TP

Nicolas Vadkerti

20 mars 2020

1 HardWare

Pour commencer, voici le montage qui nous permettra d'utiliser le moteur fourni celon les caractèristiques suivante :
Tension d'alimentaton du moteur : 24 Volts
Courant consommé par le moteur : 500 mA

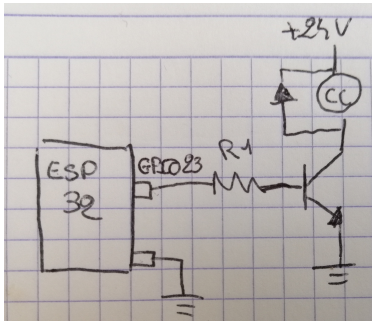

Figure 1 – Montage

Il faut donc calculer R1.
On suppose que le β de notre transistor est de $\beta = 100$ donc si on veut 500mA pour notre moteur, il faut appliquer à la base 0,005 A car $i_b = \frac{i_c}{\beta} = \frac{500mA}{100} = 5mA$
Donc $R_1 = \frac{3{,}3V - 0{,}7V(valeurTypiquepouruntransistor)}{0{,}005} = 520\Omega$

Pour la diode de roue libre, il suffit quelle soit capable de laisser traverser 500mA et 24V Ainsi nous pouront utilser notre moteur en utilisant un signal PWM via notre pin GPIO. En effetle transistor sera passant quand le GPIO sera à l'etat Haut, et le transistor sera bloquant quand le gpio sera a l'etat bas. Ainsi le signal pwm sera bien appliquer au moteur

2 Communiquer avec l'objet communiquant

L'idée est simple, l'ESP32 devra faire office de point d'accées Wifi. Il devra donc remplir les service de base pour etre facile d'utilisation pour le client. En effet, il devra, au moins faire serveur DHCP et serveur HTTP, pour que le client communique facilement avec celui à l'aide d'une page Web. Ainsi mis en place, L'esp32 mettra à jour regulierement les informations sur la page web qu'il "pushera" au client et lira les requetes "POST" du clients, et modifira donc la rotations du moteurs en conséquence.

3 Le SoftWare pour l'ESP32

On crée un serveur HTTP sur l'ESP32, on fait deux boutons, qui renvoie vers des urls distincts et quand l'utilisateur fais une requete sur cette url, la vitesse du moteurs change voici comment procéder :

```
#include <WiFi.h>
#include <WebServer.h>
#define CHANNEL 5
const char *ssid="NicolasV";
const char *pass="12345678";
IPAddress ip(192,168,42,1);
IPAddress gateway(192,168,42,254);
IPAddress subnet(255,255,255,0);
WebServer server(80);
int frequence = 1000;
int resolution = 8;
int canal = 0;
int valMot = 0;

void wifiAPSetup() {
  Serial.println("wifiAPSetup...");
  WiFi.mode(WIFI_AP);
  WiFi.softAPConfig(ip,gateway,subnet); // Configuration DHCP
  //WiFi.softAP(ssid,pass); // DHCP on 192.168.4.0/24 by default if no WiFi.softAPConfig
      ()
  WiFi.softAP(ssid,pass,CHANNEL);
  IPAddress apIP=WiFi.softAPIP();
  Serial.printf("IP address: %s\r\n",apIP.toString().c_str());
  Serial.printf("BSSID: %s\r\n",WiFi.softAPmacAddress().c_str());
}
void handleNotFound() {
  server.send(404,"text/plain","Fichier non trouve !");
}
void handleRoot() {
    String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n<
        html>\r\n ";
s +="<h1>Vitesse moteur actuel :"+String(valMot)+" </h1>\n";
s +="p><a href=\"/plus\"><button class=\"button\"> Plus</button></a></p>";
s +="p><a href=\"/moins\"><button class=\"button\"> Moins</button></a></p>";
s += "</html>\n";
server.send(200,"text/html",s);

}

void handlePlus(){
    valMot = valMot + 10;
    ledcWrite(canal, valMot);
}
void handleMoins(){
    valMot = valMot - 10;
    ledcWrite(canal, valMot);
}


void webSetup() {
  Serial.println("WEB server setup...");
  server.on("/",handleRoot);
  server.on("/plus",handlePlus);
  server.on("/moins", handleMoins);
  server.onNotFound(handleNotFound);
  server.begin();
  Serial.println("WEB server running...");
}
void setup() {
  delay(1000);
  Serial.begin(115200);
  Serial.println("Setup...");
  wifiAPSetup();
  webSetup();
  Serial.println("Setup done.");
  pinMode(23, OUTPUT);
  ledcSetup(canal, frequence, resolution);
  ledcAttachPin(23, canal);


}
void loop() {
  server.handleClient();
}
```

Listing 1 – Mon Code

Le code compile, mais vu les conditions due au COVID-19, je ne peux pas tester si cela est fonctionnel.