



深圳四博智联科技有限公司 Shenzhen Four Primus Union Technology Co., Ltd
<http://www.doit.am> <http://www.smartarduino.com> Tel: 186 7666 2425



深圳四博智联科技有限公司

基于 DoitCar 的 STA 案例

二〇一五年五月



本案例详细讲解了基于 DoitCar 的 STA 模式操作过程。程序使 NodeMCU 的工作在 STA 模式，连接到无线路由器。通过建立 TCP 客户端，连接到远程服务器，实现微信、网页以及手机 APK 的远程控制。

本案例程序源码见资料包，包括：init.lua、sta.lua 和 DoitCarControlSTA.lua 三个文件。

1 init.lua 和 sta.lua 文件

NodeMCU 在启动时预留了 init.lua 作为应用程序入口，如果没有该文件则忽略，如果存在则开始执行该文件。利用这个特性可以在 init.lua 中写入需要执行的代码，以便上电自动运行。本案例使用的 init 文件如下所示。

资料包中文件名为“init.lua”。

```
1  print("\n")
2  print("ESP8266 Started")
3
4  local exeFile="sta"
5  local luaFile = {exeFile..".lua","DoitCarControlSTA.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()
```

程序第 1、2 行打印字符信息。

第 3 行定义编译后需要执行的 lc 文件名，注意不包含后缀 “.lc” 或者 “.lua”。

第 5 行定义需要编译的 lua 文件名。

第 6 行是使用 for 循环完成多个文件的操作。

第 7 行判断文件是否存在，如果存在则执行编译。如果不存在则忽略。

第 8 行是关闭已经打开的文件。

第 9~12 行完成编译，自动生成 “DoitCarControl.lc” 文件。

第 16~20 行判断文件是否存在，如果存在则执行刚刚编译完成的 lc 文件。

第 21~22 行是回收内存。



“sta.lua”文件源码如下:

```
1 print("Ready to Set up wifi mode")
2 wifi.setmode(wifi.STATION)
3
4 wifi.sta.config("MERCURY_1013","123456789")--ssid and password
5 wifi.sta.connect()
6 local cnt = 0
7 tmr.alarm(3, 1000, 1, function()
8     if (wifi.sta.getip() == nil) and (cnt < 20) then
9         print("Trying Connect to Router, Waiting...")
10        cnt = cnt + 1
11    else
12        tmr.stop(3)
13        if (cnt < 20) then print("Config done, IP is "..wifi.sta.getip())
14        else print("Wifi setup time more than 20s, Please verify wifi.sta.config() function.
15        Then re-download the file.")
16        end
17        cnt = nil;
18        collectgarbage();
19        dofile("DoitCarControlSTA.lc");
20    end
21 end)
```

其中第 4 行是需要设置自己的无线路由器的 SSID 及密码，以便 NodeMCU 连接到该路由器。

第 18 行是在完成连接后执行 DoitCarControlSTA.lc 程序。

2 DoitCarControlSTA.lua 文件

在 DoitCarControlSTA.lua 文件中完成 GPIO 端口初始化，TCP 客户端的创建、周期性尝试连接，设置定时器对速度进行调节。在连接成功并收到数据后，对数据进行解析，实现 DoitCar 小车电机的控制。

源码如下。

```
1 --GPIO Define
2 function initGPIO()
3 --1,2EN D1 GPIO5
4 --3,4EN D2 GPIO4
5 --1A ~2A D3 GPIO0
6 --3A ~4A D4 GPIO2
7
8 gpio.mode(0,gpio.OUTPUT);--LED Light on
9 gpio.write(0,gpio.LOW);
10
11 gpio.mode(1,gpio.OUTPUT);gpio.write(1,gpio.LOW);
12 gpio.mode(2,gpio.OUTPUT);gpio.write(2,gpio.LOW);
```



```
13
14 gpio.mode(3,gpio.OUTPUT);gpio.write(3,gpio.HIGH);
15 gpio.mode(4,gpio.OUTPUT);gpio.write(4,gpio.HIGH);
16
17 pwm.setup(1,1000,1023);--PWM 1KHz, Duty 1023
18 pwm.start(1);pwm.setduty(1,0);
19 pwm.setup(2,1000,1023);
20 pwm.start(2);pwm.setduty(2,0);
21 end
22
23 --Control Program
24 print("Start DoitRobo Control");
25 initGPIO();
26
27 spdTargetA=1023;--target Speed
28 spdCurrentA=0;--current speed
29 spdTargetB=1023;--target Speed
30 spdCurrentB=0;--current speed
31 stopFlag=true;
32
33 tmr.alarm(1, 200, 1, function()
34   if stopFlag==false then
35     spdCurrentA=spdTargetA;
36     spdCurrentB=spdTargetB;
37     pwm.setduty(1,spdCurrentA);
38     pwm.setduty(2,spdCurrentB);
39   else
40     pwm.setduty(1,0);
41     pwm.setduty(2,0);
42   end
43 end)
44
45 local flagClientTcpConnected=false;
46 print("Start TCP Client");
47 tmr.alarm(3, 5000, 1, function()
48   if flagClientTcpConnected==false then
49     print("Try connect Server");
50     local conn=net.createConnection(net.TCP, false)
51     conn:connect(6005,"182.92.178.210");
52     conn:on("connection",function(c)
53       print("TCPClient:conneted to server");
54       flagClientTcpConnected = true;
55     end)
56     conn:on("disconnection",function(c)
```



```
57     flagClientTcpConnected = false;
58     conn=nil;
59     collectgarbage();
60     end)
61     conn:on("receive", function(conn, m)
62         print("TCPClient: "..m);
63         if string.sub(m,1,1)=="b" then
64             conn:send("cmd=subscribe&topic=.."car".."r\n");
65         elseif string.sub(m,1,1)=="0" then --stop
66             pwm.setduty(1,0)
67             pwm.setduty(2,0)
68             stopFlag = true;
69             conn:send("ok\r\n");
70         elseif string.sub(m,1,1)=="1" then --forward
71             gpio.write(3,gpio.HIGH)
72             gpio.write(4,gpio.HIGH)
73             stopFlag = false;
74             conn:send("ok\r\n");
75         elseif string.sub(m,1,1)=="2" then --backward
76             gpio.write(3,gpio.LOW)
77             gpio.write(4,gpio.LOW)
78             stopFlag = false;
79             conn:send("ok\r\n");
80         elseif string.sub(m,1,1)=="3" then --left
81             gpio.write(3,gpio.LOW)
82             gpio.write(4,gpio.HIGH)
83             stopFlag = false;
84             conn:send("ok\r\n");
85         elseif string.sub(m,1,1)=="4" then --right
86             gpio.write(3,gpio.HIGH);
87             gpio.write(4,gpio.LOW);
88             stopFlag = false;
89             conn:send("ok\r\n");
90         elseif string.sub(m,1,1)=="6" then --A spdUp
91             spdTargetA = spdTargetA+50;if(spdTargetA>1023) then
spdTargetA=1023;end
92             conn:send("ok\r\n");
93         elseif string.sub(m,1,1)=="7" then --A spdDown
94             spdTargetA = spdTargetA-50;if(spdTargetA<0) then spdTargetA=0;end
95             conn:send("ok\r\n");
96         elseif string.sub(m,1,1)=="8" then --B spdUp
97             spdTargetB = spdTargetB+50;if(spdTargetB>1023) then
spdTargetB=1023;end
98             conn:send(spdTargetA.." "..spdTargetB.."r\n");
```



```
99     elseif string.sub(m,1,1)=="9" then --B spdDown
100         spdTargetB = spdTargetB-50;if(spdTargetB<0) then spdTargetB=0;end
101         conn:send(spdTargetA.." "..spdTargetB.."\\r\\n");
102     else print("Invalid Command: "..m);end;
103         collectgarbage();
104     end)
105 end
106 end)
```

程序第 1~21 行定义 `initGPIO()` 函数，初始化 GPIO 口。

第 25 行执行 `initGPIO()` 函数。

第 27~30 行定义了四个变量，用于记录左轮、右轮的当前速度和目标设置速度。

第 31 行定义一个标志，用于记录停止状态。

第 33~43 行启动了周期性定时器 1，每隔 200 毫秒对当前速度和目标速度进行计算，实现速度控制。其基本思路是：`apk` 设置目标速度，程序在定时器中，将当前速度直接作为 PWM 占空比设置值进行输出。

第 45 行启用 `flagClientTcpConnected` 变量，记录 TCP 客户端的连接状态。

第 47 行启动周期性定时器 3，每隔 5 秒钟检查一次 TCP 连接情况，并进行相应的处理。通过 `flagClientTcpConnected` 判断是否需要向服务器发起连接请求。本案例的服务器地址为：“182.92.178.210”，端口为“6005”。

第 52~60 行是分别注册了 TCP 客户端的“`connection`”和“`disconnection`”事件。

第 61~104 行是数据接收函数的实现代码。在这里面判断接收数据，并根据不同数据执行响应的动作。

其中，第 64 行是向服务器发送设备名称。当 NodeMCU 连接到远程服务器时，服务器会返回字符“`b`”，此时需要向服务器报告设备名称。该设备名称用于手机 `apk`、网页或者微信控制。本教程设置名称为“`tank`”。

第 103 行，使用了 `collectgarbage()` 函数显示的回收内存。

3 程序 Log

程序运行 Log 如下。

```
1  NodeMCU 0.9.6 build 20150406  powered by Lua 5.1.4
2
3
4  ESP8266 Started
5  Compile File:sta.lua
6  Remove File:sta.lua
7  Compile File:DoitCarControlSTA.lua
8  Remove File:DoitCarControlSTA.lua
9  Ready to Set up wifi mode
10 > Trying Connect to Router, Waiting...
11 Trying Connect to Router, Waiting...
12 Config done, IP is 192.168.1.111
13 Start DoitRobo Control
14 Start TCP Client
```



```
15 Try connect Server
16 TCPClient:conneted to server
17 TCPClient:b
18
19 TCPClient:cmd=subscribe&res=1
20
21 Invalid Command:cmd=subscribe&res=1
22
23 TCPClient:1
24
25 TCPClient:2
26
27 TCPClient:3
28
29 TCPClient:4
30
```

doit