Source Code

#include <SoftwareSerial.h>

SoftwareSerial BT\_Serial(2, 3); // RX, TX

#include<Ultrasonic.h>

#include <IRremote.h>

const int RECV\_PIN = A5;

IRrecv irrecv(RECV\_PIN);

decode\_results results;

#define enA 10//Enable1 L298 Pin enA

#define in1 9 //Motor1 L298 Pin in1

#define in2 8 //Motor1 L298 Pin in1

#define in3 7 //Motor2 L298 Pin in1

#define in4 6 //Motor2 L298 Pin in1

#define enB 5 //Enable2 L298 Pin enB

#define servo A4

#define R\_S A0 //ir sensor Right

#define L\_S A1 //ir sensor Left

#define echo A2 //Echo pin

#define trigger A3 //Trigger pin

int distance\_L, distance\_F = 30, distance\_R;

long distance;

int set = 20;

int bt\_ir\_data; // variable to receive data from the serial port and IRremote

int Speed = 130;

int mode=0;

int IR\_data;

void setup(){ // put your setup code here, to run once

pinMode(R\_S, INPUT); // declare if sensor as input

pinMode(L\_S, INPUT); // declare ir sensor as input

pinMode(echo, INPUT );// declare ultrasonic sensor Echo pin as input

pinMode(trigger, OUTPUT); // declare ultrasonic sensor Trigger pin as Output

pinMode(enA, OUTPUT); // declare as output for L298 Pin enA

pinMode(in1, OUTPUT); // declare as output for L298 Pin in1

pinMode(in2, OUTPUT); // declare as output for L298 Pin in2

pinMode(in3, OUTPUT); // declare as output for L298 Pin in3

pinMode(in4, OUTPUT); // declare as output for L298 Pin in4

pinMode(enB, OUTPUT); // declare as output for L298 Pin enB

irrecv.enableIRIn(); // Start the receiver

irrecv.blink13(true);

Serial.begin(9600); // start serial communication at 9600bps

BT\_Serial.begin(9600);

pinMode(servo, OUTPUT);

for (int angle = 70; angle <= 140; angle += 5) {

servoPulse(servo, angle); }

for (int angle = 140; angle >= 0; angle -= 5) {

servoPulse(servo, angle); }

for (int angle = 0; angle <= 70; angle += 5) {

servoPulse(servo, angle); }

delay(500);

}

void loop(){

if(BT\_Serial.available() > 0){ //if some date is sent, reads it and saves in state

bt\_ir\_data = BT\_Serial.read();

Serial.println(bt\_ir\_data);

if(bt\_ir\_data > 20){Speed = bt\_ir\_data;}

}

if (irrecv.decode(&results)) {

Serial.println(results.value,HEX);

bt\_ir\_data = IRremote\_data();

Serial.println(bt\_ir\_data);

irrecv.resume(); // Receive the next value

delay(100);

}

if(bt\_ir\_data == 8){mode=0; Stop();} //Manual Android Application and IR Remote Control Command

else if(bt\_ir\_data == 9){mode=1; Speed=130;} //Auto Line Follower Command

else if(bt\_ir\_data ==10){mode=2; Speed=255;} //Auto Obstacle Avoiding Command

analogWrite(enA, Speed); // Write The Duty Cycle 0 to 255 Enable Pin A for Motor1 Speed

analogWrite(enB, Speed); // Write The Duty Cycle 0 to 255 Enable Pin B for Motor2 Speed

if(mode==0){

//===============================================================================

// Key Control Command

//===============================================================================

if(bt\_ir\_data == 1){forword(); } // if the bt\_data is '1' the DC motor will go forward

else if(bt\_ir\_data == 2){backword();} // if the bt\_data is '2' the motor will Reverse

else if(bt\_ir\_data == 3){turnLeft();} // if the bt\_data is '3' the motor will turn left

else if(bt\_ir\_data == 4){turnRight();} // if the bt\_data is '4' the motor will turn right

else if(bt\_ir\_data == 5){Stop(); } // if the bt\_data '5' the motor will Stop

//===============================================================================

// Voice Control Command

//===============================================================================

else if(bt\_ir\_data == 6){turnLeft(); delay(400); bt\_ir\_data = 5;}

else if(bt\_ir\_data == 7){turnRight(); delay(400); bt\_ir\_data = 5;}

}

if(mode==1){

//===============================================================================

// Line Follower Control

//===============================================================================

if((digitalRead(R\_S) == 0)&&(digitalRead(L\_S) == 0)){forword();} //if Right Sensor and Left Sensor are at White color then it will call forword function

if((digitalRead(R\_S) == 1)&&(digitalRead(L\_S) == 0)){turnRight();}//if Right Sensor is Black and Left Sensor is White then it will call turn Right function

if((digitalRead(R\_S) == 0)&&(digitalRead(L\_S) == 1)){turnLeft();} //if Right Sensor is White and Left Sensor is Black then it will call turn Left function

if((digitalRead(R\_S) == 1)&&(digitalRead(L\_S) == 1)){Stop();} //if Right Sensor and Left Sensor are at Black color then it will call Stop function

}

if(mode==2){

//===============================================================================

// Obstacle Avoiding Control

//===============================================================================

distance\_F = Ultrasonic\_read();

Serial.print("S=");Serial.println(distance\_F);

if (distance\_F > set){forword();}

else{Check\_side();}

}

delay(10);

}

long IRremote\_data(){

if(results.value==0xFF02FD){IR\_data=1;}

else if(results.value==0xFF9867){IR\_data=2;}

else if(results.value==0xFFE01F){IR\_data=3;}

else if(results.value==0xFF906F){IR\_data=4;}

else if(results.value==0xFF629D || results.value==0xFFA857){IR\_data=5;}

else if(results.value==0xFF30CF){IR\_data=8;}

else if(results.value==0xFF18E7){IR\_data=9;}

else if(results.value==0xFF7A85){IR\_data=10;}

return IR\_data;

}

void servoPulse (int pin, int angle){

int pwm = (angle\*11) + 500; // Convert angle to microseconds

digitalWrite(pin, HIGH);

delayMicroseconds(pwm);

digitalWrite(pin, LOW);

delay(50); // Refresh cycle of servo

}

//\*\*\*\*\*\*\*Ultrasonic\_read\*\*\*\*\*\*\*\*\*

long Ultrasonic\_read(){

digitalWrite(trigger, LOW);

delayMicroseconds(2);

digitalWrite(trigger, HIGH);

delayMicroseconds(10);

distance = pulseIn (echo, HIGH);

return distance / 29 / 2;

}

void compareDistance(){

if (distance\_L > distance\_R){

turnLeft();

delay(350);

}

else if (distance\_R > distance\_L){

turnRight();

delay(350);

}

else{

backword();

delay(300);

turnRight();

delay(600);

}

}

void Check\_side(){

Stop();

delay(100);

for (int angle = 70; angle <= 140; angle += 5) {

servoPulse(servo, angle); }

delay(300);

distance\_L = Ultrasonic\_read();

delay(100);

for (int angle = 140; angle >= 0; angle -= 5) {

servoPulse(servo, angle); }

delay(500);

distance\_R = Ultrasonic\_read();

delay(100);

for (int angle = 0; angle <= 70; angle += 5) {

servoPulse(servo, angle); }

delay(300);

compareDistance();

}

void forword(){ //forword

digitalWrite(in1, HIGH); //Right Motor forword Pin

digitalWrite(in2, LOW); //Right Motor backword Pin

digitalWrite(in3, LOW); //Left Motor backword Pin

digitalWrite(in4, HIGH); //Left Motor forword Pin

}

void backword(){ //backword

digitalWrite(in1, LOW); //Right Motor forword Pin

digitalWrite(in2, HIGH); //Right Motor backword Pin

digitalWrite(in3, HIGH); //Left Motor backword Pin

digitalWrite(in4, LOW); //Left Motor forword Pin

}

void turnRight(){ //turnRight

digitalWrite(in1, LOW); //Right Motor forword Pin

digitalWrite(in2, HIGH); //Right Motor backword Pin

digitalWrite(in3, LOW); //Left Motor backword Pin

digitalWrite(in4, HIGH); //Left Motor forword Pin

}

void turnLeft(){ //turnLeft

digitalWrite(in1, HIGH); //Right Motor forword Pin

digitalWrite(in2, LOW); //Right Motor backword Pin

digitalWrite(in3, HIGH); //Left Motor backword Pin

digitalWrite(in4, LOW); //Left Motor forword Pin

}

void Stop(){ //stop

digitalWrite(in1, LOW); //Right Motor forword Pin

digitalWrite(in2, LOW); //Right Motor backword Pin

digitalWrite(in3, LOW); //Left Motor backword Pin

digitalWrite(in4, LOW); //Left Motor forword Pin

}