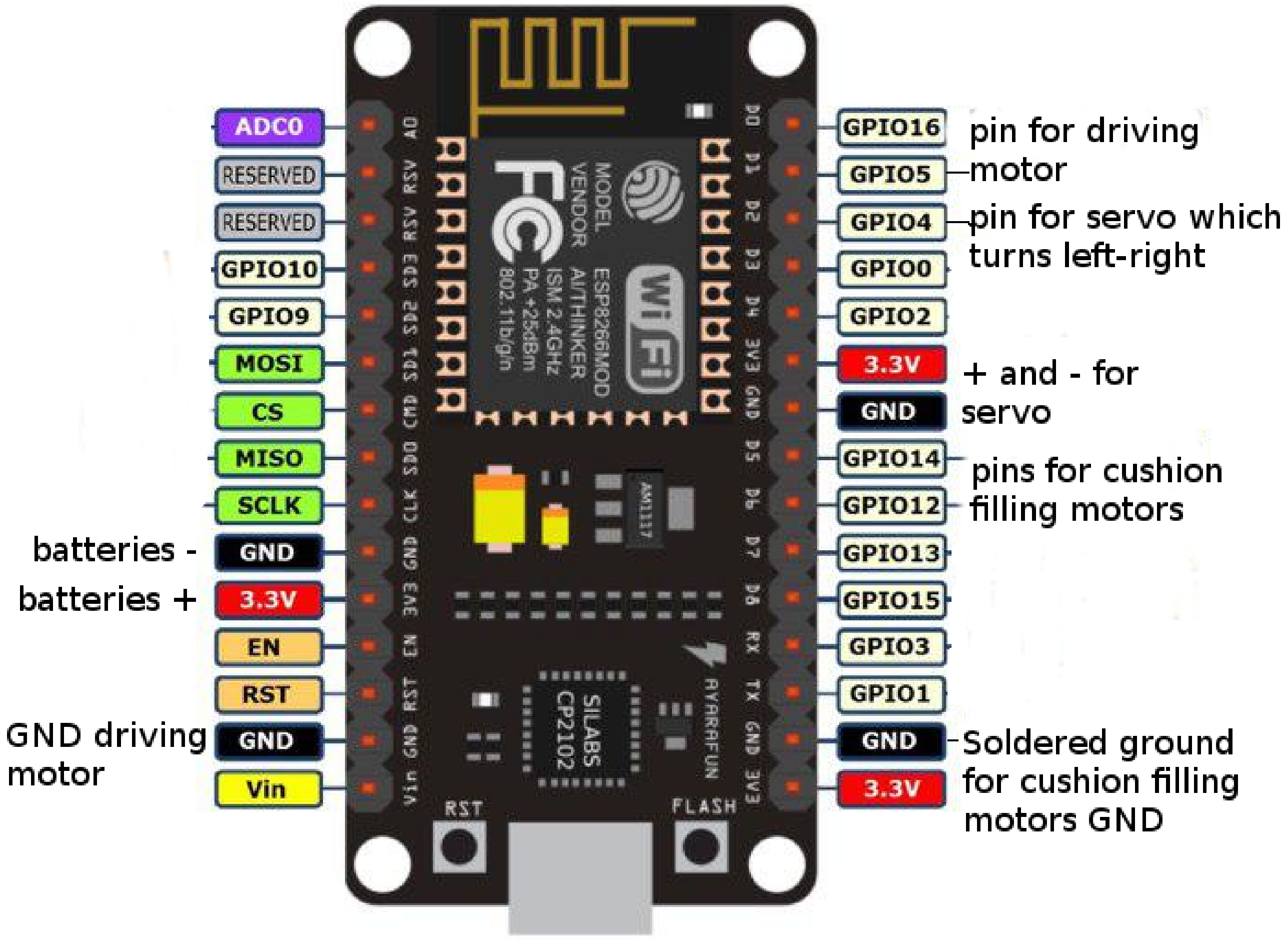
You will need :
- Nodemcu ESP8266 module
- 3 drivers (depended from hull)
- 3 motors (depended from hull)
- 3 propelers for them (depended from hull)
- pin connecters
- soldering iron
- accamulator with 5000mA + (or 2x 2500mA), for 4-5 minutes working
- battery socket
- batteries with 2x1.5v or 1x3v
- any android device
- arduino IDE and microusb connector
- * hull

Scheme :



pin for driving motor

pin for servo which turns left-right

+ and - for servo

pins for cushion filling motors

Soldered ground for cushion filling motors GND

batteries -

batteries +

GND driving motor

Wifi moudule documentation : https://github.com/esp8266/Arduino

Fast android app creation : http://ai2.appinventor.mit.edu/

1. You will need to upload scheme to arduino IDE
*If yo use linux, see official documentation for arduino ide for correct port using
(you probably must add your user to group that controls USB ports as superuser)

1) Add this link to Preferences – Additional Boards Manager URL
http://arduino.esp8266.com/stable/package_esp8266com_index.json
2) Restart arduino ide
3) Go to Tools – Board Manager and install esp8266 by ESP8266 community board
4) After installing restart arduino ide again
5) Now you can use NodeMCU 1.0 board (ESP-12E module)

2. You need to create hostpot, or use any, just put correct ssid and password into

`

```
const char * ssid = "YOUR_SSID";
const char * password = "YOUR_PASSWORD";
```

`

Perfect choise is to make it on the smartphone which will send signals to your controller

3. Then you need to create android app, we used http://ai2.appinventor.mit.edu/

4. After it you need to see ip of your wifi-module in this wifi hostpot
You can do this with

`

```
#include <ESP8266WiFi.h>
WiFiClient client;
WiFiServer server(80);
const char* ssid = "YOUR SSID";
const char* password = "YOUR PASSWORD";

void setup()
{
  Serial.begin(115200);
  connectWiFi();
  server.begin();
}

void loop()
{}

void connectWiFi()
{
  Serial.println("Connecting to WIFI");
  WiFi.begin(ssid, password);
  while ((!(WiFi.status() == WL_CONNECTED)))
  {
    delay(300);
    Serial.print("..");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("NodeMCU Local IP is : ");
  Serial.print((WiFi.localIP()));
}
```

`

5. After you know ip you can send message to this ip from your android app and receive it with controller

```
 `
String checkControllerAction (void) {

  while(!controller.available()) delay(1);

  String request = controller.readStringUntil('\r');

//  Serial.println(request);

  // delete http:/someIp/ from http:/someIp/command
  request.remove(0, 5);

  request.remove(request.length()-9,9);

  return request;
}
`
```

String returned from this method will give you the same as you sended from app, so you can do any action after receiving it.

Must be used in loop(), and the best option is make if-else constructions for every command, or, if project is small – minimal event system;

6. Recieved signal you use as you want, for example for driver-motor actions, for any led test or else;

7. After making controller app and uploading it to esp module you need to make electronic scheme
* Here you can see our controller sample :
https://github.com/progbase/hovercraft/tree/master/Project/wifiServerController

The problem is, all motors need pin, accamulator and ground. And if you have many pins, GND pin count on wifi board will make you sad;

Cause, one ground goes to connect wifi module to batteries;
One to servomotor; (or servomotors, that also you need sold to one connection pin)

And then you will need to connect 2-3 (or how many you want) driver-motors ground pins to only one GND pin on wifi module board;

We solved this problem by simply solder GND from 2 drivers-motors which filled the airbag with air to one pin connector;
And we have one free GND for other motor;

The main idea is batteries GND pin that power the wifi module must not be solded to any device GND pin;
And better that not similar device types will be connected to different GND pins;

8. In this point you must calibrate all your devices and to assemble the whole scheme together.

9. And after charging and cheking accamulators you can finnaly test your project

***if you use two accamulators connected to one power bullet, then they must be in the same charge and voltage**
cause in the other case power from one will go to other and that can make an unexpected situation for you when the whole project burns down

Contacts :
bogdan.goncharuck@gmail.com
telegram.me/danGoncharuck
telegram.me/omblazhko