Departmentul Calculatoare
Universitatea Tehnică din Cluj-Napoca

Proiect de laborator la Proiectarea cu Microprocesoare

Roboți comandați sincron prin WIFI și Bluetooth

Nume:Hura Abel Jonathan
Grupa:30235
Email:huraabeljonathan@gmail.com

Descrierea problemei

Proiectul are în vedere construirea unui robot sau mașină care se poate comanda prin intermediul unui modul WIFI , iar acesta la rândul său să trimită aceeași comandă mai departe la un alt robot printr-un modul Bluetooth. Astfel, se dorește să se proiecteze o modalitate de comunicare între utilizator și robot și abilitatea acestora de a se mișca sincron.

Utilizatorul va comunica cu un singur robot prin intermediul WIFI, va putea să aleagă ce mișcare să efectueze prin selectarea unor butoane pe o pagină web. Robotul comandat nu numai că va efectua mișcarea ce primește de la utilizator, dar va avea rol de master în comandarea unui alt robot prin intermediul unui modul Bluetooth. Comanda primită de utilizator deci va fi transmis prin Bluetooth la un alt robot care va efectua aceeași mișcare, amândoi mișcându-se în mod sincron.

Proiectul va fi realizat utilizînd plăci Arduino MEGA 2560, module Bluetooth HC-05, modul WIFI ESP8266, punte H dublu L298N, motoare DC.

Soluția găsită

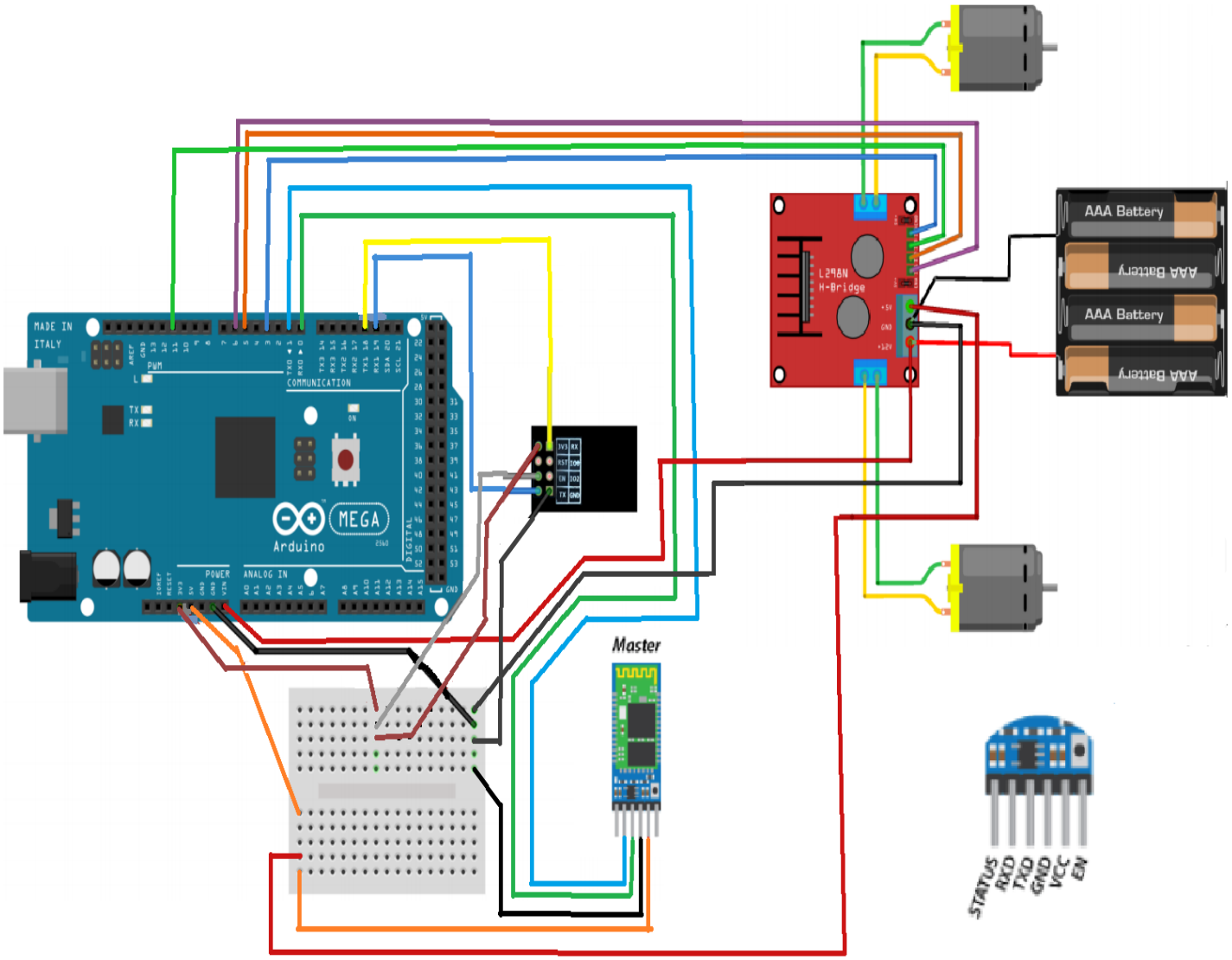
În primul rând, vom avea nevoie de un robot master cu care vom comunica prin intermediul WIFI.Vom conecta modulul ESP8266 și îl vom configura, acest proces este descris în îndrumătorul de laborator [1]. Apoi, va trebuie să configurăm modulul Bluetooth ca și master. Pentru acesta va trebui să conectăm modulul HC-05 în modul de comenzi AT, cu pinul EN conectat la 5V. Acest proces se repetă și pentru dispozitivul slave. Pe bluetoothul slave se setează rolul 0 (slave), și se află adresa acestuia. La această adresă se va va conecta dispozitivul master. Procesul de configurarea ca master și slave se poate consulta pe pagina web din referința [2].

Conectarea celor două module Bluetooth are success dacă beculețele picăie de 2 ori la 2 secunde. În această fază se pot transmite comenzi de la master la slave.

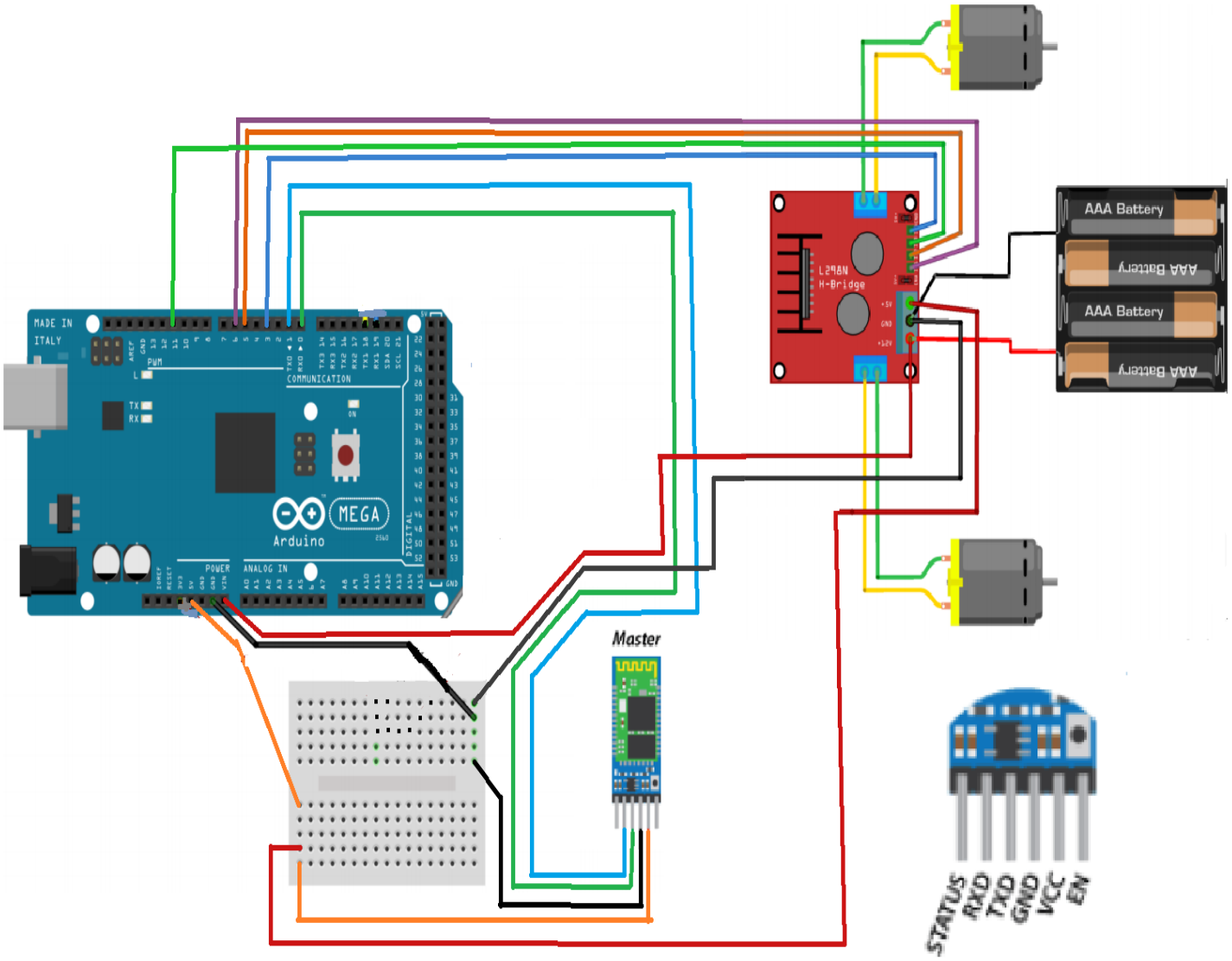
Manual de utilizare

După conectarea corectă a firelor, pe master vom lăsa firele ce se leagă de pinii 1 și 0 al plăcii Arduino ATMEGA 2560 neconectați, după care putem apăsa butonul Upload în Arduino IDE. După ce s-a transmis codul pe plăcuță vom putea să conectăm cele două fire conform schemei. Ne vom conecta la modulul WIFI asftel: deschidem din Tools Serial Monitor, setăm la baud rate de 38400, cu opțiunea Both NL & CR. Va apărea adresa IP și numele, de exemplu 192.168.4.1 și AI-THINKER3DC39D. Vom folosi aceaste date ca să ne conectăm cu telefonul. După ce ne-am conectat, vom intra pe pagina 192.168.4.1 în browserul ales de noi de pe telefon. Vor apărea 4 butoane. Apăsând butoanele vom transmite prin WIFI la modulul nostru Bluetooth de la dispozitivul master comanda de mișcare. Această comandă va fi transmisă prin Bluetooth la dispozitivul slvae care va efectua sicnron aceeași mișcare ca și dispozitivul master.

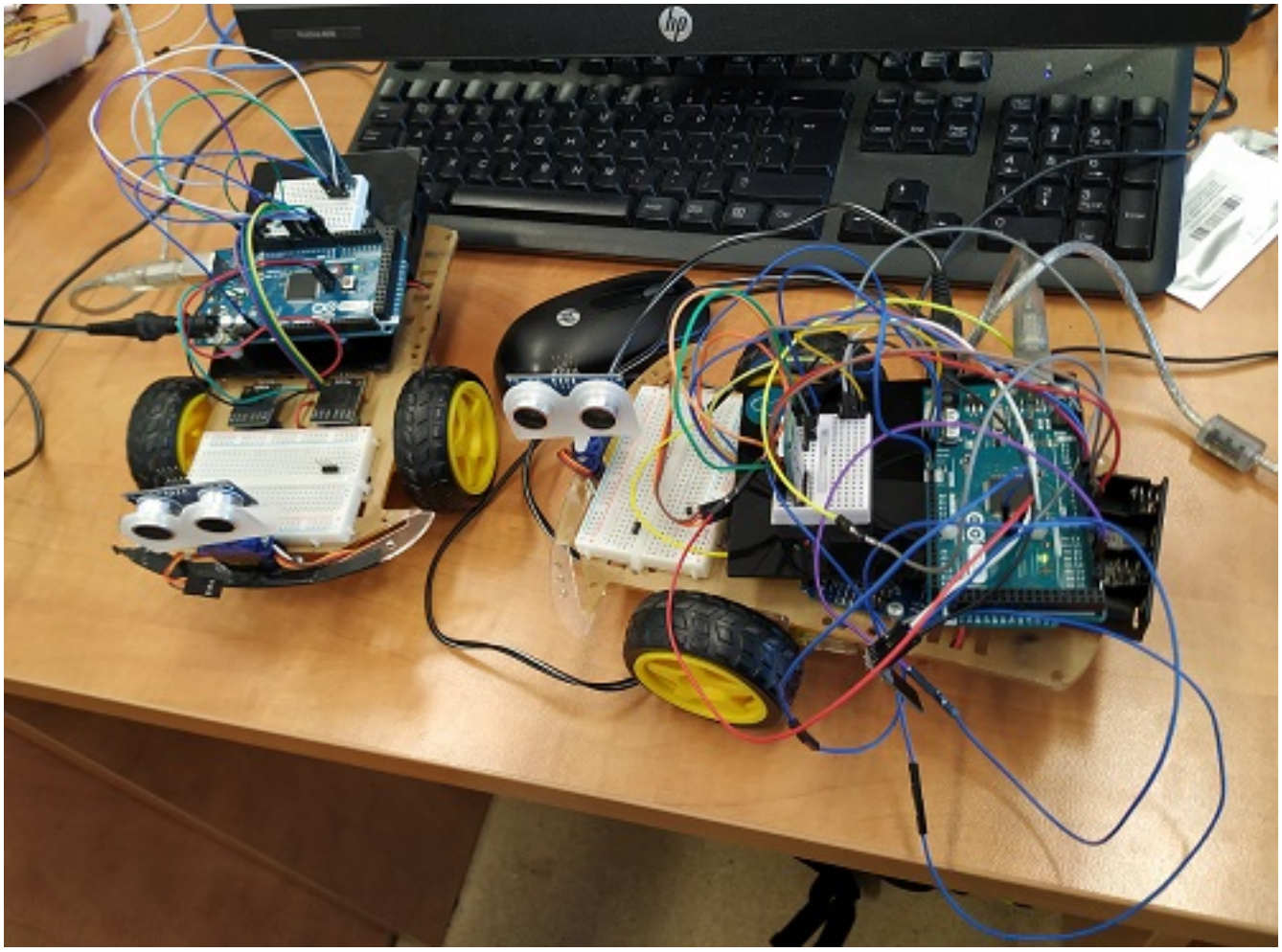
Circuitul Master



Circuitul Slave



Proiectul



Bibliografie

[1] Proiectarea cu Microprocesoare Îndrumător de laborator , pp. 86-100,
https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/336-3.pdf

[2] Conectarea a două module Bluetooth HC-05 https://howtomechatronics.com/tutorials/arduino/how-to-configure-pair-two-hc-05-bluetooth-module-master-slave-commands/

Cod Master

```
#define DEBUG true
#define mpin00 5
#define mpin01 6
// Pinii motor 2
#define mpin10 3
#define mpin11 11
void setup() {

 digitalWrite(mpin00, 0);
 digitalWrite(mpin01, 0);
 digitalWrite(mpin10, 0);
 digitalWrite(mpin11, 0);

 pinMode (mpin00, OUTPUT);
 pinMode (mpin01, OUTPUT);
 pinMode (mpin10, OUTPUT);
 pinMode (mpin11, OUTPUT);

  //BLUETOOTH
  Serial.begin(38400); // Default communication rate of the Bluetooth module


  //WIFI
  Serial1.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW);
  sendData("AT+RST\r\n", 2000, false); // resetare modul
  sendData("AT+CWMODE=2\r\n", 1000, false); // configurare ca
  //access point
  sendData("AT+CIFSR\r\n", 1000, DEBUG); // citeste adresa IP
  sendData("AT+CWSAP?\r\n", 2000, DEBUG); // citeste informa ia
  //SSID (nume re ea)
  sendData("AT+CIPMUX=1\r\n", 1000, false); // configurare
  //conexiuni multiple
  sendData("AT+CIPSERVER=1,80\r\n", 1000, false); // pornire
  //server pe port 80

}

void StartMotor (int m1, int m2, int forward, int speed)
{
```

```
 if (speed==0) // oprire
 {
        digitalWrite(m1, 0);
        digitalWrite(m2, 0);
 }
 else
 {
         if (forward)
        {
                digitalWrite(m2, 0);
                analogWrite(m1, speed); // folosire PWM
        }
 else
 {
        digitalWrite(m1, 0);
        analogWrite(m2, speed);
 }
 }
}

// Functie de siguran
// Executat oprire motoare, urmat de delay
void delayStopped(int ms)
{
 StartMotor (mpin00, mpin01, 0, 0);
 StartMotor (mpin10, mpin11, 0, 0);
 delay(500);
}

void loop() {

  // Pentru modulul WIFI
  if (Serial1.available()) {
    if (Serial1.find("+IPD,")) {
      delay(500);
      int connectionId = Serial1.read() - 48; // functia
      //read() returneaz valori zecimale ASCII
      // si caracterul    0    are codul ASCII 48
      String webpage =
      "<h1>Comanda Robotii</h1><a href=\"/l0\"><button>Left</button></a>";
      String cipSend = "AT+CIPSEND=";
      cipSend += connectionId;
      cipSend += ",";
      webpage += "<a href=\"/l1\"><button>Forward</button></a>";
      webpage += "<a href=\"/l2\"><button>Back</button></a>";
      webpage += "<a href=\"/l3\"><button>Right</button></a>";

      if (readSensor() > 0) {
        webpage += "<h2>Millis:</h2>";
```

```
        webpage += readSensor();
      }
      cipSend += webpage.length();
      cipSend += "\r\n";
      sendData(cipSend, 100, DEBUG);
      sendData(webpage, 150, DEBUG);

      String closeCommand = "AT+CIPCLOSE=";
      closeCommand += connectionId; //se adaug
      //identificatorul conexiunii
      closeCommand += "\r\n";
      sendData(closeCommand, 300, DEBUG);
    }
  }
}

String sendData(String command, const int timeout, boolean debug)
{
  String response = "";
  Serial1.print(command); // trimite comanda la esp8266
  long int time = millis();
  while ((time + timeout) > millis()) {
    while (Serial1.available()) {
      char c = Serial1.read(); // cite te caracter urm tor
      response += c;
    }
  }
  if (response.indexOf("/l0") != -1) {
    digitalWrite(LED_BUILTIN, HIGH);
    //trimite comanda la slave prin BLUETOOTH
    Serial.write('!');
    //efectueaza miscarea aleasa
    StartMotor (mpin00, mpin01, 0, 128);
        StartMotor (mpin10, mpin11, 0, 128);

        delay (500); // C t timp e motorul pornit
        delayStopped(500); // C t timp e oprit
  }
  if (response.indexOf("/l1") != -1) {
    digitalWrite(LED_BUILTIN, LOW);
    //trimite comanda la slave prin BLUETOOTH
    Serial.write('@');
    //efectueaza miscarea aleasa
    StartMotor (mpin00, mpin01, 1, 128);
        StartMotor (mpin10, mpin11, 1, 128);

        delay (500);
        delayStopped(500);

  }
```

```
  if (response.indexOf("/l2") != -1)
  {
    digitalWrite(LED_BUILTIN, HIGH);
    //trimite comanda la slave prin BLUETOOTH
    Serial.write('#');
    //efectueaza miscarea aleasa
    StartMotor (mpin00, mpin01, 0, 128);
        StartMotor (mpin10, mpin11, 1, 128);

        delay (500);
        delayStopped(500);

  }

  if (response.indexOf("/l3") != -1)
  {
    digitalWrite(LED_BUILTIN, LOW);
    //trimite comanda la slave prin BLUETOOTH
    Serial.write('$');
    StartMotor (mpin00, mpin01, 1, 128);
        StartMotor (mpin10, mpin11, 0, 128);

        delay (500);
        delayStopped(500);
  }

  if (debug) {
    Serial.print(response);
  }

  return response;
}

unsigned long readSensor() {
  return millis();
}
```

Cod Slave

```
#define ledPin9 9
// Pinii motor 1
#define mpin00 5
#define mpin01 6
// Pinii motor 2
#define mpin10 3
#define mpin11 11
int state = 0;

void setup() {

  digitalWrite(ledPin9, LOW);
  digitalWrite(mpin00, 0);
  digitalWrite(mpin01, 0);
  digitalWrite(mpin10, 0);
  digitalWrite(mpin11, 0);

  pinMode (mpin00, OUTPUT);
  pinMode (mpin01, OUTPUT);
  pinMode (mpin10, OUTPUT);
  pinMode (mpin11, OUTPUT);

  Serial.begin(38400); // Default communication rate of the Bluetooth module
}

void StartMotor (int m1, int m2, int forward, int speed)
{
  if (speed == 0) // oprire
  {
    digitalWrite(m1, 0);
    digitalWrite(m2, 0);
  }
  else
  {
    if (forward)
    {
      digitalWrite(m2, 0);
      analogWrite(m1, speed); // folosire PWM
    }
    else
    {
```

```
      digitalWrite(m1, 0);
      analogWrite(m2, speed);
    }
  }
}

void delayStopped(int ms)
{
  StartMotor (mpin00, mpin01, 0, 0);
  StartMotor (mpin10, mpin11, 0, 0);
  delay(ms);
}

void loop() {
  if (Serial.available() > 0) {
  // Verifica daca vine mesaj de pe portul serial - de la dispozitivul master
    state = Serial.read(); // Citeste din serial port
  }
  // Controlling the LED
  if (state == '!') {
    StartMotor (mpin00, mpin01, 0, 128);
    StartMotor (mpin10, mpin11, 0, 128);
    delay (500); // C t timp e motorul pornit
    delayStopped(500);
  }
  else if (state == '@') { // fata
    StartMotor (mpin00, mpin01, 1, 128);
    StartMotor (mpin10, mpin11, 1, 128);
    delay (500);
    delayStopped(500);
  }
  else if (state == '#') {
    StartMotor (mpin00, mpin01, 0, 128);
    StartMotor (mpin10, mpin11, 1, 128);
    delay (500);
    delayStopped(500);
  }
  else if (state == '$') {
    StartMotor (mpin00, mpin01, 1, 128);
    StartMotor (mpin10, mpin11, 0, 128);
    delay (500);
    delayStopped(500);
  }
}
```