int trigPin = 7; // Trigger (White)

int echoPin = 6; // Echo (Yellow)

long duration, Dist;

int RedLED = 2; // Interior LED

int GreenLED = 3;

int OrangeLED = 4;

int GreenSignal = 12; // Signal Exterior

int RedSignal = 13;

int EmergencyStop = A5;

int EButton = 0;

#include <Servo.h>

int WaterPin = 8;

int SoapPin = 9;

int ScrubPin = 10;

Servo Servo1; //Wash

Servo Servo2; //Soap

Servo Servo3; //Scrubber

const int servoMinDegrees = 0;// min and max degree of rotation

const int servoMaxDegrees = 180;

int ServoIncrease = 10; // increment of rotation by servo

int reverse = -1; // allows the servo motor to move in a reverse direction

float WaterStartTime = -1;

float WaterTime = 0;

int Servo1Pos = 90; // stop servo 1

float SoapStartTime = -1;

float SoapTime = 0;

int Servo2Pos = 90;

float ScrubStartTime = -1;

float ScrubTime = 0;

int Servo3Pos = 90;

float WashStartTime = -1;

float WashTime = 0;

boolean isDistZero = true; // calibration purposes (for proximity sensor)

boolean Stop = false; // emergency stop is off currently

void setup() {

//Serial Port begin

Serial.begin (9600);

//Define inputs and outputs

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(EmergencyStop, INPUT);

Servo1.write (Servo1Pos); // move the servo to initial position (servowrite function)

Servo1.attach(8);

Servo2.write (Servo2Pos);

Servo2.attach(9);

Servo3.write (Servo3Pos);

Servo3.attach (10);

}

void loop() {

// sending out + receiving of signals from proximity sensor to get readings

digitalWrite(trigPin, LOW);

delayMicroseconds(5);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// emergency stop process, when activated, the whole process would stop operation. Red LED light up

EButton = analogRead (A5);

if (EButton == 1023){

if (Stop){ // stop is true (initial condition : false)

Restart();

return;

}

Serial.println("Warning!");

Serial.print ("EMERGENCY STOP!");

Serial.println();

digitalWrite (GreenLED, LOW);

digitalWrite (RedLED, HIGH);

digitalWrite (OrangeLED, LOW);

Servo1.write (90);

Servo2.write (90);

Servo3.write (90);

Stop = true; // new condition to show that emergency stop is activated

delay(1000);

return;

} else if (Stop == true) { // loop function to ensure the emergency process works and whole operation stops

return;

}

duration = pulseIn(echoPin, HIGH);

// Convert the time into a distance

Dist = (duration/2) / 29.1; // Changing the time into centimeters

if(isDistZero == true){ // calibration of the distance via proximity sensor

if (Dist > 0){

isDistZero = false;

} else {

return;

}

}

if (Dist > 15){

//Servo1.write (90);

//Servo2.write (90);

//Servo3.write (90);

digitalWrite (GreenLED, HIGH);

digitalWrite (RedLED, LOW);

digitalWrite (OrangeLED, LOW);

digitalWrite (GreenSignal, HIGH);

digitalWrite (RedSignal, LOW);

Serial.print ("Drive In");

Serial.println();

delay (500);

} else if (Dist <= 15 && Dist >= 10) {

//Servo1.write (90);

//Servo2.write (90);

//Servo3.write (90);

digitalWrite (GreenLED, LOW);

digitalWrite (RedLED, LOW);

digitalWrite (OrangeLED, HIGH);

digitalWrite (GreenSignal, LOW);

digitalWrite (RedSignal, LOW);

Serial.print ("Slow Down!");

Serial.println();

delay (500);

} else { // car inside washing bay. Washing Process Begins

digitalWrite (GreenLED, LOW);

digitalWrite (RedLED, HIGH);

digitalWrite (OrangeLED, LOW);

digitalWrite (GreenSignal, LOW);

digitalWrite (RedSignal, HIGH);

Serial.print ("Stop");

Serial.println();

delay (10);

if (WashStartTime == -1){ // ensures that the washing process at stage 1&4 does not clash

SprayWater();

} else {

Wash();

}

}

}

void SprayWater(){

if (WaterStartTime == -1){

WaterStartTime = millis(); // start timing of operation (take note of the time)

}

WaterTime = (millis() - WaterStartTime)/1000; //tine the water comes out - initial start timing -> to time the duration of water

if (WaterTime <=30){ // process will run if lesser than 30 secs. 30 secs max

Serial.println();

Serial.print ("Car Wash in Progress! Spraying Water");

Servo1Pos += ServoIncrease; // two lines allowing servo to rotate (servo 1 position + servo increase = servo 1 new position)

Servo1.write (Servo1Pos); // new position

Serial.println();

Serial.print (WaterTime); // info for serial monitor

Serial.println();

} else { // water time more than 30

Servo1Pos = 90; // stop servo 1

Servo1.write (Servo1Pos);

SpraySoap(); // once wash is complete, starting soaping process

}

if (Servo1Pos >= servoMaxDegrees){ // ensures that the servo is able to rotated CW and CCW in the same operation

ServoIncrease = ServoIncrease\*reverse; // max (upper limit), need to reduce back to lower limit

}

if (Servo1Pos <= servoMinDegrees){

ServoIncrease = ServoIncrease\*reverse; // lower limit

}

}

void SpraySoap(){

if (SoapStartTime == -1){

SoapStartTime = millis();

}

SoapTime = (millis() - SoapStartTime)/1000;

if (SoapTime <=30){

Serial.println();

Serial.print ("Car Wash in Progress! Spraying Soap");

Servo2Pos += ServoIncrease;

Servo2.write (Servo2Pos);

Serial.println();

Serial.print (SoapTime);

Serial.println();

} else {

Servo2Pos = 90;

Servo2.write (Servo2Pos);

Scrub();

}

if (Servo2Pos >= servoMaxDegrees){

ServoIncrease = ServoIncrease\*reverse;

}

if (Servo2Pos <= servoMinDegrees){

ServoIncrease = ServoIncrease\*reverse;

}

}

void Scrub(){

if (ScrubStartTime == -1){

ScrubStartTime = millis();

}

ScrubTime = (millis() - ScrubStartTime)/1000;

if (ScrubTime <=30){

Serial.println();

Serial.print ("Car Wash in Progress! Scrubbing Time!");

Servo3Pos += ServoIncrease;

Servo3.write (Servo3Pos);

Serial.println();

Serial.print (ScrubTime);

Serial.println();

} else {

Servo3Pos = 90;

Servo3.write (Servo3Pos);

Wash();

}

if (Servo3Pos >= servoMaxDegrees){

ServoIncrease = ServoIncrease\*reverse;

}

if (Servo3Pos <= servoMinDegrees){

ServoIncrease = ServoIncrease\*reverse;

}

}

void Wash(){

if (WashStartTime == -1){

WashStartTime = millis();

}

WashTime = (millis()-WashStartTime)/1000;

if (WashTime <=30){

Serial.println();

Serial.print ("Car Wash in Progress! Last rinse!");

Servo1Pos += ServoIncrease;

Servo1.write (Servo1Pos);

Serial.println();

Serial.print (WashTime);

Serial.println();

} else {

FinishWash();

}

if (Servo1Pos >= servoMaxDegrees){

ServoIncrease = ServoIncrease\*reverse;

}

if (Servo1Pos <= servoMinDegrees){

ServoIncrease = ServoIncrease\*reverse;

}

}

void FinishWash () {

Servo1.write (90);

Servo2.write (90);

Servo3.write (90);

digitalWrite (GreenLED, HIGH);

digitalWrite (RedLED, LOW);

digitalWrite (OrangeLED, LOW);

Serial.print ("Car Wash Completed.");

Serial.println();

Serial.print("Please Exit Washing Bay. Thank You!");

Serial.println();

WaterStartTime = -1;

WaterTime = 0;

SoapStartTime = -1;

SoapTime = 0;

ScrubStartTime = -1;

ScrubTime = 0;

WashStartTime = -1;

WashTime = 0;

delay (10000);

}

void Restart(){

if (Stop == true){ // when emergnecy stop activated

if (EButton == 1023){ // press button once more

Serial.println ("Restarting System");

Serial.println ("Stay Clear!");

delay (5000);

Stop = false; // restart entire system

}

}

}