

第一章 ES8266 WIFI 模块开发环境搭建

选择安信可的 AiThinker_IDE 的原因是，它不需要搭建 linux 和虚拟机，非常方便。解压后在 windows 环境下直接就能用。现在支持 XP，win7 了。

第一步，安装安信可一体化开发环境

ESP8266 SKD 的开发环境有很多，官方的开发环境配置比较复杂，这里介绍使用安信可出品的集成开发环境 ESP8266 IDE，其实就是 eclipse 集成了相关的开发插件，用起来还是挺方便的。

安信可一体化开发环境有以下特点：

- 支持 ESP8266 NONOS 和 FreeRTOS 环境开发
- 支持 ESP31B/ESP32 FreeRTOS 环境开发
- 下载即用，无需另外配置环境
- 可直接编译所有乐鑫官方推出的 SDK 开发包

下载地址：https://pan.baidu.com/s/1RmgxkCJQ0P6ZijEWDK7_pg
(百度云盘) 下载红框内的最新版本。其它的老版本不用下了。

<input type="checkbox"/>	 最新版本ESP32 GCC v5.2 OpenOCD 支持XP V0.5	-
<input type="checkbox"/>	 支持ESP8266 不支持ESP32 支持XP V0.1	-
<input type="checkbox"/>	 支持ESP8266 ESP31B ESP32 GCC4.8 不支持XP V0.2	-



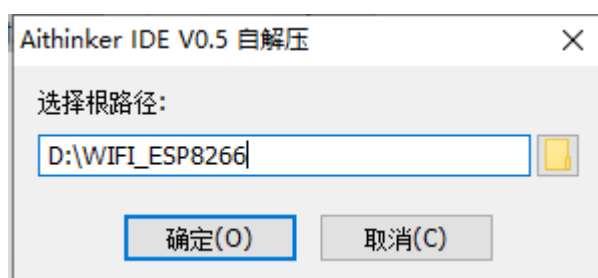
AiThinkerIDE_
V0.5_Setup



ReadMe

解压后：

双击 AiThinkerIDE_V0.5_setup 安装程序，选择一个解压目录，默认是 D 盘。我在 D 盘新建个文件夹 “WIFI_ESP8266” ， **这里要注意文件夹名称不要有中文和空格和小数点，要不后面编译程序时不会成功。**



解压完成后，打开目录

名称	修改日期	类型	大小
config	2018/12/28 12:59	文件夹	
cygwin	2018/12/28 12:59	文件夹	
docs	2018/12/28 12:59	文件夹	
eclipse	2018/12/28 13:00	文件夹	
examples	2018/12/28 12:59	文件夹	
firmware	2018/12/28 13:00	文件夹	
tools	2018/12/28 12:59	文件夹	
AiThinker_IDE	2017/1/26 17:35	应用程序	284 KB
ConfigTool	2017/4/10 21:54	应用程序	580 KB
DotNetFx40 Framework	2017/2/25 17:16	Internet 快捷方式	1 KB
OpenWorkSpace	2017/2/17 21:21	Windows 批处理...	1 KB
ReadMe	2017/4/10 21:58	文本文档	1 KB

下面几个名字了解下，其中安信可的 AiThinker_IDE.exe 将一些烦琐的步骤简化了。

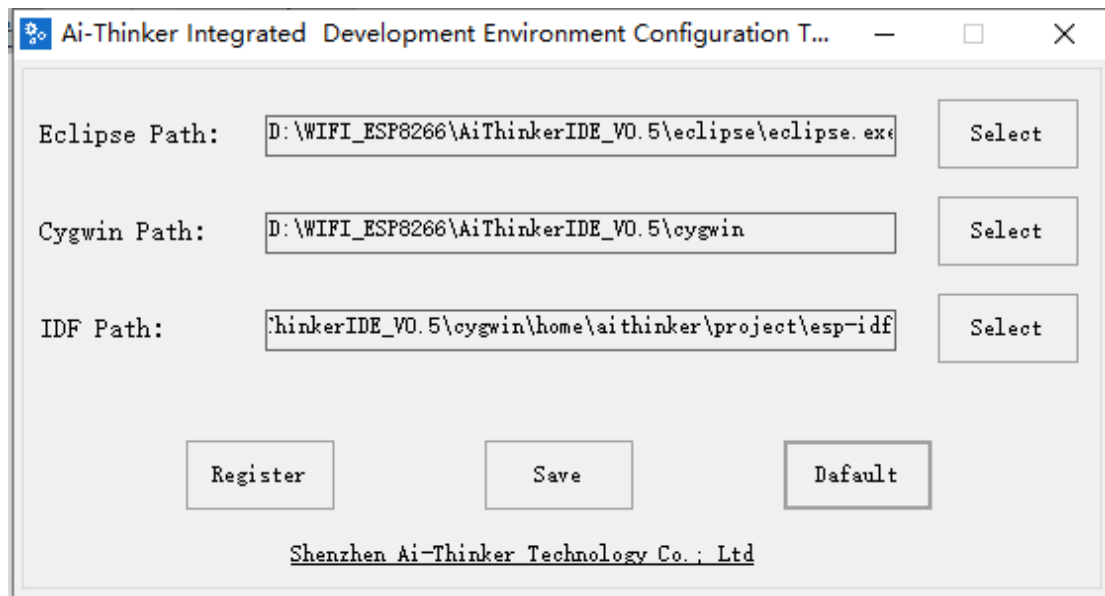
文件名	描述	备注
Cygwin.exe	cygwin和xtensa编译器环境	含ESP8266 和 ESP31B/ESP32 开发环境
Eclipse.exe	eclipse c/c++ 开发工具，仅在官网版本基础上打包了jre运行环境	本机已经安装 eclipse cdt 环境的的无需重复下载
AiThinker_IDE.exe	一体化开发环境配置工具	

开始整合编译环境

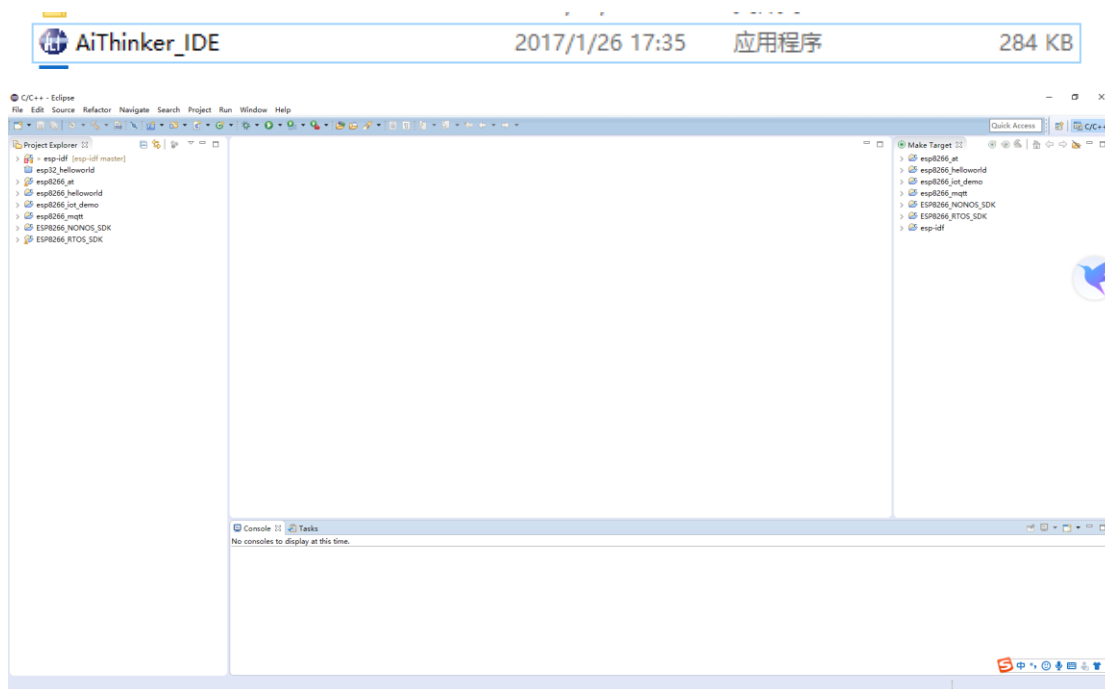
第一步：双击这个

ConfigTool 2017/4/10 21:54 应用程序 580 KB

第二步：先点击“Default”按钮，将用到的程序自动关联起来。然后点击“Save”保存设置。



第三步 2：运行



第二章 使用安信可 ESP 系列一体化开发环境

以导入一个 SDK 为例，按官方的步骤有问题，反正我没成功。我总结了一下，有两种导入方式有些差别：

第一种，当 File->Import 弹出对话框后，选“General”里的“Existing Projects into Workspace”导入。第二种，选“C/C++”里的“Existing Code as Makefile Project”导入。当第一种导入不成功时，可尝试用第二种导入。机智云生成的 SDK 都是用第二种方法导入的。网络上 eclipse 大都用的第一种，为什么这样我没有去深究，我觉得还是先学好怎么用吧。

两种方法的共同点是，所需的项目文档一定要导

AiThinkerIDE_V0.5\cygwin\home\aitthinker\project 文件夹，这个文件夹类似 eclipse 官方资料常提到的 workspace 文件夹，否则编译不成功，不成功，不成功（重要的事说三遍）。

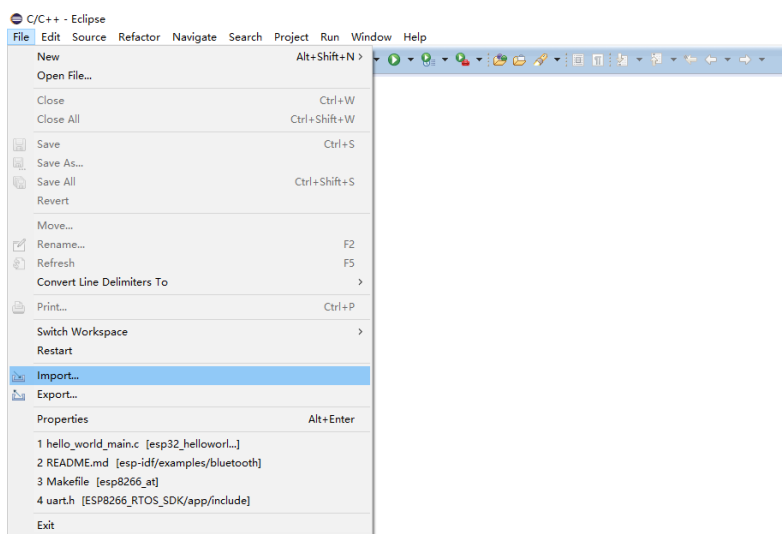
安信可 IDE 安装好后，已经给我们准备好了打开的快捷方式。我们打开刚才解压缩的“AiThinkerIDE_V0.5”文件夹，可以看到有个“OpenWorkSpace”

名称	修改日期	类型	大小
config	2018/12/28 13:09	文件夹	
cygwin	2018/12/28 12:59	文件夹	
docs	2018/12/28 12:59	文件夹	
eclipse	2018/12/28 13:11	文件夹	
examples	2018/12/28 12:59	文件夹	
firmware	2018/12/28 13:00	文件夹	
tools	2018/12/28 12:59	文件夹	
AiThinker_IDE	2017/1/26 17:35	应用程序	284 KB
ConfigTool	2017/4/10 21:54	应用程序	580 KB
DotNetFx40 Framework	2017/2/25 17:16	Internet 快捷方式	1 KB
OpenWorkSpace	2017/2/17 21:21	Windows 批处理...	1 KB
ReadMe	2017/4/10 21:58	文本文档	1 KB

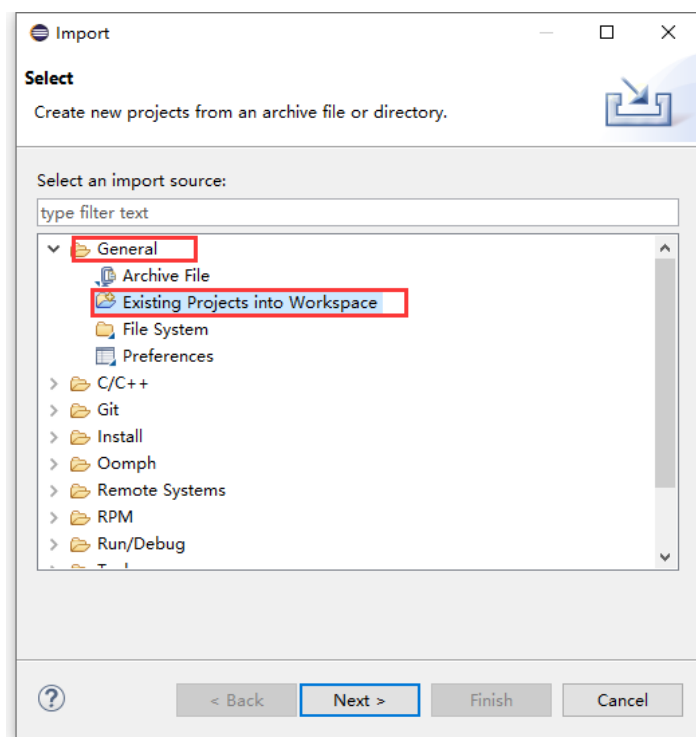
批处理文件。双击打开，会自己定位到所要导入的文件夹，该文件夹内，就是已经导入好的项目 SDK。

先说第一种导入方式：以安信可官方下载的 SDK 为例

步骤 1：双击打开 AiThinker_IDE。 “File” -> "Import"。项目栏里右击选择 “Import” 也可以。



步骤 2：选择 “General” -> "Existing Projects into Workspace" 点击 “next”

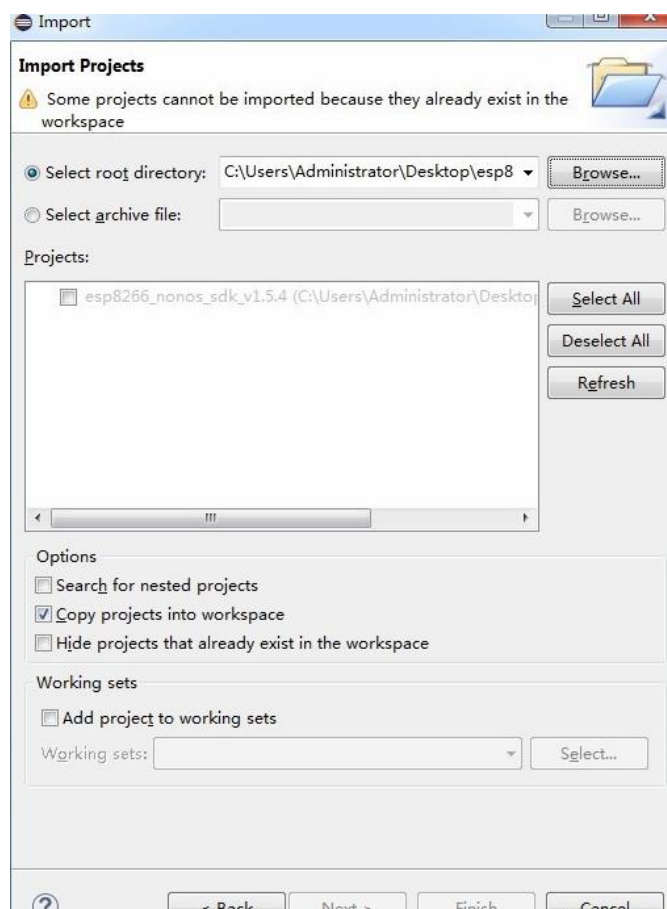


步骤 3:

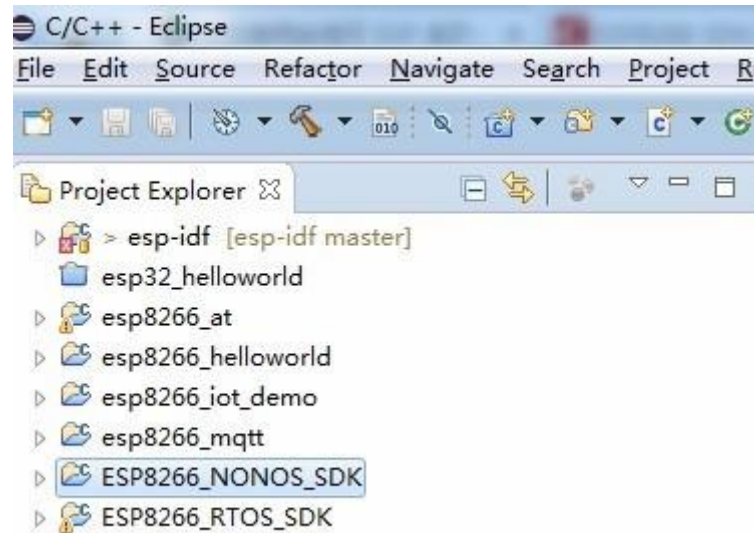
如图，选择桌面上解压好的 esp8266_nonos_sdk_v1.5.4 文件夹。下面又要注意了，**文件夹名称不能有小数点，不能是中文名**，并且将“Options”栏里，**"Copy projects into workspace"**，**前打勾**。意思是将该文件夹复制进文章开头提到的 project 文件夹内。**一定要打勾，否则编译又是不成功！不成功！不成功！。**

就是该项目文件夹一定要放到 AiThinkerIDE_V0.5\cygwin\home\aitinker\project 文件夹内。可以自己手动复制，也可以让软件自己复制。如果出现这种样子，就是文件夹内有同名文件了。

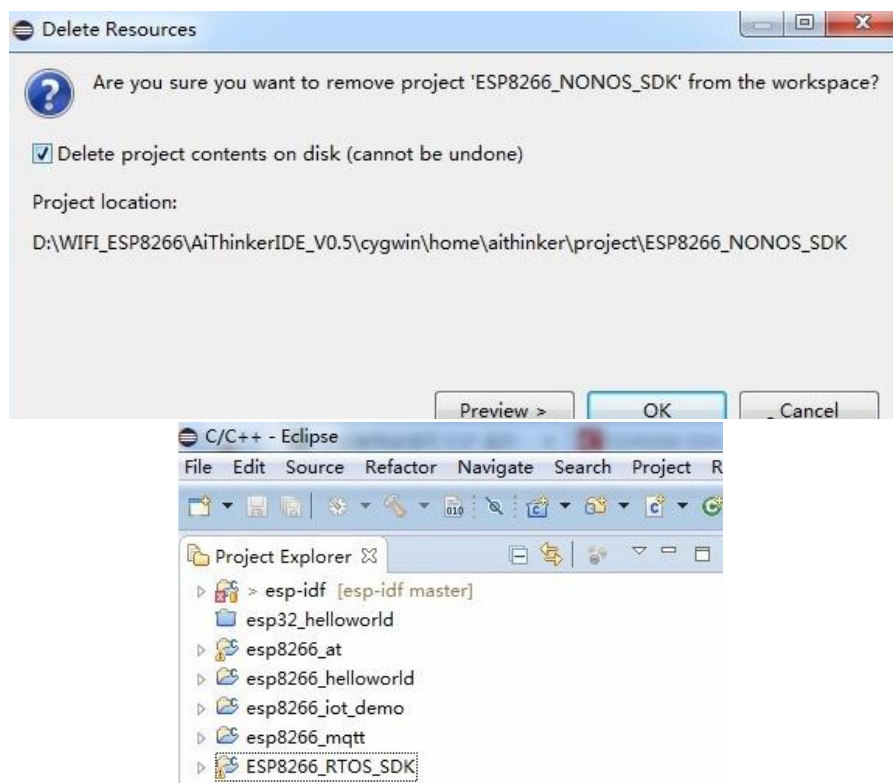
如果出现这种样子，就是文件夹内有同名文件了。



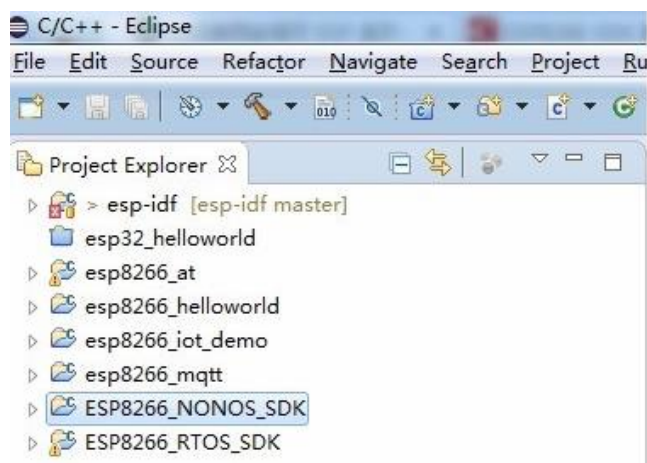
于程序默认已加载了“ESP8266_NONOS_SDK”，跟我们现在要导入的文档是一样的，重复了，所以要把以前的删除。按下“del”键，或者右键单击，选择“Delete”。



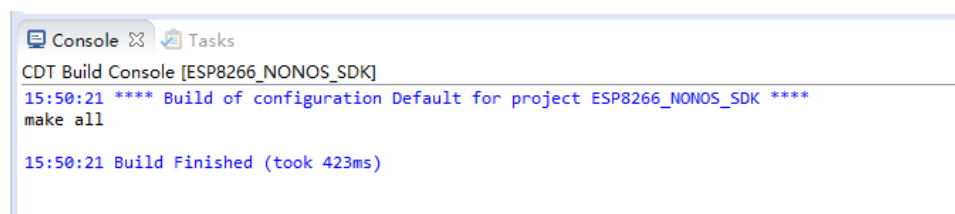
在弹出的对话框内如下图，将“Delete project contents on disk.....”前打上勾。然后点击 OK。就将项目名称连同 workspace 文件夹内的文档全部删除了。



重新按以上步骤导入下，成功后，先选择“clean project”，防止出错。然后再右键单击选择“bulid project”或者工具栏上的那个小锤子。



OK，开始编译。完成后，如下图



第二种导入方式

针对第三方云平台项目用第一种方法无法导入问题。例如“机智云”生成的项目文件夹。

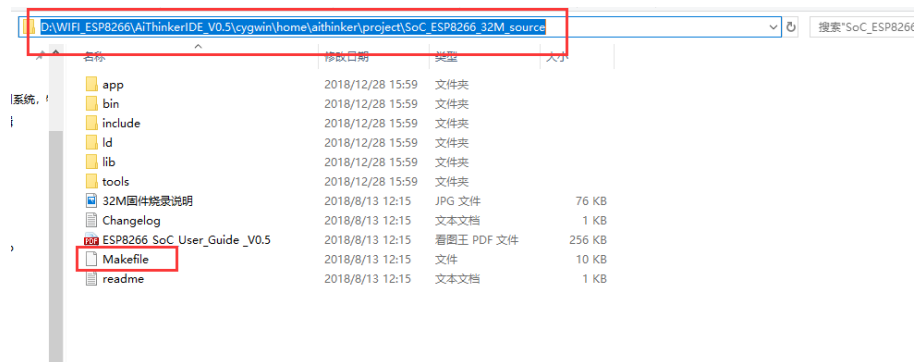
步骤 1：解压从机智云下载的项目并复制到 project 文件夹内。双击“OpenWorkspace”批处理文件，自动打开。简化步骤

名称	修改日期	类型	大小
config	2018/12/28 13:09	文件夹	
cygwin	2018/12/28 12:59	文件夹	
docs	2018/12/28 12:59	文件夹	
eclipse	2018/12/28 13:11	文件夹	
examples	2018/12/28 12:59	文件夹	
firmware	2018/12/28 13:00	文件夹	
tools	2018/12/28 12:59	文件夹	
AIThinker_IDE	2017/1/26 17:35	应用程序	284 KB
ConfigTool	2017/4/10 21:54	应用程序	580 KB
DotNetFx40 Framework	2017/2/25 17:16	Internet 快捷方式	1 KB
OpenWorkspace	2017/2/17 21:21	Windows 批处理...	1 KB
ReadMe	2017/4/10 21:58	文本文档	1 KB

步骤 2：将项目手动复制进去。下图选中的就是我刚才复制的项目文件夹。

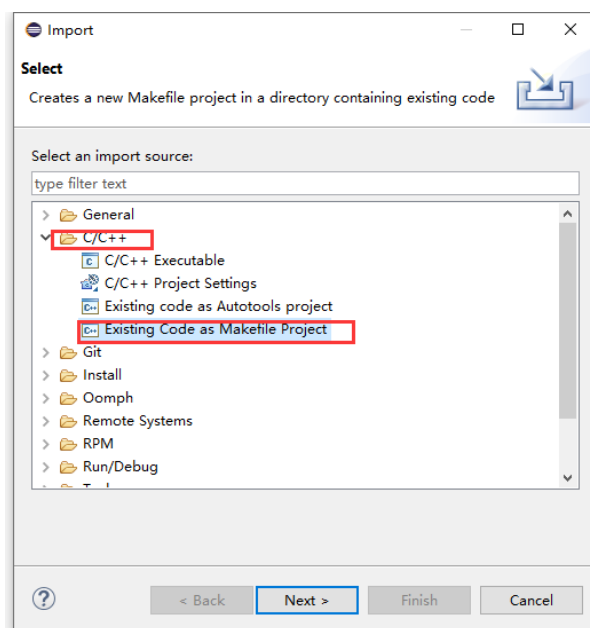
.metadata	2018/12/28 12:59	文件夹
esp8266_at	2018/12/28 12:59	文件夹
esp8266_helloworld	2018/12/28 12:59	文件夹
esp8266_iot_demo	2018/12/28 12:59	文件夹
esp8266_mqtt	2018/12/28 12:59	文件夹
ESP8266_NONOS_SDK	2018/12/28 12:59	文件夹
ESP8266_RTOS_SDK	2018/12/28 12:59	文件夹
esp-idf	2018/12/28 12:59	文件夹
RemoteSystemsTempFiles	2018/12/28 12:59	文件夹
SoC_ESP8266_32M_source	2018/12/28 15:59	文件夹
common_nonos.mk	2017/4/10 21:01	MK 文件 11 KB

步骤 3：双击打开该文件夹，直到定位到有“MakeFile”的文件夹处。并复制文件路径，到安信可 IDE 内用。注意路径上有文件名，不能出现中文。

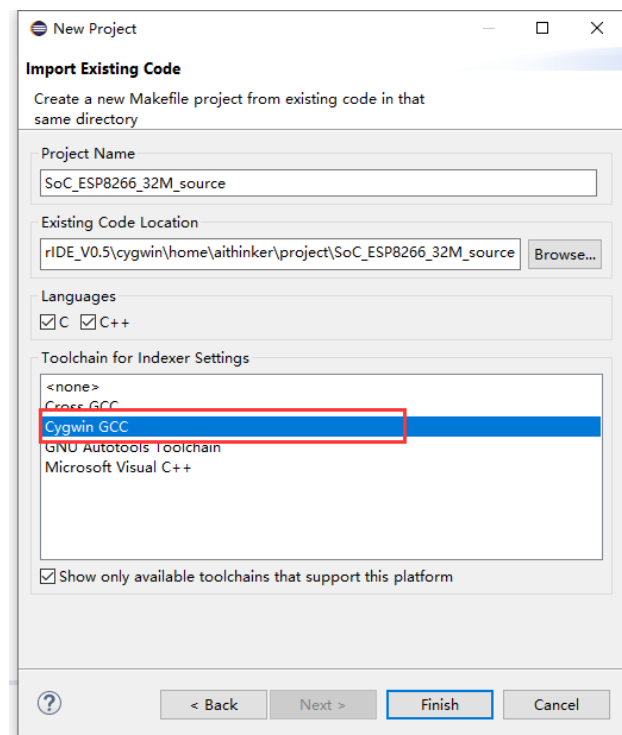


步骤 4：双击打开 AiThinker_IDE。 “File” -> "Import"。图略了。

步骤 5：双击下图的 “Existing Code as Makefile Project”



步骤 6: 将刚才复制的路径直接粘贴到 “Existing Code Location” 处，省地点 “Browse” 按钮一层层找了。下面选 “Cygwin GCC”。 点 “Finish” 。



导入成功，但还有些地方需要修改，否则编译不成功。我是从机智云论坛上看到的，目前我还没整明白。就是双击打开 MakeFile，将下面这段程序参数修改下。

BOOT?=none

APP?=0

SPI_SPEED?=40

SPI_MODE?=QIO

SPI_SIZE_MAP?=0

改成

BOOT?=new

APP?=1

SPI_SPEED?=40

SPI_MODE?=QIO

SPI_SIZE_MAP?=6

boot 赋值为 new，意思就是可以升级

APP 赋值为 1，就可以生成 user1 的 bin 文件 如果需要 OTA 云端升级，就要用到 user2 文件，只要在这里将 APP 值赋为 2 就行了。

SPI_SPEED 是下载的速度

SPI_MODE 是下载模式

SPI_SIZE_MAP 是 flash 的大小这个可以看上面那个文档详细介绍 赋值为 0 表示使用的 flash 是 512k

有一处没搞清楚，安信可官方的 SPI_SIZE_MAP?=2 机智云论坛上的那位大神是 SPI_SIZE_MAP?=4，而下面有位版主说要改成 SPI_SIZE_MAP?=6 反正都能编译成功。

具体可以查看这人的文章

<http://blog.csdn.net/chen244798611/article/details/51375337>

试验了下。

编译后生成的 bin 文件名称不同。看 user1.4096.new.4.bin 那行。

SPI_SIZE_MAP?=6 Flash 容量是 4096Kb

```
Console Tasks
CDT Build Console [SoC_ESP8266_32M_source]
!!!
-1763380647
1763380646
Support boot_v1.4 and +
Generate user1.4096.new.6.bin successully in folder bin/upgrade.
boot.bin----->0x000000
user1.4096.new.6.bin--->0x01000
!!!
make[1]: Leaving directory '/home/aithinker/project/SoC_ESP8266_32M_source/app'

16:11:28 Build Finished (took 41s.797ms)
```

user1.1024.new.2.bin SPI_SIZE_MAP?=2 Flash 容量是 1024Kb

```
Console Tasks
CDT Build Console [SoC_ESP8266_32M_source]
!!!
-487164665
487164664
Support boot_v1.2 and +
Generate user1.1024.new.2.bin successully in folder bin/upgrade.
boot.bin----->0x000000
user1.1024.new.2.bin--->0x01000
!!!
make[1]: Leaving directory '/home/aithinker/project/SoC_ESP8266_32M_source/app'

16:14:17 Build Finished (took 30s.125ms)
```

第三章 烧录

关于烧录方面，这里提一下。烧录文件在项目文件 bin/upgrade 文件夹内，文件格式是.bin。烧录地址是后面的 0x01000。

软件：[ESPFlashDownloadTool_v3.4.4](#)

在安信可官方也可以下载到，搜文件名。

烧录前先讲些基本知识。

软件编译好后，烧录文件为 bin 格式的文件。例如：boot.bin 和 user1.1024.new.2.bin 按生成的地址为准。

设置好以后 我们可以看看编译的结果如下

2.4. 烧录固件

根据实际使用的 ESP8266 硬件模块 Flash 大小，对应烧录地址如下表。

表 2-2. FOTA 固件下载地址（单位：KB）

BIN 文件	各个 Flash 容量对应的下载地址					
	512	1024	2048		4096	
			512+512	1024+1024	512+512	1024+1024
master_device_key.bin	0x3E000	0x7E000	0x7E000	0xFE000	0x7E000	0xFE000
blank.bin（烧录位置一）	0x7B000	0xFB000	0x1FB000		0x3FB000	
esp_init_data_default.bin	0x7C000	0xFC000	0x1FC000		0x3FC000	
blank.bin（烧录位置二）	0x7E000	0xFE000	0x1FE000		0x3FE000	
boot.bin	0x00000					
user1.bin	0x01000					

表 2-3. FOTA 固件说明

BIN 文件	说明
master_device_key.bin	用户从乐鑫云申请，依此享受乐鑫云端服务。 存放于用户参数区，储存地址由用户应用程序自定义。 表 2-2 中的烧录位置为 <i>IoT_Demo</i> 程序按照章节 2.2.1 设定的示例位置。
blank.bin（烧录位置一）	初始化 <i>RF_CAL</i> 参数区。 烧录位置由应用程序中的 <i>user_rf_cal_sector_set</i> 设置决定。 表 2-2 中的烧录位置为 <i>IoT_Demo</i> 程序中设定的示例位置。 由乐鑫官方提供，位于 <i>ESP266_SDK/bin</i> 路径下。

```

CDT Build Console [SoC_ESP8266_32M_source]
!!!
-487164665
487164664
Support boot_v1.2 and +
Generate user1.1024.new.2.bin successfully in folder bin/upgrade.
boot.bin----->0x00000
user1.1024.new.2.bin--->0x01000
!!!
make[1]: Leaving directory '/home/aithinker/project/SoC_ESP8266_32M_source/app'

16:14:17 Build Finished (took 30s.125ms)

```

从这里可以看到 boot 的开始地址是 0x00000 支持的版本必须要 1.2 版本或以上版本

BIN 文件	说明
<i>esp_init_data_default.bin</i>	初始化其他射频参数区，至少烧录一次。 当 <i>RF_CAL</i> 参数区初始化烧录时，本区域也需烧录。 由乐鑫官方提供，位于 <i>ESP266_SDK\bin</i> 路径下。
<i>blank.bin</i> （烧录位置二）	初始化系统参数区。 由乐鑫官方提供，位于 <i>ESP266_SDK\bin</i> 路径下。
<i>boot.bin</i>	主程序，由乐鑫官方提供，位于 <i>ESP266_SDK\bin</i> 路径下。
<i>user1.bin</i>	主程序，编译应用程序生成，位于 <i>ESP266_NONOS_SDK\bin\upgrade</i> 路径下。

过程省略！！！！