```
#include  <ESP8266WiFi.h>
#include  <WiFiClient.h>
#include  <ESP8266WebServer.h>

const char* ssid = "4Bots";
const char* password = "pra@12345";

#define IN_1  D3      // L298N in1 motors Right        GPIO0(D3
#define IN_2  D4     // L298N in2 motors Right        GPIO2(D4)
#define IN_3  D6    // L298N in3 motors Left         GPIO12(D6)
#define IN_4  D7    // L298N in4 motors Left         GPIO13(D7)

String  command;
int v = 150;
int vl = 40;

ESP8266WebServer server(80);

void setup() {

   Serial.begin(115200);

 pinMode(IN_1,  OUTPUT);
 pinMode(IN_2,  OUTPUT);
 pinMode(IN_3,  OUTPUT);
 pinMode(IN_4,  OUTPUT);



   WiFi.mode(WIFI_AP);
   WiFi.softAP(ssid, password);

  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
   Serial.println(myIP);

 // Starting WEB-server
```

```
      server.on ( "/", HTTP_handleRoot );
      server.onNotFound ( HTTP_handleRoot );
      server.begin();
}

void loop() {
     server.handleClient();

       command = server.arg("State");
            if (command == "F") goForward();
       else if (command == "B") goBack();
       else if (command == "L") goLeft();
       else if (command == "R") goRight();
       else if (command == "I") goForwardRight();
       else if (command == "G") goForwardLeft();
       else if (command == "J") goBackRight();
       else if (command == "H") goBackLeft();
       else if (command == "0") v = 100;
       else if (command == "1") v = 120;
       else if (command == "2") v = 140;
       else if (command == "3") v = 160;
       else if (command == "4") v = 180;
       else if (command == "5") v = 200;
       else if (command == "6") v = 215;
       else if (command == "7") v = 230;
       else if (command == "8") v = 240;
       else if (command == "9") v = 255;
       else if (command == "S") stop();
}

void HTTP_handleRoot(void) {

if( server.hasArg("State") ){
         Serial.println(server.arg("State"));
  }
  server.send ( 200, "text/html", "" );
  delay(1);
}

void  goForward(){

       analogWrite(IN_1, v-2);
       analogWrite(IN_2, 0);

       analogWrite(IN_3, v);
       analogWrite(IN_4, 0);

  }

void  goBack(){

       analogWrite(IN_1, 0);
       analogWrite(IN_2, v);

       analogWrite(IN_3, 0);
       analogWrite(IN_4, v);
  }

void  goRight(){

       analogWrite(IN_1, 0);
       analogWrite(IN_2, v);

       analogWrite(IN_3, v);
       analogWrite(IN_4, 0);

  }

void  goLeft(){

       analogWrite(IN_1, v);
       analogWrite(IN_2, 0);
       analogWrite(IN_3, 0);
       analogWrite(IN_4, v);
  }

void  goForwardLeft(){

       analogWrite(IN_1, v);
       analogWrite(IN_2, 0);

       analogWrite(IN_3, v-vl);
       analogWrite(IN_4, 0);
   }

void  goForwardRight(){

       analogWrite(IN_1, v-vl);
       analogWrite(IN_2, 0);

       analogWrite(IN_3, v);
       analogWrite(IN_4, 0);
  }

void  goBackLeft(){

       analogWrite(IN_1, 0);
       analogWrite(IN_2, v);

       analogWrite(IN_3, 0);
       analogWrite(IN_4, v-vl);
  }

void  goBackRight(){

       analogWrite(IN_1, 0);
       analogWrite(IN_2, v-vl);
       analogWrite(IN_3, 0);
       analogWrite(IN_4, v);
  }

void  stop(){

       analogWrite(IN_1, 0);
       analogWrite(IN_2, 0);

       analogWrite(IN_3, 0);
       analogWrite(IN_4, 0);
 }
```