Faculty of Information Technology

University of Moratuwa 2023

Final Evaluation Report (IN 1901)

Level 1

Symbol Drawer

Group 40

Group members

Abesekara R.J.D.	214002D
Gunathilaka B.H.M.M.P.M	214075C
Dissanayake D.M.K.M.B.	214059G
Sadhunhari W.P.D.	214176L
Yasundara G.A.O.	214241H

Mrs. Chamalka Rajapakshe Date of submission

…………………………………………. ……………………………..

Table of Contents

Introduction

Symbols are visual or tactile representations that hold significant importance in human communication and understanding. They provide us with a universal language that transcends linguistic and cultural barriers, enabling individuals from diverse backgrounds to interact with their surroundings effectively. Their versatility extends to facilitating accessibility for individuals with disabilities, promoting inclusivity, and ensuring consistency in conveying information across diverse contexts and cultures. Indoor public spaces serve as the canvases of human interaction, each adorned with a myriad of symbols that communicate vital information, guide our actions, and foster a sense of order and understanding.

In the tapestry of indoor public spaces, symbols drawn on the floor serve as silent guides, seamlessly choreographing the movements of individuals and defining the functionality of these dynamic environments. These floor-based symbols are, in essence, visual orchestrators, shaping our interactions, ensuring safety, and enhancing the overall experience within these spaces.

Transitioning from manual symbol placement to an automated indoor floor symbol drawer introduces a paradigm shift in managing and utilizing symbols within indoor public spaces. The significance of this transition cannot be overstated. Manual symbol placement is a labor-intensive, time-consuming, and often error-prone process. It demands substantial human resources and meticulous attention to detail, especially in large and complex environments. In contrast, automation introduces efficiency, precision, and adaptability.

Automation allows for symbols to be dynamically arranged and adjusted in real time, catering to changing needs, events, or accessibility requirements. This adaptability is particularly crucial in today's fast-paced world, where spaces need to swiftly transform to accommodate various functions and unexpected situations. Moreover, ensuring consistent symbol alignment and placement, reducing the potential for confusion among users, and enhancing overall safety.

Cost efficiency is a significant advantage as well. Automation reduces labor costs associated with manual placement and ongoing maintenance, resulting in long-term savings. It also minimizes disruptions caused by manual adjustments, allowing indoor spaces to function smoothly without unnecessary downtime.

Furthermore, an automated system responds rapidly to emergencies and dynamic scenarios, providing clear and up-to-date guidance to ensure safety and effective crowd management. It can swiftly reconfigure symbols during events, exhibitions, or even public health emergencies like social distancing requirements.

This introduction highlights the critical role of symbols drawn on floors in indoor public places and sets the stage for introducing our "Floor Symbol Drawer" hardware project as an innovative solution to enhance this essential aspect of public environments

LITERATURE SURVEY

The current technological landscape in Sri Lanka predominantly comprises machines and equipment for the creation of symbols in various media. These technologies are widely employed in signage, advertising, and design industries. However, we observed that specific machines tailored for drawing symbols directly on the floor are not prevalent in the market.

Sri Lankan graphic design firms and sign-making companies frequently employ vinyl cutting machines to craft adhesive vinyl floor decals with symbols. While versatile, these machines are primarily designed for other purposes, such as signage and vehicle graphics.

Notably, our survey revealed that the majority of floor symbol drawing methods in Sri Lanka remain manual. Skilled technicians manually apply symbols, safety markings, and directional cues to indoor and outdoor floors. This manual approach is resource-intensive and time-consuming. For example, Stencil-based floor painting is a common practice, where technicians use stencils made from various materials to apply paint on the floor to create symbols. This method is often used in factories, warehouses, and parking areas.

While automated floor symbol drawing machines tailored explicitly for this purpose do not appear to be prevalent in Sri Lanka, there is a clear opportunity for innovation in this space. Emerging technologies, coupled with local expertise in design and fabrication, could potentially pave the way for the development of automated floor symbol drawing solutions suited to the specific needs of Sri Lankan public spaces.

AIM AND OBJECTIVES

Aim
Draw any symbol(Including texts) that is given by a G-code in two colors on the ground in a 2*2 feet frame that automatically moves forward two feet at a time.

Objectives
1. Automatically moving forward.
   Ability to autonomously move forward and complete a given symbol on the floor. This entails creating a sophisticated system that can navigate and position itself accurately to draw symbols with precision, all without the need for manual intervention.

2. Giving the G-code related to the symbol to the system.
   Facilitates the input and interpretation of G-code related to symbols into the system. G-code, a standardized programming language widely used in the manufacturing and CNC machining industries, will serve as the command language for instructing the automated floor symbol drawing machine.

3. Detecting the frame limit where the symbol is drawn.
   Revolves around the implementation of a frame detection system within a 2x2 grid, where the symbol drawing process takes place. The primary goal is to equip the floor symbol drawing machine with the capability to autonomously detect the frame limits within this grid and subsequently initiate movements to complete the symbol.

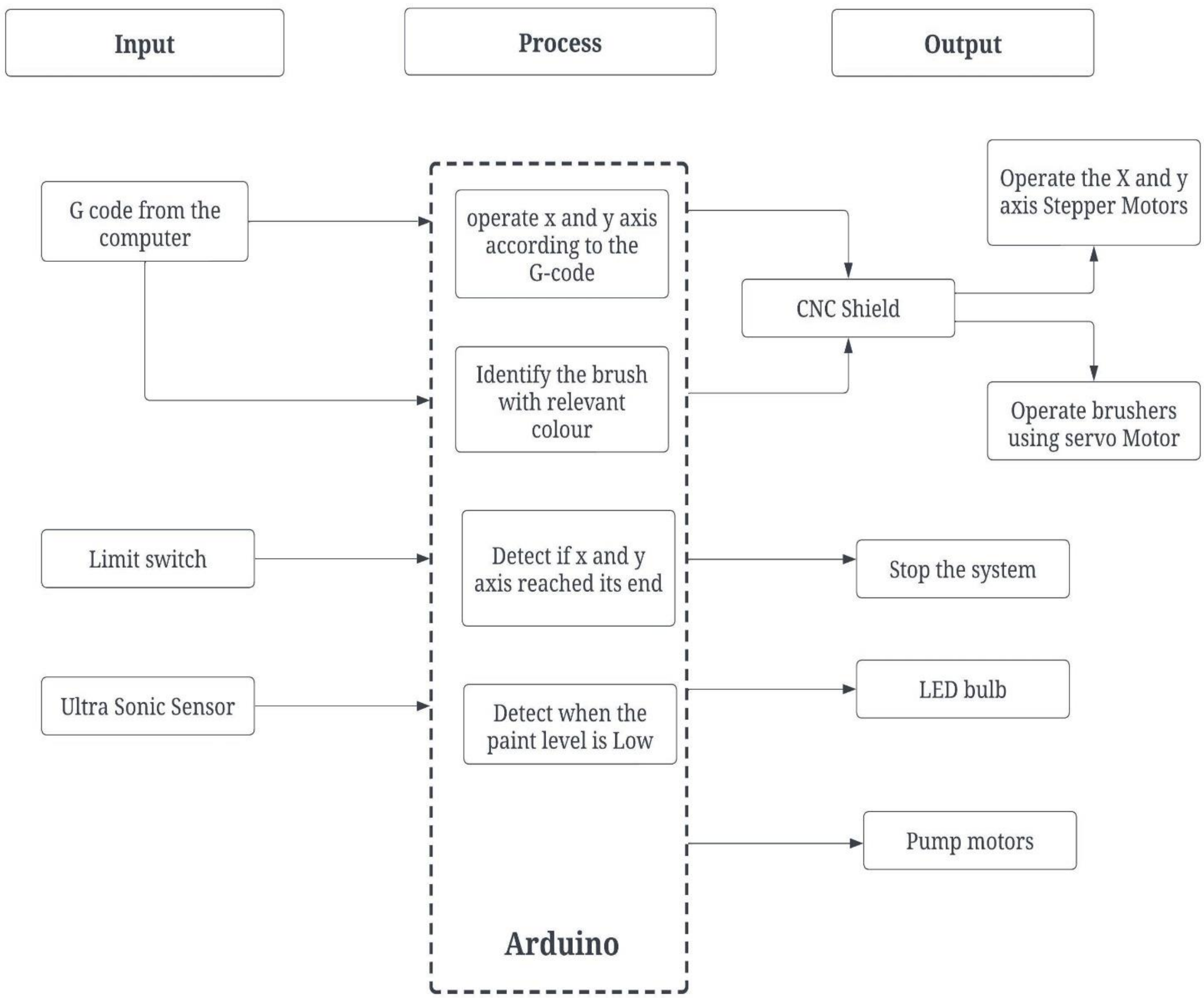
4. Apply the paint from the containers to the two brushes through pump motors.
   The project focuses on creating a reliable paint dispensing system. This system will consist of containers that store the paint required for symbol drawing. Each container will be equipped with a dedicated pump motor responsible for drawing paint from the container and transferring it to the brushes.

5. Draw the symbol using the paintbrushes by moving the x and y axes mechanically. (Paintbrushes are moving in the z-axis).
   This objective entails leveraging the CNC plotter technology to execute the drawing of symbols with meticulous control over mechanical movement along the x and y axes, while simultaneously adjusting the brushes in the z-axis.

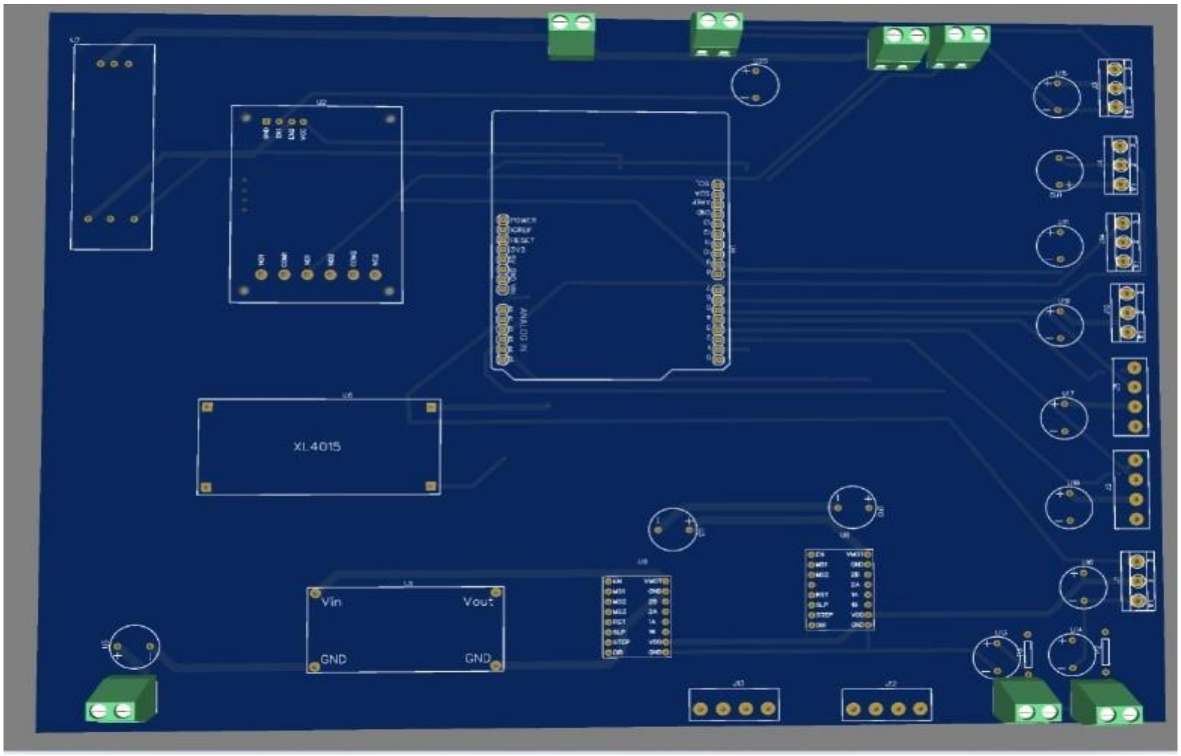
6. Detecting the paint level in pain containers and activating an LED bulb when the paint container level decreases.
   This is the paint level detection system within the paint containers, and it activates an LED indicator when the paint level decreases beyond a predefined threshold. This innovative feature aims to enhance operational efficiency and minimize disruptions during the symbol drawing process.

ANALYSIS AND DESIGN

Block Diagram of the machine



Schematic Diagram Of the machine



TESTING AND IMPLEMENTATION

All team members have successfully studied the modules assigned to them. For our project, we have selected the Arduino Uno board as our microcontroller of choice. Each team member has thoroughly researched the Arduino Uno board and its extensive range of features, ensuring that we are well-prepared to leverage its capabilities effectively in the execution of our project.

The circuit diagram was drawn by the members of the team. The codes for each module is coded by the particular member of the group.

The first phase of our hardware implementation revolved around designing a comprehensive circuit diagram, a blueprint that guided the integration of electronic components. We leveraged the user-friendly EasyEDA software to create this diagram, ensuring that our hardware architecture aligned seamlessly with our project's objectives.

At the core of our machine's intelligence lay the Arduino Uno microcontroller, a versatile choice that served as the brain of our system. Extensive research and exploration of the Arduino Uno's capabilities were conducted, enabling us to maximize its potential effectively.

One of the standout features of our machine was the inclusion of an Infrared (IR) sensor. Positioned strategically, this sensor played a pivotal role in ensuring the precision of our symbol drawing process. When the machine completed drawing a symbol within one frame and returned to its designated starting point, the IR sensor came into play. It detected the machine's arrival and, initiated movement forward to complete the remaining portion of the symbol.

Our project's versatility was further amplified through the integration of two ultrasonic sensors, a dual setup meticulously designed to monitor paint levels in our containers. However, the initial stages of our project posed a significant challenge. The ultrasonic sensors encountered accuracy issues due to the small diameter of our initial paint containers and the inherent difficulty in setting the sensor's degree. In response, we showcased adaptability and problem-solving by transitioning to larger containers.

The heart of our project lay in the CNC (Computer Numerical Control) system, which brought our symbol drawing capabilities to life. We utilized a CNC shield to simplify our processes, streamlining the integration of CNC functionalities. To communicate with the Arduino Uno board, we employed G-code, a standardized programming language widely used in CNC machining. To generate G-code instructions for symbols, we turned to Inkscape software, a versatile tool that allowed us to create intricate symbol designs effortlessly.

Once all the hardware components were ready, we embarked on the coding phase, aligning each component's functionality with its corresponding code. This meticulous coding process involved writing and testing code snippets for the Arduino Uno, IR sensor, ultrasonic sensors, and the CNC shield. These codes were carefully integrated to ensure seamless communication and operation. The culmination of this effort resulted in a well-coordinated and responsive system.

TOTAL COST AND EXPENDITURE

Item	Quantity	Price
Stepper Motor Nema 17	4	8800
Stepper Driver	4	1520
Servo Motor	1	1800
Linear Rail shaft	2	3000
Power supply	1	3850
IR sensor	1	280
Pump motors	2	3000
Ultra Sonic Sensors	2	720
Arduino uno	2	5900
CNC shield	1	1080
PCB design	1	4000
Buck converters	2	1240
Relay module	2	640
Wheels	4	2000
Other components		25 000
Total cost		62 830

FURTHUR WORK

We have identified that adding the following features will improve the process of the machine and it can be a more useful machine. To improve the machine to achieve our goals more effectively, we would like to suggest these features.

1.) Remote Control and Monitoring:
Implement a remote control and monitoring system that allows operators to control the machine and monitor its progress from a distance. This feature could be especially useful for large-scale installations or situations where real-time adjustments are needed.

2.) Wireless Connectivity:
Integrate wireless communication capabilities, such as Wi-Fi or Bluetooth, to enable seamless communication between the machine and a control interface. This would enhance the machine's versatility and ease of use.

3.) Improved Symbol Customization:
Enhance the software to enable more intricate symbol designs and customization options. This could include the ability to import vector graphics or designs from external sources.

4.) Variable Paint Application:
Implement a system for adjusting the paint application thickness or color intensity based on user-defined parameters. This would allow for more creativity in symbol design.

5.) Multiple Color Support:
Add support for multiple paint containers with different colors, enabling the creation of multi-colored symbols and designs.

6.) Real-Time Error Handling:
Develop a real-time error detection and correction system to address issues like paint spillage, sensor malfunctions, or obstructions on the floor. The machine could automatically pause and notify operators when such issues arise.

7.) Mobile Application Integration:
Create a mobile application that enables users to design symbols on a smartphone or tablet and send them to the machine for drawing. This would simplify the design process and broaden the machine's user base.

8.) Battery-Powered Option:
Design a battery-powered version of the machine for outdoor use or locations without readily available power sources. Implement energy-efficient components and power management systems to extend battery life.

9.) Path Planning Algorithms:

Incorporate advanced path planning algorithms to optimize the machine's movement, reduce paint consumption, and minimize drawing time. This could include algorithms for efficient symbol placement and path smoothing.

10.) User-Friendly Interface:
Improve the user interface to make it more intuitive and user-friendly. Consider adding a touchscreen display for easier setup and control.

11.) Safety Features:
Enhance safety features, such as obstacle detection and emergency stop mechanisms, to ensure safe operation in public spaces.

12.) Maintenance Alerts:
Implement a maintenance alert system that notifies operators when routine maintenance tasks, such as cleaning or replacing components, are required to keep the machine in optimal condition.

These future enhancements can help expand the capabilities and market potential of our automated floor symbol drawing machine while addressing specific user needs and requirements.

REFERENCES

- https://www.arduino.cc/
- https://easyeda.com/
- https://www.duino.lk/product/cnc-shield-v3-0-stepper-motor-drivers-shield/
- https://robocraze.com/blogs/post/what-is-motor-driver
- https://cnc-apps.com/en/
- https://inkscape.org/
- https://lastminuteengineers.com/
- https://www.fanucamerica.com/products/cnc/cnc-software/programming-simulation-software/cnc-guide-simulation-software
- https://resources.pcb.cadence.com/blog/2023-an-introduction-to-printed-circuit-boards
- https://www.alibaba.com/premium/computer_components.html?p4phangyebuliu=1&src=sem_ggl&field=UG&from=sem_ggl&cmpgn=18084521248&adgrp=141549188618&fditm=&tgt=kwd-10827891&locintrst=&locphyscl=1009919&mtchtyp=b&ntwrk=g&device=c&dvcmdl=&creative=617983648514&plcmnt=&plcmntcat=&aceid=&position=&gclid=Cj0KCQjwjt-oBhDKARIsABVRB0w2lLRv2XgWsos0RBVy1dE2ft8ycXFSV1aUYBso1pP8Y2JU8CAIArAaAl9aEALw_wcB
- https://tronic.lk/
- https://maxbotix.com/blogs/blog/how-ultrasonic-sensors-work#:~:text=An%20ultrasonic%20sensor%20is%20an,information%20about%20an%20object's%20proximity.

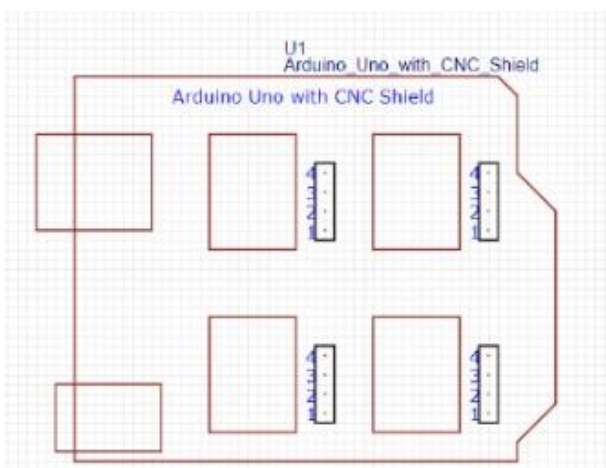
INDIVIDUAL CONTRIBUTION

1.) Abesekara R.A.D   214002D

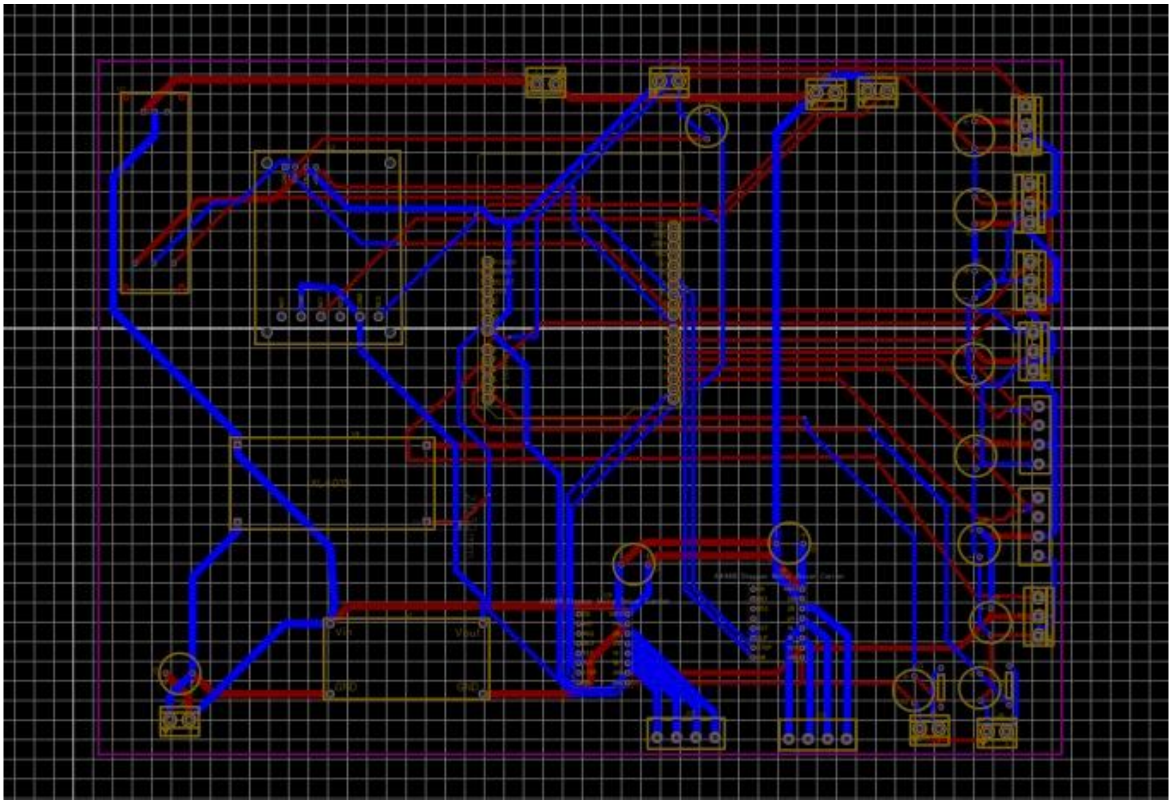
My primary responsibility revolved around the precise control of the machine's X and Y axes. To accomplish this, I meticulously configured and integrated a CNC shield into our hardware setup. The CNC shield served as the interface between the Arduino Uno microcontroller and the stepper motors responsible for moving the machine's paintbrushes along the X and Y axes. Through careful calibration and programming, I ensured that the machine could execute intricate symbol designs with utmost precision. This control system not only laid the foundation for our project's functionality but also contributed significantly to its accuracy and versatility.

Our project's success was the generation and execution of G-code instructions to direct the CNC system. My role extended to mastering the operation of Inkscape software, a powerful tool for creating intricate symbol designs and converting them into G-code. With Inkscape, I seamlessly transformed artistic concepts into machine-readable instructions, enabling the automated floor symbol drawing machine to translate these designs onto the floor surface with unparalleled finesse.

Beyond software and CNC control, I assumed the responsibility of overseeing the design of the PCB, a foundational element of our project's hardware architecture. With meticulous attention to detail, I collaborated with our team to design a PCB that seamlessly integrated our chosen components, ensuring optimal functionality and durability. And also I have assigned to do the power supply system of the machine. We want 5V and 12V for the machine, I have designed and simulated the power supply system to get those voltages from the AC current.



Arduino Uno with CNC Shield


Vin
Vout
GND
GND

2.) Gunathilaka B.H.M.M.P.M   214075C

In this collaborative endeavor to develop an automated floor symbol drawing machine, my role was pivotal in ensuring the smooth forward movement of the machine and the effective utilization of Nema 17 stepper motors with A4988 drivers. These responsibilities required a deep understanding of motor control, precision engineering, and system reliability.

My primary responsibility was to oversee the forward movement of the machine during the symbol drawing process, closely integrating it with our IR sensor system. When the IR sensor received the signal indicating the symbol's completion, it triggered the forward movement of the machine. This smart mechanism ensured that the machine seamlessly continued its journey to complete the remaining portion of the symbol.

A significant part of my role involved the management of Nema 17 stepper motors, a key component of our machine's mechanical system. These motors were paired with A4988 drivers, known for their reliability and efficiency in controlling stepper motors. I meticulously configured and integrated the A4988 drivers into our system, ensuring that they could handle the precise control requirements of our project. This involved setting microstepping resolutions, current limits, and other parameters to optimize motor performance. My expertise in managing these motors contributed to the machine's ability to execute intricate symbol designs with smooth and consistent movement.

```
#include <ContinuousStepper.h>

//limit detection
#define IRSensor_count 9
short IR_count = 0;
int IR_state = 0;
int lastIR_state = 0;

// steper moters
const uint8_t step1Pin = 0;
const uint8_t dir1Pin = 1;
const uint8_t step2Pin = 12;
const uint8_t dir2Pin = 11;

ContinuousStepper stepper1;
ContinuousStepper stepper2;

void setup() {
  //IR sensor
  pinMode(IRSensor_count, INPUT);

 //nema17
```

```
  stepper1.begin(step1Pin, dir1Pin);
  stepper2.begin(step2Pin, dir2Pin);
}

void loop() {
  checkIR();
}

void checkIR(){
  IR_state = digitalRead(IRSensor_count);

  if(IR_state!=lastIR_state){
    if(IR_state == LOW){
      move2feet();
    }
  }

  lastIR_state = IR_state;
  delay(50);
}

void move2feet(){
  int s = -10; //speed (s) : keep at -10
  int t = 30; //move time (t) in seconds | 10s = 6 inch

  //move forward
  unsigned long moveTime = millis()+(t*1000);
  stepper1.spin(s);
  stepper2.spin(s);

  while(millis()<moveTime){
    stepper1.loop();
    stepper2.loop();
  }

  delay(1000);
}
```

3.) Dissanayake D.M.K.   214059G

My responsibilities encompassed the management of the paint supply system, ensuring a steady and reliable flow of paint for the symbol drawing process.

My primary role revolved around the management of ink containers and the delivery of paint to the machine's brushes. A noteworthy aspect of this task was the implementation of an optimized paint delivery schedule. Instead of continuous operation, the pump motors were programmed to work in cycles, activating once every 10 minutes. I meticulously configured and synchronized the pump motors with our system to ensure that they were activated precisely at the designated intervals.

To facilitate the controlled operation of the pump motors within the designated time intervals, I implemented relay modules into our system. Relay modules served as the intermediary between the microcontroller and the pump motors, allowing for safe and efficient switching of power at the predetermined intervals. I programmed the relay modules to respond to commands from the microcontroller, ensuring that the pump motors activated and deactivated according to the established schedule.

```
//ink pumping

#define pumpPin_1 7

#define pumpPin_2 8

// PUMP Interval in milli seconds

long interval_duration = 600000L; //every 5min

// Variables to store the time values for pumping

long previous = 0L;

long current = 0L;

// Flags to track the state of the relays

bool pump1_on = false;

bool pump2_on = false;

void setup() {

//Ink Pumps

  pinMode(pumpPin_1,OUTPUT);

  pinMode(pumpPin_2,OUTPUT);
```

```
  digitalWrite(pumpPin_1,HIGH);
  digitalWrite(pumpPin_2,HIGH);
}
void loop() {
 pumpInk();
}
void pumpInk(){
  int pumpTime = 2000; //pump ON time in miliseconds
  current = millis();
  if((current-previous)>=interval_duration){
    digitalWrite(pumpPin_1,LOW);
    digitalWrite(pumpPin_2,LOW);
    delay(pumpTime);
    digitalWrite(pumpPin_1,HIGH);
    digitalWrite(pumpPin_2,HIGH);
    previous = current;
  }
}
```

4.) Sadhunhari W.P.D

A cornerstone of our project's success lay in the seamless integration of an IR sensor system. My role involved the careful selection and placement of IR sensors to detect specific events during the symbol drawing process. One of the key functionalities was the detection of symbol completion within a single frame. When the IR sensor received the signal indicating that the symbol had been drawn within the frame, it played a pivotal role in initiating the machine's forward movement to complete the remaining portion of the symbol. I meticulously calibrated and programmed the IR sensors to respond with precision, ensuring that the machine's actions were synchronized with symbol drawing requirements.

Another critical aspect of my contribution was the management of the Z-axis, specifically its movement to switch between two color paintbrushes. To achieve this, I integrated servo motors into our system, precisely controlling the vertical movement of the paintbrushes. When transitioning between two color brushes, the servo motors played a key role in lifting and lowering the brushes.

The code I designed and implemented facilitated communication between the microcontroller and these components, ensuring that they responded accurately and promptly to changing conditions during the symbol drawing process.

```
const int irSensorPin = 2;  // Pin connected to the IR sensor
int count = 0;            // Press count

void setup() {
  pinMode(irSensorPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  // Read the state of the IR sensor
  int irSensorState = digitalRead(irSensorPin);

  // Check if the sensor detects an object (LOW level)
  if (irSensorState == LOW) {
    // Increment the count
    count++;

    // Print the count to the serial monitor
    Serial.print("Object detected! Count: ");
    Serial.println(count);
```

```
 // Delay to avoid counting multiple times for a single object
 delay(1000);

   if (count == 2) {
 // Move the motor 2 feet ahead
Serial.print("Move the Simba ");
   delay(5);  // Delay between steps (you may need to adjust this value)
 }
}

}
```

5.) <u>Yasundara G.A.O 214241H</u>

In the collaborative effort to develop our automated floor symbol drawing machine, my role revolved around two essential aspects: detecting the ink level using ultrasonic sensors and managing the G-code operation through Inkscape software.

A key component of our project's success was the accurate monitoring of ink levels in the containers. To achieve this, I was entrusted with the task of integrating ultrasonic sensors into our system and programming them to provide real-time feedback on ink levels. These ultrasonic sensors were strategically positioned within the paint containers, enabling them to measure the remaining paint volume accurately. My role extended to the precise calibration of the ultrasonic sensors to account for container size and shape variations. The innovation in this system lay in the method used to detect ink levels: the ultrasonic sensors measured the decreasing amount of ink by timing how long it took for the ultrasonic signal to bounce back from the ink layer's surface. When the ink levels approached a predefined threshold, the ultrasonic sensors triggered alerts and actions. An essential feature of this system was the activation of an LED indicator. This LED served as a visual signal to alert users and operators that the paint level was decreasing.

I played a vital role in the generation and management of G-code instructions for symbol designs. I leveraged Inkscape software, a versatile tool known for its vector graphics capabilities, to design intricate symbols and convert them into G-code. My expertise in Inkscape allowed us to offer users the flexibility to create custom symbol designs effortlessly.

```
//ink level detection
#define trigPin_1 3
#define echoPin_1 2
#define buzzerPin 6
#define trigPin_2 5
#define echoPin_2 4
#define LEDpin_1 A3
#define LEDpin_2 A4
long previousInk = 0L;
long currentInk = 0L;
long intervalInk = 1000L;


void setup() {
  //UltraSonic
  pinMode(trigPin_1, OUTPUT);
  pinMode(echoPin_1, INPUT);
  pinMode(trigPin_2, OUTPUT);
  pinMode(echoPin_2, INPUT);
```

```
  //LEDs
  pinMode(LEDpin_1, OUTPUT);
  pinMode(LEDpin_2, OUTPUT);
  digitalWrite(LEDpin_1,LOW);
  digitalWrite(LEDpin_2,LOW);

  // put your setup code here, to run once:

}

void loop() {
  inkLevel();
  // put your main code here, to run repeatedly:

}

//ink level
detection====================================================================>
void inkLevel(){
  currentInk = millis();

  if((currentInk-previousInk)>=intervalInk){

    long duration1, distance1;
    digitalWrite(trigPin_1, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin_1, HIGH);
    delayMicroseconds(4);
    digitalWrite(trigPin_1, LOW);
    duration1 = pulseIn(echoPin_1, HIGH);
    distance1 = (duration1 / 2) / 29.1;

    Serial.print("D1:");
    Serial.print(distance1);

    long duration2, distance2;
    digitalWrite(trigPin_2, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin_2, HIGH);
    delayMicroseconds(4);
    digitalWrite(trigPin_2, LOW);
    duration2 = pulseIn(echoPin_2, HIGH);
    distance2 = (duration2 / 2) / 29.1;
```

```
  Serial.print(" D2:");
  Serial.println(distance2);

  int inkLevel = 5;

  if (distance1 < inkLevel) {
   digitalWrite(buzzerPin,LOW);
   digitalWrite(LEDpin_1,LOW);
  }else{
   //not optimal conditions
   digitalWrite(buzzerPin, HIGH);
   digitalWrite(LEDpin_1,HIGH);
  }

  if (distance2 < inkLevel) {
   digitalWrite(buzzerPin,LOW);
   digitalWrite(LEDpin_2,LOW);
  }else{
   //not optimal conditions
   digitalWrite(buzzerPin, HIGH);
   digitalWrite(LEDpin_2,HIGH);
  }
   previousInk = currentInk;
  }
}
```