Shenzhen Doctors of Intelligence & Technology (SZDOIT)

User Manual for ESP12E DevKit

UART-WiFi Transparent Transmission Module Based on Lua

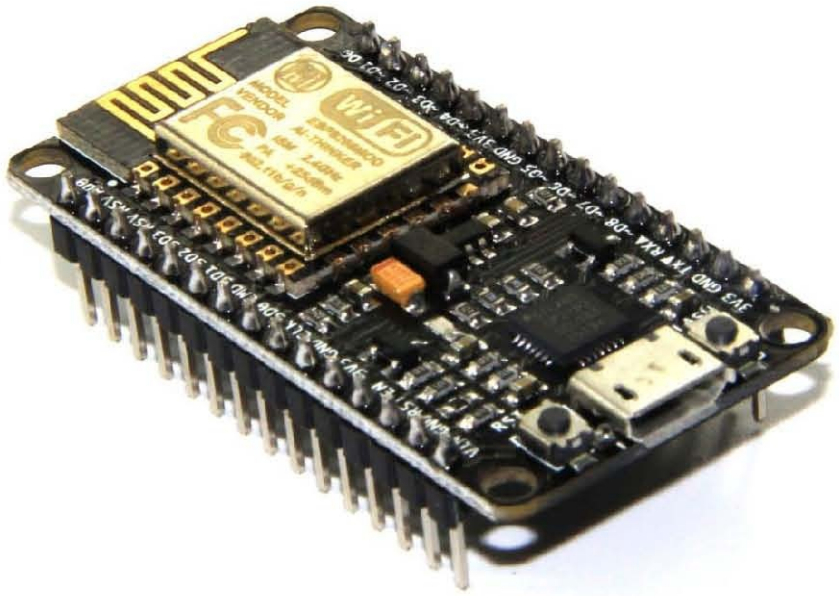


Table of Contents

Introduction

ESP-12E is designed and developed by Shenzhen Doctors of Intelligence & Technology (SZDOIT) based on the Ultra-low power consumption UART-WiFi ESP8266, which is specially for mobile devices and application of IoT (Internet of Things). Now, ESP-12E is widely applied to internet, communication in local area, intelligent home, industrial control, handed-devices, and etc.

ESP-12E DevKit is used the design of on-board antenna and encapsulated by 2.54 direct insertion. It is very convenient to debug and install device.

In ESP-12E DevKit, Hardware API operation is encapsulated by Lua language, which can avoid the hardware difficulty for sorfware engineers, and then can speed the develop of products.

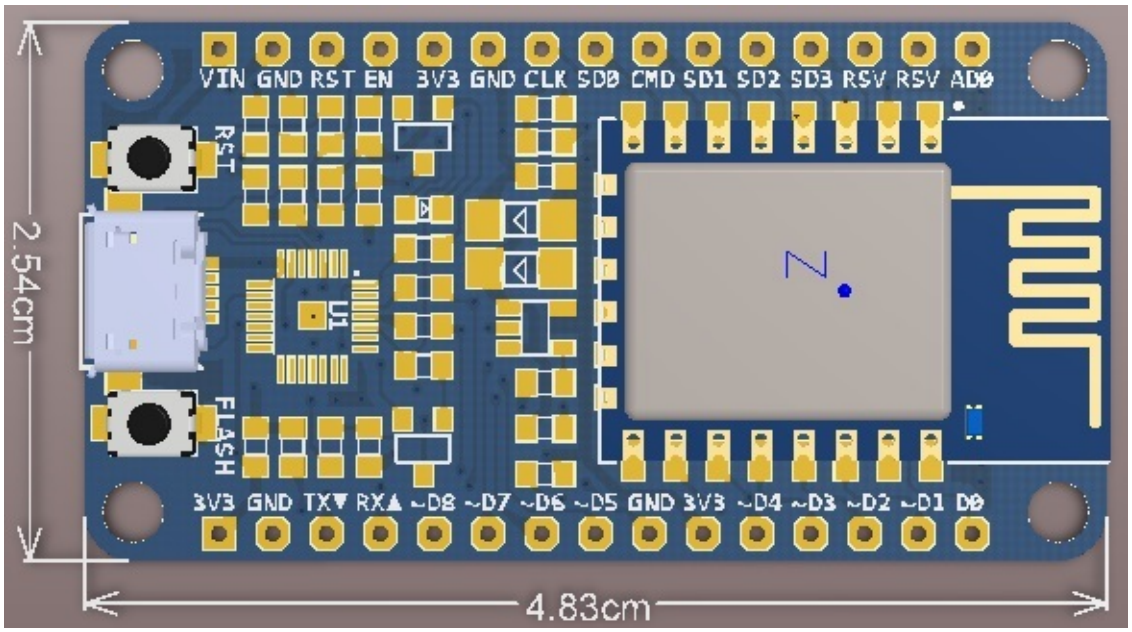
For more information, please visit http://www.doit.am, http://www.smartarduino.com.

Technical Specifications

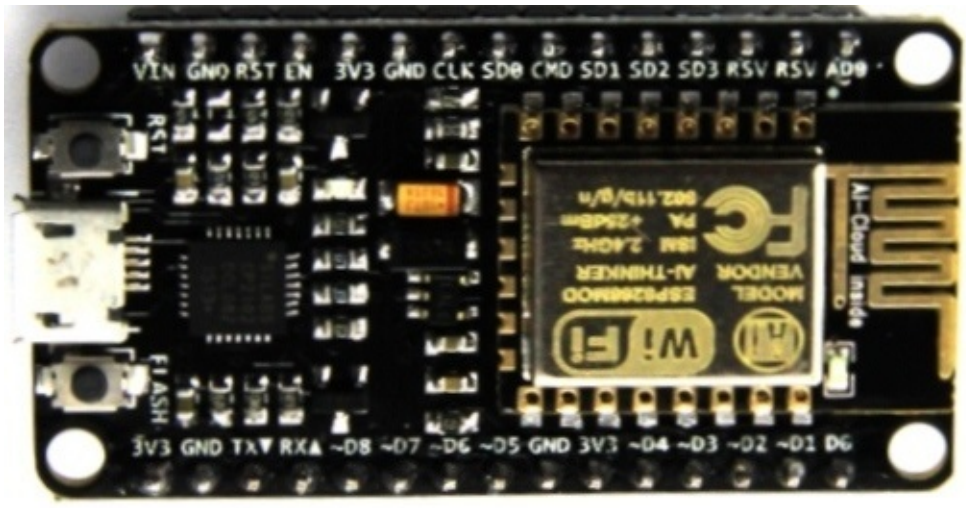
- Support STA/AP/STA+AP 3 working modes;
- Built-in TCP/IP protocol stack, support multiple-channel TCP Client connection (max 5);
- 0～D8, SD1～SD3: used for GPIO, PWM, IIC, ect; the driven ability can be arrived at 15mA;
- AD0: one-way ADC;
- Power input: 4.5V~9V(10VMAX), support USB powered and USB debug;
- Working current: ≈70mA(200mA MAX, continue), standby＜200uA;
- Transmission data rate: 110-460800bps;
- Support UART/GPIO data communication interface;
- Support update firmware remotely (OTA);
- Support Smart Link;
- Working temperature:-40℃～＋125℃；
- Driven mode: double large-power H bridge driven
- Weight: 7g.

Mechanical Size

PCB view for ESP-12E DevKit



Product view for ESP-12E DevKit



Interface and Functions

For this board, pins are screen-printed on the board, and defined as follows.

Table Definition of pins

Name	Function	Input/output
AD0	Analog sample	input
RSV	reserve	-
SD2	GPIO	Input/output
SD3	GPIO	Input/output
SD1	SPI INT	-
CMD	SPI MOSI	-
SD0	SPI MISO	-
CLK	SPI CLK	-
EN	Chip enable	input
RST	ESP12E rest	input
D0	数字 IO	Input/output
D1～8	Digital IO with PWM function	Input/output
3V3	3.3V	-
GND	GND	-
VIN	Power 4.5-9V	input
RX	UART receive	input
TX	UART output	output

Start quickly

ESP-12E DevKit is already built-in Lua fireware with AP mode, together with the following steps:

1. Let ESP-12E DevKit connect to your computer with MicroUSB. ESP-12E module would be brighten with blue light, and then the light is out;

2. If the driver is not installed in your PC, an information would be pop-up to show an unknown device. At this time, a serial-driver for CP2102 must be installed by the OS.

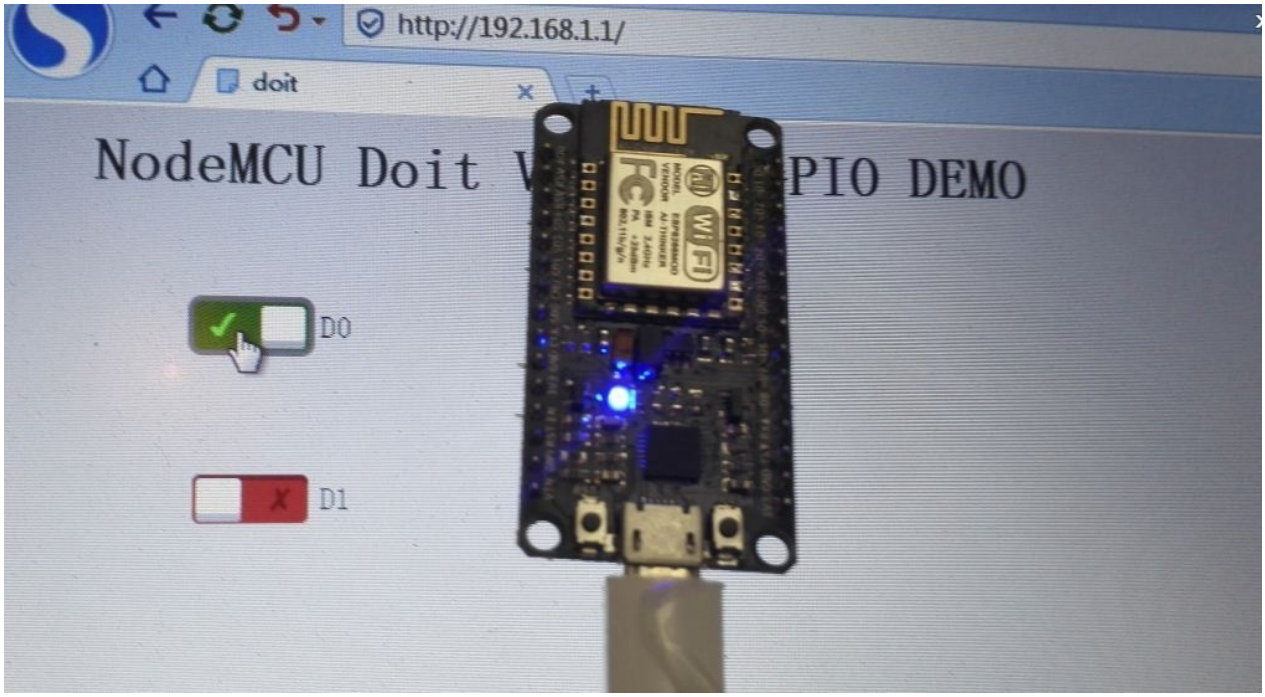
3. Open the wireless network to look for a WiFi wireless network named as DoitWiFi (its SSID), and connect it. If use

mobile phone brower, please visit the source code in this book.

4. The initial passowrd is 12345678.

5. Input 192.168.1.1 in the brower, then have

6. Click "D0", the light is brighten with blue color;



7. Test is completed.

For more information about ESP8266, please see www.ai-thinker.com, and more information for Lua, please visit www.doit.am.

Revision History

Version	Content	Date
1.0	DrALt Version	2015-05-19

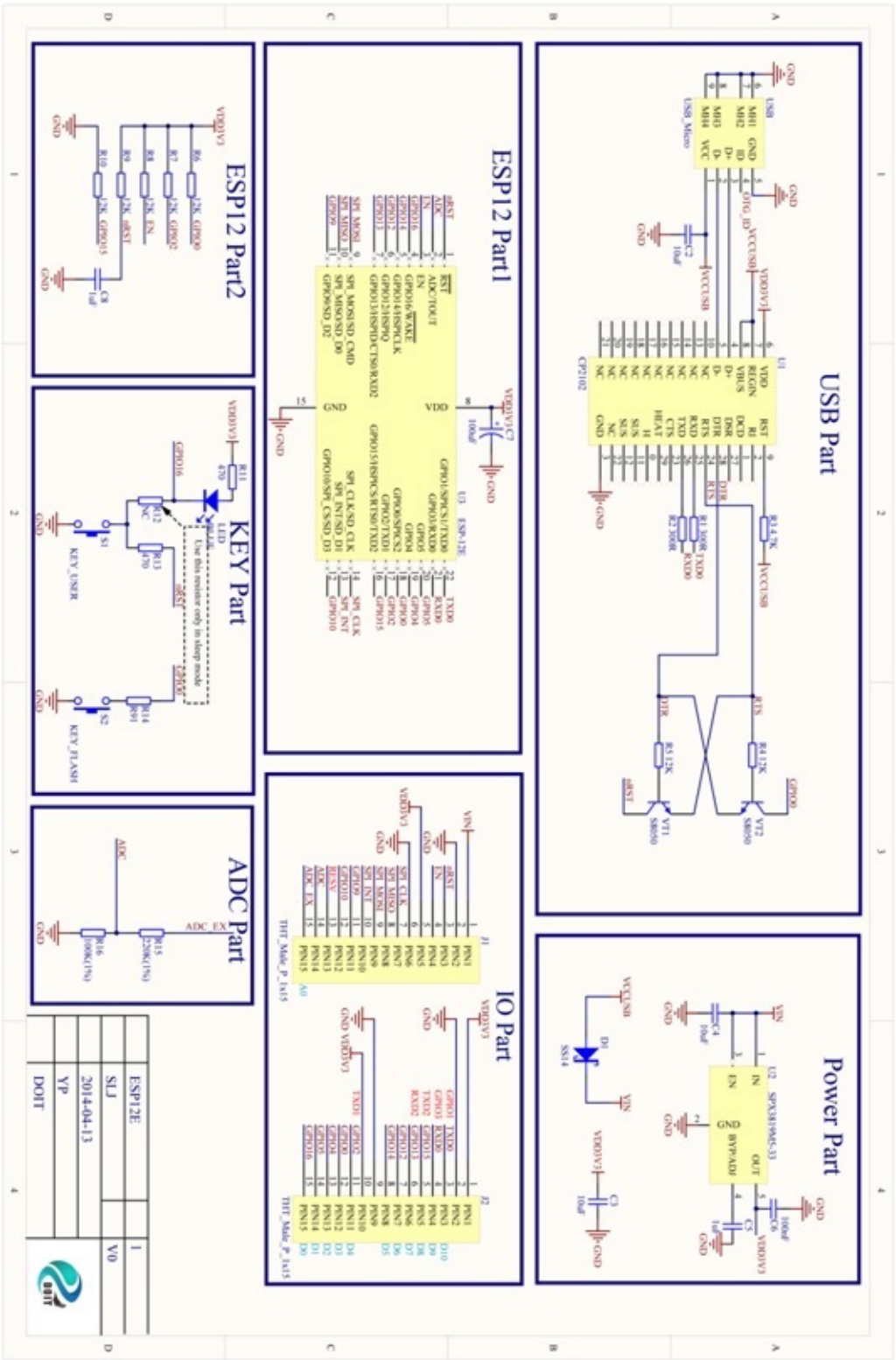
Technique Support

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

Schematics for ESP-12E DevKit



Source Code

File1：init.lua

```
--Doit WiFi Robo Car Ctronl Demo
--ap mode
--Created @ 2015-05-13 by Doit Studio
--Modified: null
--Global Site:    http://doit.am/
--China Site:    http://cn.doit.am/
--Global Shop:    http://www.smartarduino.com/
--China Shop:    http://szdoit.taobao.com/
--Chinese BBS:     bbs.iot.fm

print("\n")
print("ESP8266 Started")

local exefile="webserver"
local luaFile = {exefile..".lua"}
for i, f in ipairs(luaFile) do
    if file.open(f) then
      file.close()
      print("Compile File:"..f)
      node.compile(f)
      print("Remove File:"..f)
      file.remove(f)
    end
 end

if file.open(exefile..".lc") then
    dofile(exefile..".lc")
else
    print(exefile..".lc not exist")
end
exefile=nil;luaFile = nil
collectgarbage()
```

File2：WebServer.lua

```
--Doit WiFi Robo Car Ctronl Demo
--ap mode
--Created @ 2015-05-13 by Doit Studio
--Modified: null
--Global Site:    http://doit.am/
--China Site:    http://cn.doit.am/
--Global Shop:    http://www.smartarduino.com/
--China Shop:    http://szdoit.taobao.com/
--Chinese BBS:     bbs.iot.fm
--[ is used to replace 《

print("Start soft AP")

wifi.setmode(wifi.SOFTAP)
local 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)

start_init = function()
gpio.mode(0, gpio.OUTPUT);
gpio.mode(1, gpio.OUTPUT);
gpio.write(0,gpio.HIGH);
```

```
gpio.write(1,gpio.HIGH);
D1_state=0;
D0_state=0;
end

sendFileContents = function(conn, filename)
if file.open(filename, "r") then
    --conn:send(responseHeader("200 OK","text/html"));
    repeat
    local line=file.readline()
    if line then
        conn:send(line);
    end
    until not line
    file.close();
    else
    conn:send(responseHeader("404 Not Found","text/html"));
    conn:send("Page not found");
    end
end

responseHeader = function(code, type)
    return "HTTP/1.1 " .. code .. "\r\nConnection: close\r\nServer: nunu-Luaweb\r\nContent-Type: " ..
    type .. "\r\n\r\n";
end

httpserver = function ()
    start_init();
    srv=net.createServer(net.TCP)
    srv:listen(80,function(conn)
    conn:on("receive",function(conn,request)
    conn:send(responseHeader("200 OK","text/html"));
    if string.find(request,"gpio=0") then
        if D0_state==0 then
           D0_state=1;gpio.write(0,gpio.LOW);
        else
           D0_state=0;gpio.write(0,gpio.HIGH);
        end
    elseif string.find(request,"gpio=1") then
        if D1_state==0 then
           D1_state=1;gpio.write(1,gpio.LOW);
        else
           D1_state=0;gpio.write(1,gpio.HIGH);
        end
    else
        if D0_state==0 then
            preset0_on="";
        end
        if D0_state==1 then
            preset0_on="checked=\"checked\"";
        end
        if D1_state==0 then
            preset1_on="";
        end
        if D1_state==1 then
            preset1_on="checked=\"checked\"";
        end

        sendFileContents(conn,"header.htm");
        conn:send("[div>[input type=\"checkbox\" id=\"checkbox0\" name=\"checkbox0\" class=\"switch\" onclick=\"loadXML
        conn:send("[label for=\"checkbox0\">D0[/label>[/div>");
        conn:send("[div>[input type=\"checkbox\" id=\"checkbox1\" name=\"checkbox1\" class=\"switch\" onclick=\"loadXML
        conn:send("[label for=\"checkbox1\">D1[/label>[/div>");
        conn:send("[/div>");
    end
        print(request);
    end)
        conn:on("sent",function(conn)
                    conn:close();
                    conn = nil;
                    end)
    end)
    end

    httpserver()
```

File3: header.htm

```
<html>
<head>
<title>doit</title>
<style>
body
{
font-family: sans-serif;
font-weight: normal;
margin: 10px;
color: #555;
background-color: #eee;
}
form
{
margin: 40px 0;
}
div
{
clear: both;
margin: 0 50px;
}
input.switch:empty
{
margin-left: -999px;
}
input.switch:empty ~ label
{
position: relative;
float: left;
line-height: 1.6em;
text-indent: 4em;
margin: 2em 0;
cursor: pointer;
-webkit-user-select: none;
-moz-user-select: none;
-ms-user-select: none;
user-select: none;
}
input.switch:empty ~ label:before,
input.switch:empty ~ label:after
{
position: absolute;
display: block;
top: 0;
bottom: 0;
left: 0;
content: '\2718';
width: 3.6em;
text-indent: 2.4em;
color: #900;
background-color: #c33;
border-radius: 0.3em;
box-shadow: inset 0 0.2em 0 rgba(0,0,0,0.3);
}
input.switch:empty ~ label:after
{
content: ' ';
width: 1.4em;
top: 0.1em;
bottom: 0.1em;
text-align: center;
text-indent: 0;
margin-left: 0.1em;
color: #f88;
background-color: #fff;
border-radius: 0.15em;
box-shadow: inset 0 -0.2em 0 rgba(0,0,0,0.2);
-webkit-transition: all 100ms ease-in;
transition: all 100ms ease-in;
}
/* toggle on */
input.switch:checked ~ label:before
{
content: '\2714';
text-indent: 0.5em;
color: #6f6;
background-color: #393;
}
input.switch:checked ~ label:after
{
margin-left: 2.1em;
color: #6c6;
}
/* focus styles */
input.switch:focus ~ label
{
color: #000;
}
input.switch:focus ~ label:before
{
box-shadow: 0 0 0 3px #999;
}
</style>
<script language="javascript">
function loadXMLDoc(gpio)
{
var xmlhttp;
if (window.XMLHttpRequest)
{
xmlhttp=new XMLHttpRequest();
}
else
{
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.open("GET","gpio=" + gpio+".pht",true); xmlhttp.send()
}
</script>
</head>
<body>
<div>
<h1>NodeMCU Doit Version GPIO DEMO</h1>
```

How to Get it

The NodeMCU kit can get at: http://www.smartarduino.com/nodemcu-development-kit-nodemcu-motor-shield-l293d_p94573.html