInfoInfo();         // get  Network Info
        PMMGetLengthAndNumberOfPieces(); // Get Flying Shear Settings
    }
    Debugprintln("..."); // Initaliz
}
```

Figure 4.10 Extracting the Cutting Settings from EEPROM.

The Figures below shows the void loop() function, this is where the bulk of the cutting operation is executed. The program starts directly after the opening curly bracket (}), runs until it sees the closing curly bracket (}), and jumps back up to the first line in loop() and starts all over.

```
void loop()
{
  server.handleClient();
  if (counter == NumOfPieces)
  {
    machineStart=false;
    stepper.stop();
    stepper2.stop();
    stepper3.stop();
    stepper4.stop();
  }
```

Figure 4.11 Checking if number of cuts equals to the predefined value

```
else if (digitalRead(machineStart)) //Turning ON Controller
{
  digitalWrite(benchRollPin, HIGH);
  while (!digitalRead(PiecePosition))
  {
    stepper.run();
  }
  if (digitalRead(flyingShearPositionPin, HIGH))
  {
    stepper2.run();
    counter++;

  }
  if (place)
  {
    stepper3.run();
     digitalWrite(TellBreakerTwoPin,LOW);
    digitalWrite(TellBreakerOnePin, HIGH);
    place = false;
  }
  else
  {
    stepper4.run();
     digitalWrite(TellBreakerOnePin, LOW);
    digitalWrite(TellBreakerTwoPin, HIGH);
    place = true;
  }
}
}
```
Figure 4.12 Cutting Operation is Executed.

CHAPTER 5

TESTING AND EVALUATION

The hardware is tested thoroughly which includes all the mechanical and electrical aspects such as the product's enclosure, heat dissipation, components selection, placement. The software is tested thoroughly as well including the basic web interface is functional properly, the software functions are giving the true values and all connections and networks are successfully connected with no disturbances.

The Figure 3.17. In Chapter 3, The Block Diagram of Flying Shear Controller will be our reference for hardware testing.

Power Supply Testing:

The power supply circuitry is the part of providing the electronics parts of the device with the necessary power to operate. The conventional DC form of the input power shall be converted into a rectified DC power to comply with the input needs of the electronics key parts embedded within the enclosure and the device.

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

1. Reverse Polarity + Over Voltage.
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.

1. <u>Reverse Polarity and Over Voltage Testing:</u>

Table 5.1 Reverse Polarity and Over Voltage Testing Before Connecting to the PCB.

Voltage Input Range	**Actual output voltage (drop volt 0.3)**	**Note**
10	9.72	
20	19.55	
30	29.68	
40	39.60	
50	49.66	
52	51.62	
54	53.55	
56	55.61	
58	57.66	
60		At 60 volt the TVs is failed
No load		

Table 5.2 Reverse Polarity and Over Voltage Testing After Connecting to the PCB.

Voltage Input Range	**Actual output voltage (drop volt 0.3)**	**Measurement output voltage**
10	9.7	
20	19.73	
30	29.76	
40	39.69	
50	49.68	
52	51.67	
54	53.65	
56	55.64	
58	57.65	
60	65.65	
Full load		

2. DC-DC Converter 5V Testing:

The DC-DC converter (BUCK) is the second phase operation of the power supply, where the output voltage of the rectifier must be supplied to the buck converter to step down the voltage to values that are compatible with the electronics key part.

Table 5.3 DC-DC Converter 5V Testing Before Connecting to the PCB.

Voltage Input Range	**Actual output voltage (5 volt)**	**Measurement output voltage**
10	5	5
20	5.1	5
30	5.1	5
40	5.1	5
50	5.1	5
52	5.1	
54	5.1	5
56	5.1	5
58	5.1	5
60	5.1	5
No load		

Table 5.4 DC-DC Converter 5V Testing After Connecting to the PCB.

Voltage Input Range	Actual output voltage (5 volt)	Measurement output voltage
10	5	5
20	4.98	5
30	4.76	5
40	4.98	5
50	4.96	5
52	4.87	
54	4.76	5
56	4.66	5
58	4.97	5
60	4.95	5
Full load		

3. <u>3V3 LDO Testing:</u>

A low dropout (LDO) linear voltage regulator is a type of linear voltage regulator circuit that works well even when the output voltage is very close to the input voltage, improving its power efficiency.

Table 5.5 3V3 LDO Testing Before Connecting to the PCB.

Voltage Input Range	Actual output voltage (3.3 volt)	Measurement output voltage
10	3.29	3.3
20	3.297	3.3
30	3.299	3.3
40	3.298	3.3
50	3.298	3.3
52	3.298	
54	3.298	3.3
56	3.297	3.3
58	3.298	3.3
60	3.298	3.3
No load		

Table 5.6 3V3 LDO Testing After Connecting to the PCB.

Voltage Input Range	Actual output voltage (3.3 volt)	Measurement output voltage
10	2.95	3.3
20	2.87	3.3
30	2.96	3.3
40	2.87	3.3
50	2.98	3.3
52	2.95	
54	2.93	3.3
56	2.87	3.3
58	2.91	3.3
60	2.9	3.3
Full load		

4. Boost Converter Testing:

A booster is Step-Up circuit used for voltage regulation in direct current(DC) electrical power circuits. It takes in a low voltage input of around 5 volts and gives an output of 12 volts.

Table 5.7 Boost Converter Testing.

Voltage Input Range	Actual output voltage (12 Volt)	Measurement output voltage
10	12	12
20	12.06	12
30	12.07	12
40	12.07	12
50	12.06	12
52	12.06	
54	12.06	12
56	12.06	12
58	12.06	12
60	12.06	12

Ethernet Circuit Testing:

We have tested this circuit individually before placing it on the PCB, and the output voltage from this circuit was 3.3 volt which is convenient.

Also, this circuit has been tested by connecting it to the internet, we noticed that the interface has got an IP address. After that, we disconnected the internet connection, we noticed that this interface took an IP address of an existent local printer.

Digital Input/Output Testing:

We have tested the digital input by providing a voltage of 34V to the external connection, and the internal connection (PCB Side) have to provide 3.3V approximately.

Table 5.8 Digital Input Testing.

Voltage Input Range	Actual output voltage	Note
10	3.72	
20	3.55	
30	3.68	
40	3.60	
50	3.66	
52	3.62	
54	3.55	
56	3.61	
58	3.66	
60	3.98	

We have tested the digital output by providing a voltage of 3.3V to the internal connection (PCB Side), and the external connection have to provide 24V approximately.

Table 5.9 Digital Output Testing.

Voltage Input Range	Actual output voltage	Note No note
10	23.4	
20	24	
30	23.68	
40	23.60	
50	23.66	
52	22.62	
54	23.55	
56	23.61	
58	23.66	
60		

Analog Input/Output:

Table 5.10 Analog Input Testing

Voltage digital input	CPU
1	0.15
2	0.3
3	0.45
4	0.61
5	0.76
6.02	0.92
7	1.08
8	1.23
9.08	1.39
10	1.54

Table 5.11 Analog Output Testing

CPU	Analog output voltage
0.5	1
1.02	2.02
1.53	3
2	3.95
2.5	4.9
3	5.83
3.3	6.45
4	7.78
4.5	8.78
5	9.72

CHAPTER 6

CONCLUSION

Shears are the most widely used sheet cutting equipment in various industrial sectors and are currently widely used in steel, printing, and cardboard processing industries. The Flying Shear Control System performs synchronous cutting during the material feeding process to realize high-precision cutting without stopping. The flying shear control system is the key to the shearing machine. The quality of its product control performance directly affects the product yield. There are countless industries that make use of Industrial Internet of Things (IIoT). One example is the automotive industry, which uses IIoT devices in the manufacturing process. IIoT can help proactively maintain these systems and spot potential problems before they can disrupt production. Our proposed system insures the process of cutting any manufactured material using Flying Shear Control System, integrated with Internet of Things.

REFERENCES

[1] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 978–1013, Second Quarter 2018.

[2] K. Huang, W. Liu, Y. Li, and B. Vucetic, "To sense or to control: Wireless networked control using a half-duplex controller for IIoT," accepted by Proc. IEEE Globecom 2019.

[3] Nedjeljko Peric and Ivan Petrovic, "Flying Shear Control System" IEEE Transactions on Industry Applications, vol.26, no.6, pp.1049-1056, Nov/Dec 1990.

[4] L. Qi, L. Shuo, W. Ying, L. Kai, L. Baohua, M. Huili, and L. Jinran, "DESIGN OF FLYING SHEAR DEVICE AND ALGORITHM BASED ON MOTION CONTROL SYSTEM" Topics in Chemical & Material Engineering, pp. 257–259, 2018.

[5] PCB design tool. Available at: https://easyeda.com/ (Accessed on 12/11/2021)

[6] IPC-2221A – Generic Standard on Printed Circuit Board Design. Available at: https://www.ipc.org/TOC/IPC-2221A.pdf

[7] Modicon Modbus Protocol Reference Guide. Available at: https://www.modbus.org/docs/PI_MBUS_300.pdf

[8] The RS-485 Design Guide. Available at: https://www.ti.com/lit/an/slla272d/slla272d.pdf?ts=1642750405118&ref_url=https%253A%252F%252Fwww.google.com%252F

[9] IEEE 802.11-1999 - IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan Area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Available at: https://standards.ieee.org/standard/802_11-1999.html

[10] IEEE 802.3-2015 – IEEE Standard for Ethernet. Available at: https://standards.ieee.org/standard/802_3-2015.html

[11] USB Implementers Forum – USB2.0 Specification. Available at: https://www.usb.org/document-library/usb-20-specification

[12] Ingress Protection (IP) Ratings. Available at: https://webstore.iec.ch/publication/2452

[13] ABB Motion | Application – Flying Shear Cakes. Available at: https://www.youtube.com/watch?v=RaTVUGSUYkg/ (Accessed on 1/12/2022).

[14] ABB Motion Control Products. Available at: https://new.abb.com/drives/low-voltage-ac/servo-products/ (Accessed on 12/24/2021).

[16] IoT Fundamentals: Connecting Things (a course from Cisco Networking Academy).
Available at: https://www.netacad.com/courses/iot/iot-fundamentals

[17] Datatracker.ietf.org. 2022. *rfc2616*. [online] Available at: <https://datatracker.ietf.org/doc/html/rfc2616> (Accessed on 18 February 2022).

[18] Datatracker.ietf.org. 2022. *rfc1035*. [online] Available at: <https://datatracker.ietf.org/doc/html/rfc1035> [Accessed 18 February 2022]