//Now we are defining the motor connections

int EnA = 21;

int In1 = 20;

int In2 = 19;

#include <Servo.h>

#include <RPLidar.h>

#include <limits.h>

Servo servo;

// create servo object to control a servo RPLidar lidar; // Create lidar object for lidar sensor

float OldAngle = 0;

int i = 0;

int max\_dist;

float max\_angle;

int max\_k;

unsigned long servoMillis = 0;

unsigned long motorMillis = 0;

unsigned long currentMillis;

const unsigned long servo\_interval = 500;

// other interval is 1000 const unsigned long motor\_interval = 3000;

// other intervals are 2000 and 1500

void setup()

{ Serial.begin (115200);

//we're going to use arduino serial monitor to monitor the //results, so we setup the serial monitor at 115200 baud

lidar.begin(Serial1);

// setup the motor control pins //---------------------------------------------------------

pinMode(EnA, OUTPUT); pinMode(In1, OUTPUT); pinMode(In2, OUTPUT);

// set pin modes pinMode(RPLIDAR\_MOTOR, OUTPUT);

//---------------------------------------------------------

servo.attach(23);

// attaches the servo on to pin 23 to the servo object // Print Values //--------------------------------------------------------- Serial.print("\nDist"); Serial.print("\t"); Serial.print("Angle"); Serial.print("\t"); Serial.print("Servo");

Serial.print("\t");

Serial.print("\n");

//--------------------------------------------------------- } //Now we are going to create a set of functions that will be used to determine the movement of the car //--------------------------------------------------------- void goForward() //

run motor forward { digitalWrite(In1, LOW); digitalWrite(In2, HIGH); digitalWrite(EnA, HIGH); }

void goBackward()

//

run motor backwards { digitalWrite(In1, HIGH); digitalWrite(In2, LOW); digitalWrite(EnA, HIGH); }

void goNothing()

{ digitalWrite(In1, LOW); digitalWrite(In2, LOW); }

//--------------------------------------------------------- //Here we define the processing logic function for the movement of the car. //---------------------------------------------------------

void turnSides(int givenAngle)

{ int koefL = -28;

//Edit Coefficent as -28 as the offset angle of the front wheels for the right turn orientation

if ( givenAngle >= 0.0 && givenAngle < 90.0 )

{ int totalAngle = givenAngle + 90 + koefL;

//

turn right currentMillis = millis();

if (currentMillis - servoMillis >= servo\_interval)

{ servo.write(totalAngle); servoMillis = currentMillis;

if (currentMillis - motorMillis >= motor\_interval) { goForward(); motorMillis = currentMillis; } } }

else if

(givenAngle >= 270.0 && givenAngle < 0.0) { int totalAngle = givenAngle - 270 + koefL;

// turn left currentMillis = millis(); if (currentMillis - servoMillis >= servo\_interval) { servo.write(totalAngle); servoMillis = currentMillis; if (currentMillis - motorMillis >= motor\_interval) { goForward(); motorMillis = currentMillis; } } }

else if (givenAngle >= 90.0 && givenAngle < 180.0) { int totalAngle = givenAngle + koefL;

// turn right currentMillis = millis(); if (currentMillis - servoMillis >= servo\_interval) { servo.write(totalAngle); servoMillis = currentMillis; if (currentMillis - motorMillis >= motor\_interval) { goBackward(); motorMillis = currentMillis; } } }

else if ( givenAngle >= 180.0 && givenAngle < 270.0 ) { int totalAngle = givenAngle - 180 + koefL ; // turn left currentMillis = millis(); if (currentMillis - servoMillis >= servo\_interval) { servo.write(totalAngle); servoMillis = currentMillis; if (currentMillis - motorMillis >= motor\_interval) { goBackward(); motorMillis = currentMillis; } } } }

//--------------------------------------------------------- // Put the code here to repeat in a loop

void loop()

{ float angle = 0;

int k=0;

if (IS\_OK(lidar.waitPoint()))

{ float distance = lidar.getCurrentPoint().distance;

//

distance value in mm unit angle = lidar.getCurrentPoint().angle;

//

anglue value in degree bool startBit = lidar.getCurrentPoint().startBit;

//whether this point is belong to //a new scan byte quality =

lidar.getCurrentPoint().quality;

//quality of the current //

measurement k= servo.read();

//

Gets the Servo Motor angle if (distance > max\_dist) { max\_dist = distance; max\_angle = angle; max\_k = k; } }

else { analogWrite(RPLIDAR\_MOTOR, 0);

//stop the rplidar motor // try to detect RPLIDAR... rplidar\_response\_device\_info\_t info;

if (IS\_OK(lidar.getDeviceInfo(info, 100)))

{ // detected... lidar.startScan();

// start motor rotating at max allowed speed analogWrite(RPLIDAR\_MOTOR, 255); delay(1000); } }

if (angle - OldAngle < 0) { //perform data processing here... Serial.print(max\_angle); Serial.print("\t"); Serial.print(max\_dist);

Serial.print("\t");

Serial.println(max\_k);

float anglerequired = max\_angle; turnSides(anglerequired);

max\_dist = 0;

max\_angle = 0; }

OldAngle = angle;

}