Phone remote for RC Car

Bedrading :

A picture containing electronics, connector

Description automatically generated

Project :

Bedoeling van dit project is om een afstandsbediening te maken in blynk voor een RC auto.

Alles is gelukt behalve de DC motor aansluiting, normaal zou ge ook een DC motor driver gebruiken om ook achteruit te rijden, en dan zou ik ook de afstandsbediening aangepast hebben.

Code:

//libraries

#define BLYNK\_PRINT Serial

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include <ESP32Servo.h>

//servo object

Servo myservo;

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char auth[] = "pZvbstT2o9K5yf4g\_4l7QYLlrvcFiSO0";

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "MSI 6691";

char pass[] = "123456789";

// joystick values from blynk

int Turn = 0;

int Drive = 0;

// servo and motor pins

int servoPin = 27;

int motorPin = 26;

// RGBled pins

int led\_blue = 25;

int led\_green = 32;

int led\_red = 33;

// touch sensor variables

int ctsPin = 34;

bool cts = false;

//setup

void setup() {

Serial.begin(115200);

// wifi and blynk

delay(10);

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, pass);

int wifi\_ctr = 0;

while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

Serial.println("WiFi connected");

Blynk.begin(auth, ssid, pass, "server.wyns.it", 8081);

// Servo

ESP32PWM::allocateTimer(0);

ESP32PWM::allocateTimer(1);

ESP32PWM::allocateTimer(2);

ESP32PWM::allocateTimer(3);

myservo.setPeriodHertz(50); // standard 50 hz servo

myservo.attach(servoPin, 500, 2400);

// DC motor

pinMode(motorPin, OUTPUT);

// CTS

pinMode(ctsPin, INPUT);

// RGB

pinMode(led\_blue, OUTPUT);

pinMode(led\_green, OUTPUT);

pinMode(led\_red, OUTPUT);

}

//loop

void loop() {

//reading touch sensor value

if (digitalRead(ctsPin) == LOW)

{

if (cts == true)

{

cts = false;

Serial.println("Program stop");

}

else

{

cts = true;

Serial.println("Program start");

}

delay(4000);

}

// when the Boolean of the touch sensor is false, the rest of the code won’t go active

if (cts == true)

{

Blynk.run();

// mapped degree for the servo

myservo.write(map(Turn, 0, 1023, 0, 180));

// PWM signal for DC motor

analogWrite(motorPin, Drive);

// RGB and values to blynk when certain speeds are active

if (Drive == 512)

{

digitalWrite(led\_red, HIGH);

digitalWrite(led\_blue, LOW);

digitalWrite(led\_green, LOW);

Blynk.virtualWrite(V2, 200);

Blynk.virtualWrite(V3, 0);

}

else if (Drive < 512 && Drive > 23)

{

digitalWrite(led\_red, LOW);

digitalWrite(led\_blue, HIGH);

digitalWrite(led\_green, LOW);

Blynk.virtualWrite(V2, 600);

Blynk.virtualWrite(V3, map(Drive, 512, 23, 0, 119));

}

else if (Drive < 23)

{

digitalWrite(led\_red, LOW);

digitalWrite(led\_blue, LOW);

digitalWrite(led\_green, HIGH);

Blynk.virtualWrite(V2, 100);

Blynk.virtualWrite(V3, 120);

}

}

else

{

digitalWrite(led\_red, LOW);

digitalWrite(led\_blue, LOW);

digitalWrite(led\_green, LOW);

}

}

// Blynk value receive functions for joystick

BLYNK\_WRITE(V1)

{

Turn = param.asInt();

}

BLYNK\_WRITE(V0)

{

Drive = param.asInt();

}

Demo Video link:

<https://youtu.be/SJxXsy8NJ7k>