



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 AP 案例

二〇一五年五月



本案例详细讲解了基于 DoitCar 的 AP 操作过程。程序使 NodeMCU 的工作在 AP 模式，通过建立 TCP 服务器，在指定端口监听 TCP 连接。手机软件（Car_AP.apk）可以连接到 TCP 服务器并发送控制指令实现小车的控制。

本案例程序源码见资料包。

1 init.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="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()
```

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。



2 DoitCarControl.lua 文件

在 DoitCarControl.lua 文件中完成 ap 模式的建立和启动、初始化 GPIO 端口，设置定时器对速度进行调节，建立 TCP 服务器端口并实现端口监听。在建立连接并收到数据后，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 function setupAPMode()
24 print("Ready to start soft ap")
25
26 cfg={}
27 cfg.ssid="DoitWiFi";
28 cfg.pwd="12345678"
29 wifi.ap.config(cfg)
30
31 cfg={}
32 cfg.ip="192.168.1.1";
33 cfg.netmask="255.255.255.0";
34 cfg.gateway="192.168.1.1";
35 wifi.ap.setip(cfg);
36 wifi.setmode(wifi.SOFTAP)
37
38 str=nil;
```



```
39  ssidTemp=nil;
40  collectgarbage();
41
42  print("Soft AP started")
43  end
44
45  --Set up AP
46  setupAPMode();
47
48  print("Start DoitRobo Control");
49  initGPIO();
50
51  spdTargetA=1023;--target Speed
52  spdCurrentA=0;--current speed
53  spdTargetB=1023;--target Speed
54  spdCurrentB=0;--current speed
55  stopFlag=true;
56
57  --speed control procedure
58  tmr.alarm(1, 200, 1, function()
59    if stopFlag==false then
60      spdCurrentA=spdTargetA;
61      spdCurrentB=spdTargetB;
62      pwm.setduty(1,spdCurrentA);
63      pwm.setduty(2,spdCurrentB);
64    else
65      pwm.setduty(1,0);
66      pwm.setduty(2,0);
67    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
91         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 disconnnet");
118        collectgarbage();
119    end)
120    print("TCPSrv:Client connected")
121 end)
```

程序第 1~21 行定义 initGPIO()函数，初始化 GPIO 口。

第 23~43 行定义 setupAPMode()函数，用于建立 AP 热点。SSID 设置为：“DoitWiFi”，密码设置为“12345678”。

第 46 行执行 setupAPMode()函数。

第 49 行执行 initGPIO()函数。



第 51~54 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 55 行定义一个标志，用于记录停止状态。

第 58~68 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 71 行建立 TCP 服务器，设置客户端断开超时时间为 60 秒。

第 72~121 行建立监听端口，注册连接函数、断开连接函数、数据接收函数，在数据接收函数中对数据进行解析。第 73 行是注册数据接收函数。第 116 行是注册断开连接函数。

第 74~114 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。

第 113 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  Ready to start soft ap
6  Soft AP started
7  Start DoitRobo Control
8  > TCPSrv:Client connected
9  TCPSrv:1
10
11 TCPSrv:2
12
13 TCPSrv:3
14
15 TCPSrv:4
16
17 TCPSrv:0
18
19 TCPSrv:8
20
21 TCPSrv:9
22
23 TCPSrv:6
24
25 TCPSrv:7
26
27 TCPSrv:0
28
29 TCPSrv:Client disconnet
```