Shenzhen Doctors of Intelligence & Technology (SZDOIT)

User Manual for the development of DoitCar

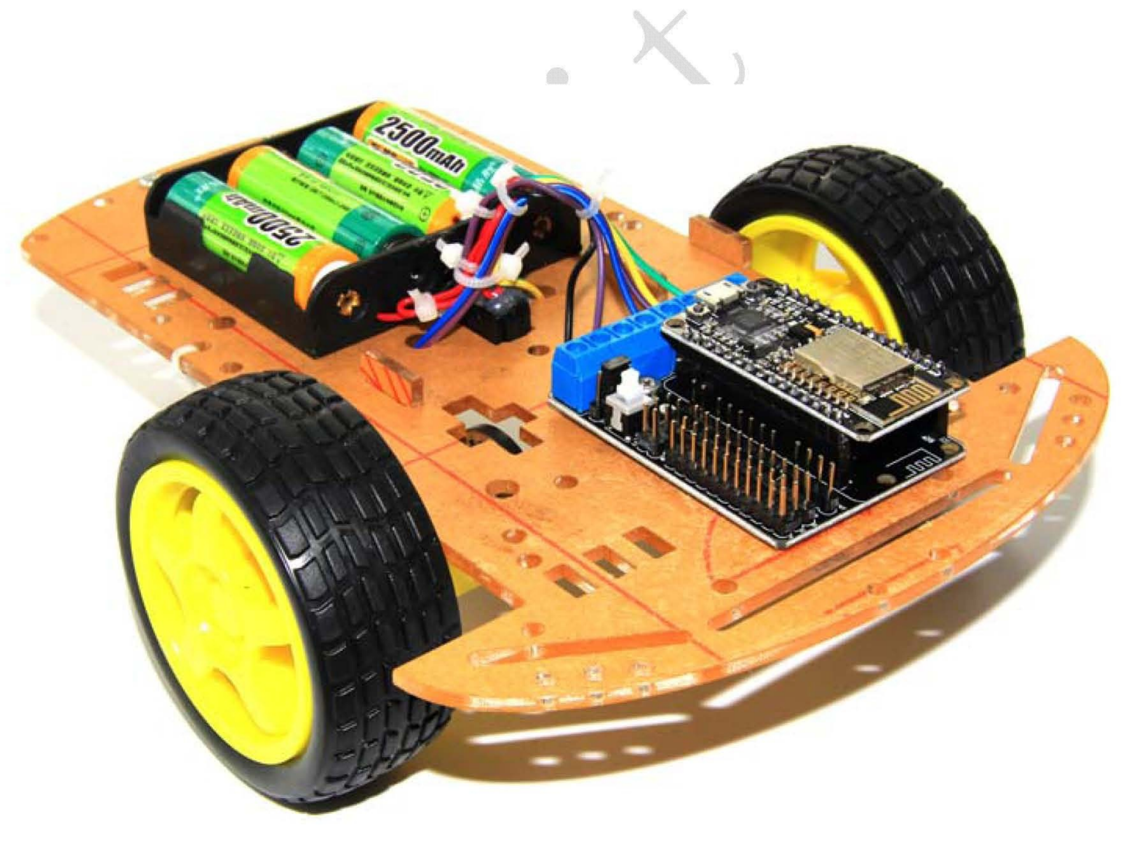


Table of Contents

Introduction

DoitCar is designed and developed by Shenzhen Doctors of Intelligence & Technology (SZDOIT), which is the most cost-effective. It is widely applied in many fields, sucha as the electronic lover, college students, Internet of Things (IoT), mobile data collection, etc. DoitCar has a great temptation for the smart car solution.

The DoitCar kit is including the car chassis, 2 pieces of 6V motors, NodeMCU WiFi board, motor driven shield board. Especially, all the codes and hardwares are open for all people.

Accordingly, the software collections are including android, Webchat, and internet. The android code is also open for all people to further develop it by your novel idea.

DoitCar is controlled by ESP-12E (as the control board) and ESP-12E Motor Shield (as the driven board). For more information about these two boards, please visit http://www.doit.am.

The develop computer language is Lua for DoitCar with large API encapsulation, which can make users design and exploit their products quickly and conveniently. In addition, DoitCar can be programmed under the condition of Arduino IDE.

For more information about DoitCar, please visit http://www.doit.am. **Skype:** yichone. **Email:** yichone@doit.am.

Technical Specifications

- Power Input

  1) Motor Power(VM): 4.5~36V, can power separately;

  2) Control Power (VIM): 4.5~9V(10V MAX), can power separately;

  3) Module having shortcut module (connect VM and VIM), thus can use one-way power publicly (4.5~9V) to power control board and motor shield board;

- Logic Working Current (Iss):≤60mA (Vi=L), ≤22mA(Vi=H);

- Driving Working Current (Io): Io:≤1.2A;
- Max Dissipation Power: 4W（T=90℃）;
- Control Signal Input Voltage: High voltage (2.3V≤VIH≤VIN); Lower voltage (-0.3V≤VIL≤1.5V);
- Working Temperature: -25℃～＋125℃
- Driven Mode: Double-way large power H-bridge driven;
- Support wireless 802.11 b/g/n standard;
- Support STA/AP/STA+AP 3-types working mode;
- Built-in TCP/IP protocol stack; Support multi-way TCP Client connection (5 MAX);
- D0～D8, SD1～SD3: used as GPIO, PWM, IIC, and etc., Port-driven ability 15mA;
- AD0: 1-channel ADC;
- Power Input: 4.5V～9V (10VMAX); Support powered-USB; Provide USB-debug interface;
- Working Current: continual send:≈70mA (200mA MAX) standby：＜200uA;
- Transmission Data:110-460800bps;
- Support UART/GPIO data communication interface;
- Support firmware by remote update;
- Support Smart Link;
- Working Temperature: -40℃～＋125℃；
- Driven Mode: double-way big-power H-bridge driven;
- Weight: about 310g (not including battery).

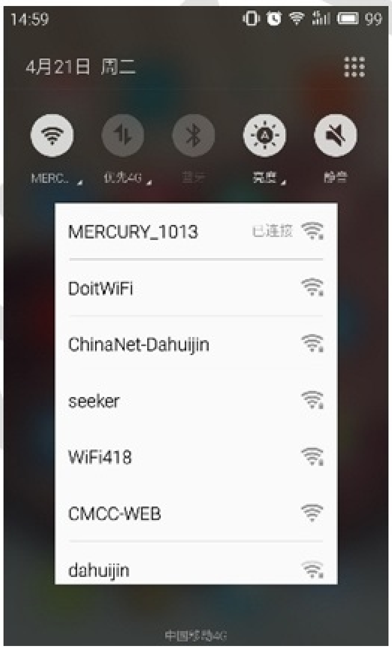
Product Function

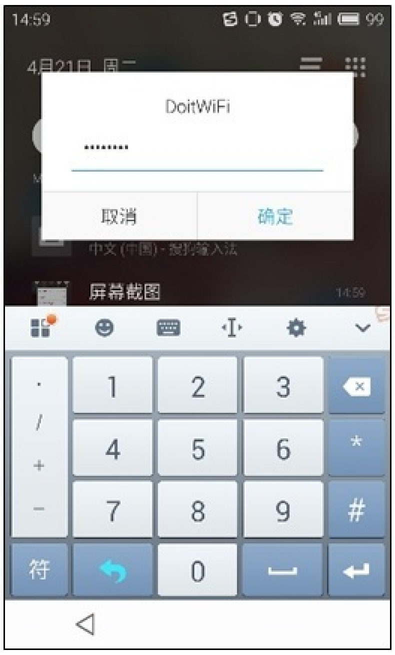
DoitCar WiFi smart car is desigened and developed based on ESP8266 chip, with two basic modes: AP (Access Point) and STA (station). Certainly,AP+STA is also supported at the same time.

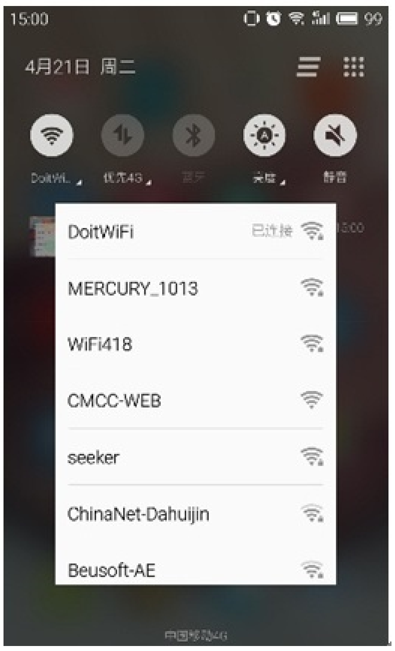
2.1 AP Mode

When get the DoitCar, the default mode is AP. Under this mode, the default SSID name is DoitWiFi, and password is 12345678.

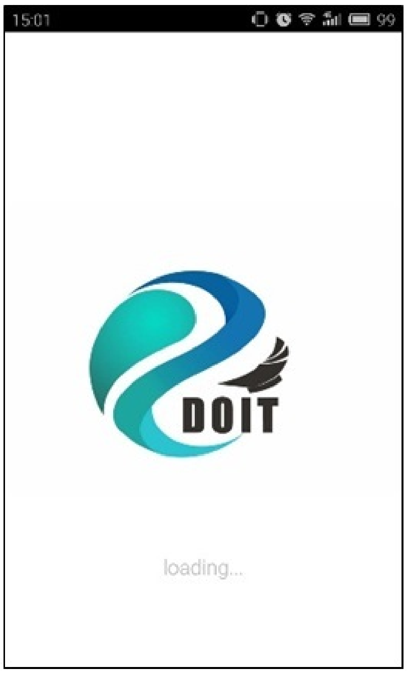
Usage Steps: (1) Open the power from the smart car; (2) Search the AP SSID name DoitWiFi, and then connect it;

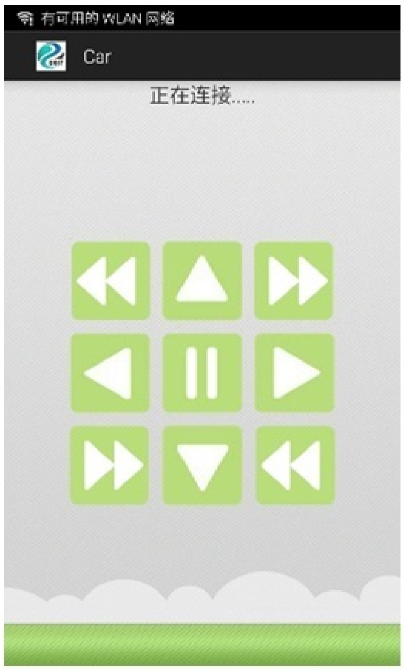


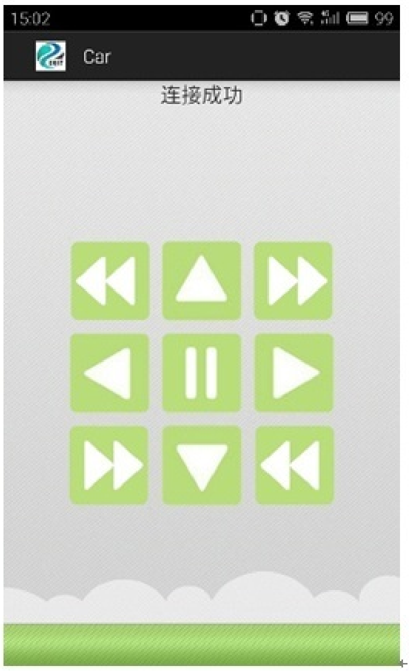




(3) Open the APP from your phone, if your has no this APP, please download (http://bbs.smartarduino.com/showthread.php?tid=4) and install it.





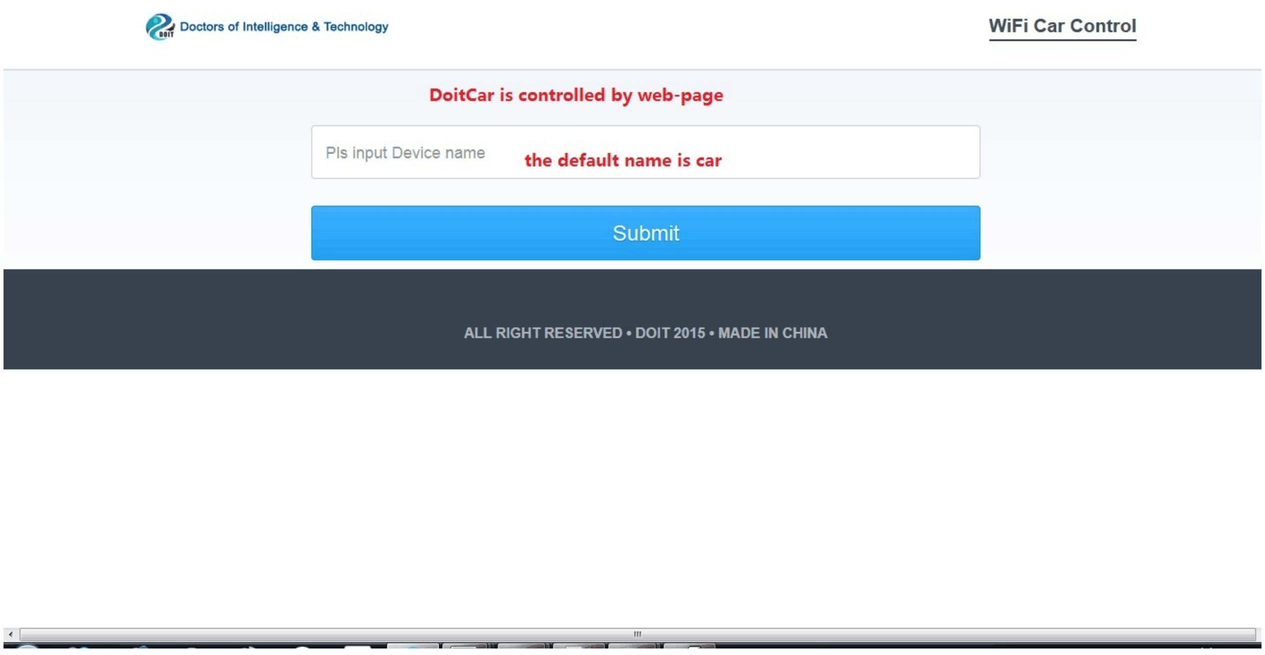


(4) After connect successfully, can let car Forward, Back, Stop, Turn Left, Turn Right, Left to Accelerate, and Right to Accelerate, etc.

2.2 STA Mode

Under the STA mode, the DoitCar can be controlled by phone APP, WeChat, and page from internet. Note that, if using STA mode, you should firstly download the DoitCarControl.lua (https://github.com/SmartArduino/DoitCar) into the DoitCar. For more details about download methods, please visit the documents on NodeMCU development (). Before download the DoitCarControl.lua into the control-board, you must let the SSID name and password in DoitCarControl.lua same as the ones in your router. In addition, in the DoitCarControl.lua, you should rename the car as the one you like (the default name is car). This name is used as the device name controlled by mobile phone, Wechat, and web-page.

The following Figure is shown that car is controlled by web-page.


Doctors of Intelligence & Technology
WiFi Car Control
DoitCar is controlled by web-page
Pls input Device name
the default name is car
Submit
ALL RIGHT RESERVED • DOIT 2015 • MADE IN CHINA

Code Parse

3.1 Code for AP case on DoitCar

This Subsection present the AP operation for the DoitCar in detail. In this case, when NodeMCU works at AP mode, it will listen the TCP connection at the designated port By using the TCP server. Then, the APP (can be downloaded at http://bbs.smartarduino.com/showthread.php?tid=4) can be connected to the TCP server, and can send the control command to control the car.

init.lua

When NodeMCU starts to work, init.lua is used as the entrance of the application. If this file exists, then all the actions can start automatically. Therefore, by this characteristics, some codes can be written here to start automatically.

The code for init. lua is shown as.

```
1     print("\n")
2     print("ESP8266 Started")
3
4     local exefile="DoitCarControl"
5     local luaFile = {exefile..".lua"}
6     for i, f in ipairs(luaFile) do
7         if file.open(f) then
8           file.close()
9           print("Compile File:"..f)
10           node.compile(f)
11           print("Remove File:"..f)
12           file.remove(f)
13         end
14      end
15
16     if file.open(exefile..".lc") then
17         dofile(exefile..".lc")
18     else
19         print(exefile..".lc not exist")
20     end
21     exefile=nil;luaFile = nil
22     collectgarbage()
```

Code Parse:

lines 1 and 2: print the characters;

line 4: deine the compiled and run lc file name. Note that, this file name is not including the suffix .lc and/or .lua;

line 5: define the need to compile .lua file name;

line 6: use for to complete the many operation for files;

line 7: judge whether the files exist. If exists, then compile, or ignore it.

line 8: close the opened file;

line 9-12: complete the compile, and generate automatically "DoitCarControl.lc" file;

line 16-20: judge whether the file exists, if exist, then run the compiled lc file;

line 21-22: release memory.

DoitCarControl.lua

In the DoitCarControl.lua document, it will complete the set-up, start, initiation for GPI, set the adjust of speed by the timer, set-up and listen the TCP server port. After receive the data when finishing the set-up, the program analyze the received data, and realize the control for DoitCar. The source code is listed as follows.

```lua
--GPIO Define
function initGPIO()
--1,2EN     D1 GPIO5
--3,4EN     D2 GPIO4
--1A  ~2A   D3 GPIO0
--3A  ~4A   D4 GPIO2

gpio.mode(0,gpio.OUTPUT);--LED Light on
gpio.write(0,gpio.LOW);

 gpio.mode(1,gpio.OUTPUT);gpio.write(1,gpio.LOW);
 gpio.mode(2,gpio.OUTPUT);gpio.write(2,gpio.LOW);

 gpio.mode(3,gpio.OUTPUT);gpio.write(3,gpio.HIGH);
 gpio.mode(4,gpio.OUTPUT);gpio.write(4,gpio.HIGH);

 pwm.setup(1,1000,1023);--PWM 1KHz, Duty 1023
 pwm.start(1);pwm.setduty(1,0);
 pwm.setup(2,1000,1023);
 pwm.start(2);pwm.setduty(2,0);
 end

 function setupAPMode()
 print("Ready to start soft ap")

 cfg={}
 cfg.ssid="DoitWiFi";
 cfg.pwd="12345678"
 wifi.ap.config(cfg)

 cfg={}
 cfg.ip="192.168.1.1";
 cfg.netmask="255.255.255.0";
 cfg.gateway="192.168.1.1";
 wifi.ap.setip(cfg);
 wifi.setmode(wifi.SOFTAP)

 str=nil;
 ssidTemp=nil;
 collectgarbage();

 print("Soft AP started")
 end

 --Set up AP
 setupAPMode();

 print("Start DoitRobo Control");
 initGPIO();

 spdTargetA=1023;--target Speed
 spdCurrentA=0;--current speed
 spdTargetB=1023;--target Speed
 spdCurrentB=0;--current speed
 stopFlag=true;

 --speed control procedure
 tmr.alarm(1, 200, 1, function()
     if stopFlag==false then
         spdCurrentA=spdTargetA;
         spdCurrentB=spdTargetB;
         pwm.setduty(1,spdCurrentA);
         pwm.setduty(2,spdCurrentB);
     else
         pwm.setduty(1,0);
         pwm.setduty(2,0);
     end
 68     end)
 69
 70     --Setup tcp server at port 9003
 71     s=net.createServer(net.TCP,60);
 72     s:listen(9003,function(c)
 73         c:on("receive",function(c,d)
 74           print("TCPSrv:"..d)
 75           if string.sub(d,1,1)=="0" then --stop
 76             pwm.setduty(1,0)
 77             pwm.setduty(2,0)
 78             stopFlag = true;
 79             c:send("ok\r\n");
 80           elseif string.sub(d,1,1)=="1" then --forward
 81             gpio.write(3,gpio.HIGH)
 82             gpio.write(4,gpio.HIGH)
 83             stopFlag = false;
 84             c:send("ok\r\n");
 85           elseif string.sub(d,1,1)=="2" then --backward
 86             gpio.write(3,gpio.LOW)
 87             gpio.write(4,gpio.LOW)
 88             stopFlag = false;
 89             c:send("ok\r\n");
 90           elseif string.sub(d,1,1)=="3" then --left
gpio.write(3,gpio.LOW)
 92             gpio.write(4,gpio.HIGH)
 93             stopFlag = false;
 94             c:send("ok\r\n");
 95           elseif string.sub(d,1,1)=="4" then --right
 96             gpio.write(3,gpio.HIGH);
 97             gpio.write(4,gpio.LOW);
 98             stopFlag = false;
 99             c:send("ok\r\n");
100  elseif string.sub(d,1,1)=="6" then --A spdUp
101             spdTargetA = spdTargetA+50;if(spdTargetA>1023) then spdTargetA=1023;end
102             c:send("ok\r\n");
103           elseif string.sub(d,1,1)=="7" then --A spdDown
104             spdTargetA = spdTargetA-50;if(spdTargetA《0) then spdTargetA=0;end
105             c:send("ok\r\n");
106           elseif string.sub(d,1,1)=="8" then --B spdUp
107             spdTargetB = spdTargetB+50;if(spdTargetB>1023) then spdTargetB=1023;end
108             c:send("ok\r\n");
109           elseif string.sub(d,1,1)=="9" then --B spdDown
110             spdTargetB = spdTargetB-50;if(spdTargetB《0) then spdTargetB=0;end
111             c:send("ok\r\n");
112           else  print("Invalid Command:"..d);c:send("Invalid CMD\r\n");end;
113           collectgarbage();
114         end) --end c:on receive
115
116         c:on("disconnection",function(c)
117             print("TCPSrv:Client disconnet");
118             collectgarbage();
119         end)
120         print("TCPSrv:Client connected")
121     end)
```

line 1~21: define initGPIO() function, init GPIO port;

line 23-43: define setupAPMode() function used to set up AP mode. SSID is set as "DoitWiFi", password is "12345678";

line 46: run setupAPMode() function;

line 49: run initGPIO() function;

line 51-54: define 4 variables used to remember the current and objective speed for left and right wheels;

line 55: define a label used to remember the stop state;

line 58-68: start timer1, compute the current and objective speed after each 200ms to control the speed. the main idea is that, apk set the objective speed, then by the timer, the current speed output as the cycle of PWM;

line 71: set up TCP server, set the disconnect time as 60s from the client;

line 72-121: set up the listening port, register connect funciton, disconnect function, data-received function. The received-data is parsed in the received function;

line 73: register the data-received function, and line 116 is the disconnection function;

line 74-114: realization of data-received function. judge the received-data, and then present different reponse by the different received data;

line 113: use collectgarbage() to show the release memory.

Log

After run, the log is shown as follows.

```
NodeMCU 0.9.6 build 20150406  powered by Lua 5.1.4


ESP8266 Started
Ready to start soft ap
Soft AP started
Start DoitRobo Control
TCPSrv:Client connected
TCPSrv:1

 TCPSrv:2

 TCPSrv:3

 TCPSrv:4

 TCPSrv:0

 TCPSrv:8

 TCPSrv:9

 TCPSrv:6

 TCPSrv:7

 TCPSrv:0

 TCPSrv:Client disconnet
```

3.2 Code for STA Case on DoitCar

This Subsection presents the STA mode in detail. NodeMCU would be work at STA mode to connect the wireless router. by setting-up TCP client, can connect to the remote server, and realize the remote control by Wechat, web-page and phone APP.

The example is including init.lua, sta.lua, and DoitCarControlSTA.lua.

init.lua and sta.lua

init.lua is the entrance when NodeMCU starts. If no init.lua, then ignore it; if has, then start to run it. Therefore, If necessary, some code can be put here to start automatically. the code for init.lua is shown as follows.

```
23     print("\n")
24     print("ESP8266 Started")
25
26     local exefile="sta"
27     local luaFile = {exefile..".lua","DoitCarControlSTA.lua"}
28     for i, f in ipairs(luaFile) do
29         if file.open(f) then
30           file.close()
31           print("Compile File:"..f)
32           node.compile(f)
33           print("Remove File:"..f)
34           file.remove(f)
35         end
36      end
37
38     if file.open(exefile..".lc") then
39         dofile(exefile..".lc")
40     else
41         print(exefile..".lc not exist")
42     end
43     exefile=nil;luaFile = nil
44     collectgarbage()
```

line 1-2: print character;

line 4: define the compiled and run lc file. Note that, not including the suffix ".lc" and/or ".lua";

line 5: define the compiled lua file name;

line 6: use for cycle to complete the operation of many files;

line 7: judge whether the file exists; if no, ignor it, or compile it;

line 8: close the opened file;

line 9-12: complete the compile, automatically generate "DoitCarControl.lc";

line 16-20: judge whether the file exists, if yes, then compile the lc file;

line 21-22: release the memory.

DoitCarControlSTA.lua

In the DoitCarControlSTA.lua, would complete the initiation of GPIO port, setting-up for TCP client, try to connect periodically, and adjust of speed by timer. after successful connection and the received-data, would parse the data, and then realize the control of DoitCar. The source code is as follows.

```
--GPIO Define
function initGPIO()
--1,2EN     D1 GPIO5
--3,4EN     D2 GPIO4
--1A  ~2A   D3 GPIO0
--3A  ~4A   D4 GPIO2

gpio.mode(0,gpio.OUTPUT);--LED Light on
gpio.write(0,gpio.LOW);

gpio.mode(1,gpio.OUTPUT);gpio.write(1,gpio.LOW);
gpio.mode(2,gpio.OUTPUT);gpio.write(2,gpio.LOW);

gpio.mode(3,gpio.OUTPUT);gpio.write(3,gpio.HIGH);
gpio.mode(4,gpio.OUTPUT);gpio.write(4,gpio.HIGH);

pwm.setup(1,1000,1023);--PWM 1KHz, Duty 1023
pwm.start(1);pwm.setduty(1,0);
pwm.setup(2,1000,1023);
pwm.start(2);pwm.setduty(2,0);
end

--Control Program
print("Start DoitRobo Control");
initGPIO();

spdTargetA=1023;--target Speed
spdCurrentA=0;--current speed
spdTargetB=1023;--target Speed
spdCurrentB=0;--current speed
stopFlag=true;

tmr.alarm(1, 200, 1, function()
    if stopFlag==false then
        spdCurrentA=spdTargetA;
        spdCurrentB=spdTargetB;
        pwm.setduty(1,spdCurrentA);
        pwm.setduty(2,spdCurrentB);
    else
        pwm.setduty(1,0);
        pwm.setduty(2,0);
    end
end)

local flagClientTcpConnected=false;
print("Start TCP Client");
tmr.alarm(3, 5000, 1, function()
    if flagClientTcpConnected==false then
    print("Try connect Server");
    local conn=net.createConnection(net.TCP, false)
    conn:connect(6005,"182.92.178.210");
    conn:on("connection",function(c)
        print("TCPClient:conneted to server");
        flagClientTcpConnected = true;
        end)
    conn:on("disconnection",function(c)
        flagClientTcpConnected = false;
        conn=nil;
        collectgarbage();
    end)
    conn:on("receive", function(conn, m)
        print("TCPClient:"..m);
        if string.sub(m,1,1)=="b" then
            conn:send("cmd=subscribe&topic=".."car".."\r\n");
        elseif string.sub(m,1,1)=="0" then --stop
            pwm.setduty(1,0)
            pwm.setduty(2,0)
            stopFlag = true;
            conn:send("ok\r\n");
        elseif string.sub(m,1,1)=="1" then --forward
            gpio.write(3,gpio.HIGH)
193                 gpio.write(4,gpio.HIGH)
194                 stopFlag = false;
195                 conn:send("ok\r\n");
196             elseif string.sub(m,1,1)=="2" then --backward
197                 gpio.write(3,gpio.LOW)
198                 gpio.write(4,gpio.LOW)
199                 stopFlag = false;
200                 conn:send("ok\r\n");
201             elseif string.sub(m,1,1)=="3" then --left
202                 gpio.write(3,gpio.LOW)
203                 gpio.write(4,gpio.HIGH)
204                 stopFlag = false;
205                 conn:send("ok\r\n");
206             elseif string.sub(m,1,1)=="4" then --right
207                 gpio.write(3,gpio.HIGH);
208                 gpio.write(4,gpio.LOW);
209                 stopFlag = false;
210                 conn:send("ok\r\n");
211             elseif string.sub(m,1,1)=="6" then --A spdUp
212                 spdTargetA = spdTargetA+50;if(spdTargetA>1023) then spdTargetA=1023;end
213                 conn:send("ok\r\n");
214             elseif string.sub(m,1,1)=="7" then --A spdDown
215                 spdTargetA = spdTargetA-50;if(spdTargetA《0) then spdTargetA=0;end
216                 conn:send("ok\r\n");
217             elseif string.sub(m,1,1)=="8" then --B spdUp
218                 spdTargetB = spdTargetB+50;if(spdTargetB>1023) then spdTargetB=1023;end
219                 conn:send(spdTargetA.." "..spdTargetB.."\r\n");
220             elseif string.sub(m,1,1)=="9" then --B spdDown
221                 spdTargetB = spdTargetB-50;if(spdTargetB<0) then spdTargetB=0;end
222                 conn:send(spdTargetA.." "..spdTargetB.."\r\n");
223             else  print("Invalid Command:"..m);end;
224                 collectgarbage();
225             end)
226         end
227     end)
```

line 1-21: define initGPIO() function, init GPIO port;

line 25: run initGPIO() function;

line 27-30: define 4 variable used to remember the current and objective speed of left and right wheels;

line 31: define a label used to remember the stop state;

line 33-34: start the periodic timer1. It would compute the current and objective speed after each 200ms to realize the control of speed. By the timer, the current speed ouputs as a PWM cycle.

line 45: use the variable flagClientTcpConnected to remember the connection state of TCP client;

line 47: use the periodic timer3 to handle the TCP connection after each 5ms. Judge whether it is necessory to send a connection requirment by the flagClientTcpConnected. In this section, the server IP IS "182.92.178.210", port="6005";

line 52-60: register the "connection" and "disconnection" case for the TCP Client, respectively;

line 61-104: show the code for realization of the received-data. By the different received-data, can do the relative response. In addtion, line 64 is sent the device name. When NodeMCU is connected to the remote server, then the character "b" is returned. At this time, the device name is need to submitted to the server. Note that, this device name can be used for the control by phone APP, web-page, and/or Wechat. In this section, the device name is tank;

line 103: use collectgarbage() function to show the release memory.

Log for this program

```
NodeMCU 0.9.6 build 20150406  powered by Lua 5.1.4


ESP8266 Started
Compile File:sta.lua
Remove File:sta.lua
Compile File:DoitCarControlSTA.lua
Remove File:DoitCarControlSTA.lua
Ready to Set up wifi mode
> Trying Connect to Router, Waiting...
Trying Connect to Router, Waiting...
Config done, IP is 192.168.1.111
Start DoitRobo Control
Start TCP Client
Try connect Server
TCPClient:conneted to server
TCPClient:b

TCPClient:cmd=subscribe&res=1

Invalid Command:cmd=subscribe&res=1

TCPClient:1

TCPClient:2

TCPClient:3

TCPClient:4
```

Revision History

Version	Content	Date
1.0	DrALt Version	2015-05-19

Technical Support

For more information about our products, please visit http://www.doit.am.

Contact Information:

Company	Shenzhen Doctors of Intelligence & Technology (SZDOIT)
Tel	+86-158 9988 0115
skype	yichone
Emails	support@doit.am;yichoneyi@163.com

How to Get it

The WiFi smart car kit is at: http://www.smartarduino.com/2wd-wifi-rc-smart-car-with-nodemcu-shield-for-esp-12e_p94572.html