#define mcw 7

#define mccw 6

#define mpwm 5

#define BallMotCW 22

#define BallMotCCW 23

#define BallMotPWM 8

#define TrigVision 10

#define Reject 11

#define ProxReject A0

int valProxReject = 0;

int sp = 50;

String RejecStatus;

int RejectPoint = 0;

void FuncTrigVision();

unsigned long time1P = 0; //time1 Previous

unsigned long time1C; //time1 Current

int t1F = 0; //time1 Finish

void rfidDetect();

void feedBasket();

//==================== RFID ID-12 Config ======================================

#include <SoftwareSerial.h> // ==============================

SoftwareSerial rfidSerial(19, 1); //(RX, TX) // ==============================

//Register your RFID tags here // ==============================

char tag1[13] = "090089C4EFAB";//Admin Card // ==============================

char tag2[13] = "010E0B8769EA";//Pruksa Card // ==============================

char tag3[13] = "55004D187A7A";//Mhee Card // ==============================

char tag4[13] = "55004CE12ED6";//Earth Card // ==============================

char tag5[13] = "010E669C6C99";//Aey Card // ==============================

char tag6[13] = "010E680FADC5";//Art // ==============================

char tag7[13] = "55004CA6843B";//Mo // ==============================

char tag8[13] = "6B003472D8F5";//Art Apartment// ==============================

char tag9[13] = "4E006C8C3A94"; //Son // ==============================

char tag10[13] = "4900C970DC2C"; // ==============================

char tagString[13]; // ==============================

int index = 0; // ==============================

boolean reading = false; // ==============================

//==================== RFID ID-12 Config ======================================

void setup()

{

pinMode(mcw,OUTPUT);

pinMode(mccw,OUTPUT);

pinMode(mpwm,OUTPUT);

pinMode(BallMotCW,OUTPUT);

pinMode(BallMotCCW,OUTPUT);

pinMode(BallMotPWM,OUTPUT);

pinMode(TrigVision,OUTPUT);

pinMode(Reject,OUTPUT);

pinMode(ProxReject,INPUT);

Serial.begin(9600);

Serial1.begin(9600);

Serial2.begin(2400);

}

void loop()

{

rfidDetect();

//feedBasket();

// digitalWrite(mcw,LOW);

// digitalWrite(mccw,HIGH);

// analogWrite(mpwm,sp);

valProxReject = digitalRead(ProxReject);

time1C = millis();

if(Serial2.available())

{

Serial.print("Serial available ");

RejecStatus = Serial2.readString();

Serial.print("RejecStatus : "); Serial.print(RejecStatus);

Serial.print("RejectPoint : "); Serial.println(RejectPoint);

if(RejecStatus.substring(0,2) == "NG")

{

RejectPoint = 1;

Serial.print("Reject");

}

else if(RejecStatus.substring(0,2) == "OK")

{

Serial.print("Product Pass");

}

}

//Serial.println(valProxReject);

if(RejectPoint == 1 && valProxReject == 0)

{

delay(500);

digitalWrite(Reject,HIGH);

delay(1000);

digitalWrite(Reject,LOW);

delay(2000);

RejectPoint = 0;

}

else if(valProxReject == 1)

{

digitalWrite(Reject,LOW);

}

//

if(Serial.available()> 0)

{

char keyCommand = Serial.read();

if(keyCommand == '+')

{

sp += 5;

if(sp>255) sp=255;

}

if(keyCommand == '-')

{

sp -= 5;

if(sp<0) sp=0;

}

if(keyCommand == 'a')

{

digitalWrite(mcw,LOW);

digitalWrite(mccw,HIGH);

analogWrite(mpwm,sp);

}

if(keyCommand == 'd')

{

digitalWrite(mcw,HIGH);

digitalWrite(mccw,LOW);

analogWrite(mpwm,sp);

}

if(keyCommand == 's')

{

digitalWrite(mcw,HIGH);

digitalWrite(mccw,LOW);

analogWrite(mpwm,0);

}

if(keyCommand == 'r')

{

digitalWrite(Reject,HIGH);

delay(1000);

digitalWrite(Reject,LOW);

}

if(keyCommand == 'f')

{

digitalWrite(Reject,LOW);

}

if(keyCommand == 't')

{

FuncTrigVision();

}

if(keyCommand == 'g')

{

digitalWrite(TrigVision,LOW);

}

}

// Serial.println(sp);

// Serial.println(millis());

}

//============================= Start RFID ID-20 Function ====================================

void rfidDetect()

{

if(Serial1.available()>= 0)

{

//Serial.println("Get Serial0");

}

while (Serial1.available())

{

int readByte = Serial1.read(); //read next available byte

//Serial.print(readByte,HEX); //Debug

delay(1);

if ((reading == true) && (readByte != 3) && (readByte != 10) && (readByte != 13)) {

//store the tag

tagString[index] = readByte;

index ++;

}

if ((readByte == 2) && (reading == false)) reading = true; //begin of tag

if ((readByte == 3) && (reading == true)) reading = false; //end of tag

}

if (reading == false) {

checkTag(tagString); //Check if it is a match

clearTag(tagString); //Clear the char of all value

index = 0;

}

}

void checkTag(char tag[])

{

if (strlen(tag) == 0) return;

if (compareTag(tag, tag1))

{

Serial2.write("Box1:IN ");

// Serial2.println(tag);// Check Card tag HEX Code

// Serial.write("B01");

FuncTrigVision();

}

else if (compareTag(tag, tag2))

{

Serial.write("Box2:IN ");

// Serial2.println(tag);

// Serial.write("B02");

FuncTrigVision();

}

else if(compareTag(tag, tag3))

{

Serial.print("Box3:IN ");

// Serial2.println(tag);

// Serial.write("B03");

FuncTrigVision();

}

else if(compareTag(tag, tag4))

{

Serial.print("Box4:IN ");

// Serial2.println(tag);

// Serial.write("B04");

FuncTrigVision();

}

else if(compareTag(tag, tag5))

{

Serial.print("Box5:IN ");

// Serial2.println(tag);

// Serial.write("B05");

FuncTrigVision();

}

else if(compareTag(tag, tag6))

{

Serial.print("Box6:IN ");

// Serial2.println(tag);

// Serial.write("B06");

FuncTrigVision();

}

else if(compareTag(tag, tag7))

{

Serial.print("Box7:IN ");

// Serial2.println(tag);

// Serial.write("B07");

FuncTrigVision();

}

else if(compareTag(tag, tag8))

{

Serial.print("Box8:IN ");

// Serial2.println(tag);

// Serial.write("B08");

FuncTrigVision();

}

else if(compareTag(tag, tag9))

{

Serial.print("Box9:IN ");

// Serial2.println(tag);

// Serial.write("B09");

FuncTrigVision();

}

else if(compareTag(tag, tag10))

{

Serial.print("Box10:IN ");

// Serial2.println(tag);

// Serial.write("B10");

FuncTrigVision();

}

else

{

// Serial.print("Unknow Box : ");

// Serial.println(tag); //read out any unknown tag

}

}

//End checkTag()

void clearTag(char one[]) { //clear the char array by filling with null - ASCII 0

//Will think same tag has been read otherwise

for (int n = 0; n < strlen(one); n++) {

one[n] = 0;

}

}

boolean compareTag(char one[], char two[])

{

//compare two value to see if same, strcmp not working 100% so we do this

if (strlen(one) == 0) return false; //empty

for (int n = 0; n < 12; n++) {

if (one[n] != two[n]) return false;

}

return true; //no mismatches

}

//============================= End RFID ID-20 Function ====================================

//=============================== FuncTrigVision =======================================

void FuncTrigVision()

{

digitalWrite(TrigVision,HIGH);

delay(300);

digitalWrite(TrigVision,LOW);

}

//================================== Feed Basket =======================================

void feedBasket()

{

if(time1C - time1P >= 2000 && t1F == 0)

{

time1P = time1C;

t1F = 1;

digitalWrite(BallMotCW,HIGH);

digitalWrite(BallMotCCW,LOW);

analogWrite(BallMotPWM,100);

}

if(time1C - time1P >= 1000 && t1F == 1)

{

time1P = time1C;

t1F = 2;

digitalWrite(BallMotCW,LOW);

digitalWrite(BallMotCCW,HIGH);

analogWrite(BallMotPWM,100);

}

if(time1C - time1P >= 1000 && t1F == 2)

{

time1P = time1C;

t1F = 0;

digitalWrite(BallMotCW,LOW);

digitalWrite(BallMotCCW,HIGH);

analogWrite(BallMotPWM,0);

}

}