

优秀不够，你是否无可替代

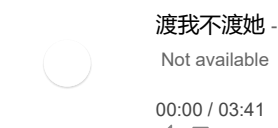
知识从未如此性感。烂程序员关心的是代码,好程序员关心的是数据结构和它们之间的关系 --QQ群: 607064330 --本人
QQ:946029359 --淘宝 <https://shop411638453.taobao.com/>

随笔 - 751, 文章 - 0, 评论 - 317, 阅读 - 183万

导航

博客园
首页
新随笔
联系
订阅 8414
管理

公告



- 1 渡我不渡她
- 2 小镇姑娘
- 3 PDD洪荒之力



昵称：杨奉武
园龄：5年10个月
粉丝：637
关注：1

搜索

 找找看
 谷歌搜索

我的标签

8266(88)
MQTT(50)
GPRS(33)
SDK(29)
Air202(28)
云服务器(21)
ESP8266(21)
Lua(18)
小程序(17)
STM32(16)
更多

随笔分类

Air724UG学习开发(2)
Android(22)
Android 开发(8)
C# 开发(4)
CH395Q学习开发(17)
CH579M学习开发(7)
ESP32学习开发(15)
ESP8266 AT指令开发(基于STC89C52单片机)(3)
ESP8266 AT指令开发(基于STM32)(1)
ESP8266 AT指令开发基础入门篇备份(12)
ESP8266 LUA脚本语言开发(13)

ESP8266 SDK开发: 开发环境搭建(RTOS 2.2.0)(建议只参考这篇文章搭建即可,教程以NONOS版本为主!)

ESP8266:SDK开发(源码见资料源码)

开发板购买链接:[开发板购买链接](#)

资料源码:<https://github.com/yangfengwu45/learn-esp8266-sdk.git>

开发软

件:https://mnifdv.cn/resource/cnblogs/Learn8266ForSDK/AiThinkerIDE_V0.5

点击加入群聊【ESP8266开发交流群】：

- 基础开源教程:ESP8266:LUA脚本开发
 - 基础开源教程:ESP8266 AT指令开发(基于51单片机)
 - 基础开源教程:Android学习开发
 - 基础开源教程:C#学习开发
 - 基础开源教程:微信小程序开发入门篇
- 需要搭配的Android, C#等基础教程如上, 各个教程正在整理。

- 1.01-准备工作-硬件说明
- 1.02-整体运行测试-APP使用SmartConfig配网绑定ESP8266,并通过MQTT远程通信控制,采集DHT11温湿度数据
- 2.01 开发环境搭建(RTOS 2.2.0)(建议只参考这篇文章搭建即可,教程以NONOS版本为主!)
- 2.01 开发环境搭建(NONOS 2.2.0)
- 2.02-外设篇-GPIO输出高低电平
- 2.03-外设篇-GPIO输入检测
- 2.04-外设篇-GPIO中断检测
- 2.05-外设篇-定时器,延时
- 2.05-外设篇-系统任务(消息队列,通知)
- 2.06-外设篇-串口
- 2.07-外设篇-PWM,呼吸灯(RTOS 2.2.0)
- 2.08-外设篇-SPI(RTOS 2.2.0)
- 2.09-外设篇-温湿度传感器-DHT11
- 2.11-外设篇-时钟芯片DS1302使用和拓展知识time.h的使用
- 2.12-外设篇-内存分布说明及Flash读写
- 3.02-网络篇-8266TCP服务器(espconn实现).(NONOS 2.2.0)
- 3.03-网络篇-8266连接路由器
- 3.04-网络篇-TCP客户端(espconn).(NONOS 2.2.0)
- 3.10-网络篇-UDP通信 - 微信小程序篇-微信小程序通过UDP实现和ESP8266局域网通信控制
- 4.01-自建MQTT服务器篇-安装MQTT服务器,ESP8266连接MQTT服务器实现通信控制
- 4.02-自建MQTT服务器篇-ESP8266配网 SmartConfig
- 4.03-自建MQTT服务器篇-APP使用SmartConfig配网绑定ESP8266,并通过MQTT远程通信控制
- 4.05-自建MQTT服务器篇-编写微信小程序连接MQTT服务器程序
- 4.10 阿里云物联网平台篇-测试MOTT调试助手和ESP8266连接阿里云物联网平

ESP8266 LUA开发基础入门篇
备份(22)
ESP8266 SDK开发(33)
ESP8266 SDK开发基础入门篇
备份(30)
GPRS Air202 LUA开发(11)
HC32F460(华大) +
BC260Y(NB-IOT) 物联网开发
(5)
NB-IOT Air302 AT指令和LUA
脚本语言开发(25)
PLC(三菱PLC)基础入门篇(2)
STM32+Air724UG(4G模组)
物联网开发(43)
STM32+BC26/260Y物联网开
发(37)
STM32+CH395Q(以太网)物
联网开发(21)
STM32+ESP8266(ZLESP8266/
物联网开发(1)
STM32+ESP8266+AIR202/30:
远程升级方案(16)
STM32+ESP8266+AIR202/30:
终端管理方案(6)
STM32+ESP8266+Air302物
联网开发(64)
STM32+W5500+AIR202/302
基本控制方案(25)
STM32+W5500+AIR202/302
远程升级方案(6)
UCOSii操作系统(1)
W5500 学习开发(8)
编程语言C#(11)
编程语言Lua脚本语言基础入
门篇(6)
编程语言Python(1)
单片机(LPC1778)LPC1778(2)
单片机(MSP430)开发基础入门
篇(4)
单片机(STC89C51)单片机开发
板学习入门篇(3)
单片机(STM32)基础入门篇(3)
单片机(STM32)综合应用系列
(16)
电路模块使用说明(11)
感想(6)
软件安装使用: MQTT(8)
更多

最新评论

1. Re:(一)Lua脚本语言入门
楼主可以分享一下这本电子书吗?
--戴思
2. Re:学习C语言-学习指针
学到了学到了, 很清晰的思
路, 给博主赞赞赞
--*夏日么么茶

阅读排行榜

1. ESP8266使用详解(AT,LUA,
SDK)(172847)
2. 1-安装MQTT服务器(Windo
ws),并连接测试(99168)
3. ESP8266刷AT固件与node
mcu固件(64823)
4. 用ESP8266+android,制作
自己的WIFI小车(ESP8266篇)
(64354)
5. 有人WIFI模块使用详解(385
49)

- 台
- [4.11-阿里云物联网平台篇-ESP8266连接阿里云物联网平台使用自定义Topic实现自定义数据的上报和数据下发](#)
- [4.12-阿里云物联网平台篇-ESP8266连接阿里云物联网平台使用物理模型Topic实现温湿度数据显示](#)
- [4.13-阿里云物联网平台篇-阿里云物联网平台加入规则引擎\(云产品流转\),让MQTT设备之间实现通信](#)
- [4.14-阿里云物联网平台篇-Android和ESP8266连接阿里云物联网平台,并通过云平台实现远程温湿度采集和继电器控制](#)
-
-
- [6.01-综合实战篇-C#上位机串口通信控制ESP8266\(RTOS 2.2.0\)](#)
- [6.02-综合实战篇-8266TCP服务器\(LWIP,RAW模式,PCB控制块实现\)\(RTOS 2.2.0\)与C#TCP客户端实现无线网络通信控制](#)
- [6.03-综合实战篇-8266TCP服务器\(espconn实现\)\(NONOS 2.2.0\)与Android TCP客户端实现无线网络通信控制](#)
-
- [9.01-常见问题及程序BUG修复](#)

前言

这节安装下编程软件,

建议安装我提供的,有问题好解决!

- 6. (一)基于阿里云的MQTT远程控制(Android 连接MQTT服务器,ESP8266连接MQTT服务器实现远程通信控制----简单的连接通信)(35982)
- 7. 关于TCP和MQTT之间的转换(33340)
- 8. C#中public与private与static(32611)
- 9. android 之TCP客户端编程(31968)
- 10. android客服端+eps8266+单片机+路由器之远程控制系统(31338)

推荐排行榜

- 1. C#委托+回调详解(9)
- 2. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(8)
- 3. 用ESP8266+android,制作自己的WIFI小车(Android 软件)(6)
- 4. ESP8266使用详解(AT,LUA,SDK)(6)
- 5. 关于TCP和MQTT之间的转换(5)

8266ForSDK

查看

此电脑 > 学习开发 (H:) > Learn8266ForSDK >

名称	修改日期
1.1 Test	2019/11/3
2.1 准备工作-开发环境搭建	2019/11/3
SDK开发软件	2019/11/3
串口驱动CH340	2019/11/3
开发板硬件资料	2019/11/3
烧录固件软件	2019/11/3
调试助手	2019/11/3
资料文档	2019/11/3
README.en.md	2019/11/3
README.md	2019/11/3

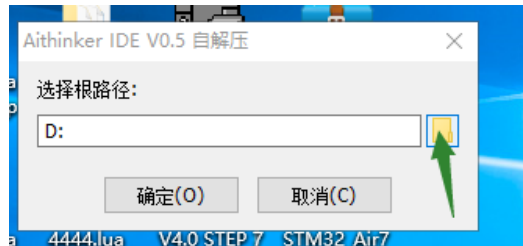
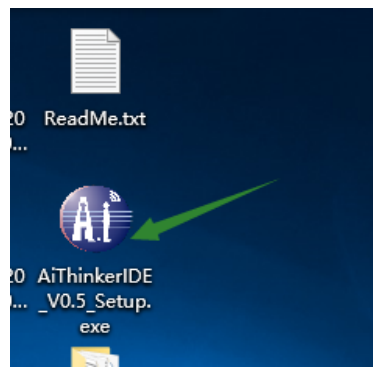
件

查看

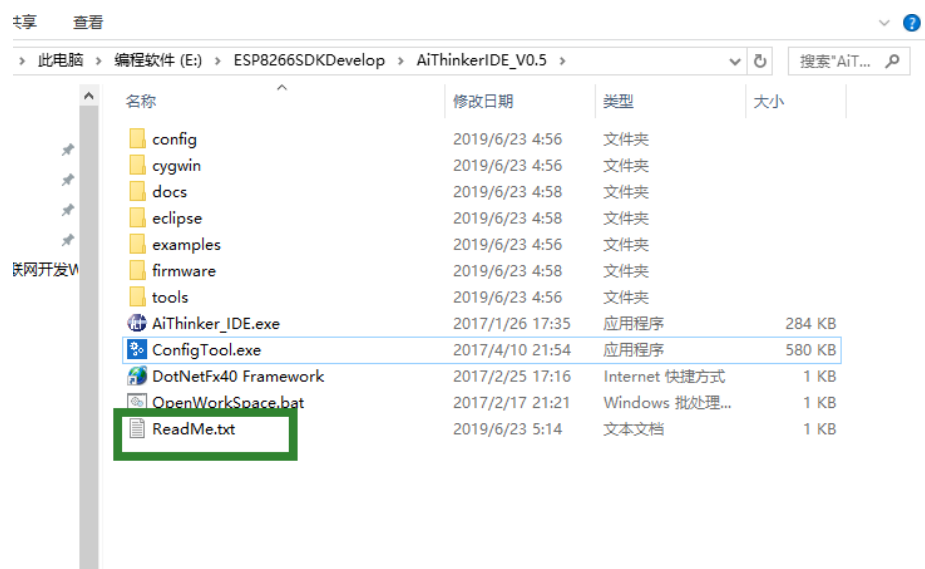
此电脑 > 学习开发 (H:) > Learn8266ForSDK > SDK开发软件

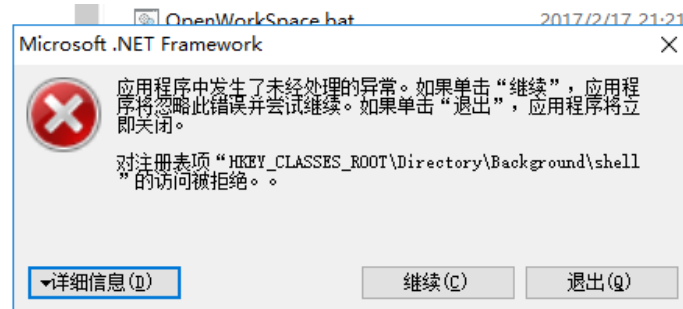
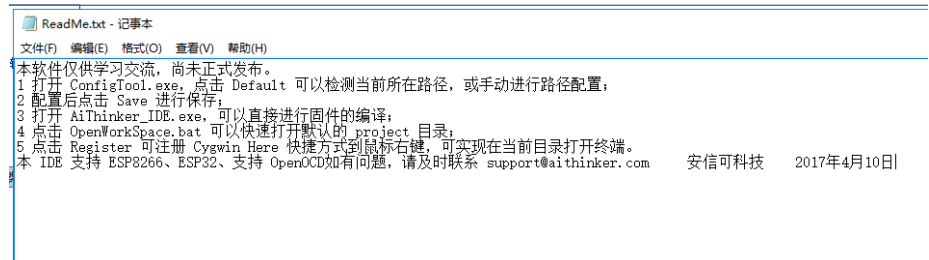
名称	修改日期
AiThinkerIDE_V0.5_Setup.exe	2019/6/23 4:13

开始安装

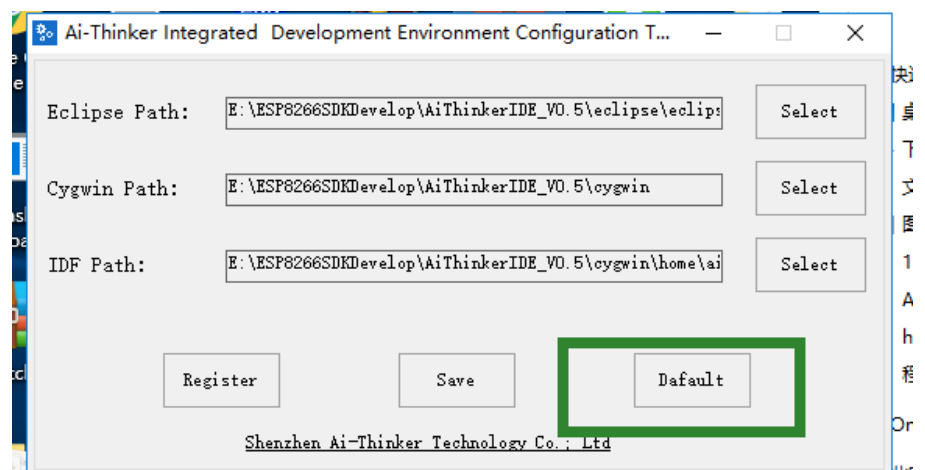
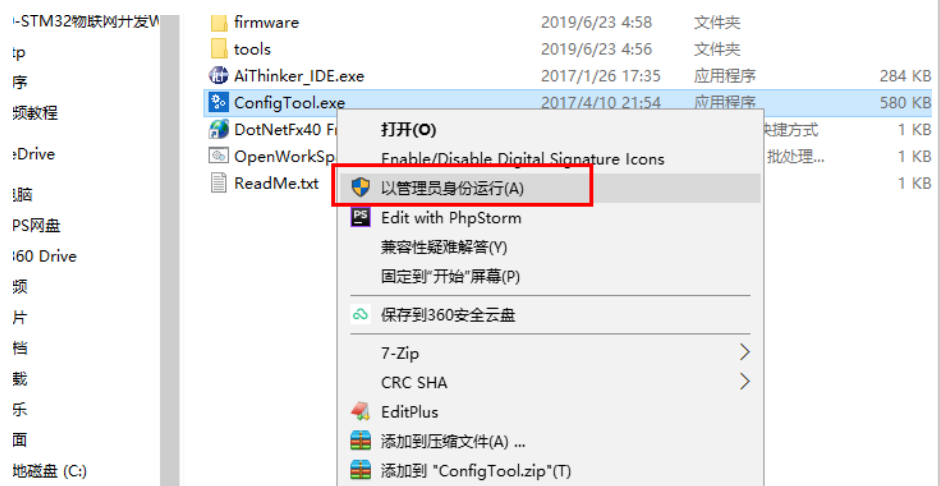


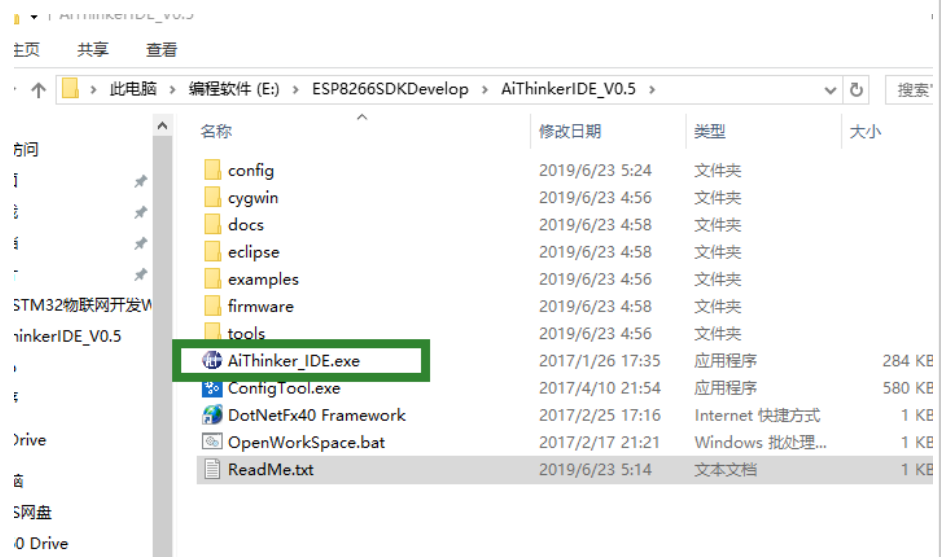
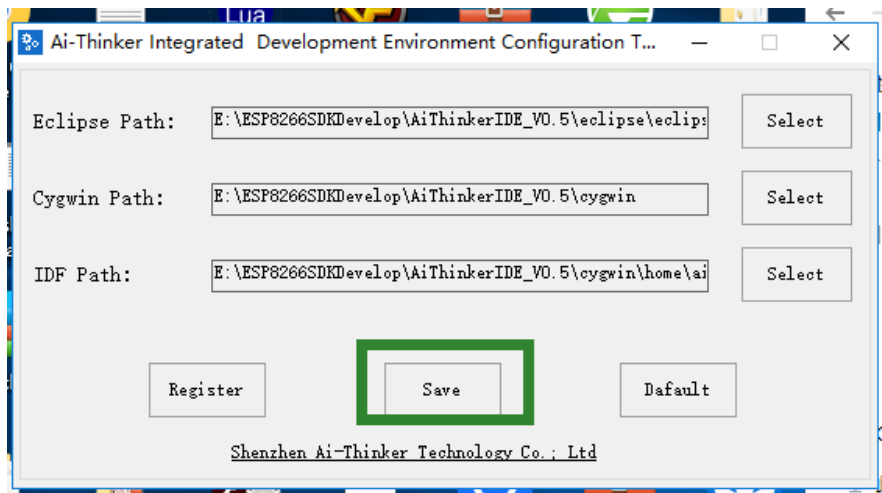
自己随意哈



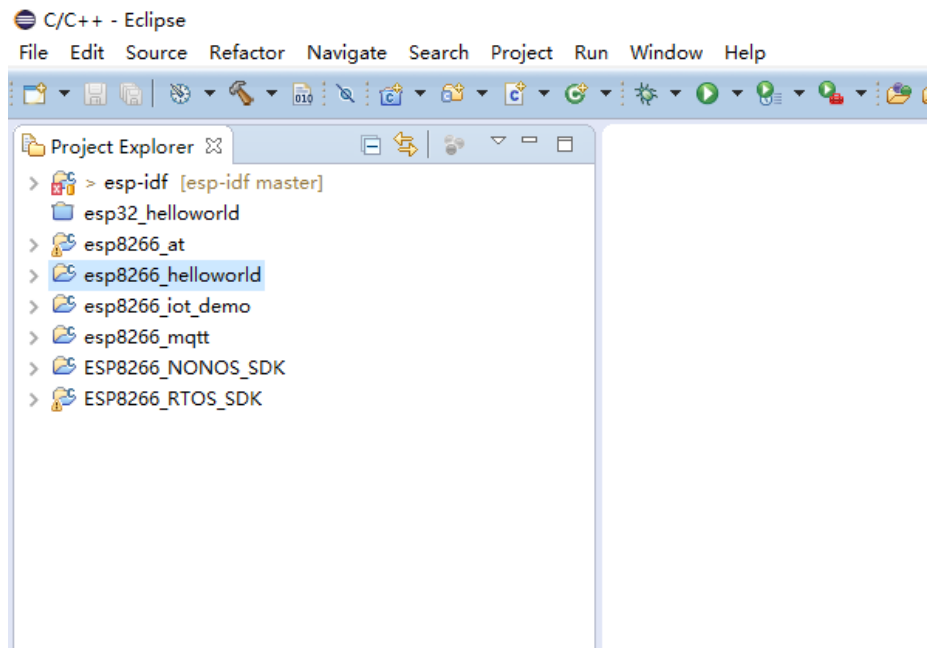


关掉程序

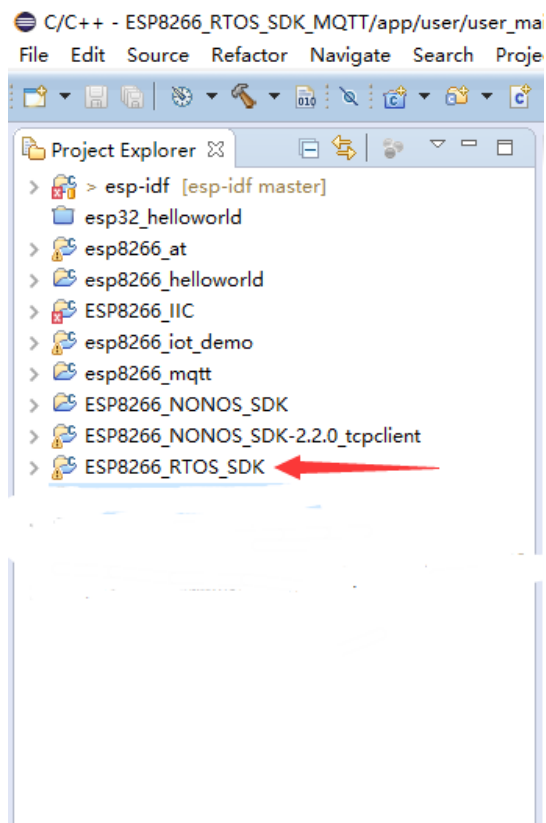




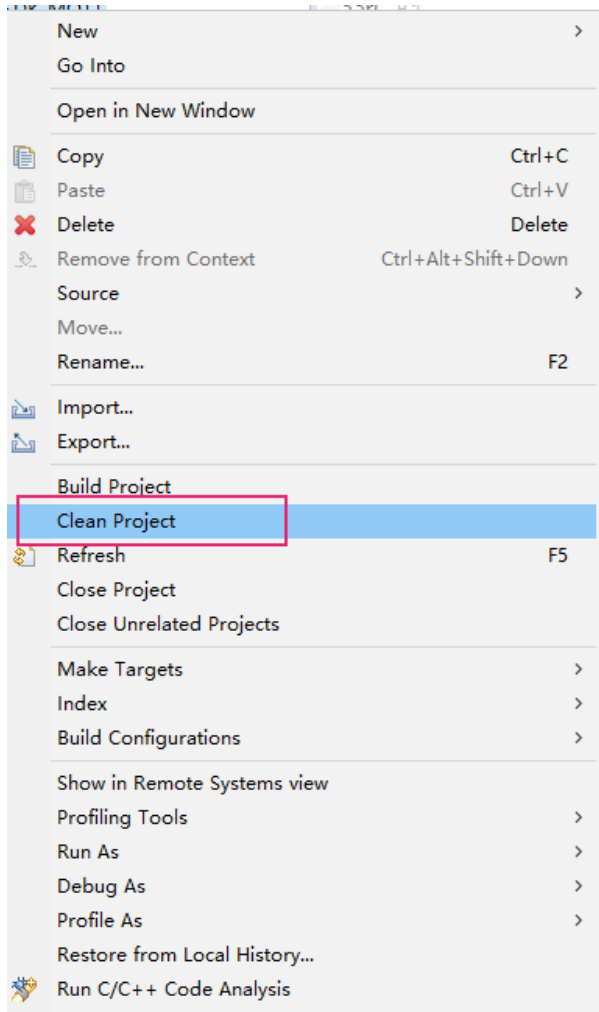
.....官方啥都给准备好了.

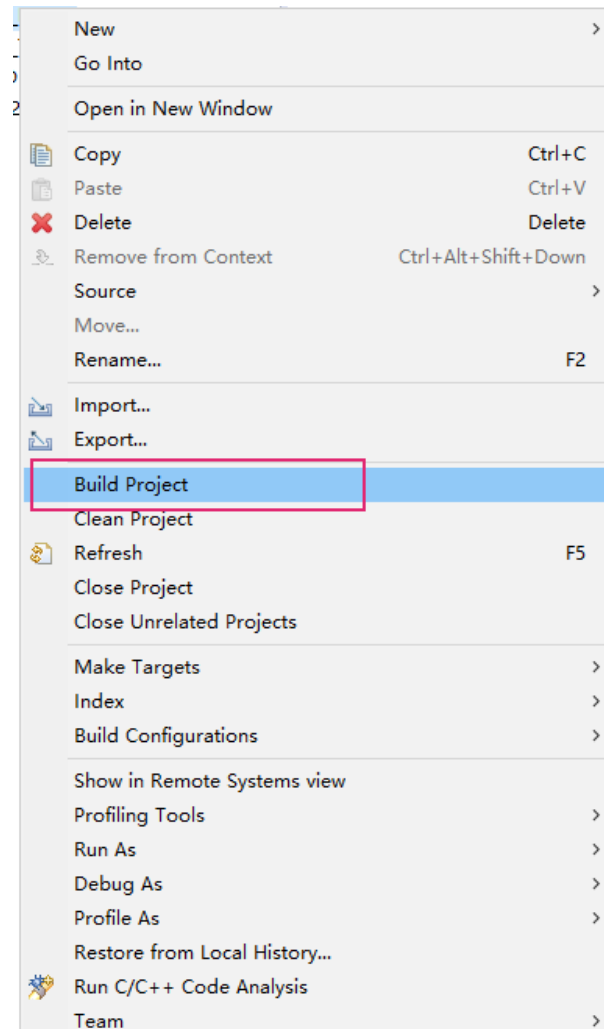


如果使用RTOS版本开发,可以使用下面这个



```
Console Tasks
CDT Build Console [esp8266_helloworld]
05:30:15 ***** Clean-only build of configuration Default for project esp8266_helloworld *****
make clean
05:30:15 Build Finished (took 324ms)
```





 Tasks  Search

CDT Build Console [ESP8266_RTOS_SDK_MQTT]

```
xtensa-lx106-elf-gcc -Os -g -Wpointer-arith -Wundef -Werror -WL,-EL -fno-inline-func:
xtensa-lx106-elf-gcc -Os -g -Wpointer-arith -Wundef -Werror -WL,-EL -fno-inline-func:
xtensa-lx106-elf-gcc -Os -g -Wpointer-arith -Wundef -Werror -WL,-EL -fno-inline-func:
xtensa-lx106-elf-gcc -Os -g -Wpointer-arith -Wundef -Werror -WL,-EL -fno-inline-func:
xtensa-lx106-elf-gcc -Os -g -Wpointer-arith -Wundef -Werror -WL,-EL -fno-inline-func:
xtensa-lx106-elf-gcc -Os -g -Wpointer-arith -Wundef -Werror -WL,-EL -fno-inline-func:
xtensa-lx106-elf-gcc -Os -g -Wpointer-arith -Wundef -Werror -WL,-EL -fno-inline-func:
xtensa-lx106-elf-gcc -Os -g -Wpointer-arith -Wundef -Werror -WL,-EL -fno-inline-func:
xtensa-lx106-elf-ar ru .output/eagle/debug/lib/libmqtt.a .output/eagle/debug/obj/MQTT
xtensa-lx106-elf-ar: creating .output/eagle/debug/lib/libmqtt.a
make[2]: Leaving directory '/home/aithinker/project/ESP8266_RTOS_SDK_MQTT/app/MQTT'
xtensa-lx106-elf-gcc -L/home/aithinker/project/ESP8266_RTOS_SDK/lib -WL,--gc-sections
```

```
!!!
SDK_PATH: /home/aithinker/project/ESP8266_RTOS_SDK
BIN_PATH: /home/aithinker/project/ESP8266_RTOS_SDK/bin/upgrade
```

默认bin文件 编译后的位置

```
bin crc: 4009f5f0
Support boot_v1.2 and +
Generate user1.1024.new.2.bin successfully in BIN_PATH
boot.bin----->0x00000
user1.1024.new.2.bin--->0x01000
!!!
make[1]: Leaving directory '/home/aithinker/project/ESP8266_RTOS_SDK_MQTT/app'
01:51:37 Build Finished (took 51s.632ms)
```

提示!

提示!

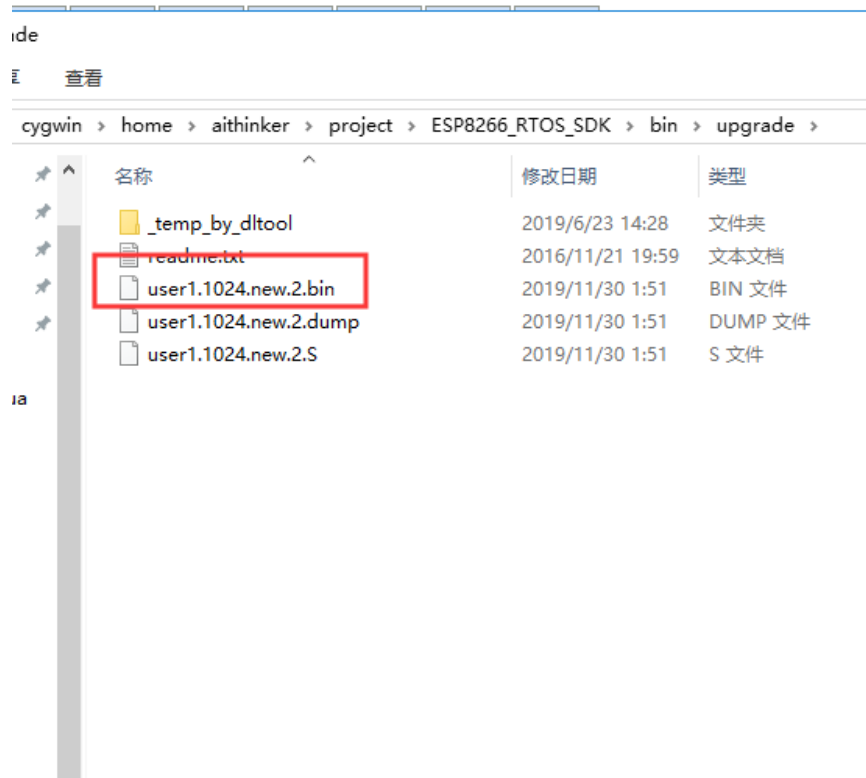
提示!

快速的打开工程目录

双击 OpenWorkSpace.bat

管理				
工程代码 (F:) > AiThinkerIDE_V0.5				
名称	修改日期	类型	大小	
config	2020/3/22 1:01	文件夹		
cygwin	2020/3/22 0:59	文件夹		
docs	2020/3/22 1:00	文件夹		
eclipse	2020/3/22 1:02	文件夹		
examples	2020/3/22 0:59	文件夹		
firmware	2020/3/22 1:00	文件夹		
tools	2020/3/22 0:59	文件夹		
AiThinker_IDE.exe	2017/1/26 17:35	应用程序		
ConfigTool.exe	2017/4/10 21:54	应用程序		
DotNetFx40 Framework	2017/2/25 17:16	Internet 快捷方式		
OpenWorkspace.bat	2017/2/17 21:21	Windows 批处理...		
ReadMe.txt	2017/4/10 21:58	文本文档		

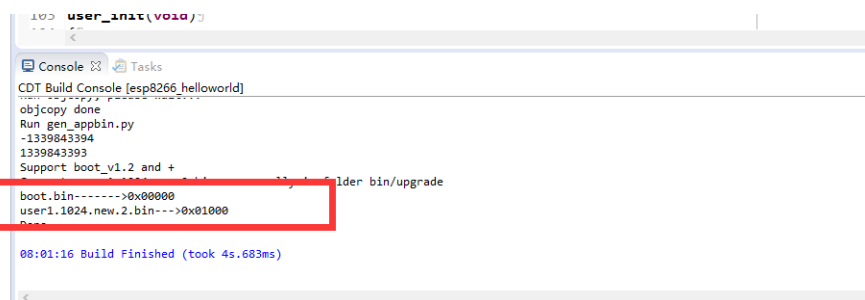
分享 查看				
<< AiThinkerIDE_V0.5 > cygwin > home > aithinker > project > ESP8266_RTOS_SDK > bin >				
名称	修改日期	类型	大小	
_temp_by_dlttool	2019/6/23 14:28	文件夹		
upgrade	2019/11/30 1:51	文件夹		
blank.bin	2016/11/21 19:59	BIN 文件	4 KB	
boot_v1.6.bin	2016/11/21 19:59	BIN 文件	4 KB	
esp_init_data_default.bin	2016/11/21 19:59	BIN 文件	1 KB	



咱就是刷这四个程序文件



现在刷程序

软件给了其中两个文件刷的位置



我用的WI-Fi是ESP8266-12F

https://wiki.ai-thinker.com/_media/esp8266/esp8266_module_list.png

ESP-12F		SMD	24.0*16.0	4	32Mbit	-	-20~80℃	PCB天线	3.0~3.6V 建议 3.3V	GPIO2	115200	22	9	-	-	-	-	-	-
ESP-12S		SMD	24.0*16.0	4	32Mbit	FCO/CE	-20~80℃	PCB天线	3.0~3.6V 建议 3.3V	GPIO2	115200	16	9	上拉	-	下拉	-	上拉	上拉

waiting for download

ESP8266 Flash 地址配置

烧录时的配置选项与编译时的配置是相关的，编译SDK时，控制台会提示bin文件的烧录地址，配置不同，烧录的地址和需要的文件也不同

我们按照烧录文件的不同分为两种情况：支持云端升级 和 不支持云端升级

另外，根据flash容量的不同，我们还要调整bin文件烧录的地址

支持云端升级的固件，在 flash 中会分为 2 个区，一个用来运行程序，一个用来保存升级固件，当运行 user1 时升级，程序会下载固件到 user2，下载完毕后，下次启动时从 use 实现云端升级。

user2 仅用作升级使用，用户烧录时无需下载 user2.bin，此处仅为说明 user2 的存放位置

注意1: 乐鑫在不同版本的 SDK 中有可能会改变这些烧录位置，以下说明仅为参考，建议以开发时的 Console 输出信息为准。

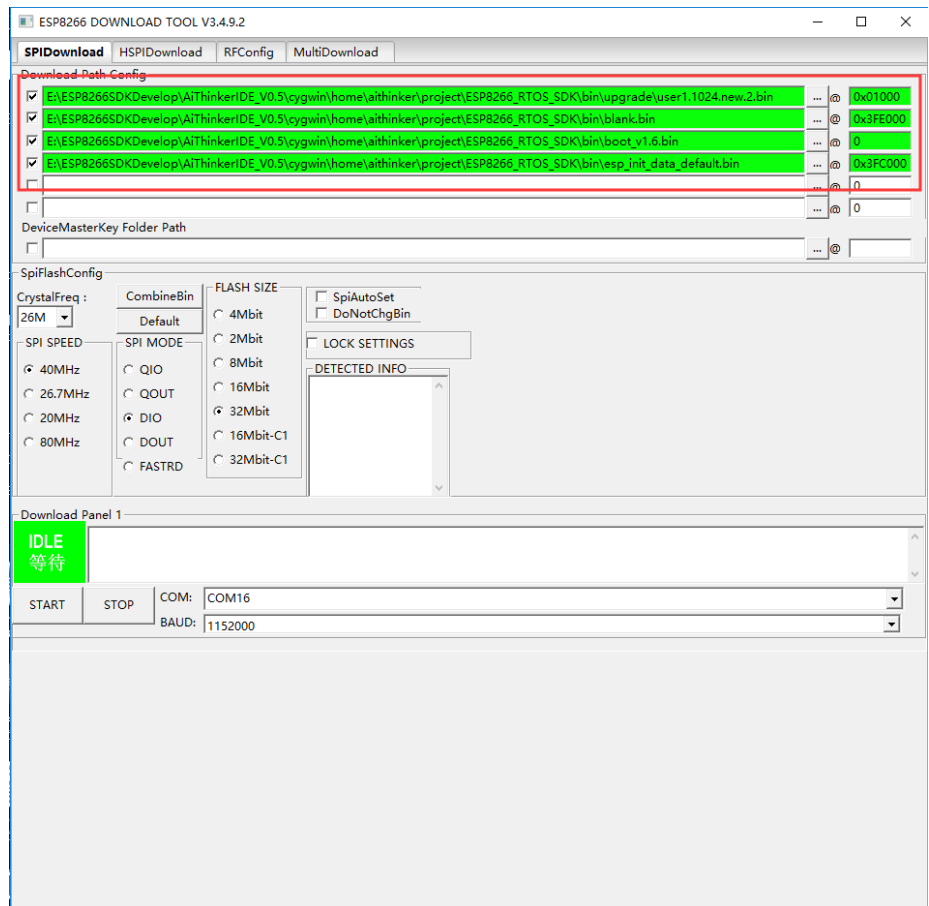
注意2: 合并后的固件内已经包含了地址信息，只需烧写到 0x0 地址即可。

不支持云端升级（NoBoot 模式）

文件名称	8Mbit 地址分配	16Mbit 地址分配	32Mbit 地址分配	备注
eagle.flash.bin	0x00000	0x00000	0x00000	主程序，由代码编译生成
eagle.irom0text.bin	0x10000	0x10000	0x10000	主程序，由代码编译生成
esp_init_data_default.bin	0xFC000	0xFC000	0xFC000	由乐鑫在 SDK 中提供
blank.bin	0xFE000	0xFE000	0xFE000	由乐鑫在 SDK 中提供

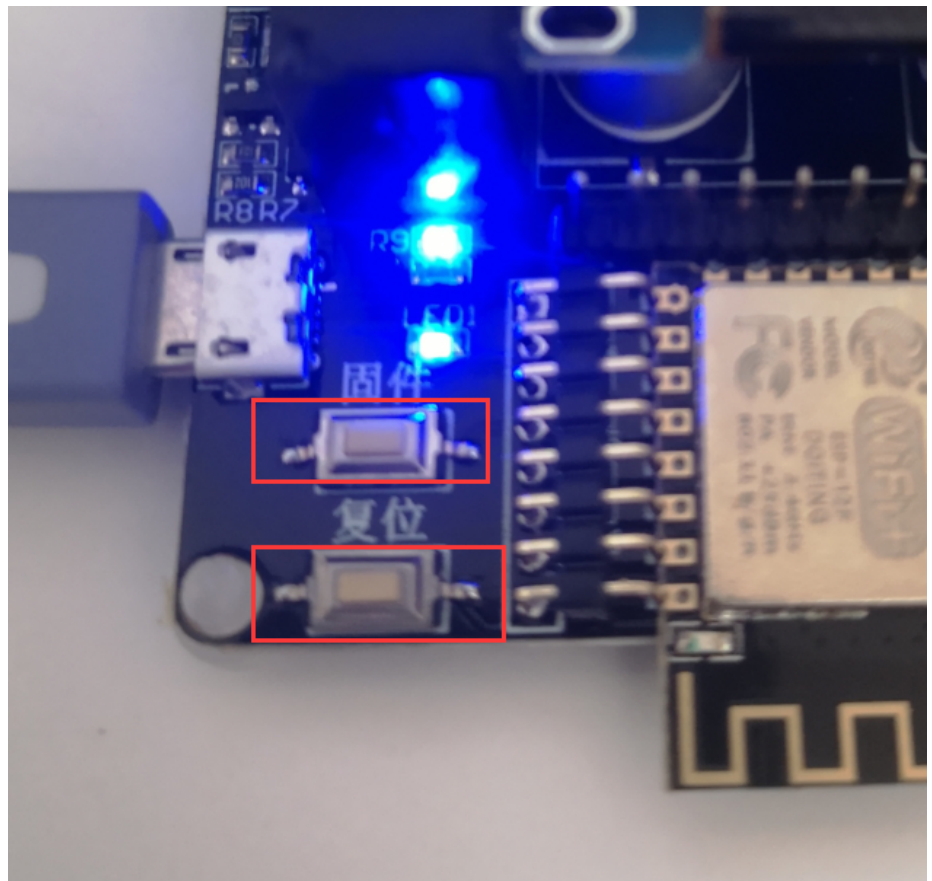
支持云端升级（Boot 模式）

文件名称	8Mbit 地址分配	16Mbit 地址分配	32Mbit 地址分配	备注
boot.bin	0x00000	0x00000	0x00000	由乐鑫在 SDK 中提供，建议一直使用最新版本
user1.bin	0x01000	0x01000	0x01000	主程序，由代码编译生成
user2.bin	0x81000	0x81000	0x81000	主程序，由代码编译生成
esp_init_data_default.bin	0xFC000	0xFC000	0xFC000	由乐鑫在 SDK 中提供
blank.bin	0xFE000	0xFE000	0xFE000	由乐鑫在 SDK 中提供



按下复位按钮 → 按下固件按钮 → 松开复位按钮 → 松开固件按钮

按照上述步骤,ESP8266进入等待刷固件状态(刷程序状态)

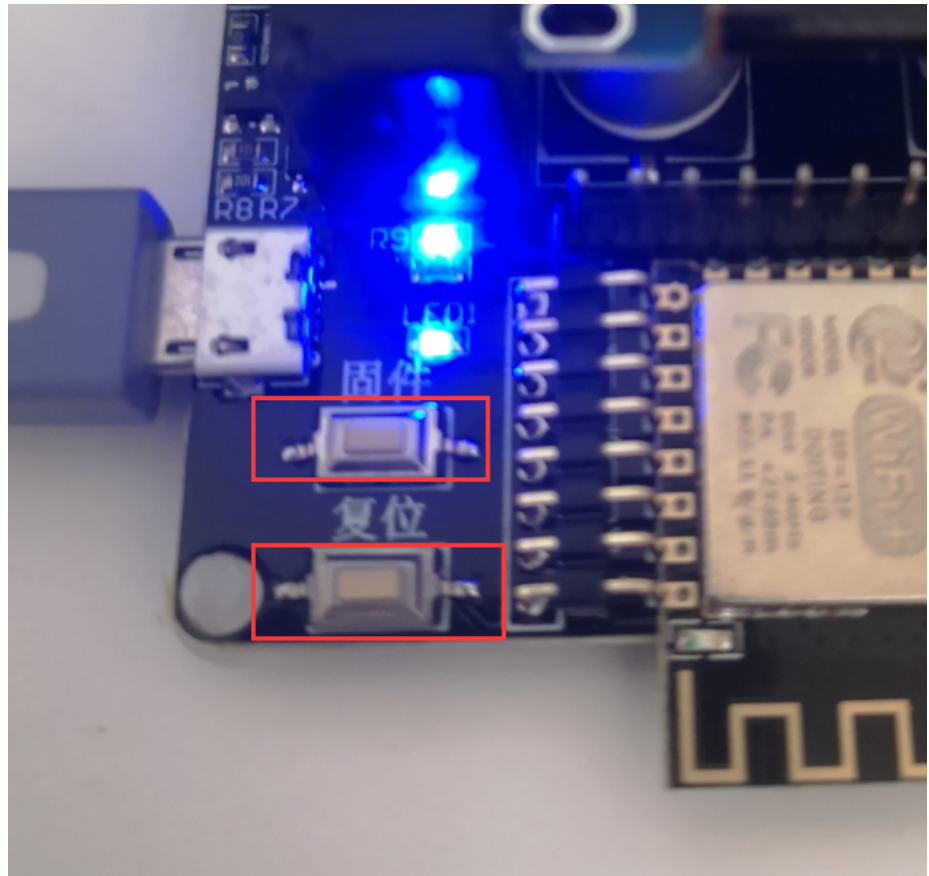


点击软件 START开始刷固件

<input type="radio"/> 26.7MHz <input type="radio"/> 20MHz <input type="radio"/> 80MHz	<input type="radio"/> QOUT <input checked="" type="radio"/> DIO <input type="radio"/> DOUT <input type="radio"/> FASTRD	<input type="radio"/> 16Mbit <input checked="" type="radio"/> 32Mbit <input type="radio"/> 16Mbit-C1 <input type="radio"/> 32Mbit-C1	flash vendor: 20h : N/A flash devID: 4016h QUAD;32Mbit crystal: 26 Mhz
Download Panel 1			
Download 下载中...	AP: DE-4F-22-10-B4-C3 STA: DC-4F-22-10-B4-C3		
START	STOP	COM: COM16	BAUD: 115200

刷入完成,复位下

CrystalFreq : 26M	CombineBin Default	FLASH SIZE <input type="radio"/> 4Mbit <input type="radio"/> 2Mbit <input type="radio"/> 8Mbit <input checked="" type="radio"/> 16Mbit <input checked="" type="radio"/> 32Mbit <input type="radio"/> 16Mbit-C1 <input type="radio"/> 32Mbit-C1	<input type="checkbox"/> SpiAutoSet <input type="checkbox"/> DoNotChgBin <input type="checkbox"/> LOCK SETTINGS DETECTED INFO flash vendor: 20h : N/A flash devID: 4016h QUAD;32Mbit crystal: 26 Mhz
Download Panel 1			
FINISH 完成	AP: DE-4F-22-10-B4-C3 STA: DC-4F-22-10-B4-C3		
START	STOP	COM: COM16	BAUD: 115200



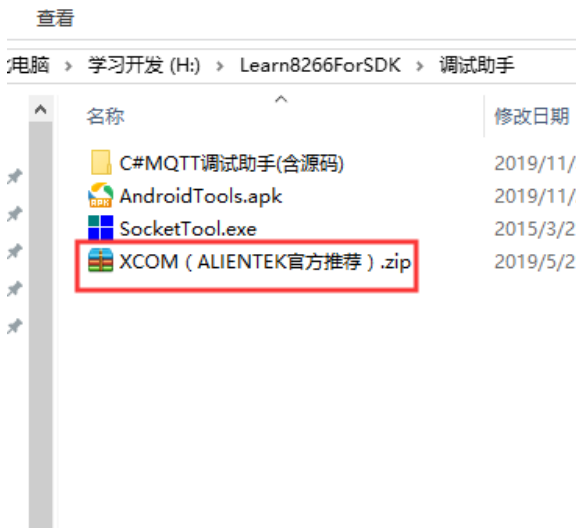
可以打开串口调试助手

5ForSDK

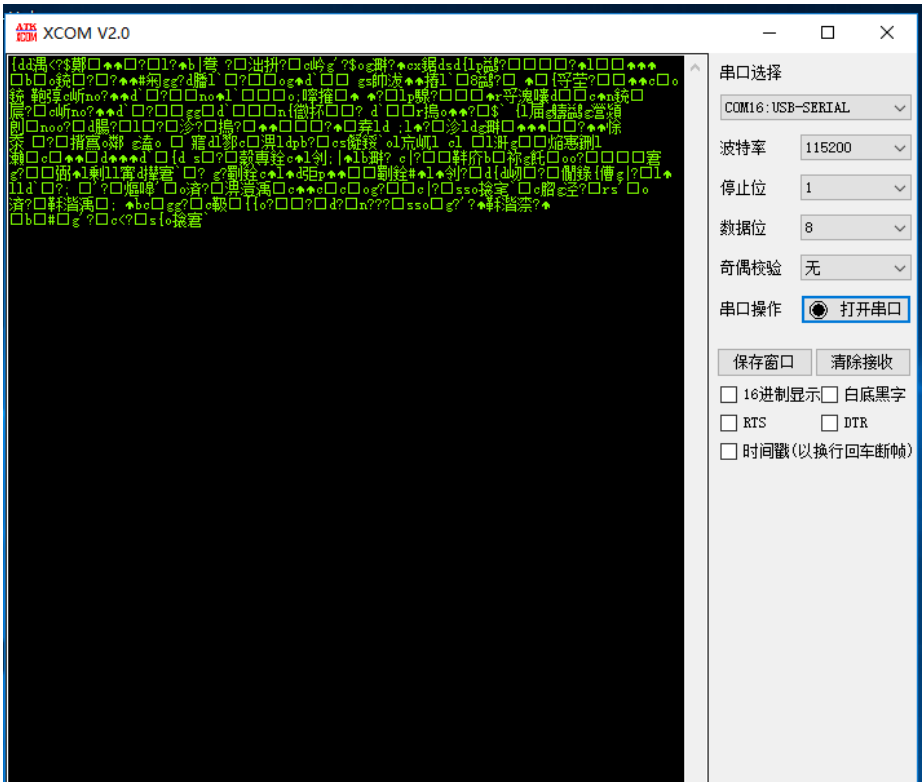
查看

电脑 > 学习开发 (H:) > Learn8266ForSDK

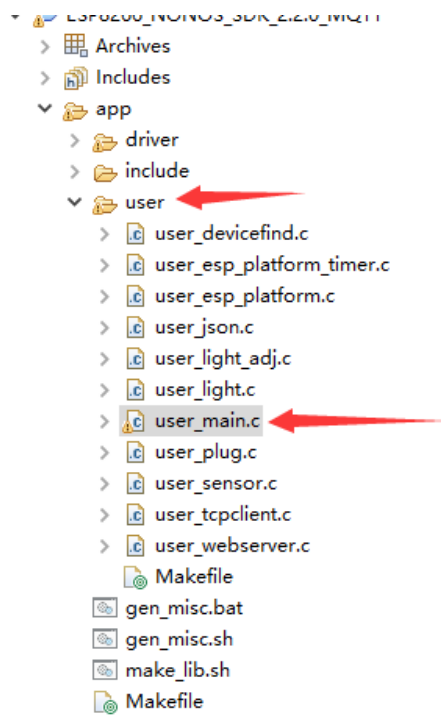
名称	修
1.1 Test	20'
2.1 准备工作-开发环境搭建	20'
SDK开发软件	20'
串口驱动CH340	20'
开发板硬件资料	20'
烧录固件软件	20'
调试助手	20'
资料文档	20'
README.en.md	20'
README.md	20'



查看打印的数据



提示



进入以下函数

```
4
5 /*****
6  *.FunctionName::user_init
7  *.Description::entry of user
8  *.Parameters::none
9  *.Returns::none
10 *****/
11 void user_init(void)
12 {
13     uart_init_new();
14     printf("SDK version:%s\n", s
15     .
16     printf("Ai-Thinker Technolog
17
18
```

为后期统一,请修改为115200波特率

```

7  ....}
3  }
3  #endif
3  @void
1  uart_init_new(void)
2  {
3  ....UART_WaitTxFifoEmpty(UART0);
4  ....UART_WaitTxFifoEmpty(UART1);
5  #endif
6  ....UART_ConfigTypeDef uart_config;
7  ....uart_config.baud_rate = BIT_RATE_74880;
8  ....uart_config.data_bits = UART_WordLength_8b;
9  ....uart_config.parity = USART_Parity_None;
10 ....uart_config.stop_bits = USART_StopBits_1;
11 ....uart_config.flow_ctrl = USART_HardwareFlowControl_None;
12 ....uart_config.UART_RxFifoThresh = 120;
13 ....uart_config.UART_InverseMask = UART_None_Inverse;
14 ....UART_ParamConfig(UART0, &uart_config);
15 #endif
16 ....UART_IntrConfTypeDef uart_intr;
17 ....uart_intr.UART_IntrEnMask = UART_RXFIFO_TOUT_INT_ENA | UART_FRM_ERR_INT_ENA | UAF
18 ....uart_intr.UART_RX_FifoFullIntrThresh = 10;
19 ....uart_intr.UART_RX_TimeOutIntrThresh = 2;
20 ....uart_intr.UART_TX_FifoEmptyIntrThresh = 20;
21 ....UART_IntrConfig(UART0, &uart_intr);
22 #endif
23 ....UART_SetPrintPort(UART0);
24 ....UART_intr_handler_register(uart0_rx_intr_handler, NULL);
25 ....ETS_UART_INTR_ENABLE();
26 #endif
27 @..../*
28 ....UART_SetWordLength(UART0, UART_WordLength_8b);
29 ....UART_SetStopBits(UART0, USART_StopBits_1);

```

修改后

```

8  }
9  #endif
10 @void
1  uart_init_new(void)
2  {
3  ....UART_WaitTxFifoEmpty(UART0);
4  ....UART_WaitTxFifoEmpty(UART1);
5  #endif
6  ....UART_ConfigTypeDef uart_config;
7  ....uart_config.baud_rate = BIT_RATE_115200;
8  ....uart_config.data_bits = UART_WordLength_8b;
9  ....uart_config.parity = USART_Parity_None;
10 ....uart_config.stop_bits = USART_StopBits_1;
11 ....uart_config.flow_ctrl = USART_HardwareFlowControl_None;
12 ....uart_config.UART_RxFifoThresh = 120;
13 ....uart_config.UART_InverseMask = UART_None_Inverse;
14 ....UART_ParamConfig(UART0, &uart_config);
15 #endif
16 ....UART_IntrConfTypeDef uart_intr;
17 ....uart_intr.UART_IntrEnMask = UART_RXFIFO_TOUT_INT_ENA | UART_FRM_ERR_IN
18 ....uart_intr.UART_RX_FifoFullIntrThresh = 10;
19 ....uart_intr.UART_RX_TimeOutIntrThresh = 2;
20 ....uart_intr.UART_TX_FifoEmptyIntrThresh = 20;
21 ....UART_IntrConfig(UART0, &uart_intr);
22 #endif
23 ....UART_SetPrintPort(UART0);
24 ....UART_intr_handler_register(uart0_rx_intr_handler, NULL);
25 ....ETS_UART_INTR_ENABLE();
26 #endif
27 @..../*
28 ....UART_SetWordLength(UART0, UART_WordLength_8b);

```

建议使用下一节NONOS版本,后期教程以NONOS版本为主!

分类: [ESP8266 SDK开发](#)

好文要顶

关注我

收藏该文



杨奉武

关注 - 1

粉丝 - 637

0

0

« 上一篇: [ESP8266 SDK开发-1.02-整体运行测试 -APP使用SmartConfig配网绑定ESP8266,并通过MQTT远程通信控制,采集DHT11温湿度数据](#)

» 下一篇: [01-STM32+W5500+AIR202基本控制篇-硬件使用说明](#)

posted on 2019-11-30 02:08 杨奉武 阅读(2879) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

发表评论

编辑 预览

B

支持 Markdown

自动补全

提交评论 退出

[Ctrl+Enter快捷键提交]

【推荐】百度智能云2021普惠上云节：新用户首购云服务器低至0.7折

【推荐】阿里云云大使特惠：新用户购ECS服务器1核2G最低价87元/年

【推荐】大型组态、工控、仿真、CAD\GIS 50万行VC++源码免费下载!

编辑推荐：

· [C# 10 完整特性介绍](#)

· [不是技术也能看懂云原生](#)

- 记一次接口慢查排查
- 一个故事看懂HTTPS
- 人人都能看懂系列：分布式系统改造方案之数据篇



最新新闻：

- 你还抢购华为吗？门店可能没有存货了
- 字节新消费版图大起底：投资自营双管齐下
- 上市破发、资金受困，理想“勇争第一”空成口号
- 百度二季度财报点评：以更高维的ESG识别其价值
- 锂电专利战争：欧美、日韩围剿，中国换道超车
- » 更多新闻...

Powered by:

博客园

Copyright © 2021 杨奉武

Powered by .NET 5.0 on Kubernetes



单片机,物联网,上位机,...

扫一扫二维码，入群聊。