AN-NAJAH NATIONAL UNIVERSITY

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY
COMPUTER ENGINEERING DEPARTMENT

Graduation Project II

Window cleaner machine

Done by:
Nada Abu Hantash
Seema Ihbesheh

Supervisor:
Dr.Luai Malhis

Presented in partial fulfillment of the requirements for a Bachelor's degree in Computer Engineering.

jan-2023

Acknowledgment

First, we would like to thank Dr. Luai Malhis a lot for his helpful, kind, patient, and taking care of us, and for making everything simple. he was always inspiring and encouraging us to move, we are feeling proud of being one of its students helps us to improve our educational level as well as improve our skills. Secondly, we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

Disclaimer Statement

This report was written by Nada Abu Hantash and Seema Ihbisheh at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, because of the assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was Commissioned.

Contents

List of Figures

1 Abstract

With the increase in the height of the buildings and their numbers, they became It important to have safe processes for cleaning windows and panes to reduce the risk of climbing tall buildings.

The solution is to create a machine that cleans building glass by using a circular brush suspended by a rail that helps it move horizontally and rollers that help it move vertically. Of course, all of this can be done automatically or via a Wi-Fi website. To enable manual control of this feature, we will also use a camera and a water pump. To ensure that the brush moves inside the glass frame, a set of ultrasonic sensors are also present. As a result, we reduce the risk of death for personnel responsible for cleaning high glazing.

The alternative available on the market is like a robot that moves too by wheels or double-sided magnets

2 Chapter 1: Introduction

2.1 Statement of the problem

While residential and one-story commercial buildings are good for do-it-yourself glass cleaning, high-rise buildings are not. High-rise glass cleaning comes with numerous safety concerns for both cleaners and pedestrians below. Working hundreds of feet in the air is not for the faint-hearted and requires extensive training and experience.

One of the most significant risks for high-rise glass washers is falling from high heights due to safety equipment failure or strong winds. Also, these glass washers pose safety concerns for pedestrians as falling tools and equipment can hit pedestrians and cause catastrophic consequences.

2.2 Objectives of the work

objective This project focuses on cleaning high-performance glass panels. The project is a cleaning brush that moves horizontally and vertically to clean the glass pane, by spraying water and moving the brush in a circular motion. To increase work efficiency, the camera was used for live broadcasting on the website to control the movement of the brushes remotely.

2.3 Significance of the work

This has the advantage that the glass can be cleaned easily and safely without the use of detergents or cranes, and it can also be cleaned remotely using the website, all to maintain an occupational health and safety system.

2.4 Organization of the report

In this report, we illustrate the idea in full detail. Going through the report you can first see the constraints we faced during our work including equipment, tools we used, and earlier coursework. Then, you can read about similar systems to get a background about the topic being discussed and what special features we have done on other systems. After that, the methodology of our work is extensively explained. The next chapter states our results and a discussion to interpret and compare the results. Ending it up with the conclusion of the whole work and what is our vision for the future to improve our work.

3 Chapter 2: Constraints, Standards/ Codes, and Earlier coursework

3.1 Constraints

The biggest difficulty in the project was dealing with the mechanical part because we, as computer students had never installed mechanical parts with this complexity, in addition to the lack of availability in the shops, and where we needed a design that could bear the weight on it and at the same time be able to move horizontally without an imbalance in the brush because of the high railway prices, we had to change the design to wooden to suit the movement.

3.2 Standards/Codes

We need external libraries to implement our project, the Stepper.h, AccelStepper.h to control the steppers motors, also we add ezButton.h to control the micro limit switch button the finals libraries which we add are esp$_{c}amera.hand < WiFi.h > tocontrolespcamera$

3.3 Earlier coursework

Working on our project depends on some courses we learned within the Computer Engineering programs such as:

Microcontroller: The microcontroller provides basic information about understanding the PIC Microcontroller and how to program hardware components, also, the lab of this course provides how-to download code on the PIC Microcontroller equipment and how to understand every pin and feature there. so, it is one of the most important materials that helped us understand how to deal with the Arduino Mega parts in the project, through our knowledge of how to deal with the microcontroller, as the laboratory of this material, contributed mainly to help us start working on the project.

Electronic circuits: This course has mainly contributed to helping us for dealing with electrical circuits and related connections, as a result of this course that provides basic information about how to deal with many different circuits and how to wire their circuits.

4 Chapter 3: Literature review

Skyscrapers form a majestic and essential part of many urban skylines. As the urban population of the world continues to grow, skyscrapers will be necessary to maximize vertical space.

In 1985, two billion people worldwide lived in cities. Today, that number has reached four billion, and projections suggest that by 2050 six billion people will have become urban residents. Unsurprisingly, projections also indicate that by 2050, the number of skyscrapers in the world will increase by a tremendous 41,000 buildings.

With their walls of sparkling glass windows, skyscrapers give the skyline an elegant look — but those windows don't stay sparkling on their own. Rain, snow, smog, salt spray, dust, debris, bird droppings, and more conspire to obscure a skyscraper's windows. Fortunately, glass cleaners are on the job to get them pristine and shiny once more.

Glass cleaning in the Middle East region is something totally different from other parts of the world as the cleaners have to face rough weather under a scorching sun. In addition to this, they have to cope with wind waves, while hanging in the air with the support of suspended ropes tied from the roof of Skyliners. glass cleaning methods vary right from Skyliners to normal commercial and institutional premises as size matters a lot in this segment.

Standing on a small platform – or dangling from a rope – hundreds of feet above the ground, removing dirt and grime from elevated glazing is probably not your idea of a normal day at work. But for some brave souls, high-rise glass cleaning is just another day at the office. As an essential part of property maintenance, it helps keep our skylines and our skyscrapers looking their very best.

5 Chapter 4: Methodology

In light of what we have discussed above, it is evident that the project has significance and advantages over traditional methods. In this project, we combined both software and hardware to control the machine. we control this machine in two ways:

- Auto
- Manual

we will talk about what components we are using in our project.

5.1 Hardware

- Arduino Mega 2560

  The following list contains the hardware components that have been used in our project: Arduino Mega 2560: The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the

  microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most

  shields designed for the Uno and the former boards Duemilanove or Diecimila. The Mega 2560 is an update to the Arduino Mega, which it replaces.

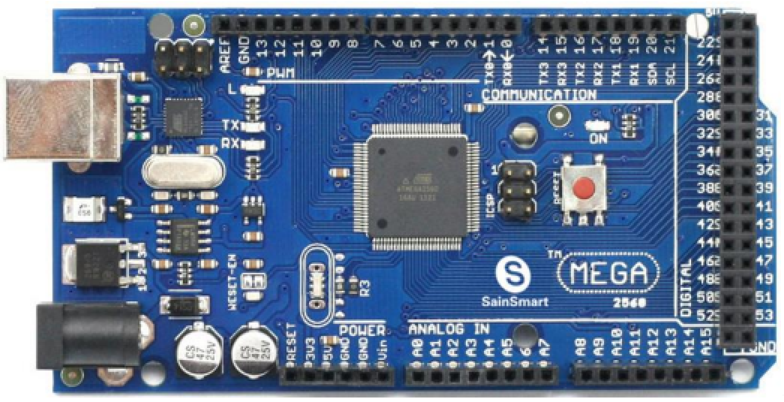

Figure 1: Figure 1: Arduino mega

- Stepper motors:

  A stepper motor is an electromechanical device it converts electrical power into mechanical power. How do our stepper motors work? Stepper motors use a cogged wheel and electro-magnets to rotate the wheel one 'step' at a time. Each HIGH pulse sent energizes the coil, attracting the teeth closest to the cogged wheel and driving the motor one step forward. The way you pulse these coils greatly affects the behavior of the motor: 1-The sequence of pulses determines the spinning direction of the motor. 2-The frequency of the pulses determines the speed of the motor. 3-The number of pulses determines how far the motor will turn.

  Why did we choose a stepper motor for our project? stepper motors offer excellent speed control, precise positioning, and repeatability of movement which we need in our project. We use 2 stepper motors in our project :

  one motor for y-axis

  one motor for x-axis

- Esp 32 Cam module controller

  The ESP32 CAM is basically an ESP32 without a CP2102 Chip. Instead, it is equipped with a 2MP OV2640 camera module. It has integrated Wi-Fi, so we used it to generate a web page that contains a live stream from its camera besides buttons to control the brush movement.

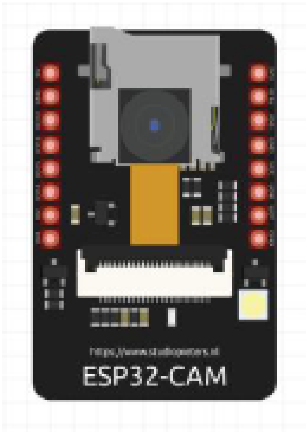

Figure 2: Figure 2:Esp 32 Cam module controller

- DC Motor

  A DC motor is used to move the brush in a circular motion, we use DC motors that require 12 volts to operate. It has been connected to the L298N to control its speed.

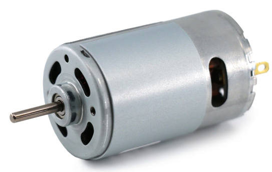

Figure 3: Figure 3: DC motor

- water pump pipe

  The water pump is an important part of the robot as it will pump water to clean the window. In this project, a small-size and lightweight water pump were used with a L298N motor driver module that takes its power from H-bridge. A pipe for spraying the facade with water pumped from that water pump.

Figure 4: Figure 4: water pump pipe

- Ultrasonic sensor

  It is a sensor for the distances through which the ends of the glass plate are located from the bottom. It has four output pins: voltage (VCC) (operate around 5V), ground pin (GND), digital output (DO), and analog output (AO).

Figure 5: Figure 5: Ultrasonic sensor

- micro limit switch button

  Switches act as a mechanism that tells the computer the limits of the machine. When the horizontal axis reaches the axis limit, the switch is activated and the motor that controls the vertical movement is activated, and then the direction of the horizontal movement is reversed.

Figure 6: Figure 6: Micro limit switch button

- soil monitor sensor

  It is a sensor for the height of the water in the water tank. When the tank becomes not full of water, it sends a notification and the system stops working until it is filled.

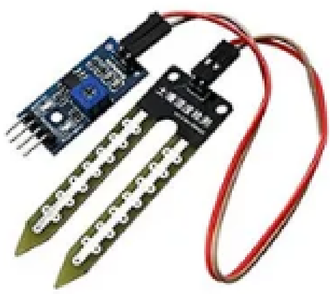

Figure 7: Figure 7: soil monitor sensor

- L298N Motor Driver Module

  is an electronic circuit that switches the polarity of a voltage applied to a load. These circuits are often used in robotics and other applications to allow DC motors to run forwards or backwards.

Figure 8: Figure 8: L298N Motor Driver Module

- Brush: It is an important part of our project, as it cleans the surface with its bristles, which are connected to the DC motor, and are controlled by it.

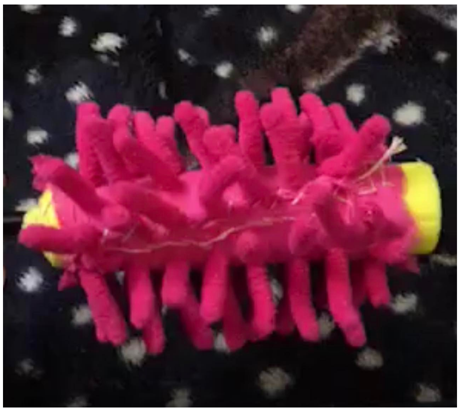

Figure 9: Figure 9: Brush

- 12 6 volts battery Connecting wires.

  They were used to complete work and delivery

5.2 Mechanical Parts:

Material selection:

- **STRUCTURER**

  -Aluminum:

  Advantages: lightweightt, stiff.

  Disadvantages: expensive, 35 per meter, kind of hard to form the structure, so we use wood instead.

  -Wood :

  Advantages: cheap, easy to form, available everywhere.

  Disadvantages: Light in weight.

- **AXES MECHANISM**

  X-Axis: Move laterally by GT2 Belt.

Figure 10: Figure 10: GT2 Belt spur gear

Y-Axis: Move vertically by pulley and rope, Created with a 3D printer

Figure 11: Figure 11: pulley

STRUCTURE:

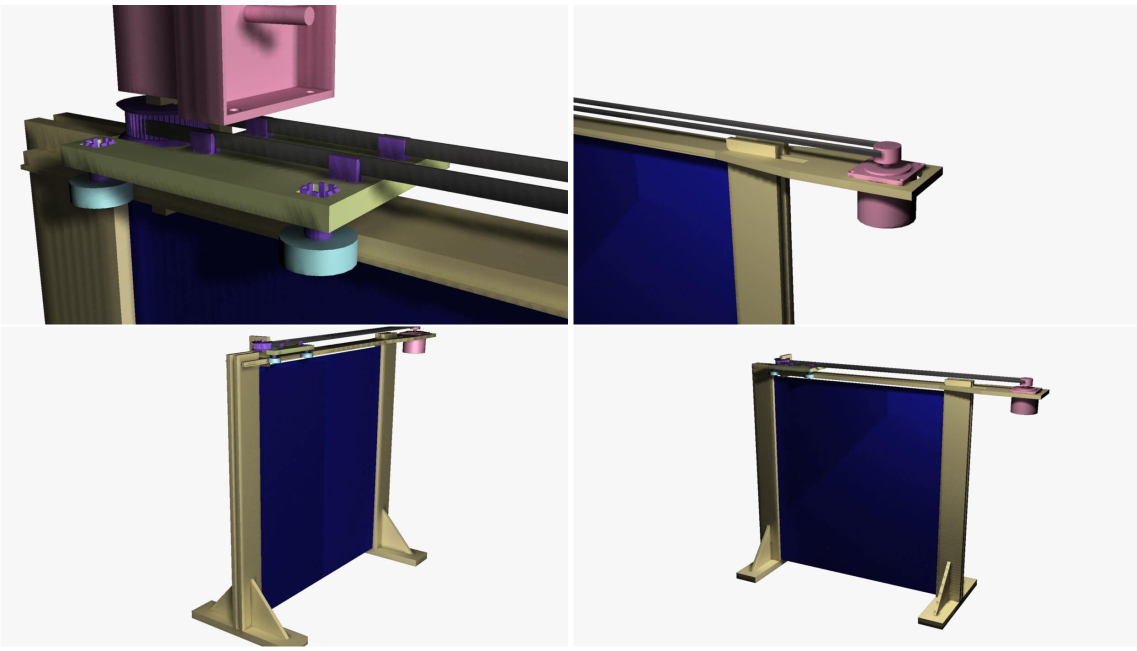

Figure 12: Figure 12: structure for the project

5.3 Programs

- We used **Arduino IDEs** for programming the microcontrollers and all used hardware.
- **circuito.io** to design the circuits.

5.4 Block Diagram

The figure below shows the block diagram with a general view of the project's parts, in our project we use Arduino Mega, source 12V for motors Arduino: In general, there are pictures of each case

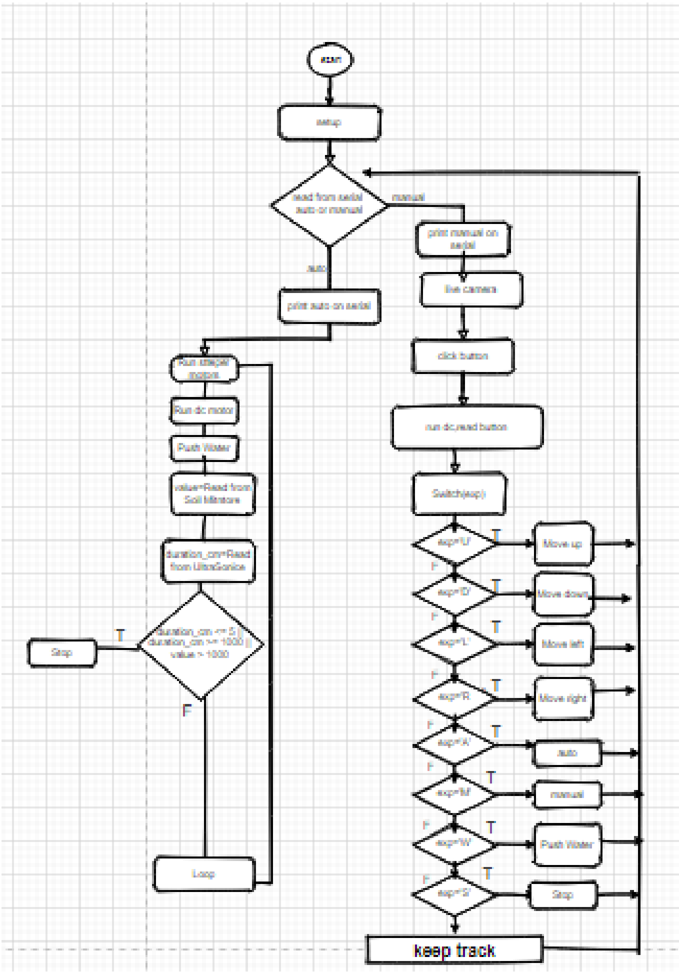

Figure 13: Figure 13: Block Diagram

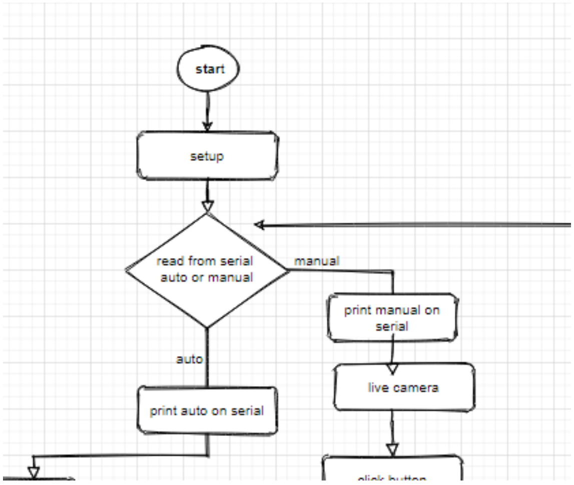

Figure 14: Figure 14: read from serial

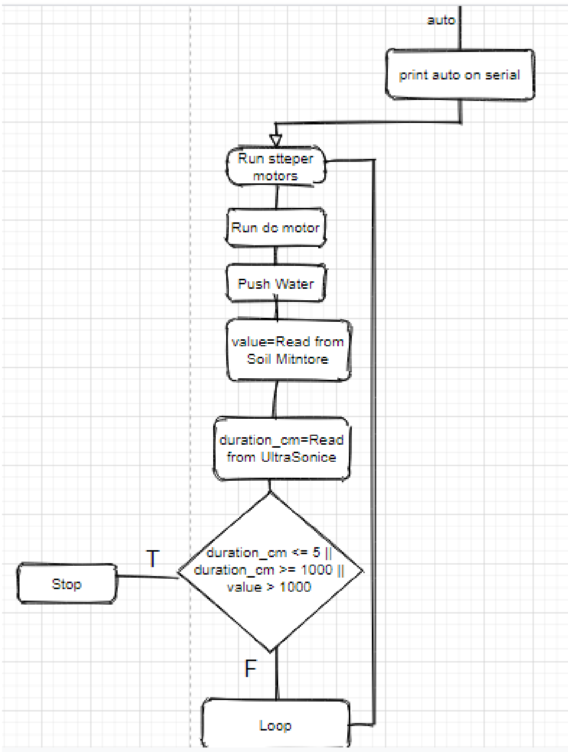

Figure 15: Figure 15: Auto Mode

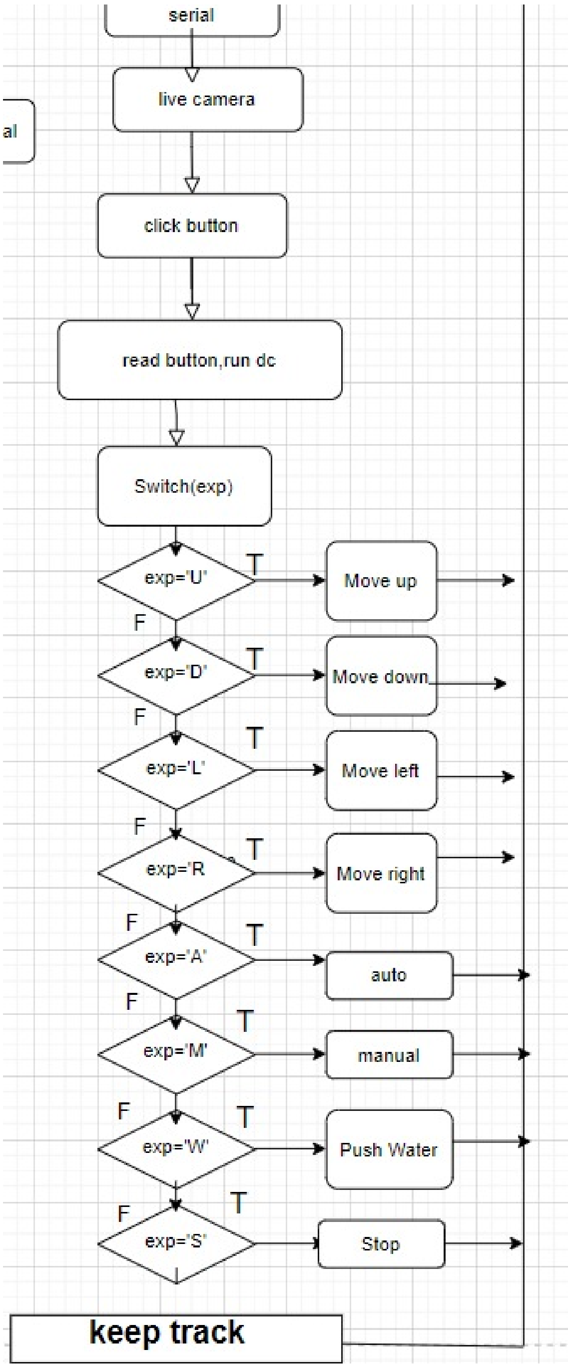

Figure 16: Figure 16: Manual Mode

5.5 Circuit Connection

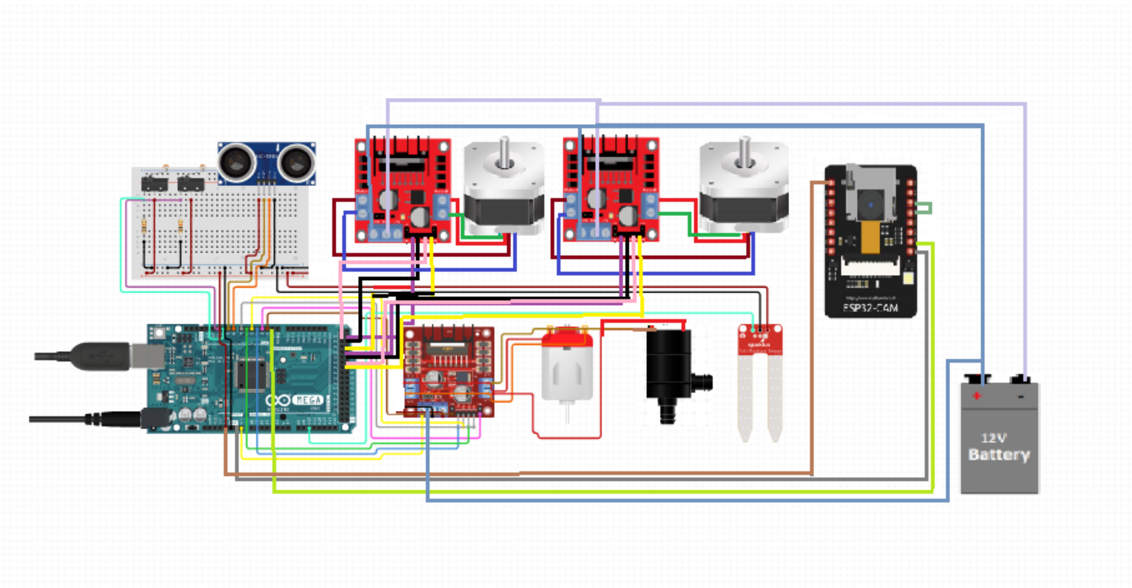

Figure 17: Figure 17: Circuit Connection

5.6 Overall Design

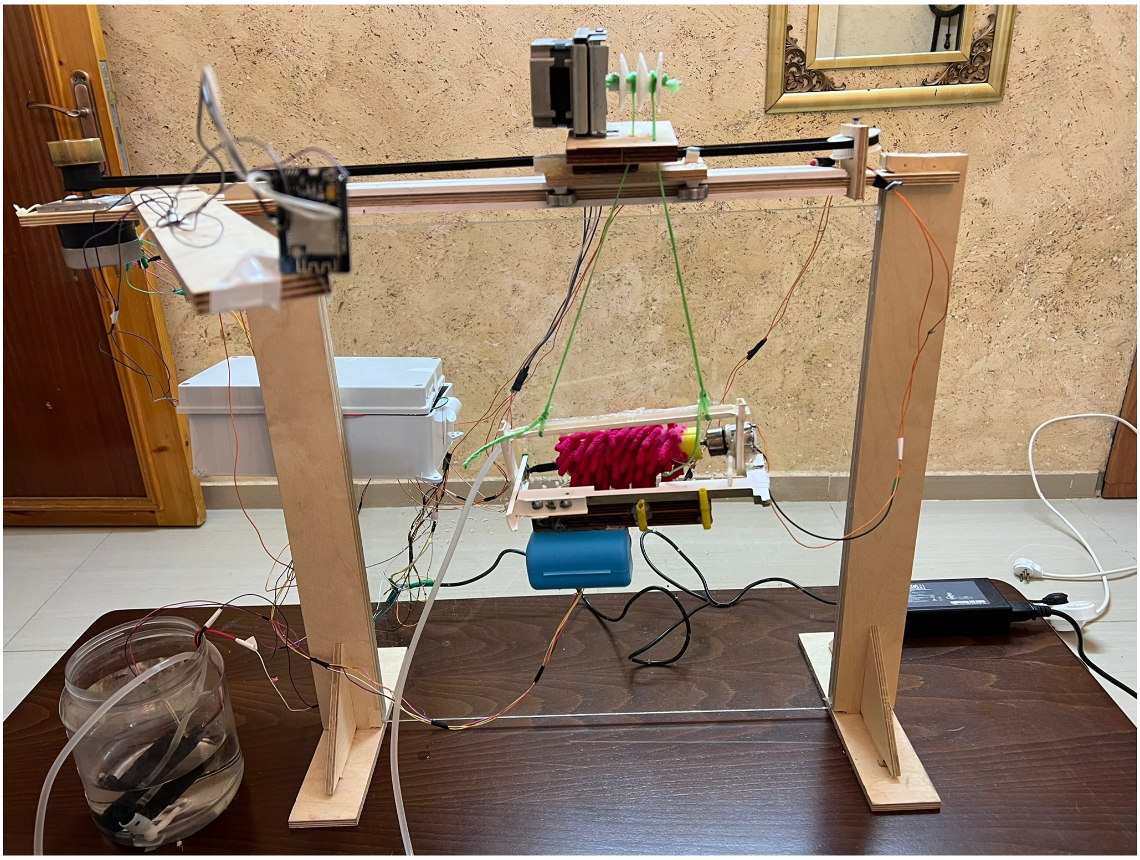

Figure 18: Figure 18: Design from front

Figure 19: Figure 19: Design from above

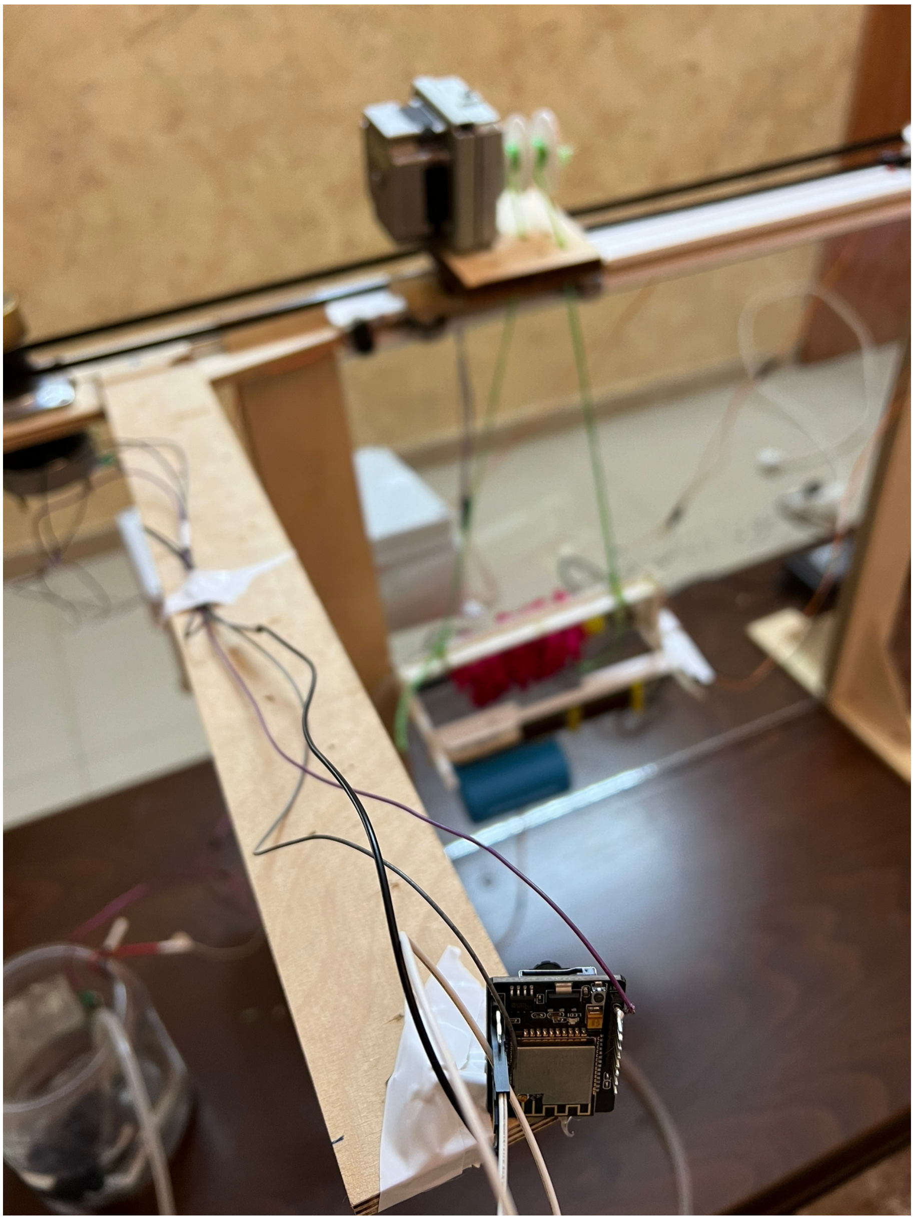

Figure 20: Figure 20: camera angle

6 Chapter 5: Software Implementation

All the following codes make sense when we look at the setup function and global variables in the Arduino code:

```
const int stepsPerRevolution6 = 130;
const int stepsPerRevolution8 = 100;
Stepper myStepper6(stepsPerRevolution6, 4 , 5 , 6, 7);
Stepper myStepper8(stepsPerRevolution8, 8 , 9 , 10, 11);

ezButton limitSwitch_1(A0); // create ezButton object that attach to pin A0;
ezButton limitSwitch_2(A1); // create ezButton object that attach to pin A1;

AccelStepper stepper(AccelStepper::HALF4WIRE, 8 , 9, 10, 11);
int stepperState = STATE_MOVE;
int direction    = DIRECTION_CW;
long targetPos   = 0;

///DC
const int ENB_PIN = 53; // the Arduino pin connected to the EN2 pin L298N
const int IN3_PIN = 52; // the Arduino pin connected to the IN3 pin L298N
const int IN4_PIN = 50; // the Arduino pin connected to the IN4 pin L298N

//soil
int value;
#define AOUT_PIN A2 // ESP32 pin GIOP36 (ADC0) that connects to AOUT pin
//of moisture sensor
const int ENA_PIN = 48; // the Arduino pin connected to the EN1 pin L298N
const int IN1_PIN = 46; // the Arduino pin connected to the IN1 pin L298N
const int IN2_PIN = 44; // the Arduino pin connected to the IN2 pin L298N
//water
int speedwater = 90;
```

```
boolean autoMovee = false;
char Value_wifi = 0;
char Old_Value_wifi = 1;


////our code seeemma nadaa
#include <Stepper.h>
////ultra
#define TRIG_PIN 21
#define ECHO_PIN 20
float duration_us;
float duration_cm;
///stper
#include <ezButton.h>
#include <AccelStepper.h>

#define DIRECTION_CCW -1
#define DIRECTION_CW   1

#define STATE_CHANGE_DIR 1
#define STATE_MOVE       2
#define STATE_MOVING     3

#define STATE_MOVE_M       1
#define stopit      2

#define MAX_POSITION 0x7FFFFFFF
```

Figure 21: Figure 21: define variables

At the beginning, the website will open and choose the resource that you want to run

```
void setup() {

  Serial3.begin(115200);
  Serial.begin(9600);
  myStepper6.setSpeed(50);
  myStepper8.setSpeed(40);
  /// ultra
  pinMode(ECHO_PIN, INPUT);
  pinMode(TRIG_PIN, OUTPUT);
///water
  pinMode(ENA_PIN, OUTPUT);
  pinMode(IN1_PIN, OUTPUT);
  pinMode(IN2_PIN, OUTPUT);
///button
  limitSwitch_1.setDebounceTime(50);
  limitSwitch_2.setDebounceTime(50);

  stepper.setMaxSpeed(500.0);   // set the maximum speed
  stepper.setAcceleration(50.0); // set acceleration
  stepper.setSpeed(100);         // set initial speed
  stepper.setCurrentPosition(0); // set position
  //////////////

  //  DC motor initialize digital pins as outputs.
  pinMode(ENB_PIN, OUTPUT);
  pinMode(IN3_PIN, OUTPUT);
  pinMode(IN4_PIN, OUTPUT);

}
```

Figure 22: Figure 22: setup function

```
void loop() {

while (!autoMovee) {
    if (Serial3.available() > 0) {
      Value_wifi = Serial3.read();
      Serial.println(Value_wifi);
      if ( Value_wifi != Old_Value_wifi ) {
        Old_Value_wifi = Value_wifi;
        DoSomethingWith(Value_wifi);
      }
    }
  }//while

  while (autoMovee) {
    if (Serial3.available() > 0) {
      Value_wifi = Serial3.read();
      Serial.println(Value_wifi);
      if ( Value_wifi == 'M') {
        autoMovee = false;
      }
    }
    autoMove();

  }


}//end loop
```

Figure 23: Figure 23: select mode

```
void DoSomethingWith(char input) {
  switch (input) {
    case 'U':
      Up_stteper6();
      break;
    case 'B':
      Down_stteper6();
      break;
    case 'L':
      left_step();
      break;
    case 'R':
      right_step();
      break;
    case 'S':
      Stop();
      break;
    case 'A':
      autoMovee = true;
      break;
    case 'M':
      autoMovee = false;
      break;
    case 'W':
      water()
      break;
    case 'P':
      dcmove();
      break;
    default : break;}}
```

Figure 24: Figure 24: Manual mode

The rest of the code is in the Appendices

7 Chapter 6: Results and Discussion

Working on this project was not without its obstacles, as we did not have any previous knowledge of Arduino, we were able to create a glass cleaner machine with all the features we had planned,after building a complete machine from scratch, we gained enough experience with Arduino , it contains features that will allow us to expand in this field and work more with Arduino. Direct discussions of the project will yield clear results.

8 Chapter 7: Conclusion

Conclusion By building the glass cleaning machine ourselves, we gained an unparalleled amount of experience:

- An understanding of various hardware components, including Arduino Mega, stepper motors, limit switches, and l289N driver.
- We learn how to deal with mechanical part as screws, pulleys, and belts.
- We learned how to control the water level in the water tank by soil monitor.
- We learned how to handle two microcontrollers, namely Arduino Esp 32 Cam.
- We learned some of the mechanics that helped us structure the project.

9 Chapter 8: Future Work

Future Work In spite of the fact that we have built a complete glass cleaning system that differs from the traditional one and has several features, As our machine is highly flexible and easy to add features to, the work can be further expanded in the future. Improved features are taken into consideration, such as :

- Add sound at the start and end of clean.
- Show metaData about the process by LCD such as height
- Add AI
- Adding another rail to cover the Z axis

10 References

- circuito.io. (n.d.). Retrieved from https://www.circuito.io/app?components=512,11061
- control-pump. (n.d.). Retrieved from arduinogetstarted: https://arduinogetstarted.com/tutorials/arduino-controls- pump
- MisterBotBreak. (2019). how-to-use-a-soil-moisture-sensor. Retrieved from projecthub: https://create.arduino.cc/projecthub/MisterBotBreak/how-to-use-a-soil- moisture-sensor- ce769b
- Jabbaar, A. A. (2019). Ultrasonic Sensor HC-SR04 with Arduino. Retrieved from projecthub: https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc- sr04-with-arduino- tutorial-327ff6
- Arduino Mega 2560 : https://docs.arduino.cc/hacking/hardware/PinMapping2560
- LastMinuteEngineers (2022) Control Stepper Motor with A4988 Driver module and Arduino .Retrieved from In-Depth: Control Stepper Motor with A4988 Driver Module Arduino (lastminuteengineers.com).
- Adi, P. D. P. (2022). Performance evaluation of ESP32 Camera Face Recognition for various projects.
- Yin, L., Wang, F., Han, S., Li, Y., Sun, H., Lu, Q., ... Wang, Q. (2016, October). Application of drive circuit based on L298N in direct current motor speed control system. In Advanced laser manufacturing technology (Vol. 10153, pp. 163-169). SPIE.

11 Appendices

Here are some other functions in auto mode and manual mode

```
void autoMove() {
  /////////////DC
  digitalWrite(IN3_PIN, HIGH); // control motor A spins clockwise
  digitalWrite(IN4_PIN, LOW);  // control motor A spins clockwise

  // for (int speed = 0; speed <= 255; speed++) {
  analogWrite(ENB_PIN, 225); // control the speed
  speedwater = 130;
  analogWrite(ENA_PIN, speedwater);
  digitalWrite(IN1_PIN, HIGH); // control motor A spins clockwise
  digitalWrite(IN2_PIN, LOW);  // control motor A spins clockwise
  limitSwitch_1.loop(); // MUST call the loop() function first
  limitSwitch_2.loop(); // MUST call the loop() function first

  if (limitSwitch_1.isPressed()) {
    stepperState = STATE_CHANGE_DIR;
    //Serial.println(F("The limit switch 1: TOUCHED"));
  }

  if (limitSwitch_2.isPressed()) {
    stepperState = STATE_CHANGE_DIR;
    //  Serial.println(F("The limit switch 2: TOUCHED"));
  }
```

Figure 25: Figure 25: Auto mode

```
switch (stepperState) {
  case STATE_CHANGE_DIR:
    direction *= -1; // change direction
    // Serial.print(F("The direction -> "));
    if (direction == DIRECTION_CW) {}else {}
    speedwater = 0;
    analogWrite(ENA_PIN, speedwater);
    digitalWrite(IN1_PIN, LOW); // control motor A spins clockwise
    digitalWrite(IN2_PIN, LOW);  // control motor A spins clockwise

    myStepper6.step(stepsPerRevolution6);

    stepperState = STATE_MOVE;
    // after changing direction, go to the next state to move the motor
    break;

  case STATE_MOVE:
    targetPos = direction * MAX_POSITION;
    stepper.setCurrentPosition(0); // set position
    stepper.moveTo(targetPos);

    stepperState = STATE_MOVING;
    // after moving, go to the next state to keep the motor moving infinity
    break;

  case STATE_MOVING:
  // without this state, the move will stop after reaching maximum position
    if (stepper.distanceToGo() == 0) { // if motor moved to the maximum position
      stepper.setCurrentPosition(0);   // reset position to 0
      stepper.moveTo(targetPos); } break; }
```

```
    stepper.run(); // MUST be called in loop() function

    value = analogRead(AOUT_PIN); // read the analog value from sensor
    digitalWrite(TRIG_PIN , LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN , HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN , LOW);
    long duration_us = pulseIn(ECHO_PIN , HIGH);

    duration_cm = duration_us * 0.017;
    Serial.print("distance: ");

    Serial.println(duration_cm);

    if (Serial3.available() > 0) {
      Value_wifi = Serial3.read();
      Serial.println(Value_wifi);
      if ( Value_wifi == 'M') {
        autoMovee = false;
      }
    }
```

Figure 26: Figure 26: Auto mode

```
void dcmove() {
  /////////////DC
  digitalWrite(IN3_PIN, HIGH); // control motor A spins clockwise
  digitalWrite(IN4_PIN, LOW);  // control motor A spins clockwise
  analogWrite(ENB_PIN, 225); // control the speed
}
```

Figure 27: Figure 27: DC Manual mode

```
viod water() {
  speedwater = 140;
  analogWrite(ENA_PIN, speedwater);
  digitalWrite(IN1_PIN, HIGH); // control motor A spins clockwise
  digitalWrite(IN2_PIN, LOW);  // control motor A spins clockwise
}
```

Figure 28: Figure 28: Water Manual mode

```
void Up_stteper6() {
  myStepper6.step(-stepsPerRevolution6);

}

void Down_stteper6() {
  myStepper6.step(stepsPerRevolution6);
}
```

Figure 29: Figure 29: Up Down stepper motor

```
void right_step() {
  myStepper8.step(stepsPerRevolution8);
}

void left_step() {
  myStepper8.step(-stepsPerRevolution8);

}
```

Figure 30: Figure 30: Lift Right stepper motor

```
void Stop() {
  /// water off

  speedwater = 0;
  analogWrite(ENA_PIN, speedwater);
  digitalWrite(IN1_PIN, LOW); // control motor A spins clockwise
  digitalWrite(IN2_PIN, LOW);  // control motor A spins clockwise

  //DRIVER dc
  digitalWrite(IN3_PIN, HIGH); // control motor A spins clockwise
  digitalWrite(IN4_PIN, LOW);  // control motor A spins clockwise
  analogWrite(ENB_PIN, 0); // control the speed\

}
```

Figure 31: Figure 31: stop function