**the code**

#include

<SoftwareSerial.h> #include<LiquidCrystal.h>

LiquidCrystal lcd(13, 12, A5, A4, A3, A2); String value;

int TxD = 5; int RxD = 6;

SoftwareSerial bluetooth(TxD, RxD);

// Pin Assignments //

#define DIRA 2 // Direction control for motor A #define PWMA 3 // PWM control (speed) for motor A #define DIRB 4 // Direction control for motor B #define PWMB 11 // PWM control (speed) for motor B void setup() {

// All pins should be setup as outputs:

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

lcd.begin(16,2); pinMode(PWMA, OUTPUT); pinMode(PWMB, OUTPUT); pinMode(DIRA, OUTPUT); pinMode(DIRB, OUTPUT);

// Initialize all pins as low: digitalWrite(PWMA, LOW); digitalWrite(PWMB, LOW); digitalWrite(DIRA, LOW); digitalWrite(DIRB, LOW); lcd.setCursor(4,0); lcd.print("WEL COME"); delay(2000);

lcd.clear();

}

void loop() {

int Lsensor,Rsensor; Lsensor=analogRead(0); Rsensor=analogRead(1);

//Serial.println(Lsensor); delay(300);

//Serial.println(Rsensor); delay(300);

if((Lsensor >= 300)||(Rsensor >= 300)){ lcd.setCursor(0,0);

lcd.print(" Car Stop "); forward(0); //stop delay(200);

}

else

{

if (bluetooth.available())

{

value = bluetooth.readString(); Serial.println(value);

if (value == "5F"){ lcd.setCursor(0,0); lcd.print(" Forword "); }

if (value == "5B"){ lcd.setCursor(0,0); lcd.print(" Backword "); backward(120);

}

if (value == "5L"){ lcd.setCursor(0,0); lcd.print(" Left "); rotate\_left(120);

}

if (value == "5R"){ lcd.setCursor(0,0); lcd.print(" Right "); rotate\_right(120);

}

if (value == "S"){ lcd.setCursor(0,0); lcd.print(" Stop "); forward(0);

} } } }

void forward(byte spd){

digitalWrite(DIRA, 0); analogWrite(PWMA, spd); digitalWrite(DIRB, 0); analogWrite(PWMB, spd);

}

void backward(byte spd){

digitalWrite(DIRA, 1); analogWrite(PWMA, spd); digitalWrite(DIRB, 1); analogWrite(PWMB, spd);

}

void rotate\_right(byte spd){

digitalWrite(DIRA, 0); analogWrite(PWMA, spd); digitalWrite(DIRB, 1); analogWrite(PWMB, spd);

}

void rotate\_left(byte spd){

digitalWrite(DIRA, 1); analogWrite(PWMA, spd); digitalWrite(DIRB, 0); analogWrite(PWMB, spd);

}

forward(120