# Arduino Code

#include "Wire.h" // For I2C

#include "LCD.h" // For LCD

#include "LiquidCrystal\_I2C.h" // Added library\*

#include <Servo.h>

#include "HX711.h"

String str;

LiquidCrystal\_I2C lcd(0x27,2,1,0,4,5,6,7);

//weight pin

#define DOUT 3

#define CLK 2

HX711 scale(DOUT, CLK);

//motor controller

int in1 = 7;

int in2 = 8;

int ENA = 5;

int SPEED = 250;

//COLOR INITIATION:

#define S0 53

#define S1 52

#define S2 51

#define S3 50

#define sensorOut 49

//frequency init

int freq\_R = 0;

int freq\_G = 0;

int freq\_B = 0;

Servo StackGate;

Servo LoadGate;

Servo WashGate;

Servo Sorter;

int angle = 60;

int wait = 1000;

int i=1; //for loop counter

int j = 0;

int k= 0;

int color=0;

int weight = 0;

int colorOK = 0;

int weightOK = 0;

void setup() {

StackGate.attach(10);

LoadGate.attach(9);

WashGate.attach(11);

Sorter.attach(4);

LoadGate.write(127);

StackGate.write(90);

WashGate.write(25);

Sorter.write(90); //motor control

pinMode(in1,OUTPUT);

pinMode(in2,OUTPUT);

pinMode(ENA,OUTPUT);

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

pinMode(S0, OUTPUT);

pinMode(S1, OUTPUT);

pinMode(S2, OUTPUT);

pinMode(S3, OUTPUT);

pinMode(sensorOut, INPUT);

// Setting frequency-scaling to 20%

digitalWrite(S0,HIGH);

//digitalWrite(S1,LOW);

//Serial.println("Press T to tare");

scale.set\_scale(-96650); //Calibration Factor obtained from first sketch

scale.tare();

Serial.begin(115200);

lcd.begin (20,4); // 20 x 4 LCD module

lcd.setBacklightPin(3,POSITIVE); // BL, BL\_POL

lcd.setBacklight(HIGH);

lcd.setCursor(0,0);

lcd.print("IOT SORTER v2.1");

}

void loop()

{ lcd.setCursor(0,1);

lcd.print("Initializing...");

delay(800);

lcd.setCursor(0,1);

lcd.print("Opening gate...");

delay(800);

StackGate.write(135);

delay(wait);

StackGate.write(90);

delay(wait);

//NOW SENSE COLOR:

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Sensing Color...");

delay(800);

col:

color = getColor();

if(color ==1 || color==2 || color==3)

{ colorOK = 1;

if(color==1)

{ lcd.setCursor(0,2);

lcd.print("C: RED");

delay(200); }

else if(color==2){

lcd.setCursor(0,2);

lcd.print("C: GREEN");

delay(200); }

else if(color==3){

lcd.setCursor(0,2);

lcd.print("C: BLUE");

delay(200);

}

else

{

;

}

}

else

{

colorOK = 0;

lcd.setCursor(0,2);

lcd.print("C: UNKNOWN");

delay(200);

}

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Releasing...");

delay(200);

delay(wait);

StackGate.write(50);

delay(wait);

StackGate.write(90);

delay(2000); //Ball Passing Time

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Releasing...");

delay(200);

LoadGate.write(127); //accept ball in load gate

delay(2000);

// NOW SENSE WEIGHT:

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Weighing...");

delay(200);

weight = getWeight();

if(weight<100 && weight >15)

{

weightOK = 1;

}

else

{

weightOK = 0;

}

lcd.setCursor(13,2);

lcd.print("W: ");

lcd.setCursor(16,2);

lcd.print(weight);

lcd.setCursor(19,2);

lcd.print("g");

delay(200);

///IF COLOR AND WEIGHT ALL ARE OK THEN LET GO OTHERWISE BLOCK:

if(colorOK ==1 && weightOK==1)

{

//LET IT GO : rotate loadgate to 30 degree position

LoadGate.write(30);

delay(2000);

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Accepted...");

delay(200);

//Serial.print("All is ok");

LoadGate.write(127);

delay(300);

}

else

{

//Block IT:rotate loadgate to 180 degree position

//Explain Which Is error:

if(colorOK==0 && weightOK!=0)

{

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Color UNKNOWN...");

delay(200);

///Serial.print("Color NOT RECOGNIZED");

}

else if(weightOK==0 && colorOK!=0)

{

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Weight not in range");

delay(200);

//Serial.print("Weight Not In Range...");

}

else if(weightOK==0 && colorOK==0)

{

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Both REJECTED...");

delay(200);

}

//change mechanism

delay(500);

LoadGate.write(135);

delay(1000);

LoadGate.write(145);

delay(1000);

LoadGate.write(180);

delay(2000);

LoadGate.write(127);

delay(2000);

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Terminating...");

delay(800);

goto end;

}

//change mechanism

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Washing...");

delay(200);

analogWrite(ENA, SPEED);

for(j=30;j<80;)

{WashGate.write(j);

delay(400);

j=j+10;}

for(j=80;j>30;)

{WashGate.write(j);

delay(400);

j=j-10;}

analogWrite(ENA, 0);

delay(300);

//align sorter and then return wash gate

if(color==1)

{

Sorter.write(30);

}

else if(color==2)

{

Sorter.write(98);

}

else if(color==3)

{

Sorter.write(145);

}

else

{

Sorter.write(90);

Sorter.write(110);

Sorter.write(98);

}

WashGate.write(180);

delay(800);

WashGate.write(30);

// Start Sorter now

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Sorting...");

delay(200);

//called earlier sorter

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Finishing...");

delay(200);

delay(500);

end:

lcd.setCursor(0,2);

lcd.print(" ");

i++;

str =String("color=")+ String(color) + String("&weight=")+String(weight);

Serial.print(str);

}

int getColor()

{

int COL2\_R = 243;

int COL2\_G =144;

int COL2\_B =138;

int COL3\_R = 414;

int COL3\_G = 419;

int COL3\_B =419;

//read RED

digitalWrite(S2,LOW);

digitalWrite(S3,LOW);

freq\_R = pulseIn(sensorOut, LOW);

delay(100);

// read Green

digitalWrite(S2,HIGH);

digitalWrite(S3,HIGH);

freq\_G = pulseIn(sensorOut, LOW);

delay(100);

// read BLUE

digitalWrite(S2,LOW);

digitalWrite(S3,HIGH);

freq\_B = pulseIn(sensorOut, LOW);

delay(100);

int send = 0;

//NOW COMPARE VALUES IF RED OR GREN OR BLUE

// RED:1 GREEN:2, BLUE:3, NO MATCH: 4

if(freq\_R < freq\_G && freq\_R < freq\_B)

{

//color = ORANGE

send = 1; //if needed

}

else if(freq\_G < freq\_R && freq\_G < freq\_B)

{

//color = GREEN

send = 2; //if needed

}

else if(freq\_B < freq\_R) //&& freq\_B < freq\_G

{

//color = BLUE

send = 3; //if needed

}

else

{

//color = UNKNOWN;

send = 4; //if needed

//Serial.println("NOT IN ANY RANGE.");

}

Serial.print(freq\_R);

Serial.print("\_: ");Serial.print(freq\_G);

Serial.print("\_: ");Serial.print(freq\_B);

Serial.println("\_\_\_");

return send; //send;

}

int getWeight()

{

int weightm = 1000\*(scale.get\_units()/24);

//Serial.print("Weight: ");Serial.print(weightm);Serial.println(" gm");

return weightm;

}