

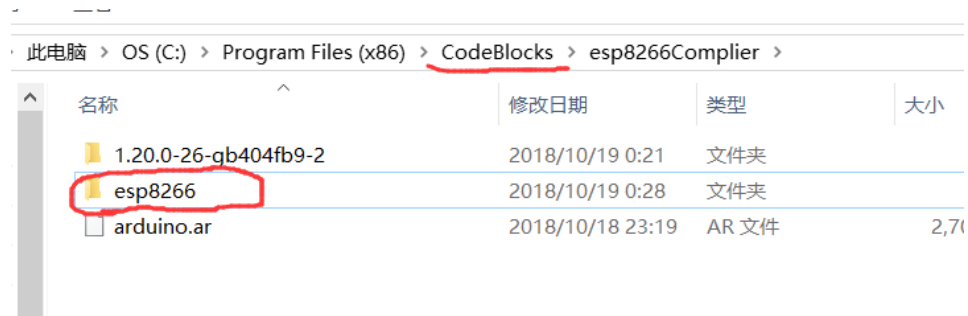
Mise en place d'une chaine de développement code blocks pour un module ESP8266 de Ai Thinker

1. Configuration de l'environnement

1.1 Télécharger le dossier Cores/Esp8266 depuis :

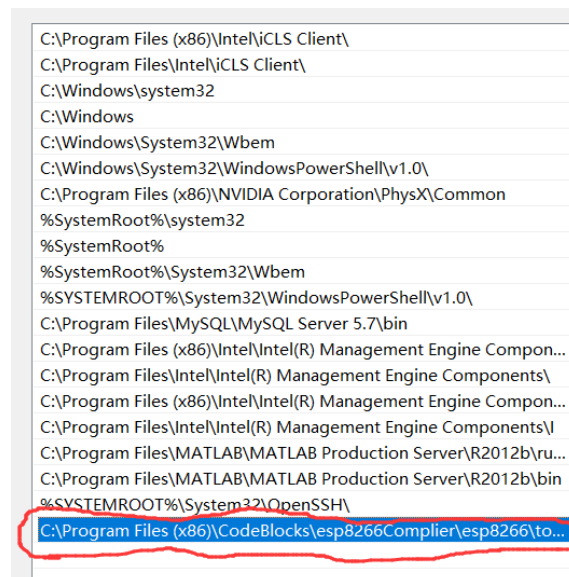
<https://github.com/esp8266/Arduino>

Déplacer le dans le répertoire d'installation de Code::blocks.

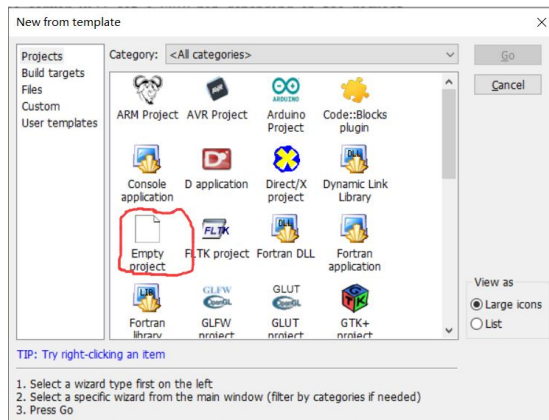


1.2 Ajoutez le répertoire où se trouve le compilateur xtensa-lx106-elf-gcc

à la variable d'environnement système



1.3 Créer un Empty project (Projet vide)



Télécharger les fichiers de bibliothèque que vous devez utiliser dans votre projet.

Depuis : <https://github.com/esp8266/Arduino/tree/master/libraries>

Puis ajouter ces fichiers au dossier source de votre projet Code::Blocks

| nom | ^ | 修改日期 | 类型 | 大小 |
|-----------------------------|---|------------------|--------|-------|
| include | | 2018/10/24 21:11 | 文件夹 | |
| BearSSLHelpers.cpp | | 2018/10/24 21:11 | C++ 文件 | 26 KB |
| BearSSLHelpers.h | | 2018/10/24 21:11 | H 文件 | 5 KB |
| CertStoreBearSSL.cpp | | 2018/10/24 21:11 | C++ 文件 | 6 KB |
| CertStoreBearSSL.h | | 2018/10/24 21:11 | H 文件 | 4 KB |
| ESP8266WiFi.cpp | | 2018/10/24 21:11 | C++ 文件 | 3 KB |
| ESP8266WiFi.h | | 2018/10/24 21:11 | H 文件 | 3 KB |
| ESP8266WiFiAP.cpp | | 2018/10/24 21:11 | C++ 文件 | 12 KB |
| ESP8266WiFiAP.h | | 2018/10/24 21:11 | H 文件 | 2 KB |
| ESP8266WiFiGeneric.cpp | | 2018/10/24 21:11 | C++ 文件 | 16 KB |
| ESP8266WiFiGeneric.h | | 2018/10/24 21:11 | H 文件 | 5 KB |
| ESP8266WiFiMulti.cpp | | 2018/10/24 21:11 | C++ 文件 | 9 KB |
| ESP8266WiFiMulti.h | | 2018/10/24 21:11 | H 文件 | 2 KB |
| ESP8266WiFiScan.cpp | | 2018/10/24 21:11 | C++ 文件 | 11 KB |
| ESP8266WiFiScan.h | | 2018/10/24 21:11 | H 文件 | 3 KB |
| ESP8266WiFiSTA.cpp | | 2018/10/24 21:11 | C++ 文件 | 20 KB |
| ESP8266WiFiSTA.h | | 2018/10/24 21:11 | H 文件 | 4 KB |
| ESP8266WiFiType.h | | 2018/8/3 9:19 | H 文件 | 5 KB |
| WiFiClient.cpp | | 2018/10/24 21:11 | C++ 文件 | 8 KB |
| WiFiClient.h | | 2018/10/24 21:11 | H 文件 | 4 KB |
| WiFiClientSecure.h | | 2018/10/24 21:11 | H 文件 | 2 KB |
| WiFiClientSecureAxTLS.cpp | | 2018/10/24 21:11 | C++ 文件 | 25 KB |
| WiFiClientSecureAxTLS.h | | 2018/10/24 21:11 | H 文件 | 4 KB |
| WiFiClientSecureBearSSL.cpp | | 2018/10/24 21:11 | C++ 文件 | 53 KB |
| WiFiClientSecureBearSSL.h | | 2018/10/24 21:11 | H 文件 | 10 KB |
| WiFiServer.cpp | | 2018/10/24 21:11 | C++ 文件 | 5 KB |
| WiFiServer.h | | 2018/10/24 21:11 | H 文件 | 3 KB |
| WiFiServerSecure.h | | 2018/10/24 21:11 | H 文件 | 1 KB |
| WiFiServerSecureAxTLS.cpp | | 2018/10/24 21:11 | C++ 文件 | 3 KB |
| WiFiServerSecureAxTLS.h | | 2018/10/24 21:11 | H 文件 | 2 KB |
| WiFiServerSecureBearSSL.cpp | | 2018/10/24 21:11 | C++ 文件 | 5 KB |
| WiFiServerSecureBearSSL.h | | 2018/10/24 21:11 | H 文件 | 3 KB |

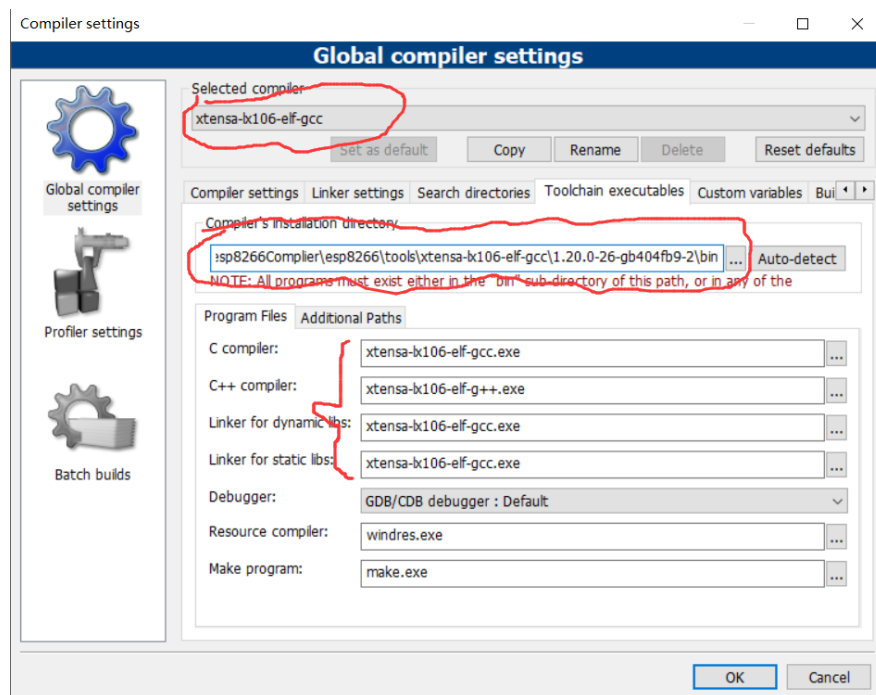
1.4 Configurer l'environnement de construction

Settings -> compiler -> toolchain executables

<Selected compiler> : Changer le nom en **xtensa-lx106-elf-gcc**

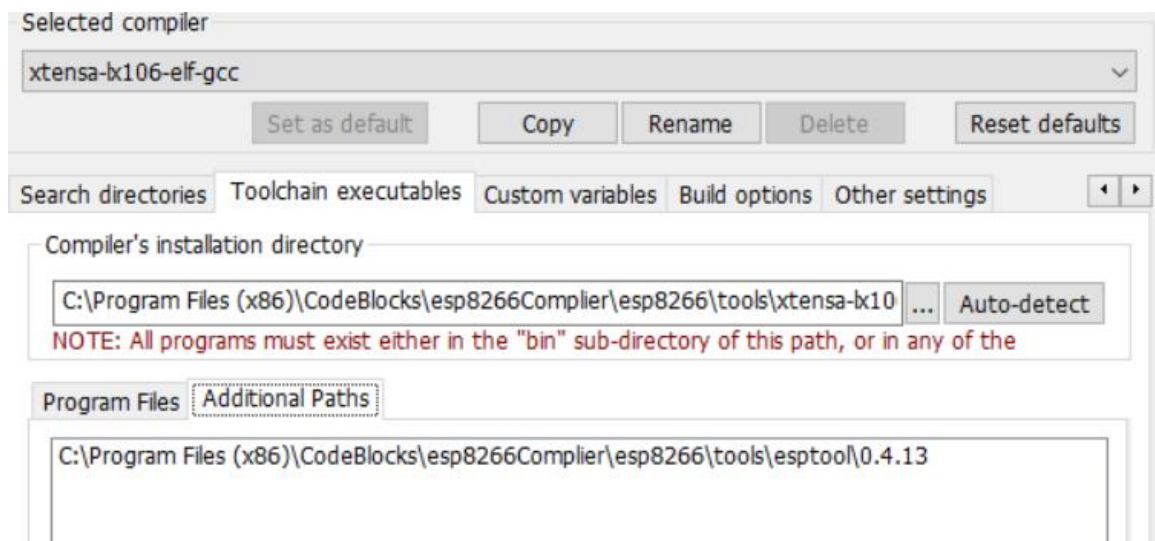
<Compiler's installation directory> : Sélectionnez le répertoire dans lequel se trouve le compilateur **xtensa-lx106-elf-gcc** dans le répertoire **esp8266**.

<Program Files> : Sélectionnez le compilateur approprié



< Additional Paths >:

Ajouter : `xxx\esp8266\tools\esptool\0.4.13`



1.5 Définir le répertoire de recherche du fichier d'en-tête et le répertoire de recherche de l'éditeur de liens

Settings—>compiler-->Search directories

<Compiler>

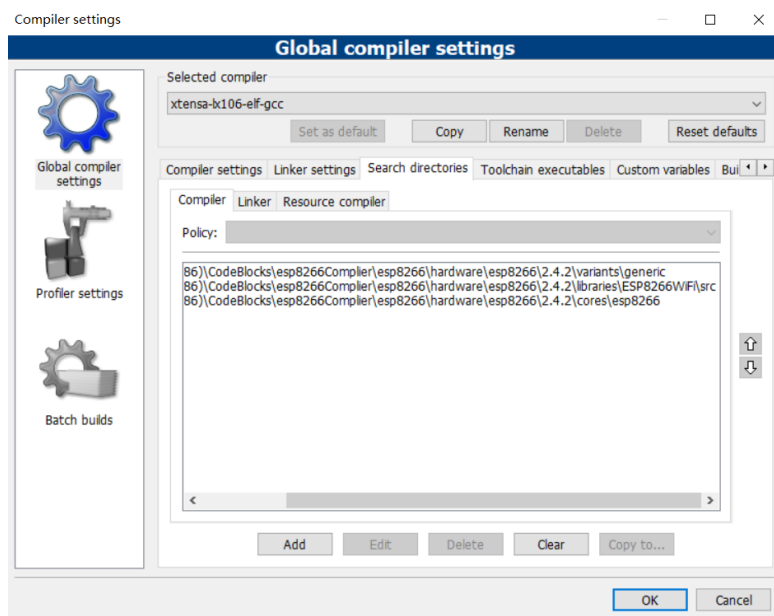
Ajouter :

Xxx\esp8266\hardware\esp8266\2.4.2\variants\generic

Xxx\ esp8266\hardware\esp8266\2.4.2\libraries\ESP8266WiFi\src

(Ce répertoire n'est pas constant. Si vous devez utiliser d'autres fichiers de bibliothèque, vous pouvez l'ajouter vous-même.)

Xxx\ esp8266\hardware\esp8266\2.4.2\cores\esp8266



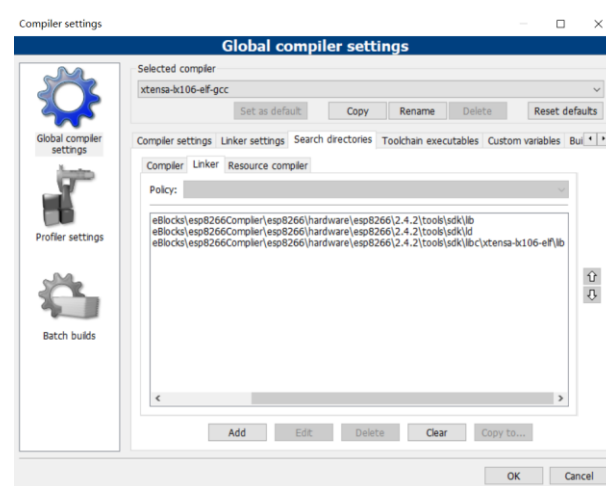
<Linker>

Ajouter :

Xxx\esp8266\hardware\esp8266\2.4.2\tools\sd\lib

Xxx\esp8266\hardware\esp8266\2.4.2\tools\sd\ld

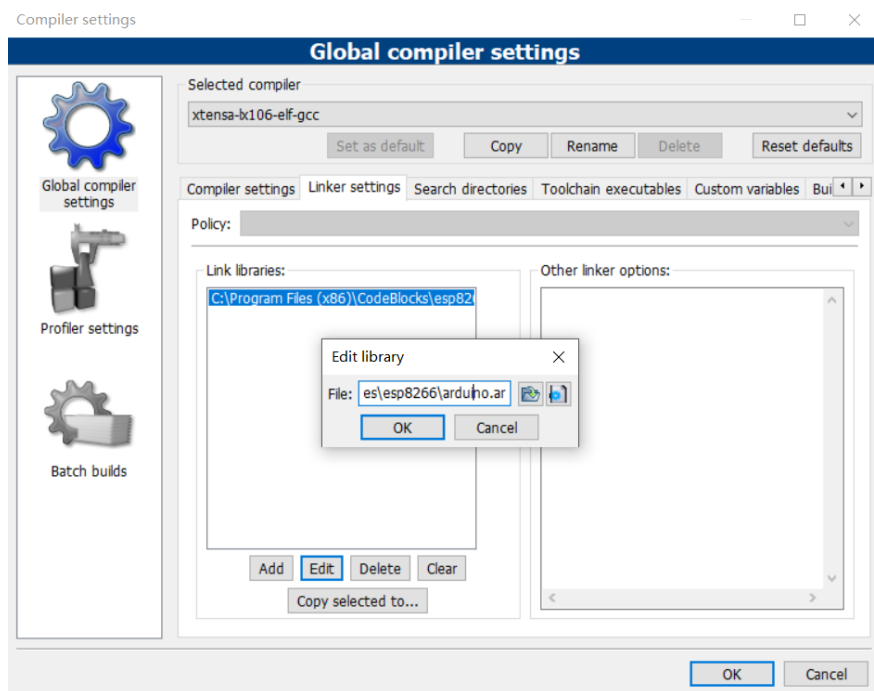
Xxx\esp8266\hardware\esp8266\2.4.2\tools\sd\libc\xtensa-lx106-elf\lib



1.6 Ajouter le chemin de la bibliothèque de liens

Settings—>compiler-->Linker settings

Ajouter : xxx\esp8266\hardware\esp8266\2.4.2\cores\esp8266\arduino.ar

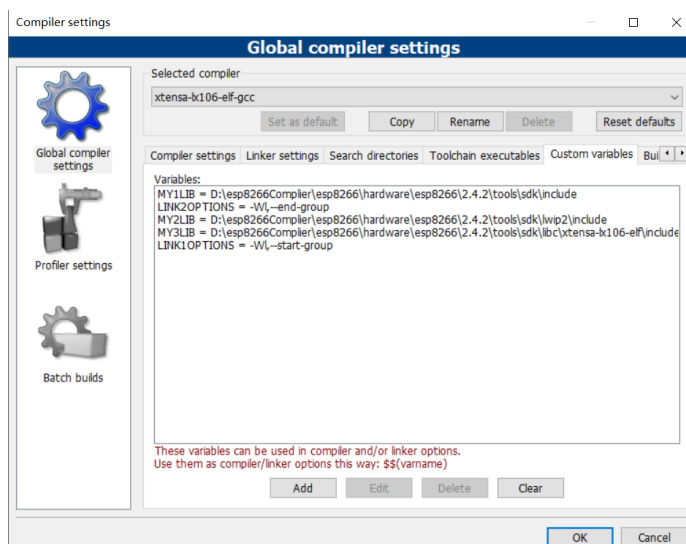


1.7 Ajouter des variables personnalisées

Settings—>compiler-->Custom variables

Ajouter:

```
MY1LIB = xxx\esp8266\hardware\esp8266\2.4.2\tools\sdh\include
MY2LIB = xxx\esp8266\hardware\esp8266\2.4.2\tools\sdh\lwip2\include
MY3LIB = xxx\esp8266\hardware\esp8266\2.4.2\tools\sdh\libc\xtensa-
        lx106-elf\include
LINK1OPTIONS = -Wl,--start-group
LINK2OPTIONS = -Wl,--end-group
```



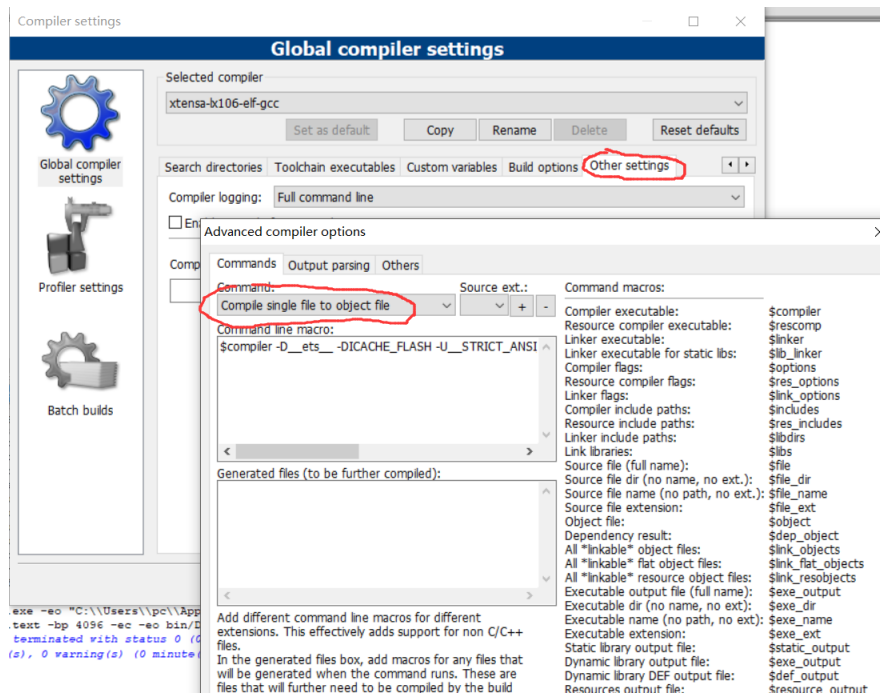
1.8 Modifier les instructions de compilation / lien

Settings—>compiler-->Other settings/Advanced compiler

Commands->Compile single file to object file

Ajouter la commands:

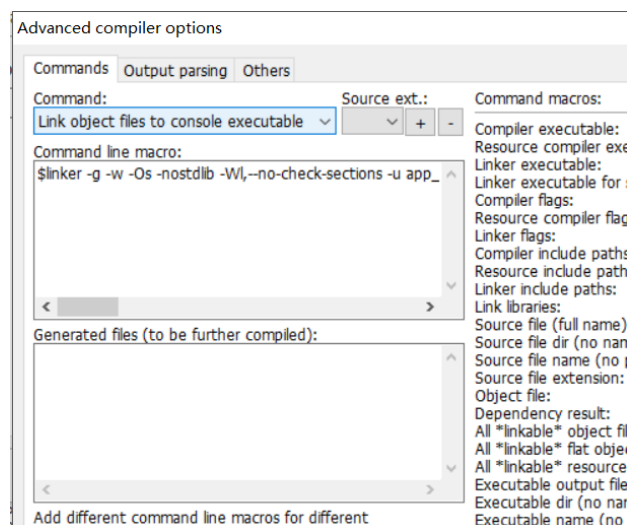
`$compiler -D_ets_ -DICACHE_FLASH -U_STRICT_ANSI_ "-I$My1Lib" "-I$My2Lib" "-I$My3Lib" -c $options $includes $file -o $object`



Commands->Link object files to console executable

Ajouter la commands:

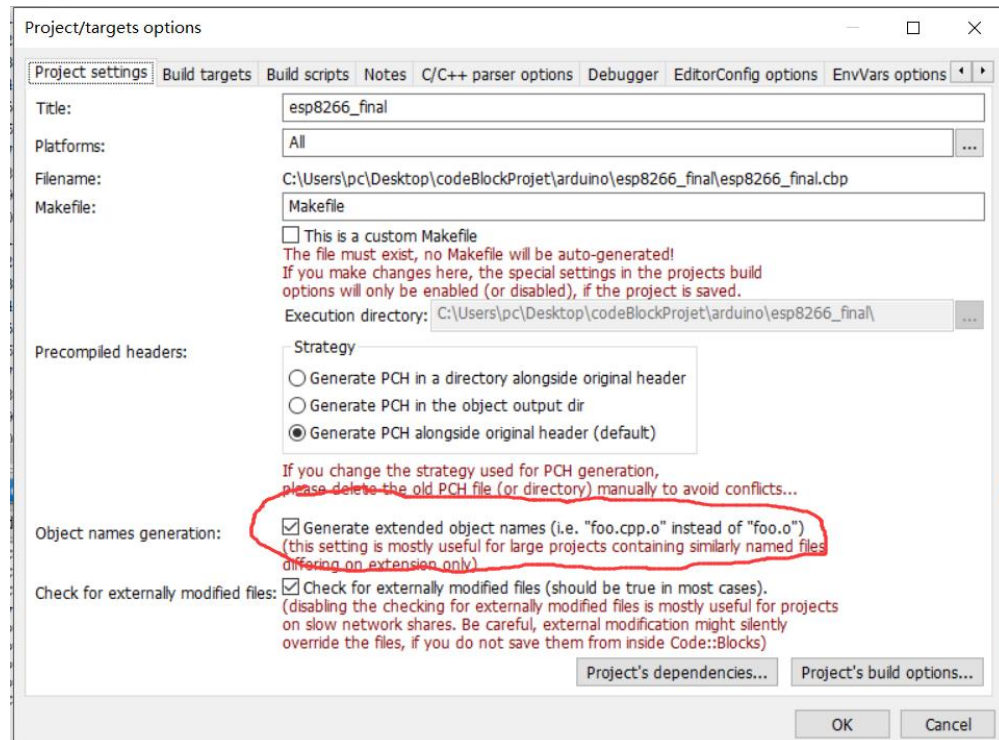
`$linker -g -w -Os -nostdlib -Wl,-no-check-sections -u app_entry -u _printf_float -u _scanf_float -Wl,-static $libdirs $link_resobjects $link_options -o $exe_output $Link1Options $link_objects $libs -lhal -lphy -lpp -lnet80211 -llwip2 -lwpa -lcrypto -lmain -lwps -lbearssl -lxtls -lespnow -lsmartconfig -lairkiss -lwpa2 -lstc++ -lm -lc -lgcc $Link2Options`



1.9 Définir les propriétés du projet

Project → properties

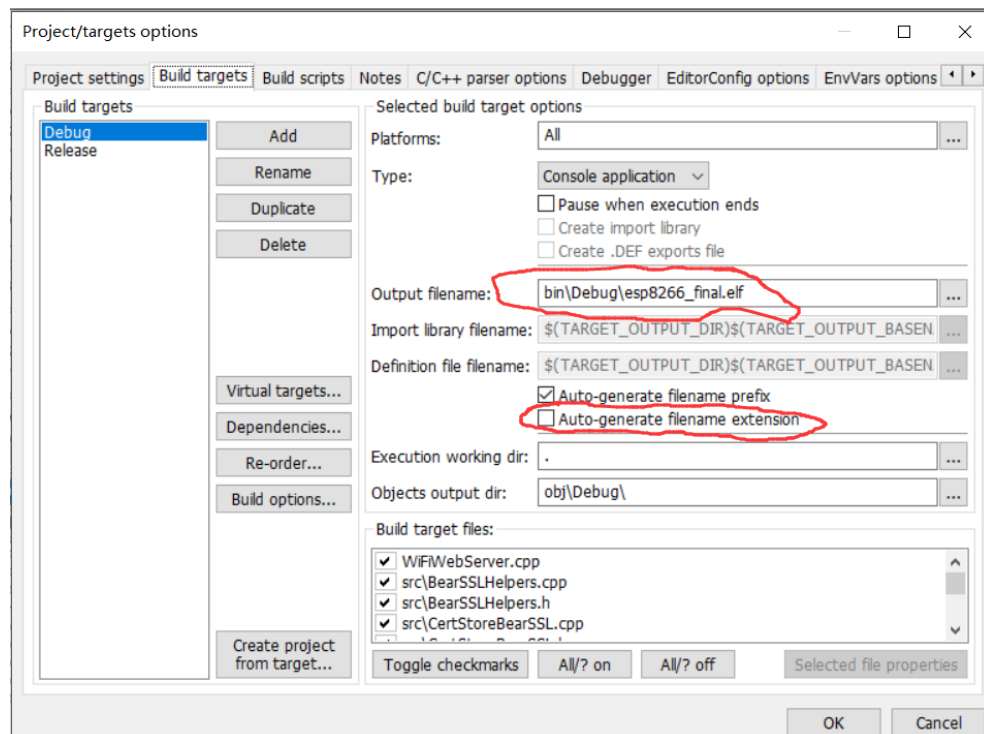
<Project settings>: Cocher Object names generation



<Build targets> :

Output filename: Changer le fichier avec le suffixe **elf**.

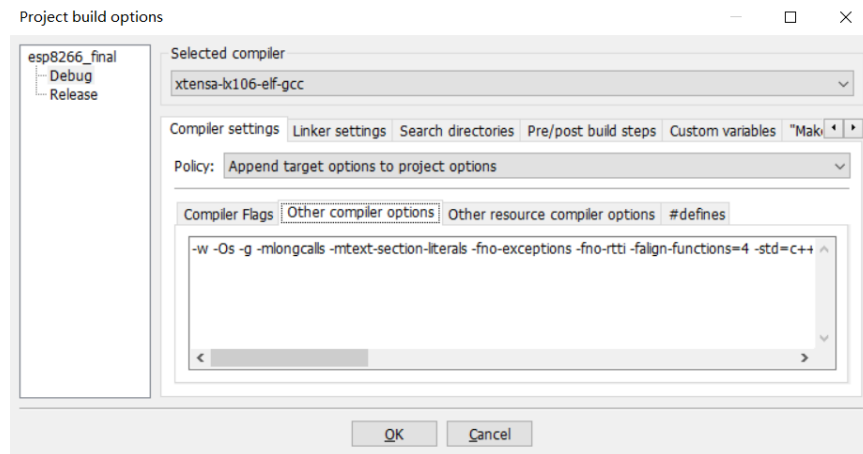
Décocher Auto-generate filename extension.



Project→Build options

<Other compiler options>

Ajouter: -w -Os -g -mlongcalls -mtext-section-literals -fno-exceptions -fno-rtti -falign-functions=4 -std=c++11 -MMD -ffunction-sections -fdata-sections -DF_CPU=80000000L -DLWIP_OPEN_SRC -DTCP_MSS=536 -DARDUINO=10807 -DARDUINO_ESP8266_ESP01 -DARDUINO_ARCH_ESP8266 -DARDUINO_BOARD="ESP8266_ESP01" -DLED_BUILTIN=2 -DESP8266

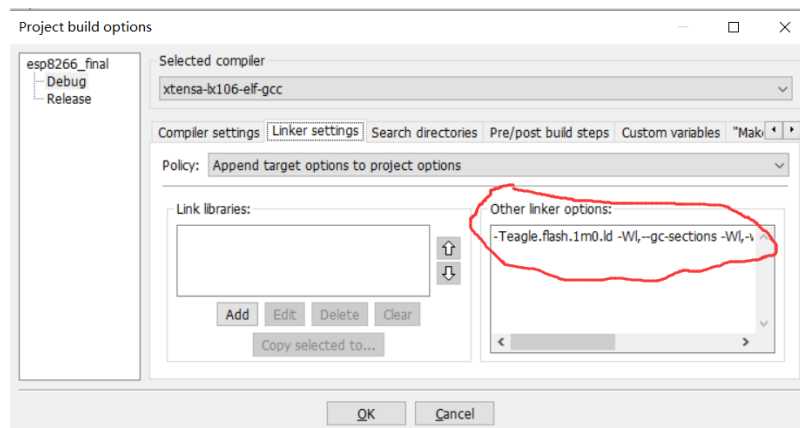


<Linker settings>

Other linker options

Ajouter:

-Teagle.flash.1m0.ld -Wl,--gc-sections -Wl,-wrap,system_restart_local -Wl,-wrap,spi_flash_read



<Pre/post build steps>

Post-build steps

Ajouter

: esptool.exe -eo "C:\\Users\\pc\\AppData\\Local\\Arduino15\\packages\\esp8266\\hardware\\esp8266\\2.4.2\\bootloaders\\eboot\\eboot.elf" -bo output.bin -bm dout -bf 40 -bz 1M -bs .text -bp 4096 -ec -eo \$exe_output -bs .irom0.text -bs .text -bs .data -bs .rodata -bc -ec

— □ ×



2 Démonstration de performance

