

SWSaver Prototype

3rd Edition

Diyar Hussein

Student of Computer Science and Engineering
at Voronezh Institute of High Technologies

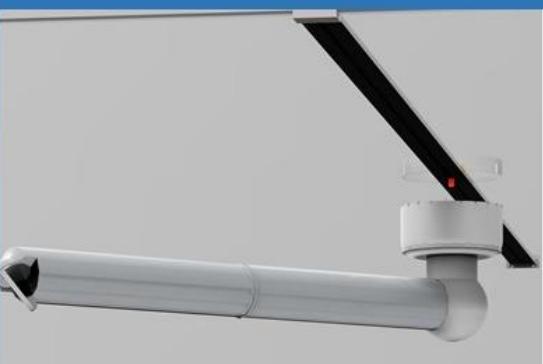
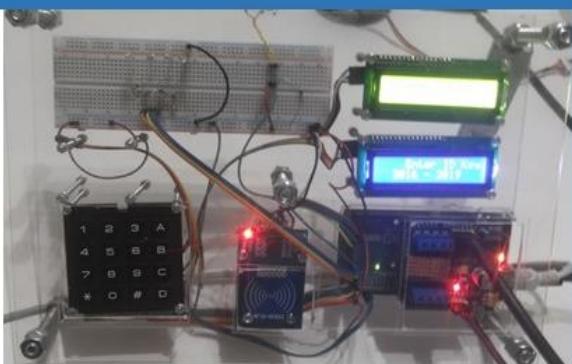


Table of Contents

Introduce	2
Chapter One: Analyzing.....	2
Wardrobe Survey	3
Diagram the model.....	5
Chapter Two: Modeling	7
First part: Basics-Wall mount place	7
Second part: Linear Motion Rail, two axis and, Ball screw	9
Third part: Microcontroller and Monitor	15
Chapter Three: Prototype	16
Hardware:	16
Arduino Library	17
Software:	25
Merging the components.....	33
Source Code:	43
Reference:	72

Introduce

Conclusion prototype for Smart Wardrobe Saver (SWSaver) from studies of personal experiences through different steps start by an accumulation of problems encountered, planning, modeling, surveying, and last but not least , prototype the device. The planning is made with Microsoft office Visio 2019, Diagram of Unified Modeling Language on <https://lucidchart.com>, Model-Based Design with Siemens PLM Community Solid Edge, using Arduino Tool kit for prototype the microcontroller to control Automation devices.

Chapter One: Analyzing

The wardrobe is part of a number of industries using it, and have many types that made it to be comfortable for user, legacy type is most costly due the place will take in premises and monthly payment for employees,



Figure 1 Legacy wardrobe model at theater

Another type is Automation Wardrobe that will assistance to get the clothes without human interaction but is still not efficient because use big space of premises and it will be costly if do customize order.

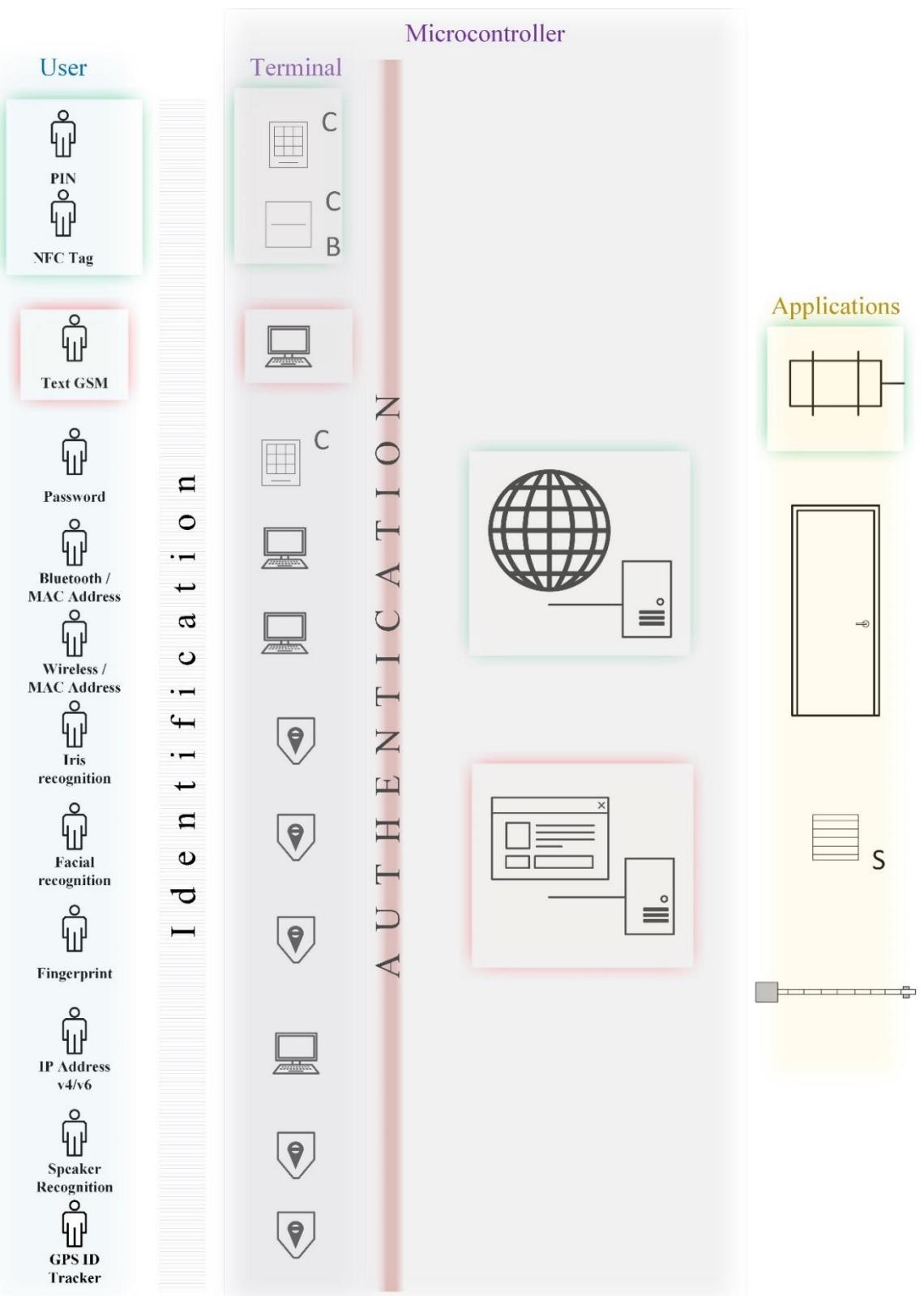


Figure 2 Automation wardrobe

And advantage of these two types are easy installation or maybe inexpensive for short-term but not secure enough to protect the clothes from accident like fire damages and etc., furthermore, are fixed shift or limited access, so, the plan is to changes these some of these legacy operations to be part of modern technology, economic growth for long-term and, availability plus be protected from any accident with covering each coat that is going to hanging on hanger.

Wardrobe Survey

Unfortunately, the data for this part is unavailable for this reason the plans are unstandardized and will focus on technology side in software section and other parts just assembled in home made to be done only for visioning the model how is efficiency and progress the steps, in figure -3- showing the plans that interested to be add part of this application with considering that colored by blue are finished work and light pink are remaining in processing the reason GSM text get failure in SIM800L microprocessor and conflict between pins of ethernet shield and other shield the website for administrative controlling / client viewer status staying incomplete project.



Created by Microsoft Office Visio Professional 2019

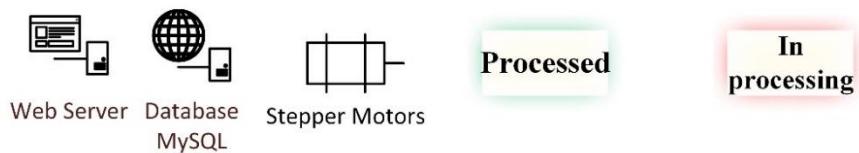


Figure 3 Planning of Smart Wardrobe Saver

Diagram the model

Use Case Diagram for interactive between the user and SWSaver device:

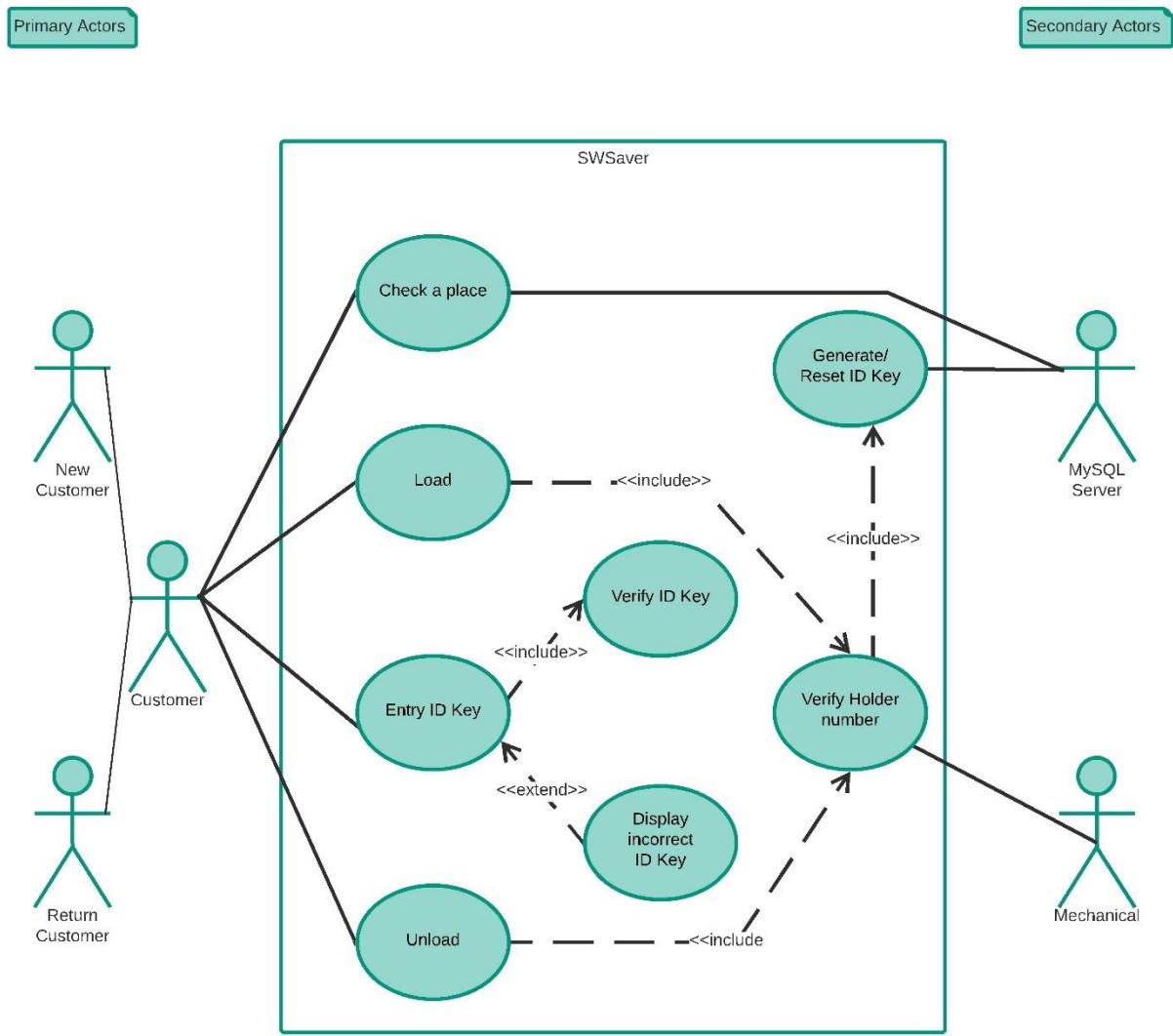


Figure 4 Use Case Diagram of Smart Wardrobe Saver

Additional, Activity Diagram for loading and unloading the coat:

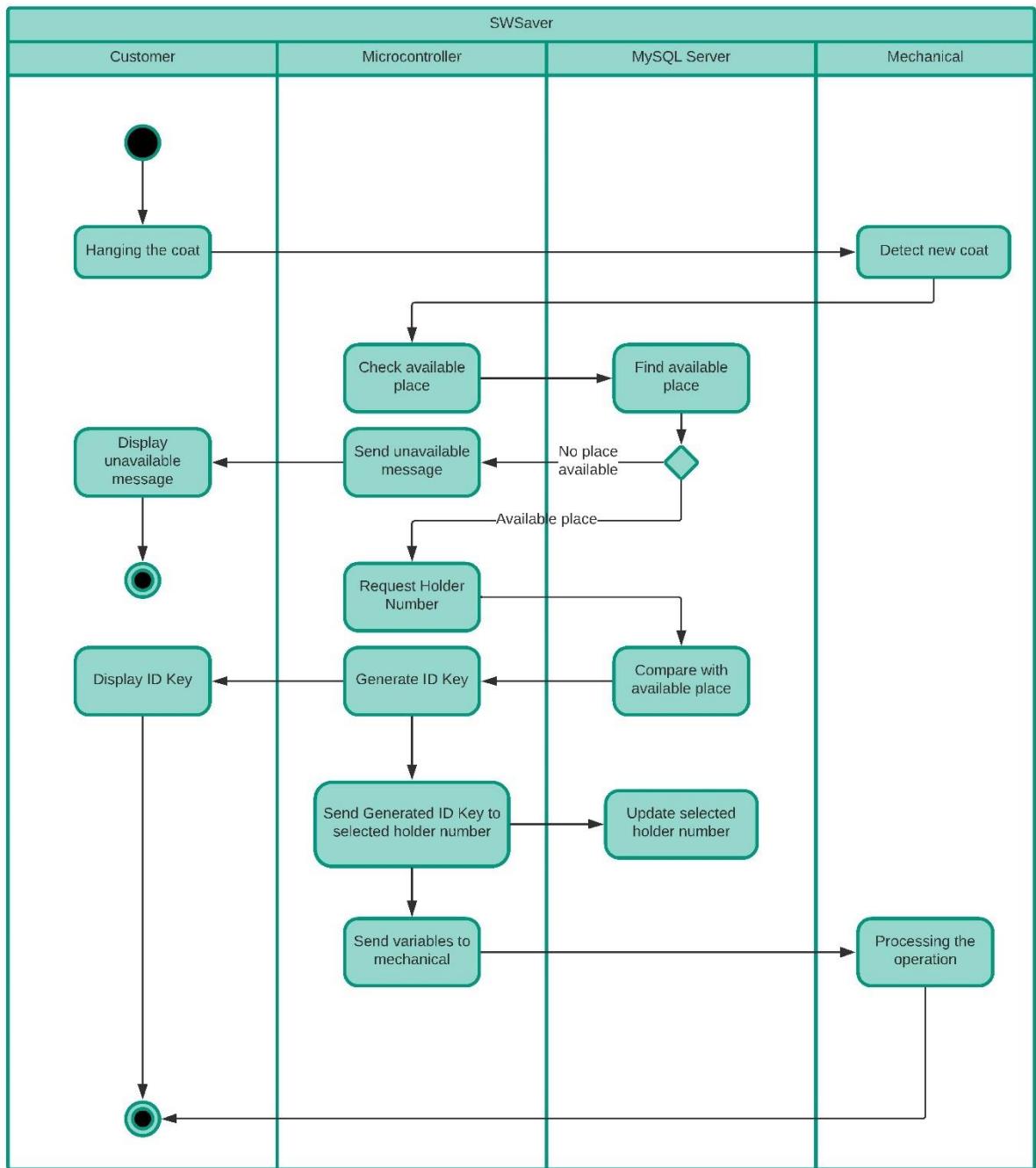


Figure 5 Activity Diagram for loading operation in SWSaver

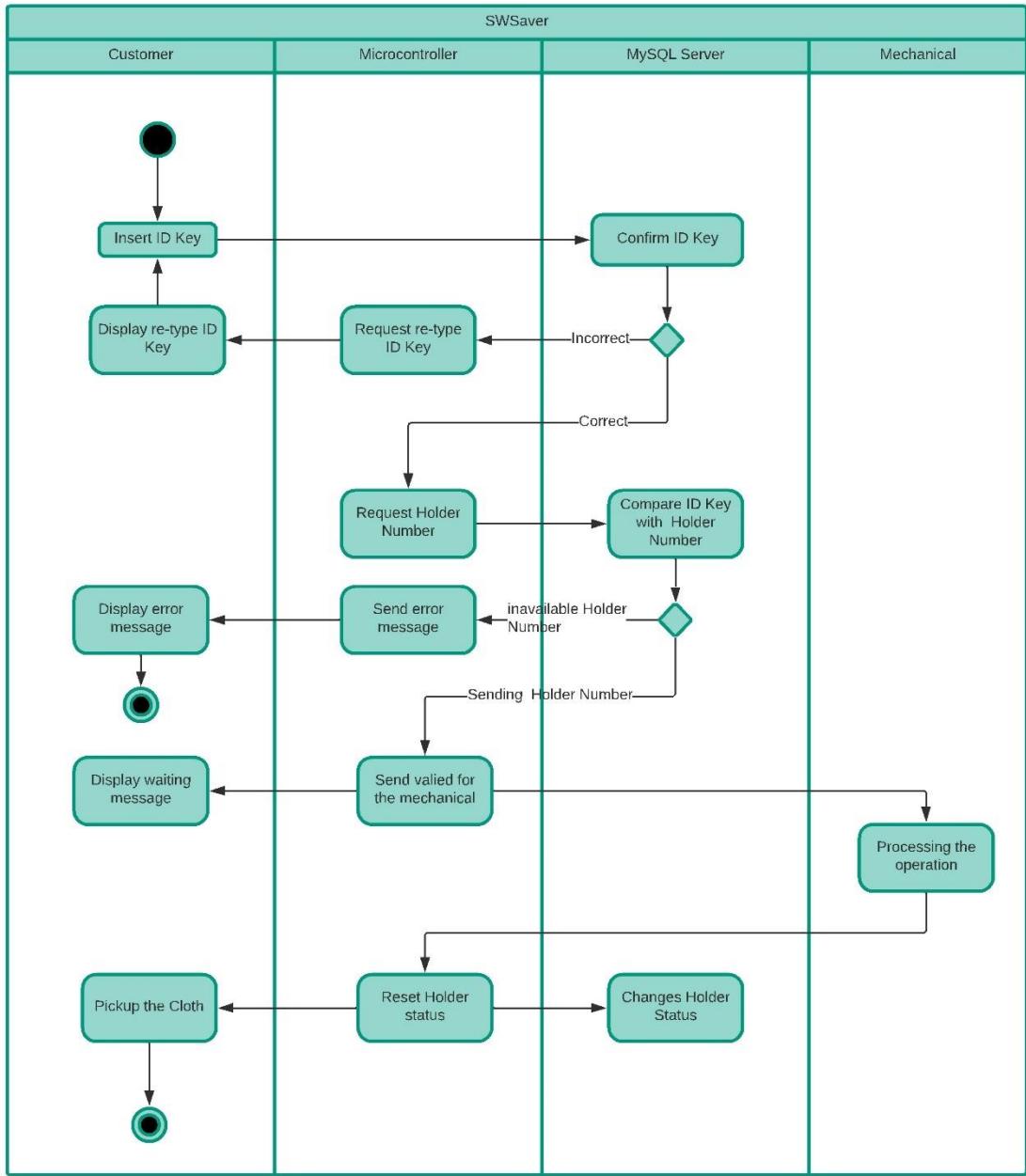


Figure 6 Activity Diagram for unloading operation in SWSaver

Chapter Two: Modeling

Designed the modeling is divided into three main parts:

First part: Basics-Wall mount place

In this part are Basics-Wall mount place included rail for mounting with Hangers holder.



Figure 7 Basics-wall mount place

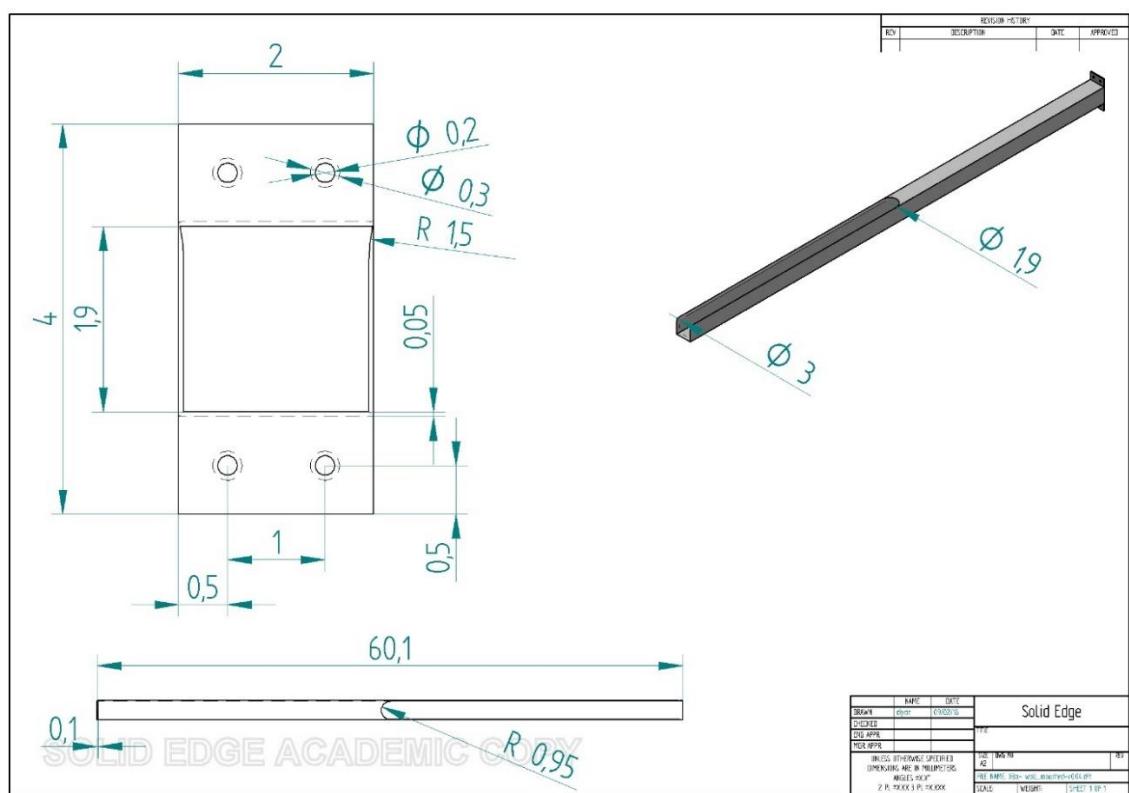


Figure 8 Basics-wall mount place part 1

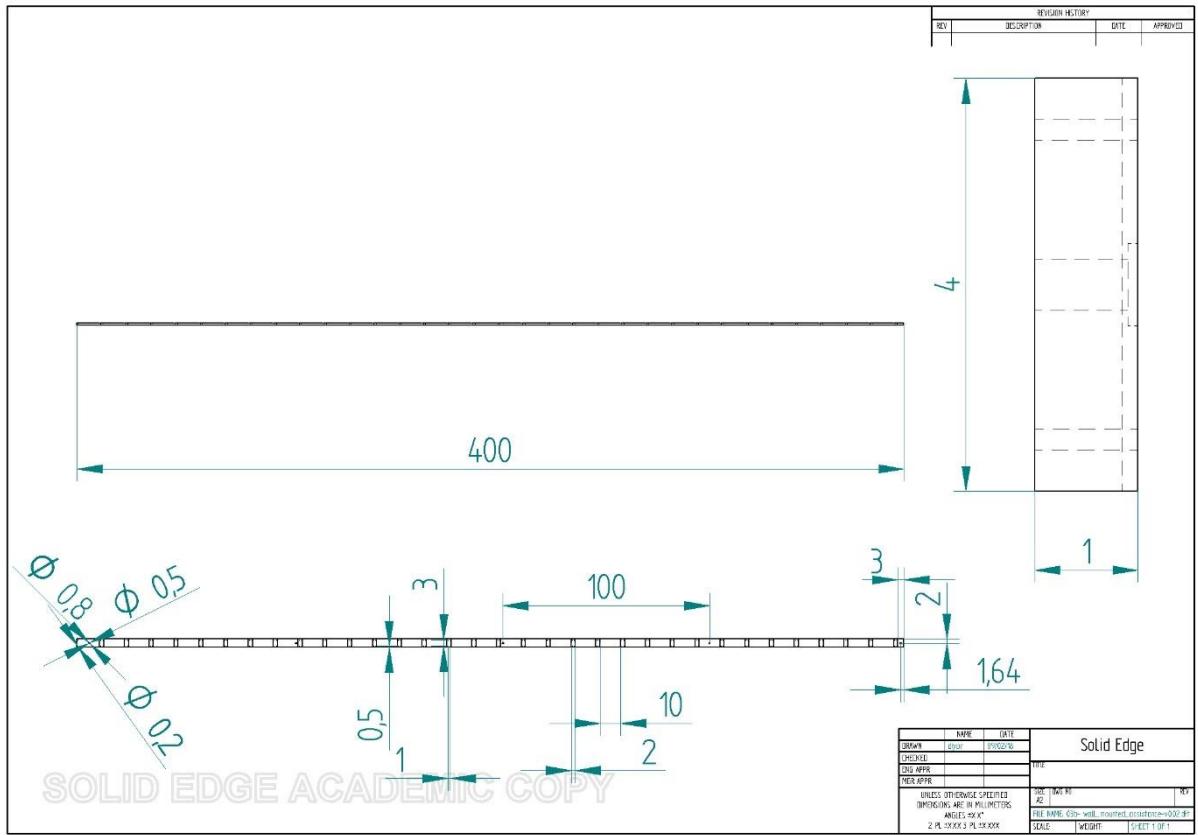


Figure 9 Basics-wall mount place part 2

Second part: Linear Motion Rail, two axis and, Ball screw

This part will pick up the hanger and it will be exchanged it between the window and places.

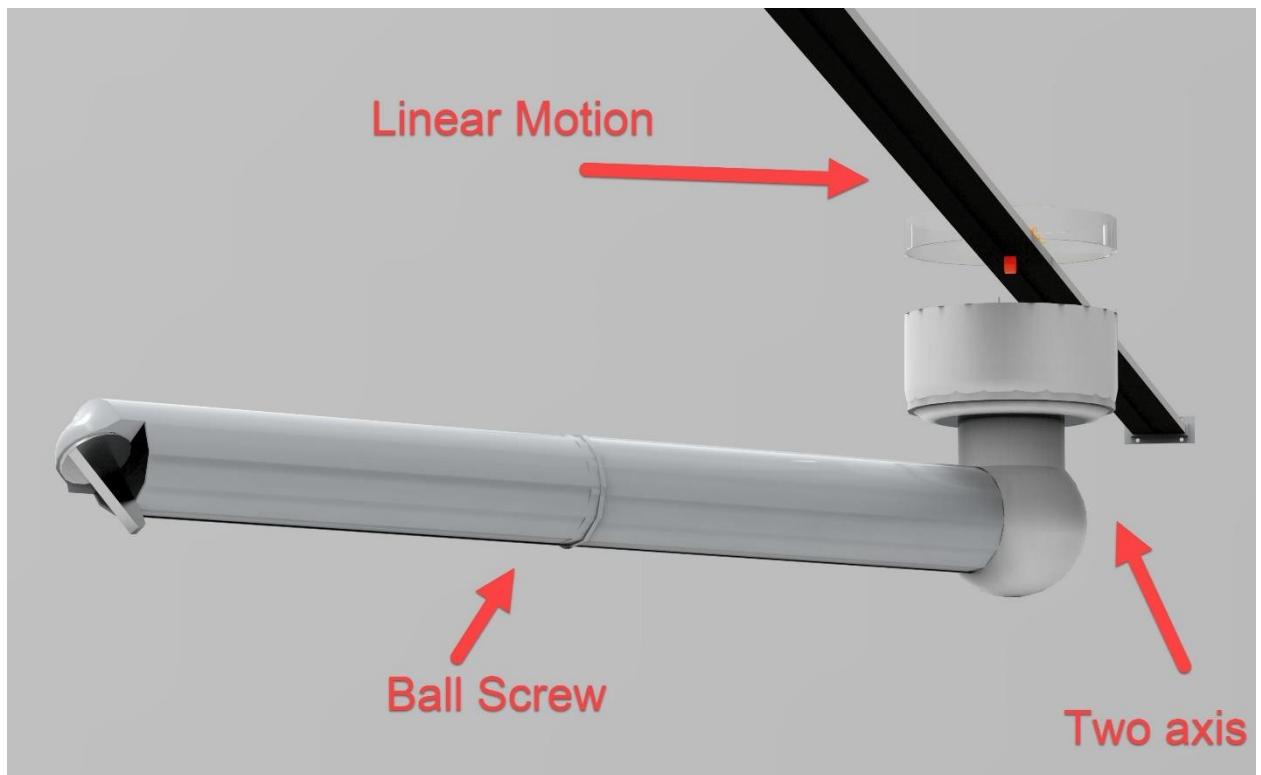


Figure 10 Model of Linear Motion, Two axis and Ball Screw

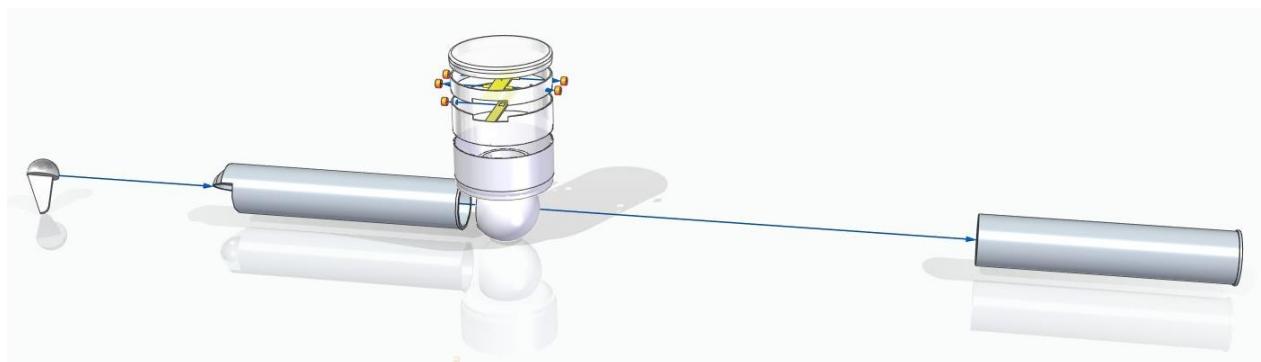


Figure 11 Parts of linear motion, two axis and ball screw

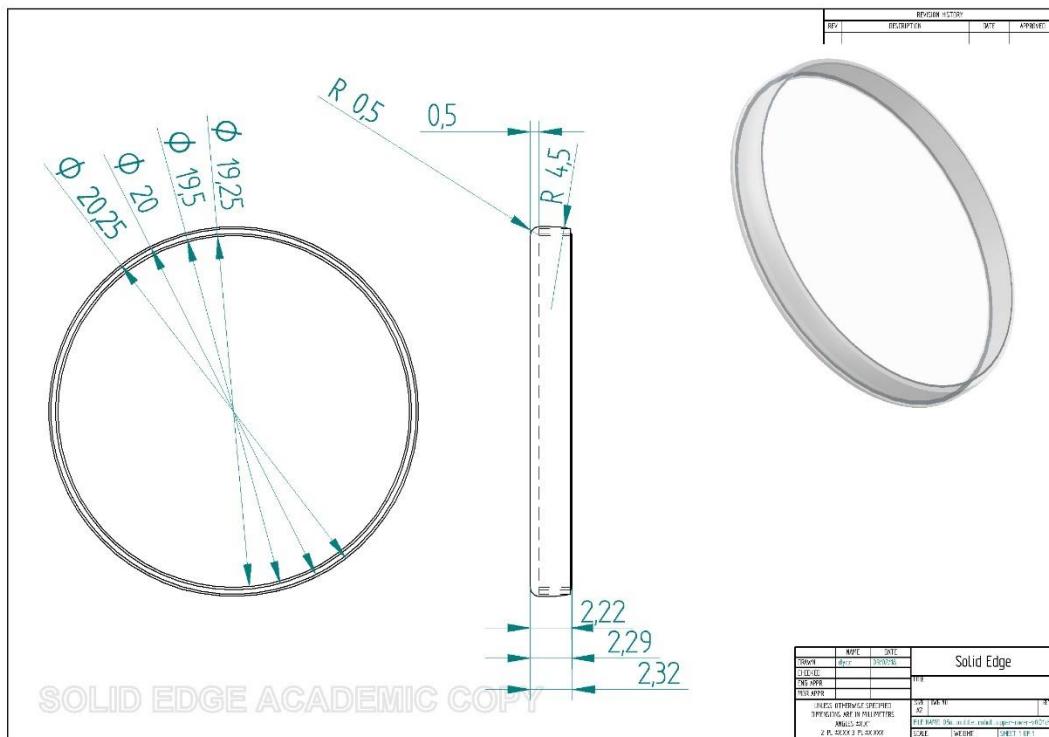


Figure 12 Linear motion, two axis and ball screw part 1

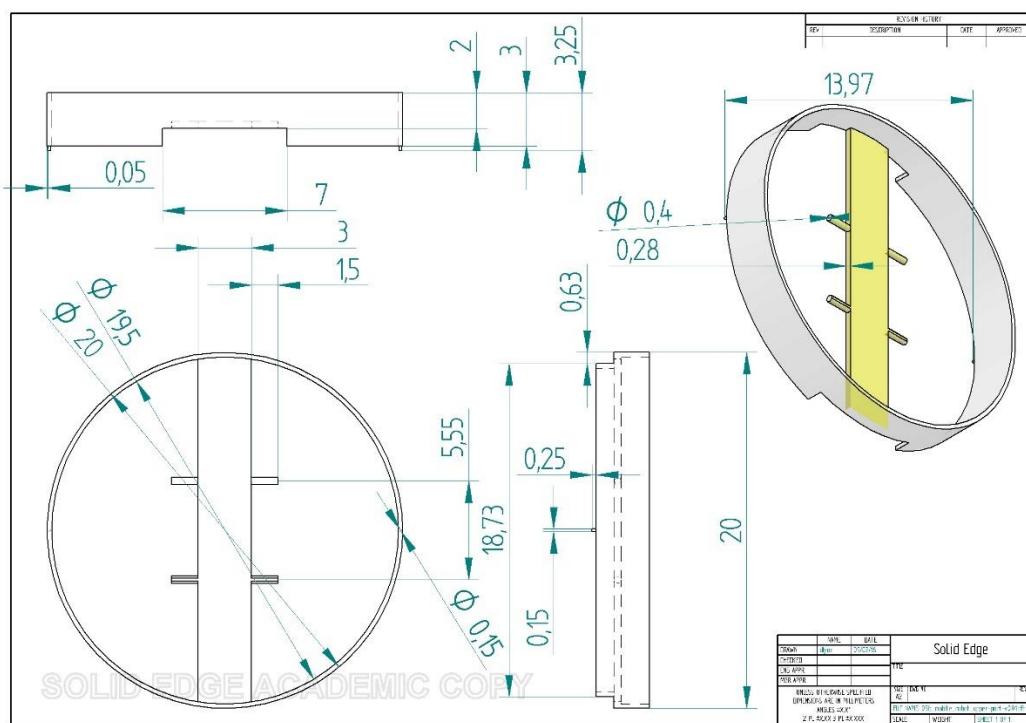


Figure 13 Linear motion, two axis and ball screw part 2

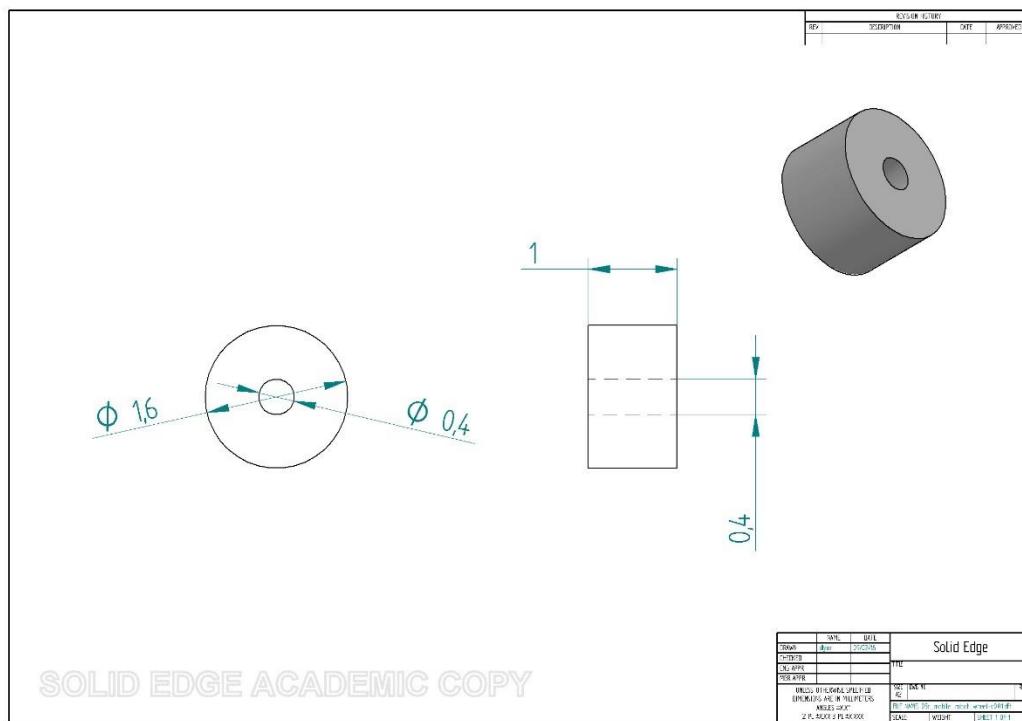


Figure 14 Linear motion, two axis and ball screw part 3

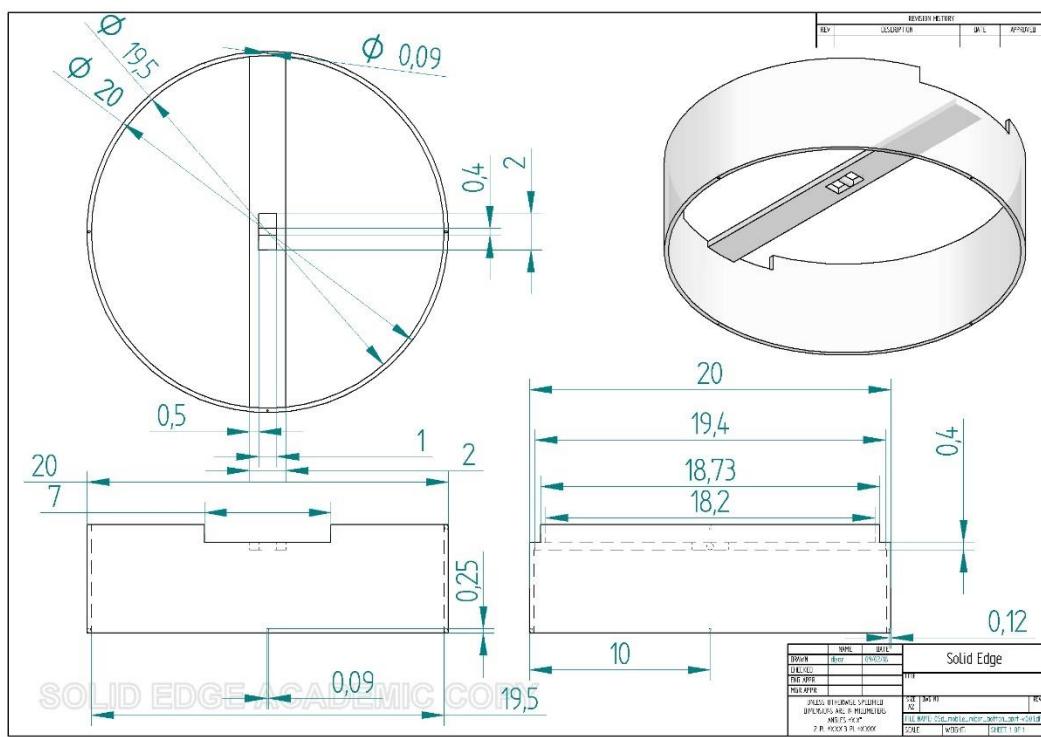


Figure 15 Linear motion, two axis and ball screw part 4

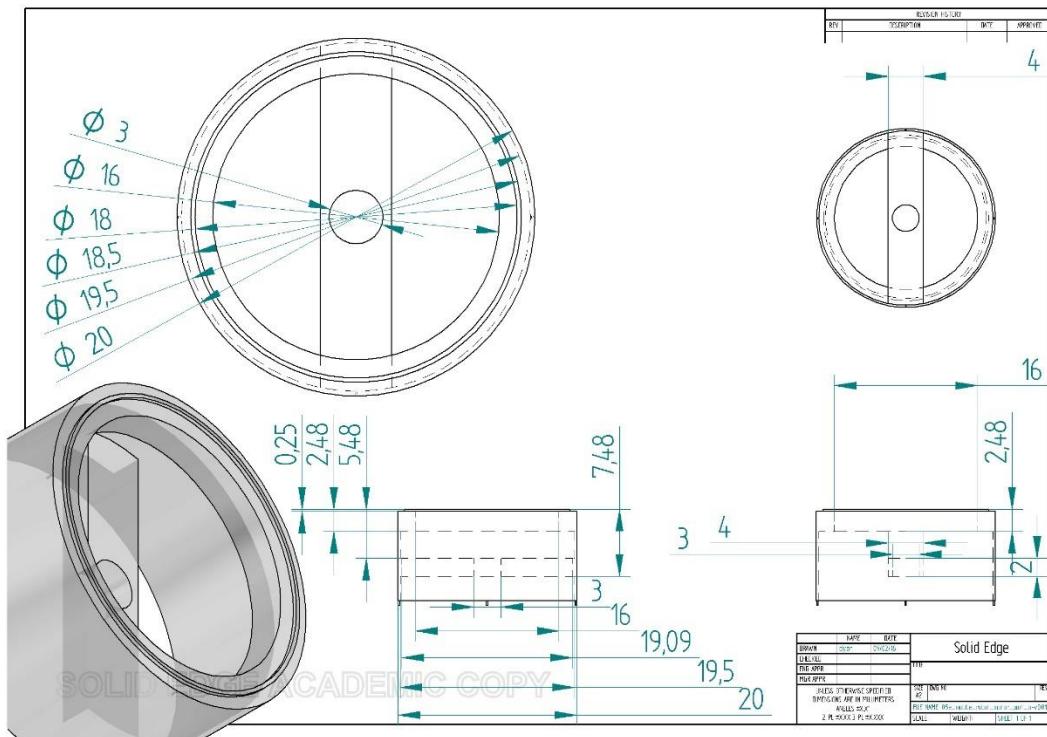


Figure 16 Linear motion, two axis and ball screw part 5

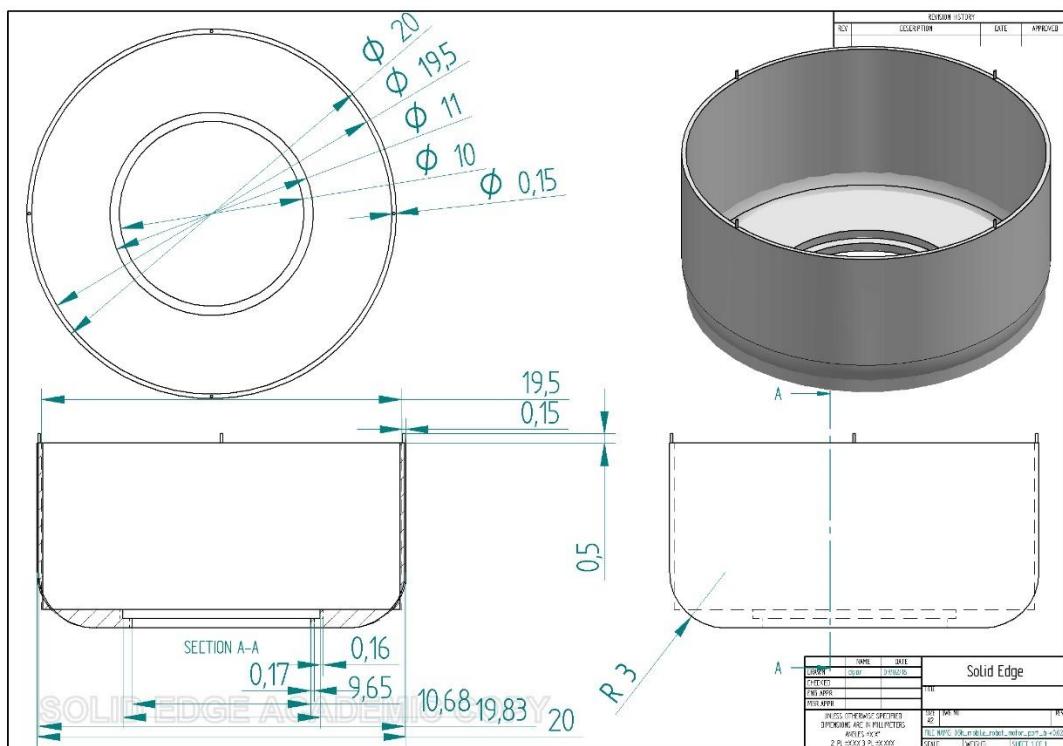


Figure 17 Linear motion, two axis and ball screw part 6

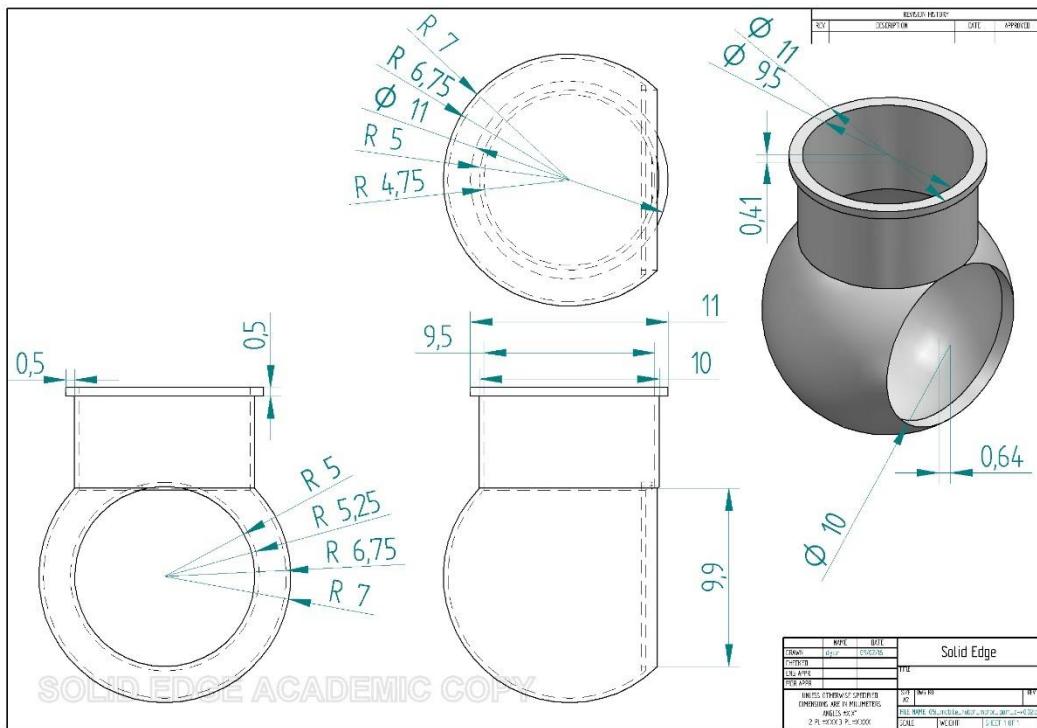


Figure 18 Linear motion, two axis and ball screw part 7

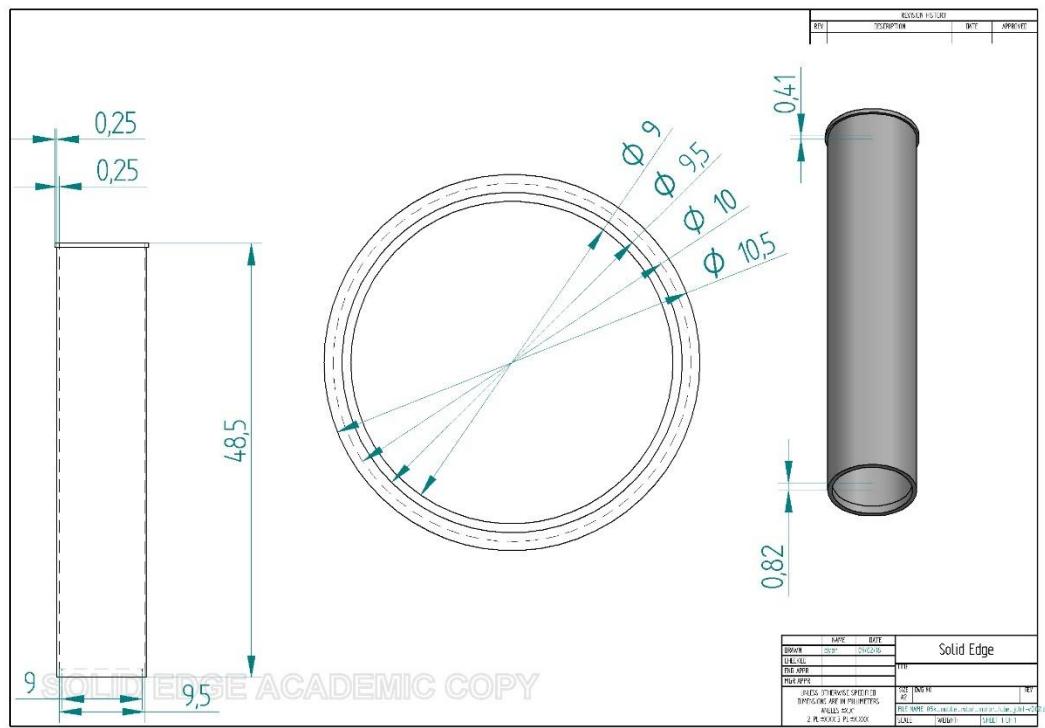


Figure 19 Linear motion, two axis and ball screw part 8

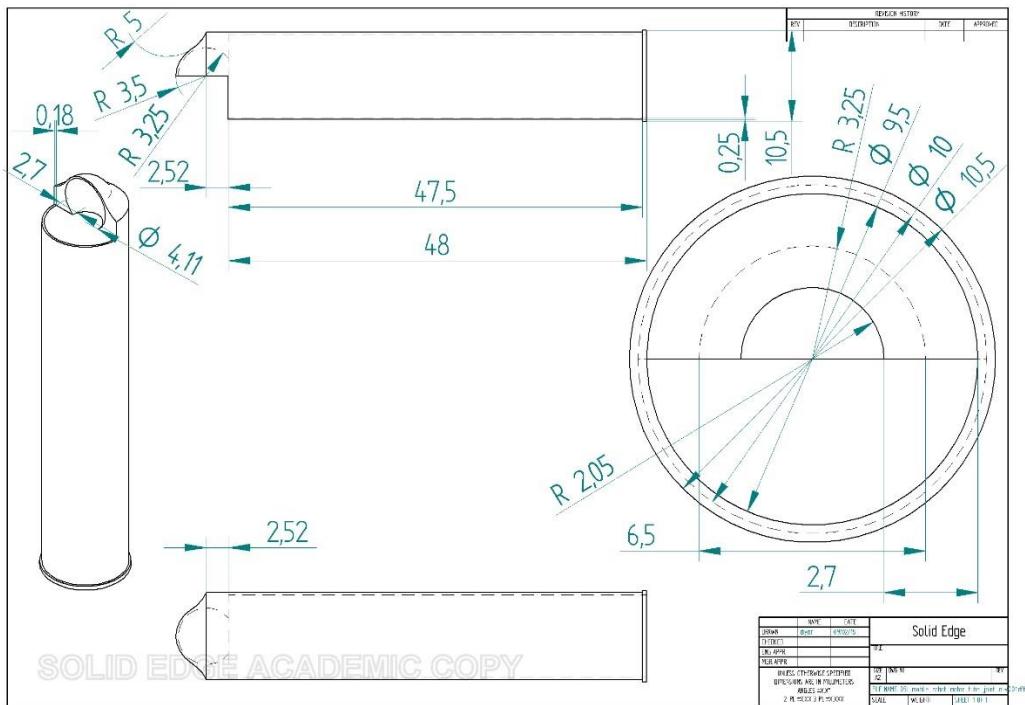


Figure 20 Linear motion, two axis and ball screw part 9

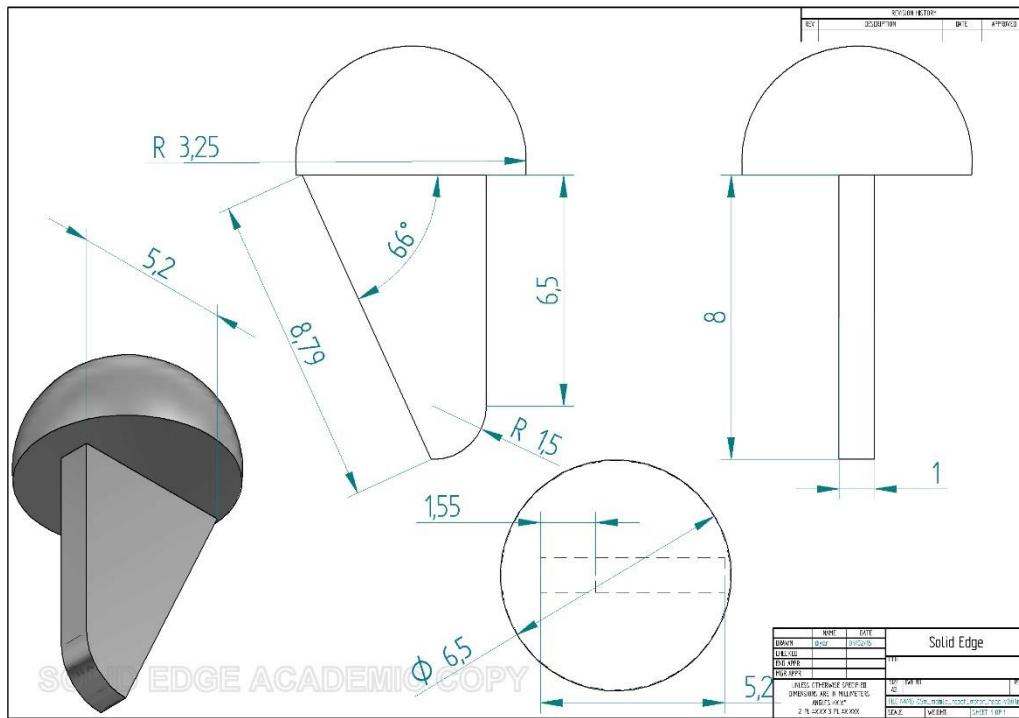


Figure 21 Linear motion, two axis and ball screw part 10

Third part: Microcontroller and Monitor

This part where user interactive for identifying during putting and pick up the coat.



Figure 22 Model of Monitor for SWSaver

Chapter Three: Prototype

List components of prototype,

Hardware:

Arduino Mega 2560 Rev3



Figure 23 Arduino Mega 2560 Rev3

The sketch of the project it is big size and needed a lot of pins for connecting the components for that Arduino Mega 2560 Rev3 it is suitable for this purpose. Example how to use this board to blinking built-in LED, File > Examples > 01.Basics > Blink:

```
void setup() {  
    // initialize digital pin LED_BUILTIN as an output.  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage  
    level)  
    delay(1000); // wait for a second  
    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage  
    LOW  
    delay(1000); // wait for a second  
}
```

After set this source code while USB is connected with board time to upload it click on Sketch > upload.

Arduino Library

To add a new Arduino library can be found in Library Manager in Arduino IDE 1.8.9 click on Sketch > Include Library > Manager Libraries and search the library need and hit on Install button as showing in figure -24-:

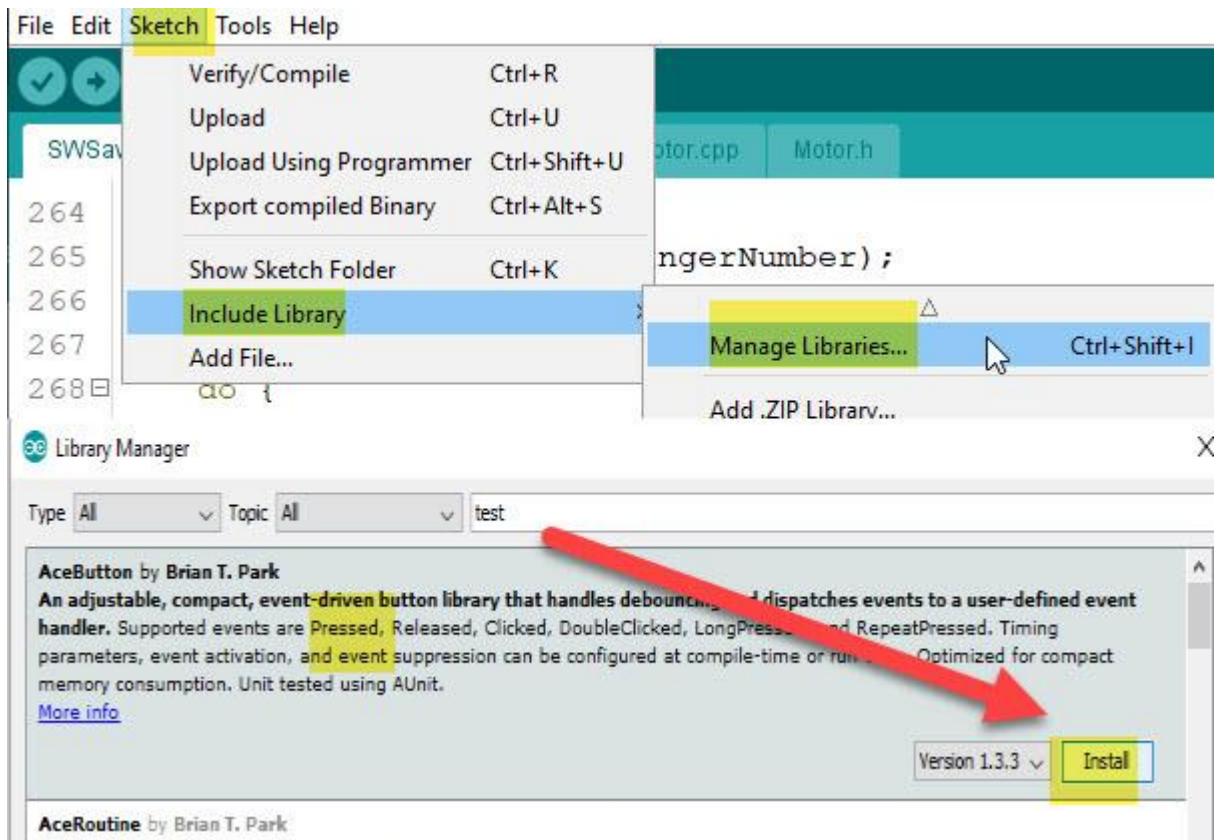


Figure 24 Manager Libraries selection

Additional, can add library manually with .zip file from Sketch > Include Library > Add .ZIP Library as showing in figure -25-

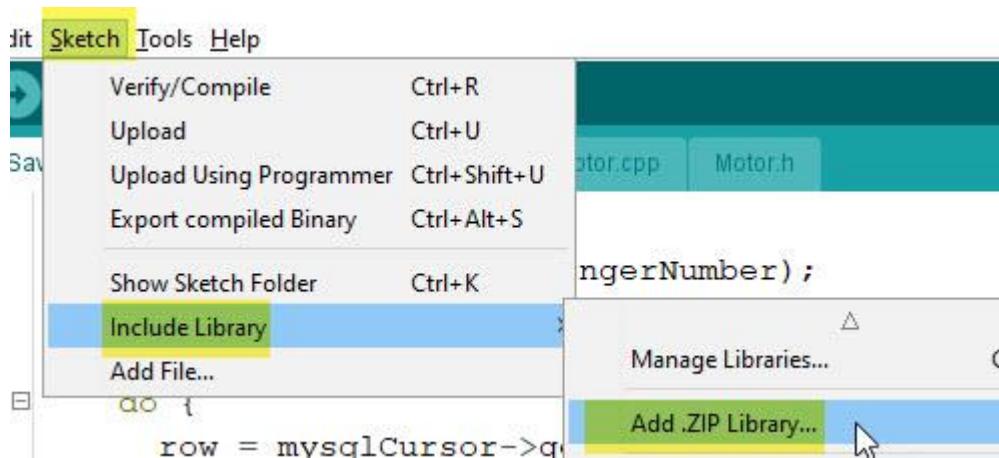


Figure 25 Upload .ZIP file library

Seeed W5500 Ethernet Shield v1.0

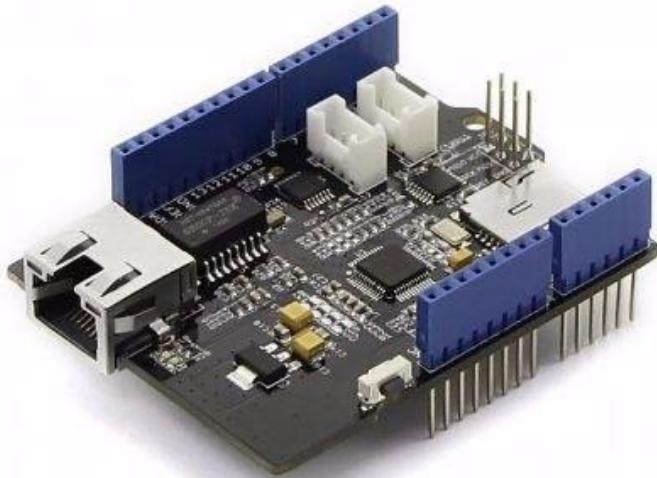


Figure 26 Seeed W5500 Ethernet Shield v1.0

W5500 support a multi protocols TCP, UDP, PPPOE, ARP, IGMP, IPv4, ICMP with micro-SD card socket for storage, example about connecting to Internet, File > Examples > Ethernet > LinkStatus:

```
#include <SPI.h>
#include <Ethernet.h>

void setup() {
    Serial.begin(9600);
}

void loop() {
    auto link = Ethernet.linkStatus();
    Serial.print("Link status: ");
    switch (link) {
        case Unknown:
            Serial.println("Unknown");
            break;
        case LinkON:
```

```

Serial.println("ON");
break;

case LinkOFF:
Serial.println("OFF");
break;
}

delay(1000);
}

```

And the output on Serial Monitor showing the status of linking:

```

| 14:57:31.400 > LINK status: ON
14:57:32.453 -> Link status: ON
14:57:33.444 -> Link status: ON
14:57:34.431 -> Link status: ON
14:57:35.448 -> Link status: ON
14:57:36.434 -> Link status: ON
14:57:37.449 <- Link status: ON

```

Figure 27 Output Monitor Serial for LinkStatus sketch

Waveshare Motor Control Shield

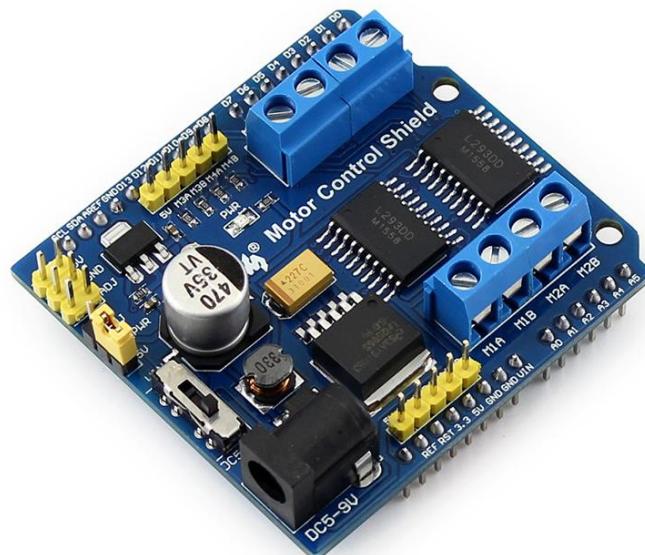


Figure 28 Waveshare motor control shield

Onboard two pieces of L293D transistors to running two Stepper motors and this shield suitable because first to controlling Stepper motor of Linear Motion and second controlling Stepper motor of basic two axis as showing in figure -29-

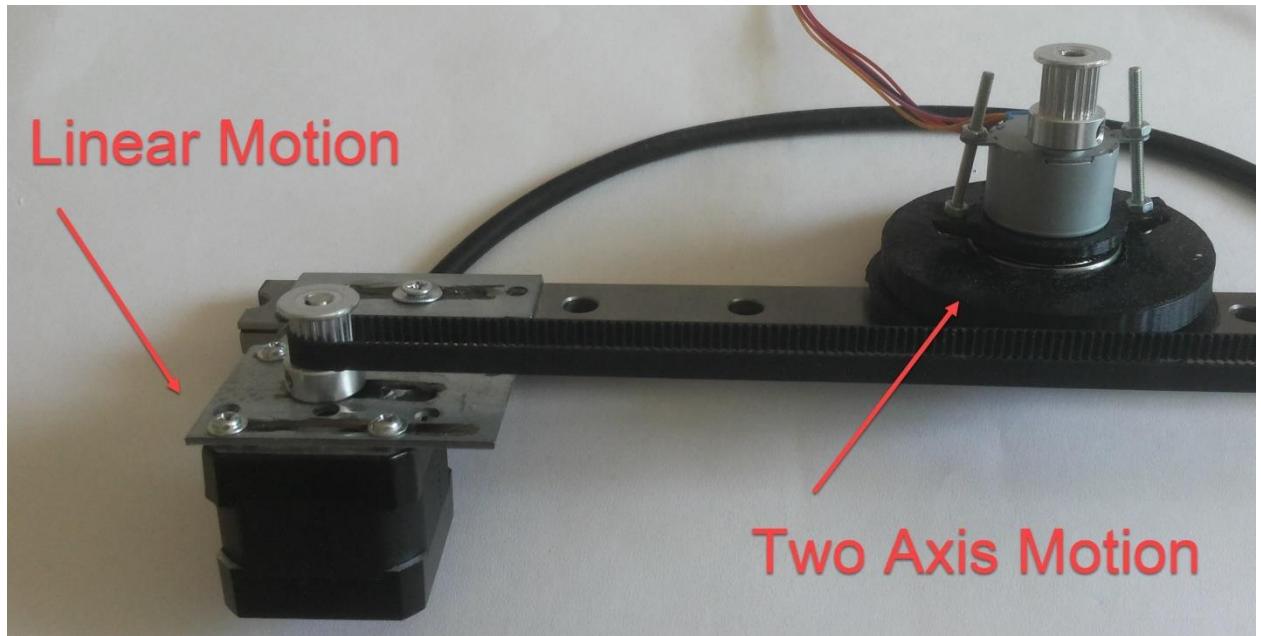


Figure 29 Linear motion and two axis motion

LCD 1602 I2C



Figure 30 LCD 1602 I2C

Is necessary two LCD 1602 to display message for user can be merging both of them to print the messages,

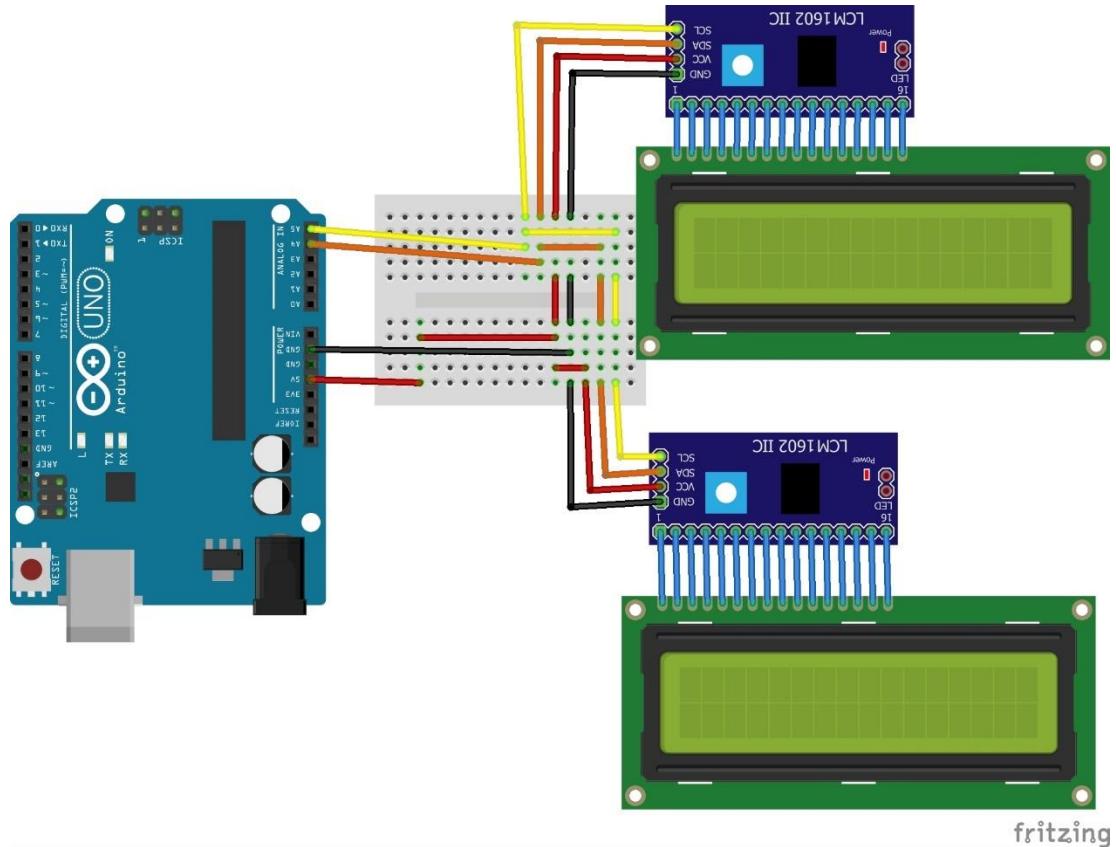


Figure 31 Merging two LCD 1602 I2C

Example to run both LCD 1602 I2C with this Sketch:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcdUp(0x27, 20, 4);
LiquidCrystal_I2C lcdDown(0x3F,20,4);

void setup()
{
    lcdUp.init();
    lcdDown.init();
    lcdUp.backlight();
    lcdDown.backlight();
```

```
lcdUp.setCursor(1,0);
lcdDown.setCursor(1,0);
lcdUp.print("Hello, world!");
lcdDown.print("Hello, World 2");
}
void loop()
{
}
```

And here is the screenshot for the Display:



Figure 32 Printing message of sketch for Hello World

MIRF-RC522



Figure 33 MFRC522 RFID-RC522

To read and write 13.56 MHz contactless Tag and Card.

Keypad 4x4



Figure 34 Keypad 4x4

To can user input the PIN Code and make selection for putting and pickup coats this Keypad is suitable to do this job,

Software:

Up to run Ubuntu Server

Steps for up to run Ubuntu Server to using it for Database, downloading the image file of Ubuntu Server beta version for developer from <http://cdimage.ubuntu.com/daily-live/current/> and click on “64-bit PC(AMD64) desktop image as showing in figure -35-

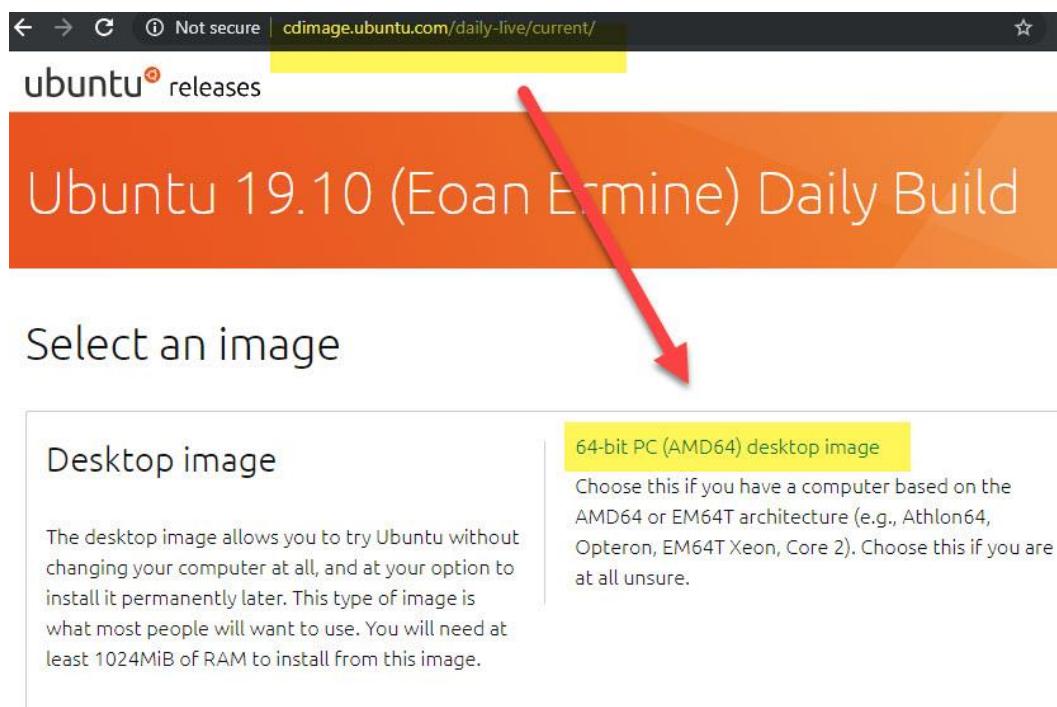


Figure 35 Download Ubuntu Server development channel

Next download VMware Workstation from <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html> to create a Virtual Machine to install Ubuntu Server on it:

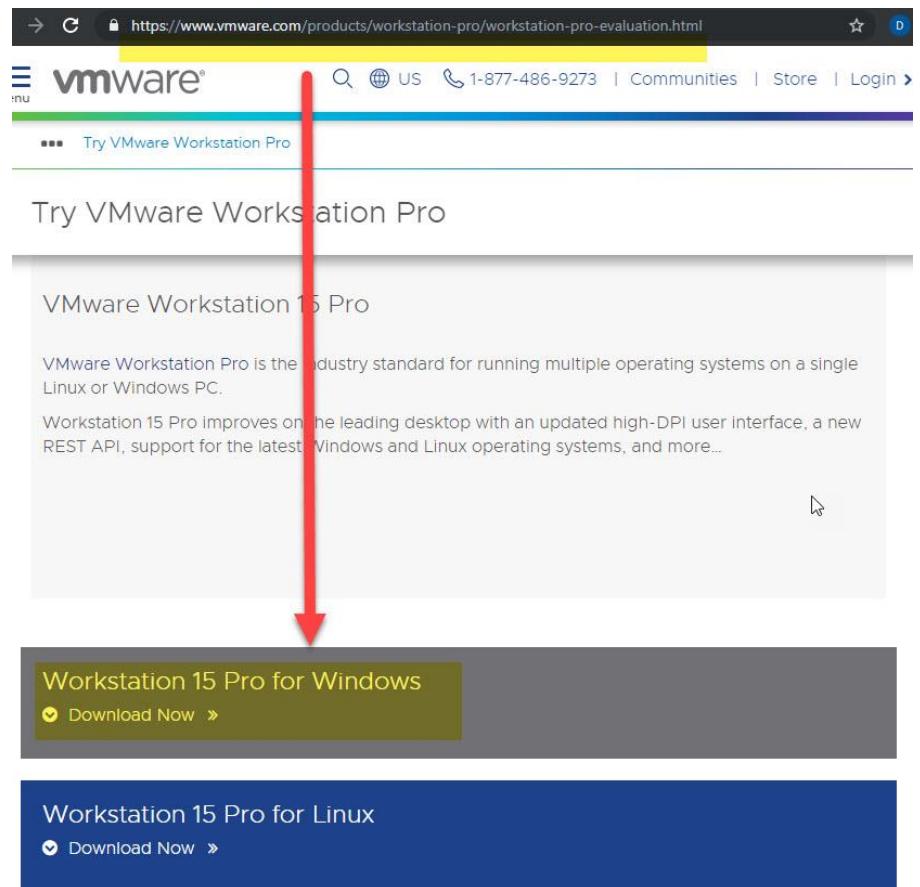


Figure 36 Download VMware Workstation 15

After installing VMware Workstation will create Virtual Machine by click on File > New Virtual Machine:

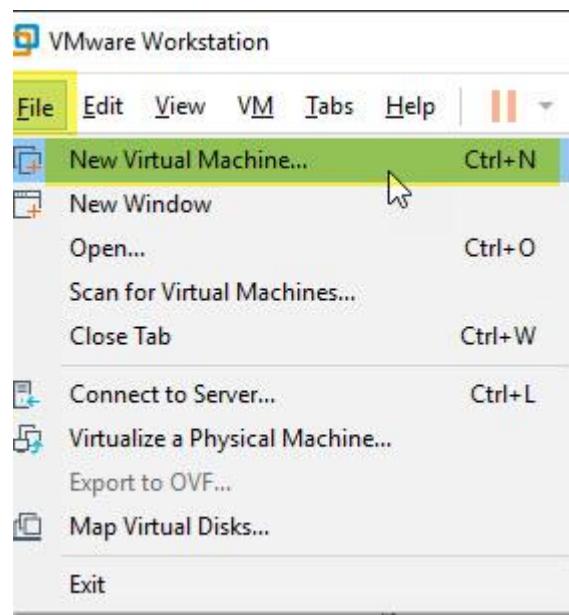


Figure 37 Selection New Virtual Machine

Leaving the selection on Typical configuration and click on Next and select the image file of Ubuntu Server that downloaded it by click on Browser button and explorer to the image file as show in figure -38-

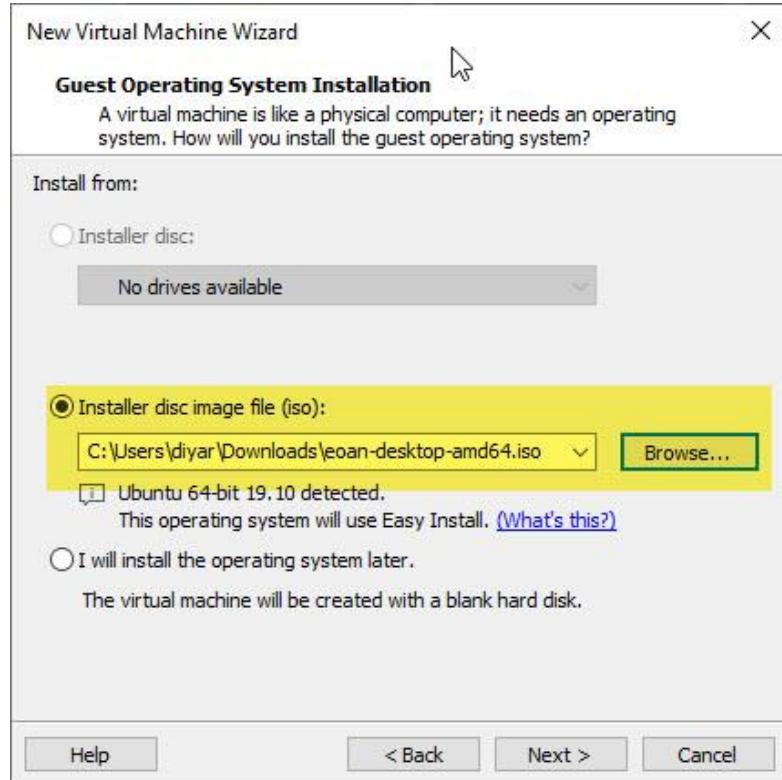


Figure 38 Selection ISO Image for Ubuntu Server

Click on next and fill the form with User name and password:

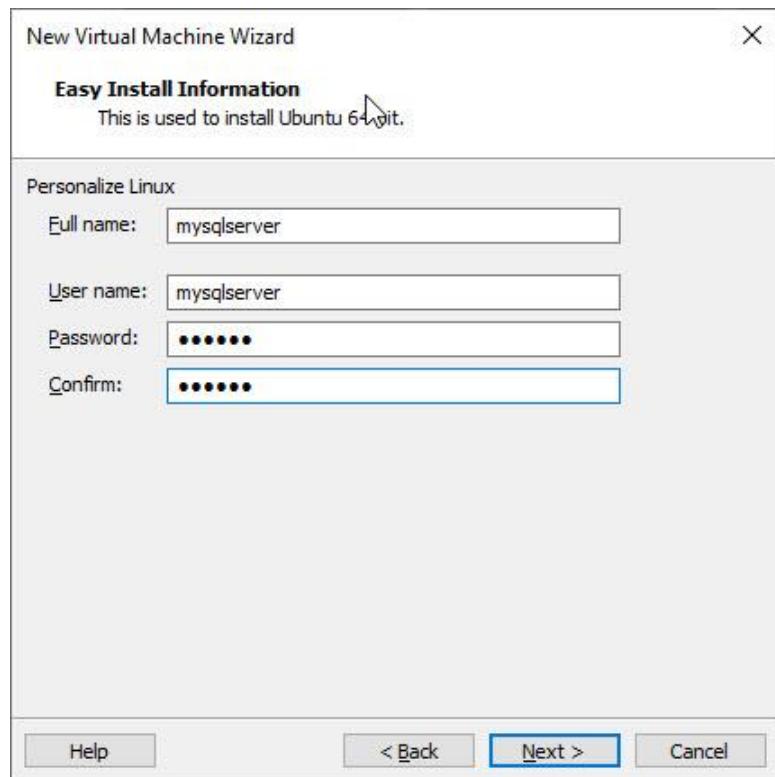


Figure 39 Set Username and password

Click on next and rename the virtual machine the click on Next and set 15 GB for disk and click Next

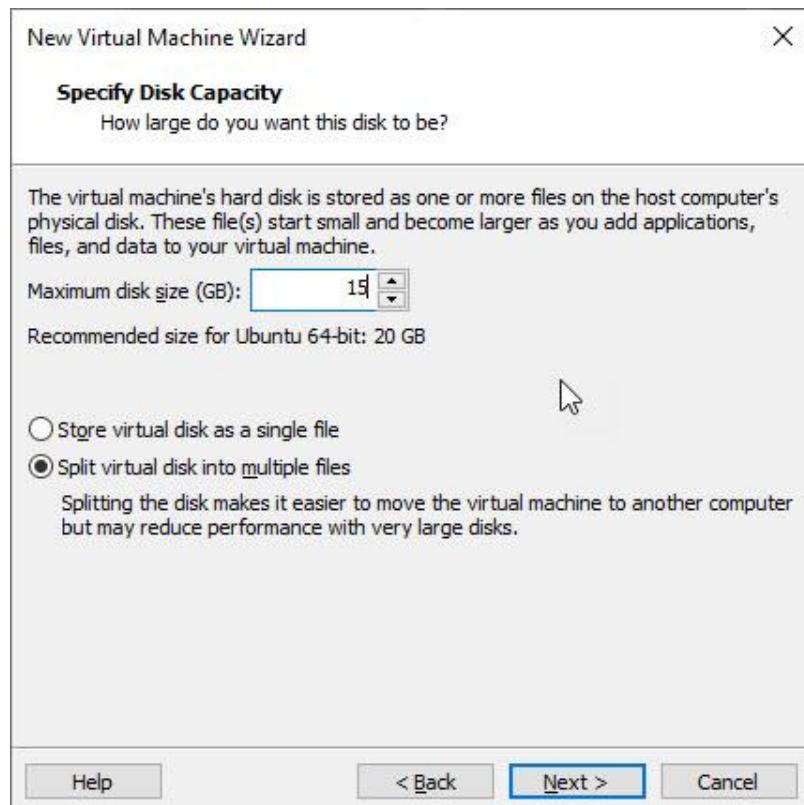


Figure 40 resize the VHD

Click on finished to start Virtual Machine and wait to boot for the first time and loading installation procedure and wait until will be finished, hit Ctrl+ Alt F3 to startup by terminal and type the username and password to login and type `sudo apt update && sudo apt dist-upgrade` for up to date the system. Install SSH to typing `sudo apt install SSH` as showing in figure -41-

```
Processing triggers for liblc bin (2.23-0ubuntu10.2) ...
Processing triggers for systemd (240-6ubuntu5) ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for dbus (1.12.12-1ubuntu1) ...
Processing triggers for shared-mime-info (1.10-1) ...
Processing triggers for fontconfig (2.13.1-2ubuntu2) ...
Processing triggers for desktop-file-utils (0.23-4ubuntu1) ...
Processing triggers for initramfs-tools (0.133ubuntu2) ...
update-initramfs: Generating /boot/initrd.img-5.0.0-13-generic
mysqlserver@ubuntu:~$ sudo apt install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libpoppler85
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere rssh ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh ssh-import-id
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 662 kB of archives.
After this operation, 6,050 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
0% [Working]^\Quit
mysqlserver@ubuntu:~$ _
```

Figure 41 Installing SSH

After that necessary to install network tools by type `sudo apt install net-tools` as showing in figure -42-

```
mysqlserver@ubuntu:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libpoppler85
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 196 kB of archives.
After this operation, 864 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu eoan/main amd64 net-tools amd64 1.60+git20180626.aebd88e-1
[196 kB]
Fetched 196 kB in 36s (5,456 B/s)
Selecting previously unselected package net-tools.
(Reading database ... 139380 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Processing triggers for man-db (2.8.5-2) ...
mysqlserver@ubuntu:~$
```

Figure 42 Installing network tools

Type `ifconfig` command to know the IP Address for Virtual Machine and using SSH client to access it and doing the configuration needed.

```
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Processing triggers for man-db (2.8.5-2) ...
mysqlserver@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.0.32 netmask 255.255.255.0 broadcast 192.168.0.255
        inet6 fe80::f816:848a:d6b9:9d8b prefixlen 64 scopeid 0x20<link>
          ether 00:0c:29:b5:6e:c5 txqueuelen 1000 (Ethernet)
            RX packets 49384 bytes 60368376 (60.3 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 17929 bytes 1134136 (1.1 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 460 bytes 39143 (39.1 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 460 bytes 39143 (39.1 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mysqlserver@ubuntu:~$
```

Figure 43 Running `ifconfig` command

Next step to installing phpMyAdmin and MySQL, typing `sudo apt install phpmyadmin mysql-server php libapache2-mod-php` and following the instruction of installing

```
Setting up lua-sql-mysqli-dev:amd64 (2.3.4-1) ...
Setting up lua-dbi-mysql-dev:amd64 (0.7.1-2) ...
Setting up libodbc-mysql-dev (2.4.0-5) ...
mysqlserver@ubuntu:/etc/mysql$ sudo apt install phpmyadmin mysql-server php libapache2-mod-php...
```

Figure 44 Installing `phpmyadmin`

Up to run Arduino and MySQL Server:

Creating Database and Table in MySQL,

```
mysql> CREATE TABLE test_arduino.hello_arduino (
    -> num integer primary key auto_increment,
    -> message char(40),
    -> recorded timestamp
    -> );
```

Figure 45 MySQL query for create database and tabl

Next insert a data by running this `INSERT` statement:

```
mysql> INSERT INTO test_arduino.hello_arduino (message) VALUES ('Hello, Arduino!');
```

Figure 46 MySQL Query for insert data

Then connecting to MySQL Database by Arduino, click on File > Examples > MySQL Connector Arduino:

```
#include <Ethernet.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>

byte mac_addr[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

IPAddress server_addr(192,168,0,32); // IP of the MySQL *server* here
char user[] = "root";           // MySQL user login username
char password[] = "123456";     // MySQL user login password

// Sample query
char INSERT_SQL[] = "INSERT INTO test_arduino.hello_arduino (message)
VALUES ('Hello, Arduino!');"

EthernetClient client;
MySQL_Connection conn((Client *)&client);

void setup() {
  Serial.begin(115200);
  while (!Serial); // wait for serial port to connect
  Ethernet.begin(mac_addr);
  Serial.println("Connecting...");
  if (conn.connect(server_addr, 3306, user, password)) {
    delay(1000);
  }
}
```

```

else
    Serial.println("Connection failed.");
}

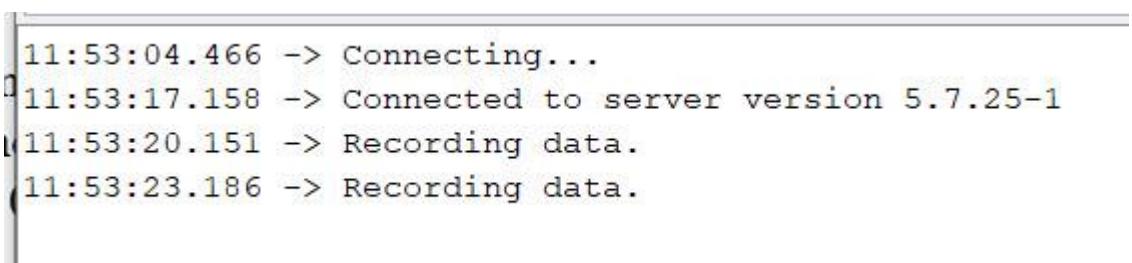
void loop() {
    delay(2000);

    Serial.println("Recording data.");

    // Initiate the query class instance
    MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
    // Execute the query
    cur_mem->execute(INSERT_SQL);
    // Note: since there are no results, we do not need to read any data
    // Deleting the cursor also frees up memory used
    delete cur_mem;
}

```

After MySQL Query apply the result in Serial Monitor showing is recorded:



The screenshot shows the Arduino Serial Monitor window. It displays four lines of text output from a MySQL connection process:

- 11:53:04.466 -> Connecting...
- 11:53:17.158 -> Connected to server version 5.7.25-1
- 11:53:20.151 -> Recording data.
- 11:53:23.186 -> Recording data.

Figure 47 Print message about the MySQL query finished

Check the result that query is applied:

```

mysql> select * from test_arduino.hello_arduino;
+----+-----+-----+
| num | message           | recorded          |
+----+-----+-----+
|   1 | Hello, Arduino! | 2019-05-21 01:32:23 |
|   2 | Hello, Arduino! | 2019-05-21 01:53:20 |
+----+-----+-----+
2 rows in set (0.00 sec)

```

Figure 48 MySQL query for selection

PIN Code Generator

To get four random digits numbers to use for PIN Code first get seed for random by reading the voltage and change to integer value and it will be changes and this mechanism use it to seed random function then set the ranges to get result 1000 – 9999 as showing in figure -49-

```

randomSeed(analogRead(A0));
unsigned int firstGenerator = random(1000, 8999);
unsigned int secondGenerator = random(100, 899);
unsigned int thirdGenerator = random(99);
unsigned int fourthGenerator = random(3);
unsigned int idKey = firstGenerator + secondGenerator + thirdGenerator + fourthGenerator;

```

Figure 49 Source code for Generating PIN Code

Merging the components

Hardware:

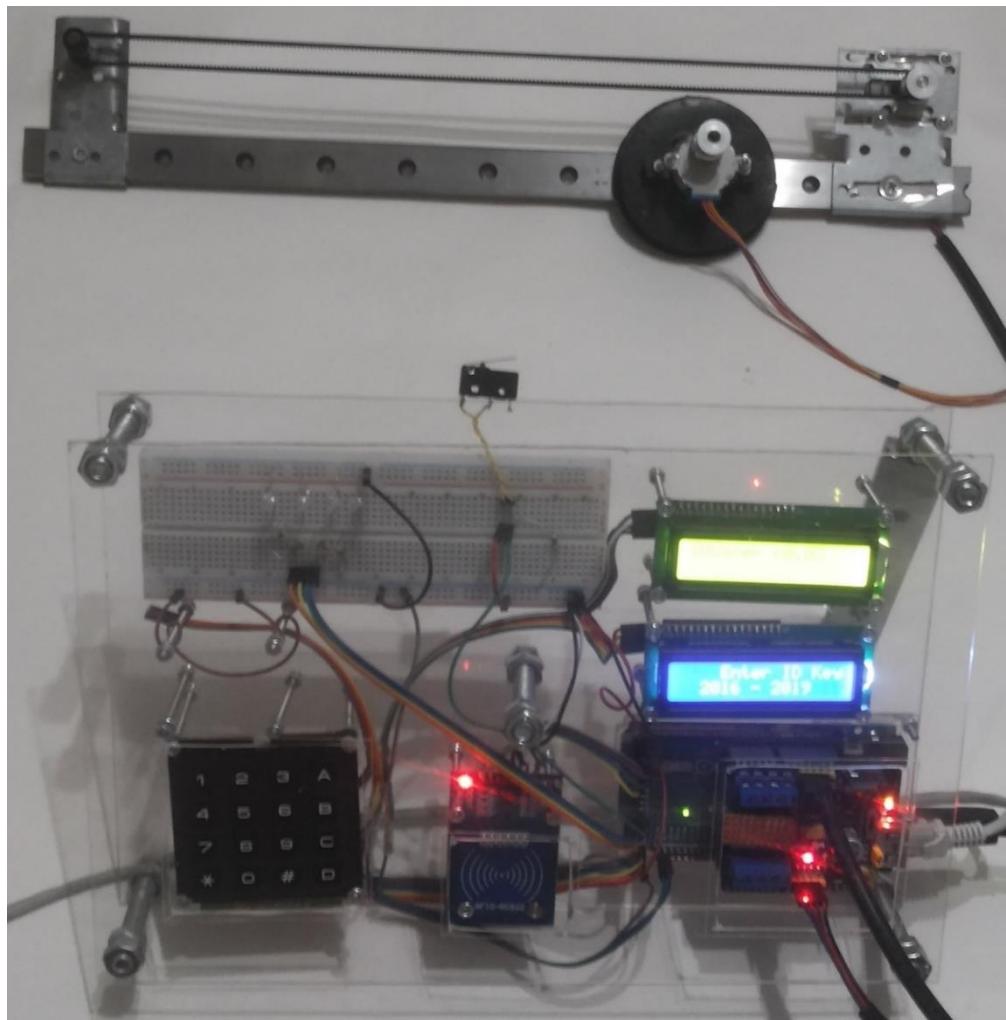


Figure 50 Fixed hardware for SWSaver prototype

In figure -50- fixed all components of the SWSaver prototype.

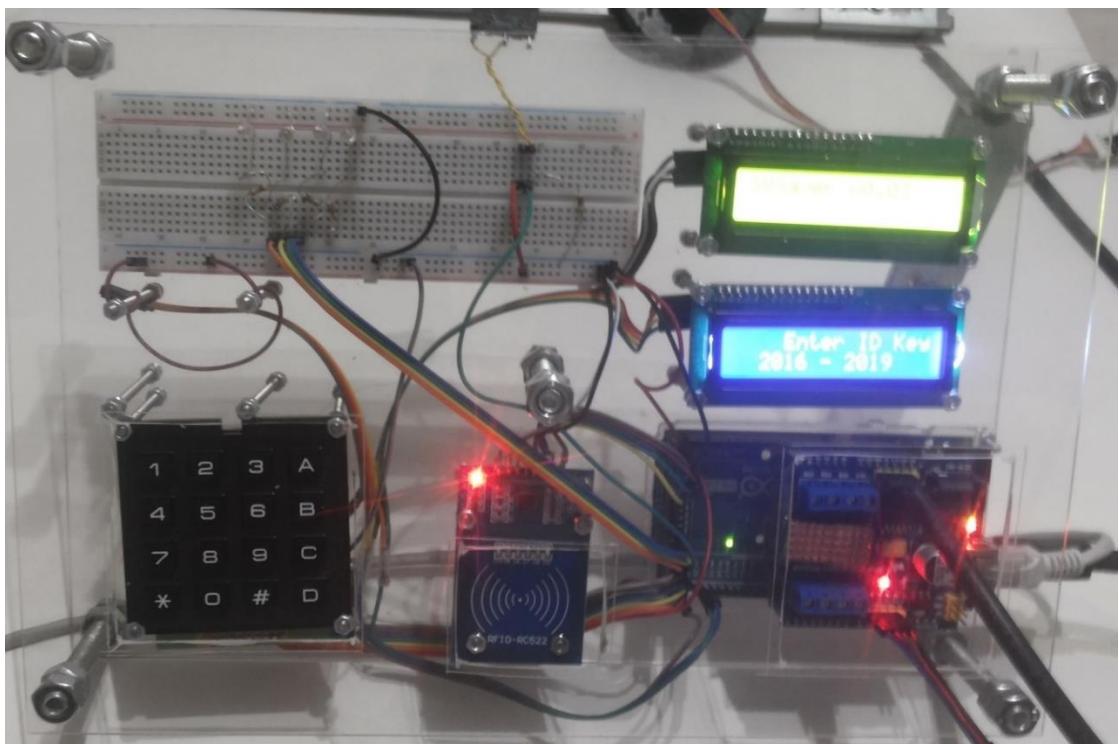


Figure 51 Main controller

In figure -51- main control for the microcontroller and monitor prototype.

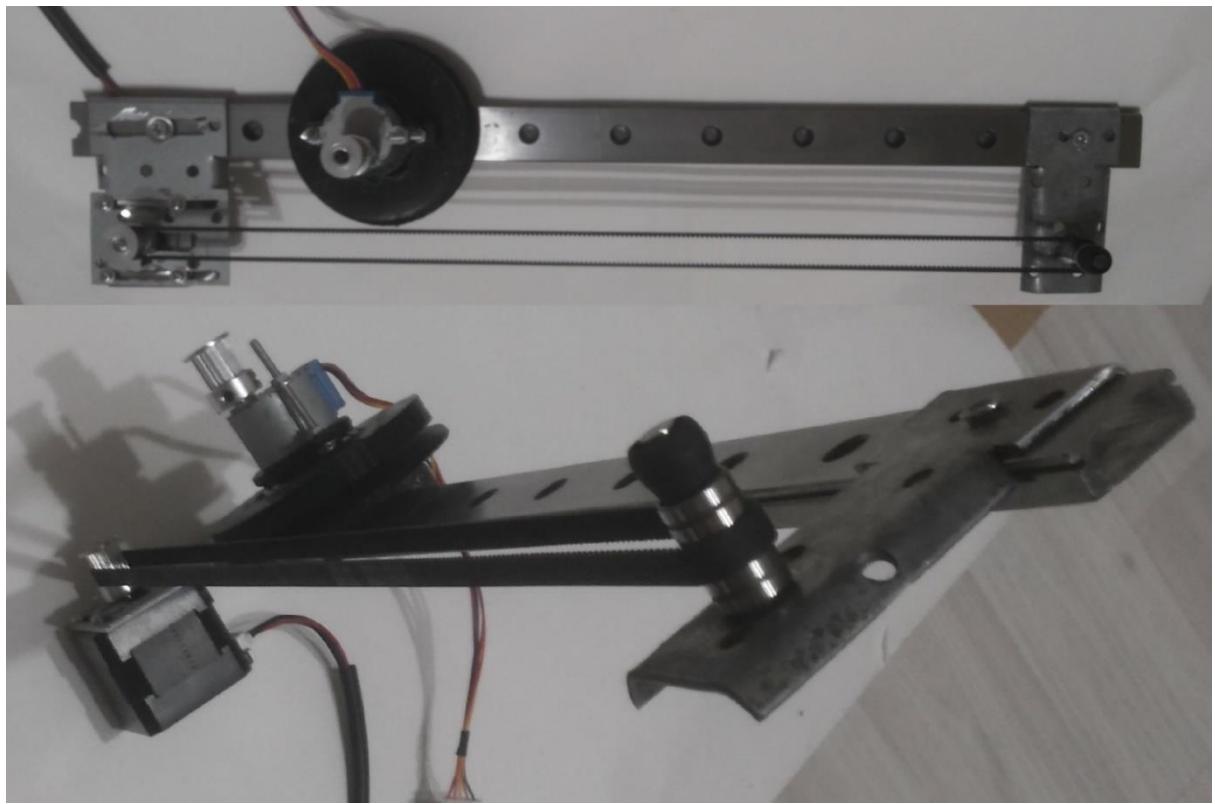


Figure 52 Linear motion and two axis motion

In figure -52- Linear motion with two axis for holding the hangers and set in places.

Software:

```
//*****  
//#####:: :: Library Part :: :#####//  
//*****  
  
#include <Keypad.h>  
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>  
#include <Ethernet.h>  
#include <SPI.h>  
#include <MySQL_Connection.h>  
#include <MySQL_Cursor.h>  
#include "DEV_Config.h"  
#include "Motor.h"  
#include <MFRC522.h>
```

Figure 53 Import libraries

In figure -53- showing import necessary libraries for the components are connected to Arduino Mega 2560 REV3 board,

```
//*****  
//#####:: :: lcd1 Part :: :#####//  
//*****  
  
LiquidCrystal_I2C lcd0(0x27, 20, 4);  
LiquidCrystal_I2C lcd1(0x3f, 20, 4);
```

Figure 54 Naming LCD 1602 I2C

In figure -54- is show to Naming the LCDs devices (lcd0, lcd1) by them addresses (0x27, 0x3f) to can initializing and use them in program.

```

//*****//
//#####:: :: Keypad Part :: ::#####//
//*****//

const byte rowsKeypad = 4;
const byte columnsKeypad = 4;
char keys[rowsKeypad][columnsKeypad] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};

byte rowKeypadPins[rowsKeypad] = {25, 24, 23, 22};
byte columnKeypadPins[columnsKeypad] = {29, 28, 27, 26};
char customKey;

Keypad customKeypad(makeKeymap(keys), rowKeypadPins, columnKeypadPins, rowsKeypad, columnsKeypad);

```

Figure 55 Preparing Keypad 4x4

In figure -55- showing how to setup the Keypad with 16 keys and name the Keypad (customKeypad) and define the list of row keys and column keys.

```

//*****//
//#####:: :: Networking Part :: ::#####//
//*****//

byte physicalAddress[] = {0xAA, 0xBB, 0xEE, 0xFF, 0xAA};
IPAddress serverAddress(192, 168, 0, 14);
EthernetClient ethernetClient;

MySQL_Connection MySQLConnection((Client *)&ethernetClient);
const char MySQLUser[] = "root";
const char MySQLPassword = "123456";

```

Figure 56 Preparing network communication

In figure -56- showing the preparing the network connection by set a random MAC Address for the Ethernet Shield and assign IP address, username and password of Server that will use it for Database.

```

//*****//  

//#####:: : RFID-RC522 Part :: :#####//  

//*****//  

#define SS_RFID 53  

#define RST_RFID 49  

MFRC522 rfid_Read(SS_RFID, RST_RFID);  

MFRC522 mfrc522(SS_RFID, RST_RFID);

```

Figure 57 Preparing RFID-RC522

In figure -57- define the pins for Slave Select (SS_RFID) to enable and disable the RFID-RC522 device as part of Serial Peripheral Interface for communicating to synchronous serial data protocol between devices quickly over short distances and (RST_RFID) for resetting option.

```

//*****//  

//#####:: : Other Parts :: :#####//  

//*****//  

#define sensorSwitch 30      // Assigned Pin Number D40 to Sensor Switch.  

#define led0 31              // For debug purpose only.  

#define led1 32              // For debug purpose only.  

#define led2 33              // For debug purpose only.  

#define led3 34              // For debug purpose only.  

int sensorSwitchStatus = 0;] // Define the Sensor Switch Status in zero value assuming is not pressed.

```

Figure 58 Define debug components

In figure -58- defined the pins for LEDs for debug purpose to check the steps of Motors and Switch as Sensor to detect cloth that putting on Cloth Hanger.

```

void setup() {
    Serial.begin(115200);
    Serial.println(F(" #sis000 Initialize and confirm Serial connected."));
    while (!Serial);
    Serial.println(F(" #sip000 Initialize Pins."));
    pinMode(led0, OUTPUT);
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(led3, OUTPUT);
    pinMode(sensorSwitch, INPUT);
    Serial.println(F(" #sil000 Initialize Liquid Crystal Display."));
    lcd0.init();
    lcd0.backlight();
    lcd0.setCursor(0, 0);
    lcd0.print("SWSaver v0.03");
    lcd1.init();
    lcd1.backlight();
    lcd1.setCursor(3, 0);
    lcd1.print("Welcome...");
    lcd1.setCursor(2, 1);
    lcd1.print("2016 - 2019");
    Serial.println(F(" #sie000 Initialize Ethernet Shield Network Module"));
    ethernetHardwareChecker();
    Serial.println(F(" #sim000 Initialize MySQL Server connection"));
    mysqlServerConnection();
    lcd0.clear();
    lcd1.clear();
    Serial.println(F(" #sir000 Initialize RFID-RC522."));
    mfrc522.PCD_Init();
}

```

Figure 59 Setup function

In figure -59- is declaration Setup function to preparing the communication and initializing the components.

```

void ethernetHardwareChecker () {
    Ethernet.begin(physicalAddress);
    // Check for Ethernet hardware present
    if (Ethernet.hardwareStatus() == EthernetNoHardware) {
        Serial.println(F(" #sie001 Ethernet shield was not found. Sorry, can't run without hardware. :("));
        while (true) {
            delay(1); // do nothing, no point running without Ethernet hardware
        }
    }
    if (Ethernet.linkStatus() == LinkOFF) {
        Serial.println(F("Ethernet cable is not connected."));
    }
    Serial.print(F(" #sie002 Assign a local IP Address for Ethernet Shield Network Module: "));
    Serial.println(Ethernet.localIP());
}

```

Figure 60 Ethernet setup

In figure -60- checking the Ethernet Shield is connected and cable is plug-in.

```

void mysqlServerConnection() {
    lcd1.clear();
    lcd1.setCursor(1, 0);
    lcd1.print("Connecting to");
    lcd1.setCursor(2, 1);
    lcd1.print("MySQL Server");
    if (MySQLConnection.connect(serverAddress, 3306, MySQLUser, MySQLPassword))
    {
        delay(500);
        lcd1.clear();
    }
    else
    {
        lcd1.clear();
        Serial.println(F(" #sim001 Failed to connect, try again!"));
        lcd1.setCursor(1, 0);
        lcd1.print("Error connect");
        lcd1.setCursor(0, 1);
        lcd1.print(" Try again...");
        delay(10000);
        lcd1.clear();
    }
}

```

Figure 61 Setup MySQL Connector

In figure -61- creating the connecting between Arduino and Database Server.

```

void loop() {
    sensorSwitchStatus = digitalRead(sensorSwitch);
    customKey = customKeypad.getKey();
    lcd0.setCursor(0, 0);
    lcd0.print("SWSaver v0.03");
    lcd1.setCursor(4, 0);
    lcd1.print("Enter ID Key");
    lcd1.setCursor(2, 1);
    lcd1.print("2016 - 2019");

    if (sensorSwitchStatus == 0) {
        unloadHanger();
    } else {
        loadHanger();
    }
    digitalWrite(RST_RFID, HIGH);
    digitalWrite(SS_RFID, HIGH);
    rfid_Read.PCD_SoftReset;
    rfid_Read.PCD_Init();
    //rfid_Read.PCD_DumpVersionToSerial(); // For debug purpose only.
    if (rfid_Read.PICC_IsNewCardPresent() && rfid_Read.PICC_ReadCardSerial()) {
        readDataFromRFIDTag();
    }
}

```

Figure 62 Loop function

In figure -62- declaration the Loop function for loading, unloading hangers, detecting RFID Tag to Read/ Write and, detect input from Keypad.

```

10:30:40.095 -> #sis000 Initialize and confirm Serial connected.
10:30:40.095 -> #sip000 Initialize Pins.
10:30:40.095 -> #sil000 Initialize Liquid Crystal Display.
10:30:42.312 -> #sie000 Initialize Ethernet Shield Network Module
10:30:44.901 -> #sie002 Assign a local IP Address for Ethernet Shield Network Module: 192.168.0.12
10:30:44.901 -> #sim000 Initialize MySQL Server connection
10:30:47.106 -> Connected to server version 5.5.5-10.1.38-MariaDB
10:30:47.744 -> #sir000 Initialize RFID-RC522.

```

Figure 63 Output of debug trace messages

In figure -63- showing the debug for processing the program in each step to can be tracing the issue when it happened, more details about debug code identify in figure -64- , -65- , -66-

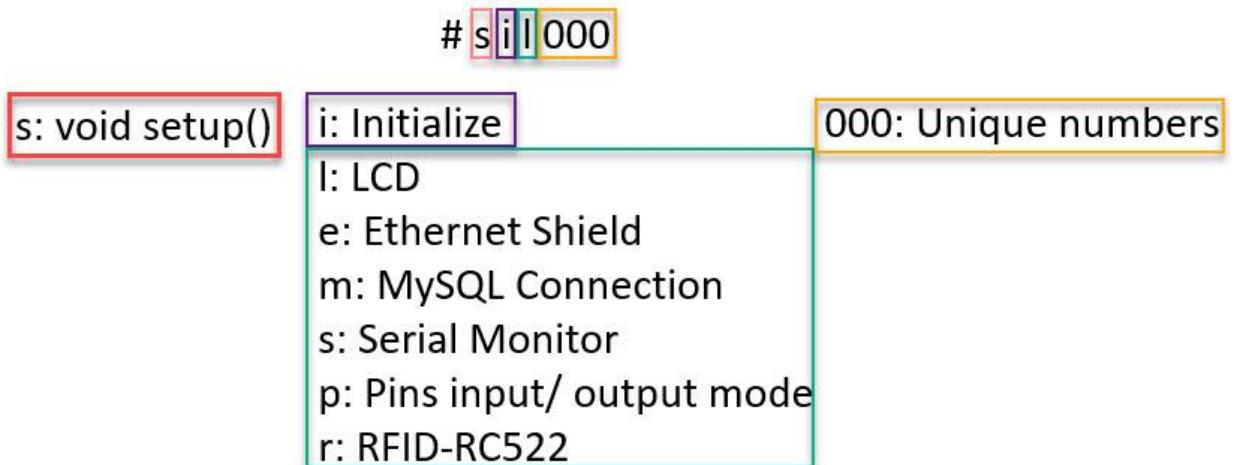


Figure 64 Debug identification for setup function



Figure 65 Debug identification for loading operation

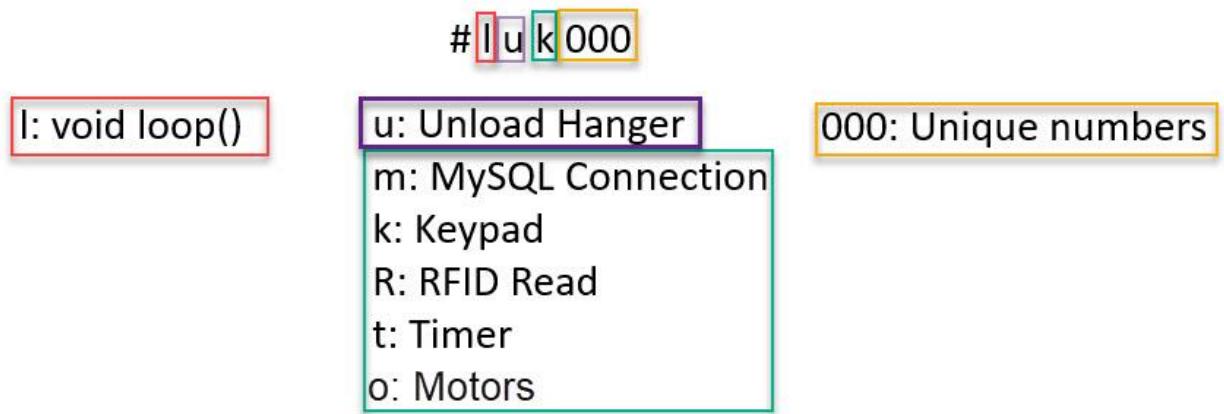


Figure 66 Debug identification for unloading operation

In figure # showing the database how setup the data for query;

```
time: ~
mysql> select * from hanger.hanger;
+----+-----+-----+-----+-----+
| id | hangerNumber | hangerStatus | idKey | timeLine |
+----+-----+-----+-----+-----+
| 1 | 1 | 2 | 6427 | 2019-05-18 13:03:35 |
| 2 | 2 | 2 | 8966 | 2019-05-17 22:35:03 |
+----+-----+-----+-----+-----+
2 rows in set (0.02 sec)

mysql> █
```

Figure 67 MySQL Query for selection statement

Source Code:

+ References:

- Waveshare Motor Control Shield - Created by Waveshare
https://www.waveshare.com/wiki/Motor_Control_Shield
 - Keypad 4x4 <https://playground.arduino.cc/code/keypad>
 - Ethernet Shield W5100 <https://www.arduino.cc/en/Reference/Ethernet>
 - LiquidCrystal I2C https://github.com/johnrickman/LiquidCrystal_I2C
 - MySQL Connector Arduino - Created by: Dr. Charles A. Bell
https://github.com/ChuckBell/MySQL_Connector_Arduino
 - RFID - RC522 <https://github.com/miguelbalboa/rfid>

+List of components:

- Arduino Mega 2560 R3 (Original).
 - Stepper Motor 28BYJ-48 v5.
 - Hybrid Stepper Motor MT-1704HS180A - NEMA 17 Stepper motor.
 - Ethernet Shield Network Module W5100 (From Aliexpress, China's clone).

- Liquid Crystal Display 16x2 (From Aliexpress, China's clone)(2 pieces).
- Motor Control Shield Waveshare SKU 9724
- Carriage MGW12H.
- Mini Rail MGW12.
- LED (4 pieces).

*/

```
////////////////////////////////////////////////////////////////////////
//#####:: :: Library Part    :: ::#####
////////////////////////////////////////////////////////////////////////
```

```
#include <Keypad.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Ethernet.h>
#include <SPI.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>
#include "DEV_Config.h"
#include "Motor.h"
#include <MFRC522.h>
```

```
////////////////////////////////////////////////////////////////////////
//#####:: :: lcd1 Part    :: ::#####
////////////////////////////////////////////////////////////////////////
```

```
LiquidCrystal_I2C lcd0(0x27, 20, 4);
LiquidCrystal_I2C lcd1(0x3f, 20, 4);
```

```

//*****//  

//#####:: :: Keypad Part :: ::#####//  

//*****//  
  

const byte rowsKeypad = 4;  

const byte columnsKeypad = 4;  

char keys[rowsKeypad][columnsKeypad] = {  

    {'1', '2', '3', 'A'},  

    {'4', '5', '6', 'B'},  

    {'7', '8', '9', 'C'},  

    {'*', '0', '#', 'D'}  

};  
  

byte rowKeypadPins[rowsKeypad] = {25, 24, 23, 22};  

byte columnKeypadPins[columnsKeypad] = {29, 28, 27, 26};  

char customKey;  
  

Keypad customKeypad(makeKeymap(keys), rowKeypadPins,  

columnKeypadPins, rowsKeypad, columnsKeypad);  
  

//*****//  

//#####:: :: ID Key Part :: ::#####//  

//*****//  
  

#define idLength 5  

#define rfidLength 2  

char tempId[idLength];  

char rfidOption[rfidLength] = "A";

```

```

byte idCounter = 0;

char testIdKey [idLength] = "1234"; // For debug purpose only.

//*****::: Networking Part :: :#####
//*****::: MySQL Query Part :: :####//

byte physicalAddress[] = {0xAA, 0xBB, 0xEE, 0xFF, 0xAA};
IPAddress serverAddress(192, 168, 0, 14);
EthernetClient ethernetClient;

MySQL_Connection MySQLConnection((Client *)&ethernetClient);
const char MySQLUser[] = "root";
const char MySQLPassword = "123456";

//*****::: MySQL Query Part :: :####//

char insertData[] = "INSERT INTO `hanger`.`hanger` ( hangerNumber,
hangerStatus, idKey) VALUES (%d, %d, %d)"; // For debug purpose only.
char hangerStatus[] = "SELECT hangerStatus FROM hanger.hanger WHERE
hangerStatus ='1' LIMIT 1";
char hangerNumber[] = "SELECT hangerNumber FROM hanger.hanger WHERE
hangerStatus = '1' LIMIT 1";
char updateHanger[] = "UPDATE hanger.hanger SET hangerStatus = %d, idKey
= %d WHERE hangerNumber = %d";
char updateHangerB[] = "UPDATE hanger.hanger SET hangerStatus = %u, idKey
= %u WHERE hangerNumber = %u"; // Backup Update Statement
char idSelector[] = "SELECT idKey FROM hanger.hanger WHERE idKey = %d";

```

```

const char unloadHa[] = "SELECT hangerNumber FROM hanger.hanger WHERE
idKey=%d";
char query[128];

//*****// ::  RFID-RC522 Part :: ::#####
//*****// ::  Other Parts :: ::#####
//*****//

#define SS_RFID 53
#define RST_RFID 49
MFRC522 rfid_Read(SS_RFID, RST_RFID);
MFRC522 mfrc522(SS_RFID, RST_RFID);

//*****// ::  Other Parts :: ::#####
//*****//


#define sensorSwitch 30 // Assigned Pin Number D40 to Sensor Switch.
#define led0 31 // For debug purpose only.
#define led1 32 // For debug purpose only.
#define led2 33 // For debug purpose only.
#define led3 34 // For debug purpose only.
int sensorSwitchStatus = 0; // Define the Sensor Switch Status in zero value
assuming is not pressed.

void setup() {
    Serial.begin(115200);

```

```

Serial.println(F(" #sis000 Initialize and confirm Serial connected."));
while (!Serial);
Serial.println(F(" #sip000 Initialize Pins."));
pinMode(led0, OUTPUT);
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
pinMode(sensorSwitch, INPUT);
Serial.println(F(" #sil000 Initialize Liquid Crystal Display."));
lcd0.init();
lcd0.backlight();
lcd0.setCursor(0, 0);
lcd0.print("SWSaver v0.03");
lcd1.init();
lcd1.backlight();
lcd1.setCursor(3, 0);
lcd1.print("Welcome...");
lcd1.setCursor(2, 1);
lcd1.print("2016 - 2019");
Serial.println(F(" #sie000 Initialize Ethernet Shield Network Module"));
ethernetHardwareChecker();
Serial.println(F(" #sim000 Initialize MySQL Server connection"));
mysqlServerConnection();
lcd0.clear();
lcd1.clear();
Serial.println(F(" #sir000 Initialize RFID-RC522."));
mfrc522.PCD_Init();
}

void loop() {

```

```

sensorSwitchStatus = digitalRead(sensorSwitch);
customKey = customKeypad.getKey();
lcd0.setCursor(0, 0);
lcd0.print("SWSaver v0.03");
lcd1.setCursor(4, 0);
lcd1.print("Enter ID Key");
lcd1.setCursor(2, 1);
lcd1.print("2016 - 2019");

if ( sensorSwitchStatus == 0) {
    unloadHanger();
} else {
    loadHanger();
}

digitalWrite(RST_RFID, HIGH);
digitalWrite(SS_RFID, HIGH);
rfid_Read.PCD_SoftReset;
rfid_Read.PCD_Init();
//rfid_Read.PCD_DumpVersionToSerial(); // For debug purpose only.
if (rfid_Read.PICC_IsNewCardPresent() &&
rfid_Read.PICC_ReadCardSerial()) {
    readDataFromRFIDTag();
}
}

void ethernetHardwareChecker () {
Ethernet.begin(physicalAddress);
// Check for Ethernet hardware present
if (Ethernet.hardwareStatus() == EthernetNoHardware) {

```

```

Serial.println(F(" #sie001 Ethernet shield was not found. Sorry, can't run
without hardware. :("));

while (true) {
    delay(1); // do nothing, no point running without Ethernet hardware
}

if (Ethernet.linkStatus() == LinkOFF) {
    Serial.println(F("Ethernet cable is not connected."));
}

Serial.print(F(" #sie002 Assign a local IP Address for Ethernet Shield Network
Module: "));

Serial.println(Ethernet.localIP());
}

```

```

void mysqlServerConnection() {
    lcd1.clear();
    lcd1.setCursor(1, 0);
    lcd1.print("Connecting to");
    lcd1.setCursor(2, 1);
    lcd1.print("MySQL Server");
    if (MySQLConnection.connect(serverAddress, 3306, MySQLUser,
        MySQLPassword))
    {
        delay(500);
        lcd1.clear();
    }
    else
    {
        lcd1.clear();
        Serial.println(F(" #sim001 Failed to connect, try again!"));
        lcd1.setCursor(1, 0);
    }
}

```

```

lcd1.print("Error connect");
lcd1.setCursor(0, 1);
lcd1.print(" Try again...\"");
delay(10000);
lcd1.clear();
}
}

void loadHanger() {
    Serial.println(F(" #ll000 Detected Sensor Switch a new coat on Hanger.\n\r
#llr000 Software turn off RFID device."));

    rfid_Read.PCD_SoftPowerDown();
    lcd1.clear();
    lcd1.setCursor(1, 0);
    lcd1.print(" Processing...\"");
    Serial.println(F(" #llm000 Running MySQL Connector operation for
hangerStatus statement."));
}

```

```

int hangerStatusCount = 0;
MySQL_Cursor *mysqlCursor = new MySQL_Cursor(&MySQLConnection);
mysqlCursor->execute(hangerStatus);
// char hangerStatus[] = "SELECT hangerStatus FROM hanger.hanger WHERE
hangerStatus ='1' LIMIT 1";
column_names *hangerStatusColumn = mysqlCursor->get_columns();
row_values *row = NULL;
do {
    row = mysqlCursor->get_next_row();
    if (row != NULL) {
        hangerStatusCount = atol(row->values[0]);
    }
}

```

```
    } while (row != NULL);
```

```
    Serial.println(String("") + "#llm001 MySQL Connector operation is finished  
and the Hanger Status available is: " + hangerStatusCount);
```

```
    if (hangerStatusCount == 1) {
```

```
        Serial.println(F(" #llm002 MySQL Connector operation for hangerNumber  
statement to select that first top available."));
```

```
    row_values *row = 0;
```

```
    int numHa = 0;
```

```
    mysqlCursor->execute(hangerNumber);
```

```
    // char hangerNumber[] = "SELECT hangerNumber FROM hanger.hanger  
WHERE hangerStatus = '1' LIMIT 1";
```

```
    column_names *hangerNumberColumn = mysqlCursor->get_columns();
```

```
    do {
```

```
        row = mysqlCursor->get_next_row();
```

```
        if (row != NULL) {
```

```
            numHa = atol(row->values[0]);
```

```
        }
```

```
    } while (row != NULL);
```

```
    Serial.println(String("") + "#llm003 MySQL Connector operation for  
hangerNumber statement is finished, and the Hanger Number is: " + numHa +  
"\n\r #llm004 MySQL Connector operation for hangerUpdate statement to set a  
new value for selected Hanger.\n\r #llm005 Generating a new 4 digits number as  
ID Key for the selected Hanger");
```

```
    randomSeed(analogRead(A0));
```

```
    unsigned int firstGenerator = random(1000, 8999);
```

```
    unsigned int secondGenerator = random(100, 899);
```

```
    unsigned int thirdGenerator = random(99);
```

```
    unsigned int fourthGenerator = random(3);
```

```
    unsigned int idKey = firstGenerator + secondGenerator + thirdGenerator +  
fourthGenerator;  
  
    Serial.println(String("") + " #ll001 Generator of ID Key is finished, the ID Key  
is: " + idKey);  
  
    int reserved = 2;  
  
    sprintf(query, updateHanger, reserved, idKey, numHa);  
  
    // updateHanger[] = "UPDATE hanger.hanger SET hangerStatus = %d, idKey =  
%d WHERE hangerNumber = %d";  
  
    mysqlCursor->execute(query);  
  
    Serial.println(F(" #llm006 MySQL Connector operation for hangerUpdate  
statement is finished, run SELECT statement for MySQL Server view the  
changes"));
```

```
/// Loop statement of Countdown Timer for User Selection
```

```
for (int timer = 0; timer <= 15;) {
```

```
    customKey = customKeypad.getKey();  
  
    lcd0.clear();  
  
    lcd1.clear();  
  
    lcd0.setCursor(0, 0);  
  
    lcd0.print("SWSaver v0.03");  
  
    lcd0.setCursor(14, 0);  
  
    lcd0.print(timer);  
  
    lcd0.setCursor(timer, 1);  
  
    lcd0.print("#####");  
  
    lcd1.setCursor(0, 0);  
  
    lcd1.print("Press A for NFC");  
  
    lcd1.setCursor(0, 1);  
  
    lcd1.print("or C to continue");
```

```

/// Waiting for decision by Keypad input or continue after deadline

if (customKey) {
    Serial.println(F(" #llk000 Detected input from Keypad.\n\r #llk001
Recording the types from Keypad."));
    tempId[idCounter] = customKey;
    lcd0.setCursor(14, 0);
    lcd0.print(tempId[idCounter]);
    idCounter++;
}

if (idCounter == rfidLength - 1) {
    lcd0.clear();
    lcd1.clear();
    if (strcmp(tempId, rfidOption)) {
        Serial.println(F(" #llk0002 Continue by display ID Key on LCD."));
        lcd0.setCursor(0, 0);
        lcd0.print("SW Saver v0.03");
        lcd1.setCursor(1, 0);
        lcd1.print("ID Key is:");
        lcd1.setCursor(12, 0);
        lcd1.print(idKey);
        lcd1.setCursor(3, 1);
        lcd1.print("2016 - 2019");
    }
    break;
}
else {
    Serial.println(F(" #llk003 Select A option for write ID Key on RFID
Tag."));
    lcd0.setCursor(0, 0);
}

```

```

lcd0.print("SWSaver v0.03");
lcd1.setCursor(0, 0);
lcd1.print("ID Key is:");
lcd1.setCursor(12, 0);
lcd1.print(idKey);
lcd1.setCursor(0, 1);
lcd1.print("Nearby NFC Tag");

//*****//
//#####:: :: RFID-RC522 Write :: :#####//
//*****//

Serial.println(F(" #llw000 Software turn on RFID device."));
mfrc522.PCD_SoftPowerUp();

// Prepare key - all keys are set to FFFFFFFFFFh at chip delivery from
the factory.

MFRC522::MIFARE_Key key;
for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

// Reset the loop if no new card present on the sensor/reader. This saves the
entire process when idle.

if ( ! mfrc522.PICC_IsNewCardPresent() {
    return;
}

// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial() {
    return;
}

```

```

Serial.print(F(" #llw001 Card UID:")); //Dump UID
for (byte i = 0; i < mfrc522.uid.size; i++) {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
}
MFRC522::PICC_Type piccType =
mfrc522.PICC_GetType(mfrc522.uid.sak);
Serial.println(String("") + " #llr002 PICC type: " +
mfrc522.PICC_getTypeName(piccType));

byte buffer[34];
byte block;
MFRC522::StatusCode status;
byte len;

Serial.setTimeout(20000L); // wait until 20 seconds for input from serial
sprintf(buffer, "%d", idKey);
block = 1;
//Serial.println(F("Authenticating using key A..."));
status =
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
block, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
    Serial.print(String("") + " #llw003 PCD_Authenticate() failed: " +
mfrc522.GetStatusCodeName(status));
    return;
}
else Serial.println(String("") + " #llw004 PCD_Authenticate() success: ");

// Write block
status = mfrc522.MIFARE_Write(block, buffer, 16);

```

```

    if (status != MFRC522::STATUS_OK) {
        Serial.print(String("") + "#llw004 MIFARE_Write() failed: " +
mfrc522.GetStatusCodeName(status));
        return;
    }
    else Serial.println(String("") + "#llw005 MIFARE_Write() success: ");

    block = 2;
    Serial.println(F(" #llw006 Authenticating using key A..."));
    status =
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
block, &key, &(mfrc522.uid));
    if (status != MFRC522::STATUS_OK) {
        Serial.print(String("") + "#llw007 PCD_Authenticate() failed: " +
mfrc522.GetStatusCodeName(status));
        return;
    }

// Write block
status = mfrc522.MIFARE_Write(block, &buffer[16], 16);
if (status != MFRC522::STATUS_OK) {
    Serial.print(String("") + "#llw008 MIFARE_Write() failed: " +
mfrc522.GetStatusCodeName(status));
    return;
}
else Serial.println(F(" #llw009 MIFARE_Write() success: "));

Serial.println(" ");
mfrc522.PICC_HaltA(); // Halt PICC
mfrc522.PCD_StopCrypto1(); // Stop encryption on PCD

```

```

        break;
    }
}

delay(800);
timer++;
}

/// End Loop statement of Countdown Timer
lcd0.clear();
lcd1.clear();
lcd0.setCursor(1, 0);
lcd0.print("SWSaver v0.03");
lcd1.setCursor(1, 0);
lcd1.print("ID Key is:");
lcd1.setCursor(12, 0);
lcd1.print(idKey);
lcd1.setCursor(3, 1);
lcd1.print("2016 - 2019");

Serial.println(F(" #llo000 Passing the Hanger Number to the Stepper
Motors"));

if (numHa == 1) {
    Serial.println(F(" #llo001 Motor Controller operating the Hanger number
1"));
    baseMotor1();
    linearMotor1();
    linearMotorReturnToHome1();
    baseMotorReturnToHome1();
}

```

```

    delay(500);

}

else if (numHa == 2) {

    Serial.println(F(" #llo002 Motor Controller operating the Hanger number
2"));

    baseMotor2();

    linearMotor2();

    linearMotorReturnToHome2();

    baseMotorReturnToHome2();

    delay(500);

}

Serial.println(F(" #ll002 Operation for checking Hanger, number Hanger, and
ID Key generating is finished."));

delay(2000);

lcd1.clear();

} else {

    Serial.println(F(" #ll003 No place availalble, Out put the message on Display
about the status!"));

    lcd1.clear();

    lcd1.setCursor(3, 0);

    lcd1.print("It Is Full");

    lcd1.setCursor(3, 1);

    lcd1.print("Try later");

    delay(5000);

}

delay(5000);

Serial.println(String("") + " #lm007 Memory usage before deleting the MySQL
Cursor / Packets: " + memoryUsage());

delete mysqlCursor;

```

```

Serial.println(String("") + " #llm008 Memory usage after deteling the MySQL
Cursor / Packets: " + memoryUsage());

lcd0.clear();
lcd1.clear();
clearTemp();
}

void unloadHanger() {
    if (customKey) {
        Serial.println(F(" #luk000 Detected input from Keypad.\n\r #lux0001
Recording the types from Keypad."));
        tempId[idCounter] = customKey;
        lcd1.setCursor(idCounter, 0);
        lcd1.print(tempId[idCounter]);
        idCounter++;
    }
    if (idCounter == idLength - 1) {
        int idTyped = atol(tempId);
        if (idTyped != 0) {
            Serial.println(String("") + " #luk001 The ID Key recorded ( " + idTyped + " )"
+ "\n\r #lum000 Running MySQL Connector operation for idSelector
Statement.");
            MySQL_Cursor *mysqlCursor = new MySQL_Cursor(&MySQLConnection);

            row_values *row = NULL;
            int idDB = 0;
            sprintf(query, idSelector, idTyped);
            mysqlCursor->execute(query);
            column_names *columns = mysqlCursor->get_columns();
            do {
                row = mysqlCursor->get_next_row();
                if (row != NULL) {

```

```
    idDB = atol(row->values[0]);  
}  
} while (row != NULL);  
  
Serial.println(F(" #lum001 MySQL Connector operation is finished.\n\r  
#lum002 Verifying the typed ID Key with Database."));
```

```
if (!strcmp(idTyped, idDB)) {  
    Serial.println(F(" #luk002 Entered correct ID Key."));  
    lcd1.clear();  
    lcd1.setCursor(0, 0);  
    lcd1.print("Please wait...");
```

```
Serial.println(String("") + " #lum003 Running MySQL Connector  
operation for unloadHa statement.\n\r ID of Database: ." + idDB + " and ID Key  
Typed: " + idTyped);
```

```
int numHa = 0;  
row_values *row = NULL;  
sprintf(query, unloadHa, idTyped);  
mysqlCursor->execute(query);  
column_names *columns = mysqlCursor->get_columns();  
do {
```

```
    row = mysqlCursor->get_next_row();  
    if (row != NULL) {  
        numHa = atol(row->values[0]);  
    }
```

```
} while (row != NULL);
```

```
Serial.println(String("") + " #lum004 The Hanger Number from unloadHa  
statement is: " + numHa + "\n\r #lum005 The ID Key belong to the Hanger  
number: ");
```

```
if (numHa == 1) {  
    Serial.println(numHa);
```

```

baseMotor1();

linearMotor1();

linearMotorReturnToHome1();

baseMotorReturnToHome1();

}

} else if (numHa == 2 ) {

Serial.println(numHa);

baseMotor2();

linearMotor2();

linearMotorReturnToHome2();

baseMotorReturnToHome2();

}

lcd1.setCursor(0, 1);

lcd1.print("Check the hanger");

delay(2500);

lcd1.clear();

Serial.println(" #lum006 MySQL Connector operation for hangerNumber
statement is finished.\n\r #lum007 Running MySQL Connector ooperation for
updateHanger statement.");

int unreserved = 1;

int idKey = 0;

sprintf(query, updateHanger, unreserved, idKey, numHa);

// updateHanger[] = "UPDATE hanger.hanger SET hangerStatus = %d,
idKey = %d WHERE hangerNumber = %d";

mysqlCursor->execute(query);

Serial.println(F(" #lum008 MySQL Connector operation is finished."));

}

else {

Serial.println(F(" #luk003 Entered incorrect ID Key."));

lcd1.clear();

lcd1.setCursor(0, 0);

```

```

lcd1.print("Incorrect ID Key");
lcd1.setCursor(1, 1);
lcd1.print("Enter it again");
delay(5000);
lcd1.clear();

}

Serial.println(String("") + " #lum009 Memory usage before deleting the
MySQL Cursor / Packets: " + memoryUsage());

delete mysqlCursor;

Serial.println(String("") + " #lum010 Memory usage after deteling the
MySQL Cursor / Packets: " + memoryUsage());

} else {

lcd1.clear();
lcd1.print("ID Key doesnot");
lcd1.setCursor(0, 1);
lcd1.print("start with/or 0");
delay(5000);
lcd1.clear();

} clearTemp();

}

//*****//
//#####:: :: RFID-RC522 Read :: ::#####//
//*****//

void readDataFromRFIDTag() {
MFRC522::MIFARE_Key KeyR;

```

```

MFRC522::StatusCode statusR;
byte bufferR[18];
byte len = 18;
byte Block = 1;

Serial.println(F(" #luR0001 Read data from RFID Tag"));
for (uint8_t i = 0; i < 6; i++) KeyR.keyByte[i] = 0xFF;
statusR =
rfid_Read.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
Block, &KeyR, &(rfid_Read.uid));
if (statusR != MFRC522::STATUS_OK) {
    statusR =
rfid_Read.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_B,
Block, &KeyR, &(rfid_Read.uid));
    if (statusR != MFRC522::STATUS_OK) {
        Serial.println(String("") + " #luR0002 Authentication failed: " +
rfid_Read.GetStatusCodeName(statusR));
        return;
    }
}
statusR = rfid_Read.MIFARE_Read(Block, bufferR, &len);
if (statusR != MFRC522::STATUS_OK) {
    Serial.println(String("") + " #luR0003 Reading failed: " +
rfid_Read.GetStatusCodeName(statusR));
    return;
}
for (uint8_t i = 0; i < 16; i++) {
    if (bufferR[i] != 32) {
        //Passing the value to out of loop statement
    }
}

```

```

Serial.println(F(" #luR0004 Convert the Byte array to unsigned Interger number
with operation '\atol\'."));

unsigned int rfid_IDKey = atol(bufferR);

Serial.println(String("") + "\n\r #luR0005 ID Key stored on Tag is: " +
rfid_IDKey + (".\n\r #luR0006 Running MySQL Connector operation for
idSelector Statement."));

MySQL_Cursor *mysqlCursor = new MySQL_Cursor(&MySQLConnection);

row_values *row = NULL;
int idDB = 0;
sprintf(query, idSelector, rfid_IDKey);
mysqlCursor->execute(query);
column_names *columns = mysqlCursor->get_columns();
do {
    row = mysqlCursor->get_next_row();
    if (row != NULL) {
        idDB = atol(row->values[0]);
    }
} while (row != NULL);

Serial.println(F(" #luR0007 MySQL Connector operation is finished.\n\r
#luR0008 Verifying the typed ID Key with Database."));

if (!strcmp(rfid_IDKey, idDB)) {
    Serial.println(F(" #luR0009 Entered correct ID Key."));
    lcd1.clear();
    lcd1.setCursor(0, 0);
    lcd1.print("Please wait...");
}

Serial.println(F(" #luR0010 Running MySQL Connector operation for
unloadHa statement."));

Serial.println(idDB);

```

```

Serial.println(rfid_IDKey);

int numHa = 0;

row_values *row = NULL;

sprintf(query, unloadHa, rfid_IDKey);

mysqlCursor->execute(query);

column_names *columns = mysqlCursor->get_columns();

do {

    row = mysqlCursor->get_next_row();

    if (row != NULL) {

        numHa = atol(row->values[0]);

    }

} while (row != NULL);

Serial.println(String("") + (" #luR0011 The Hanger Number from unloadHa
statement is: ") + numHa + ("\n\r #luR0012 The ID Key belong to the Hanger
number: "));

if (numHa == 1) {

    Serial.println(numHa);

    baseMotor1();

    linearMotor1();

    linearMotorReturnToHome1();

    baseMotorReturnToHome1();

}

} else if (numHa == 2 ) {

    Serial.println(numHa);

    baseMotor2();

    linearMotor2();

    linearMotorReturnToHome2();

    baseMotorReturnToHome2();

}

lcd1.setCursor(0, 1);

```

```

lcd1.print("Check the hanger");
delay(200);
lcd1.clear();

Serial.println(" #luR0013 MySQL Connector operation for hangerNumber
statement is finished.\n\r #luR0014 Running MySQL Connector ooperation for
updateHanger statement.");

int unreserved = 1;
int idKey = 0;
sprintf(query, updateHanger, unreserved, idKey, numHa);
// updateHanger[] = "UPDATE hanger.hanger SET hangerStatus = %d, idKey =
%d WHERE hangerNumber = %d";
mysqlCursor->execute(query);
Serial.println(F(" #luR0015 MySQL Connector operation is finished."));

}

else {
    Serial.println(F(" #luR0016 Entered incorrect ID Key."));
    lcd1.clear();
    lcd1.setCursor(0, 0);
    lcd1.print("Incorrect ID Key");
    lcd1.setCursor(1, 1);
    lcd1.print("Enter it again");
    delay(5000);
}

Serial.println(String("") + (" #luR0017 Memory usage before deleting the
MySQL Cursor / Packets: ") + memoryUsage());

delete mysqlCursor;

Serial.println(String("") + (" #luR0018 Memory usage after deteling the MySQL
Cursor / Packets: ") + memoryUsage());

rfid_Read.PICC_HaltA();
rfid_Read.PCD_StopCrypto1();

```

```
rfid_Read.PCD_Idle;  
digitalWrite(RST_RFID, LOW);  
digitalWrite(SS_RFID, LOW);  
lcd1.clear();  
}  
  
//*****  
//##### Motor Control for Position 1 #####//  
//*****
```

```
void baseMotor1()  
{  
    digitalWrite(led0, HIGH);  
    delay(500);  
    Motor_Init(MOTOR_DEV_1, MOTOR_DEV_2);  
    Motor_Trunk(MOTOR_DEV_1, 750); //750  
    digitalWrite(led0, LOW);  
}
```

```
void baseMotorReturnToHome1()  
{  
    digitalWrite(led1, HIGH);  
    delay(500);  
    Motor_Init0(MOTOR_DEV_1, MOTOR_DEV_2);  
    Motor_Trunk0(MOTOR_DEV_1, 750); //750  
    digitalWrite(led1, LOW);  
}
```

```
void linearMotor1()  
{
```

```
digitalWrite(led2, HIGH);
delay(500);
Motor_Init(MOTOR_DEV_1, MOTOR_DEV_2);
Motor_Trun(MOTOR_DEV_2, 90); //90
digitalWrite(led2, LOW);
}
```

```
void linearMotorReturnToHome1()
{
digitalWrite(led3, HIGH);
delay(500);
Motor_Init0(MOTOR_DEV_1, MOTOR_DEV_2);
Motor_Trun0(MOTOR_DEV_2, 90); //90
digitalWrite(led3, LOW);
}
```

```
////////////////////////////////////////////////////////////////////////
//##### Motor Control for Position 2 #####
////////////////////////////////////////////////////////////////////////
```

```
void baseMotor2()
{
digitalWrite(led0, HIGH);
delay(500);
Motor_Init(MOTOR_DEV_1, MOTOR_DEV_2);
Motor_Trun(MOTOR_DEV_1, 1600); //1600
digitalWrite(led0, LOW);
}
```

```
void baseMotorReturnToHome2()
{
    digitalWrite(led1, HIGH);
    delay(500);
    Motor_Init0(MOTOR_DEV_1, MOTOR_DEV_2);
    Motor_Trunk0(MOTOR_DEV_1, 1600); //1600
    digitalWrite(led1, LOW);
}
```

```
void linearMotor2()
{
    digitalWrite(led2, HIGH);
    delay(500);
    Motor_Init(MOTOR_DEV_1, MOTOR_DEV_2);
    Motor_Trunk(MOTOR_DEV_2, 90); //90
    digitalWrite(led2, LOW);
}
```

```
void linearMotorReturnToHome2()
{
    digitalWrite(led3, HIGH);
    delay(500);
    Motor_Init0(MOTOR_DEV_1, MOTOR_DEV_2);
    Motor_Trunk0(MOTOR_DEV_2, 90);
    digitalWrite(led3, LOW);
}
```

```
void clearTemp() {
    while (idCounter != 0) {
```

```
tempId[idCounter--] = 0;
}
return;
}

int memoryUsage()
{
    extern char __bss_end;
    extern char *__brkval;
    int free_memory;
    if ((int)__brkval == 0)
        free_memory = ((int)&free_memory) - ((int)&__bss_end);
    else
        free_memory = ((int)&free_memory) - ((int)__brkval);
    return free_memory;
}
```

Reference:

SIEMENS Solid Edge Training & Certification for Students
<https://www.udemy.com/solidedgecertification1/>

MECHANISMS AND MOTION - ROBOTICS FOCUS
<https://www.udemy.com/mechanisms/>

Arduino Reference

<https://www.arduino.cc/reference/en/>

Ubuntu Documentation

<https://help.ubuntu.com/community/Installation>

MySQL Documentation

<https://dev.mysql.com/doc/>

Troubleshooting

Github.com and <https://stackoverflow.com/>