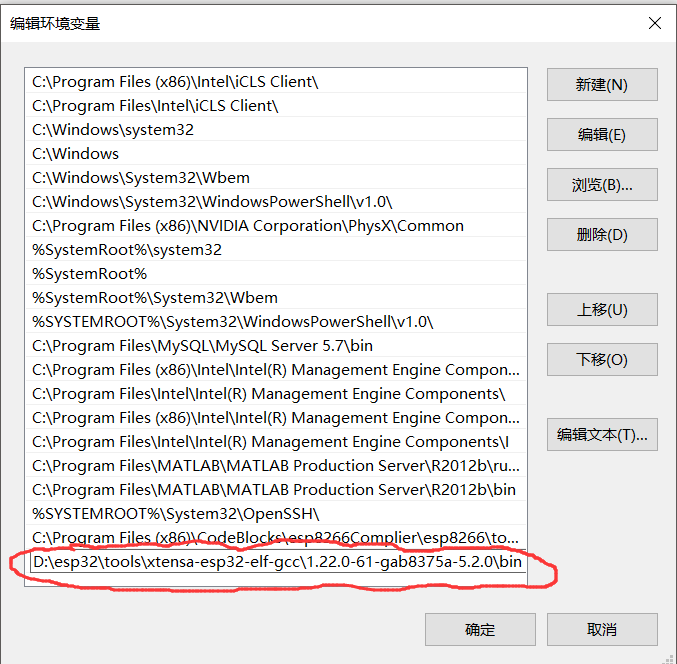
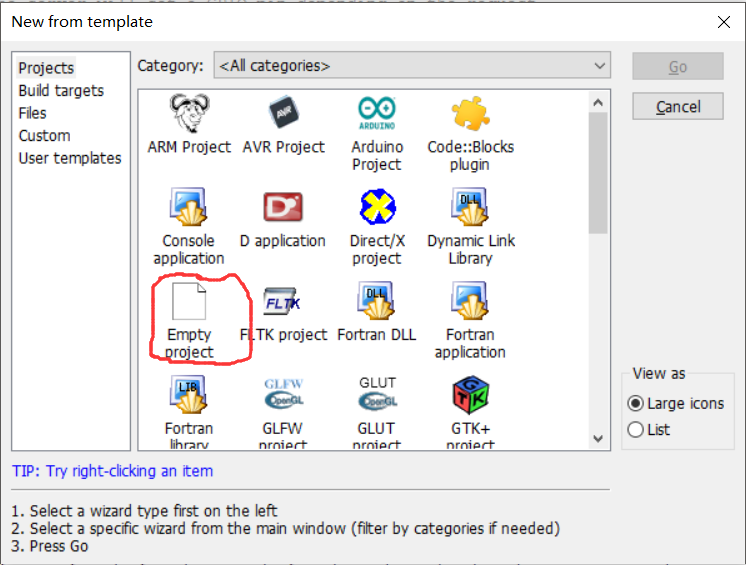
**Mise en place de la chaine de developpement sous Codeblocks d'un module Wemos, basé sur un ESP WROOM 32**

1. **Configurer l’environnement de code::bolcks**
   1. **Ajoutez le répertoire où se trouve le compilateur xtensa-esp32-elf-gcc à la variable d'environnement système**



* 1. **Créer un Empty project**



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



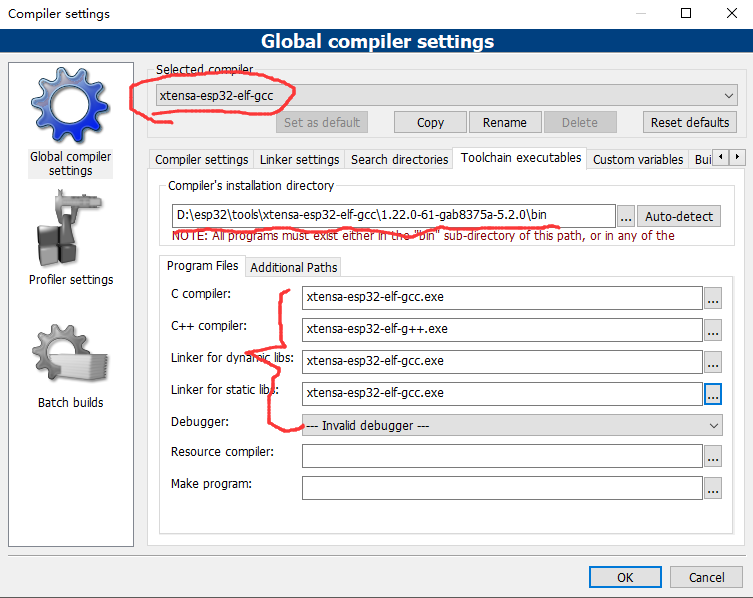
**1.3 Configurer l'environnement de construction**

Settings—>compiler-->toolchain executables

<Selected compiler>: Changer le nom en xtensa-esp32-elf-gcc

<Compiler’s installation directory>: Sélectionnez le répertoire dans lequel se trouve le compilateur xtensa-esp32-elf-gcc dans le répertoire esp32.

<Program Files>: Sélectionnez le compilateur approprié

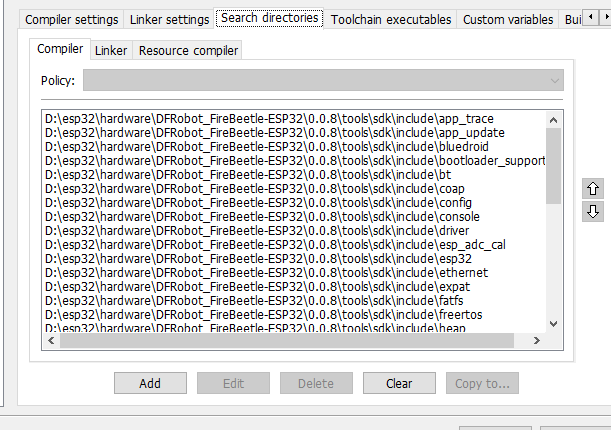


* 1. **Définir le répertoire de recherche du fichier d'en-tête et le répertoire de recherche de l'éditeur de liens**

Settings—>compiler-->Search directories

<Compiler>

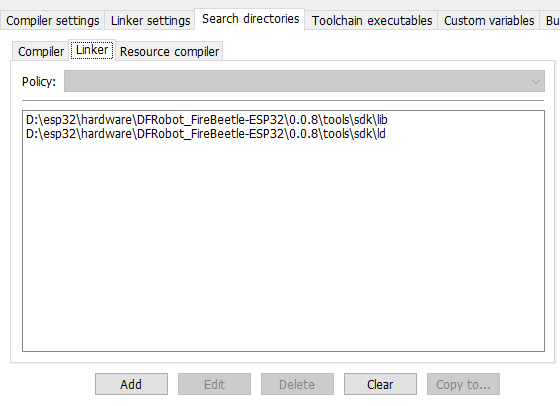
Ajoutez tous les fichiers du répertoire xxx \ esp32 \ hardware \ DFRobot\_FireBeetle-ESP32 \ 0.0.8 \ tools \ sdk \ include



<Linker>

Xxx\esp32\hardware\DFRobot\_FireBeetle-ESP32\0.0.8\tools\sdk\lib

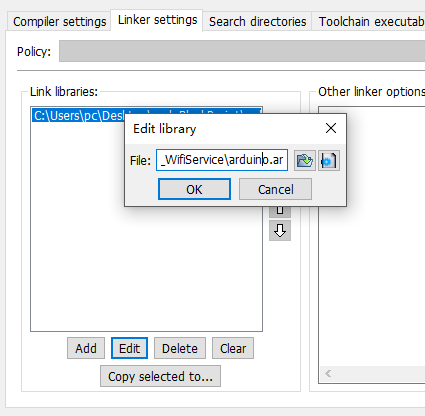
Xxx\esp32\hardware\DFRobot\_FireBeetle-ESP32\0.0.8\tools\sdk\ld



* 1. **Ajouter le chemin de la bibliothèque de liens**

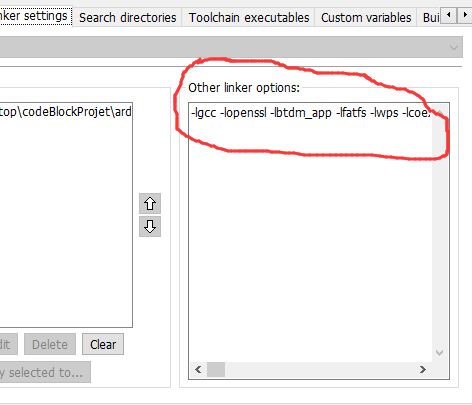
Settings—>compiler-->Linker settings

Mettez arduino.ar dans notre projet.



<Other linker options>

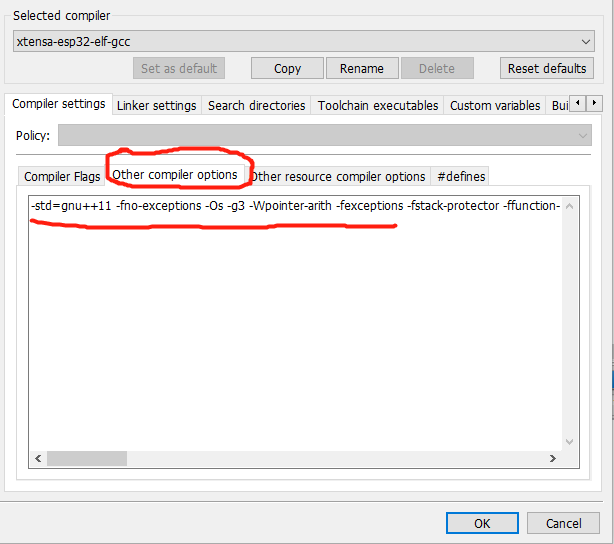
-lgcc -lopenssl -lbtdm\_app -lfatfs -lwps -lcoexist -lwear\_levelling -lhal -lnewlib -ldriver -lbootloader\_support -lpp -lmesh -lsmartconfig -ljsmn -lwpa -lethernet -lphy -lapp\_trace -lconsole -lulp -lwpa\_supplicant -lfreertos -lbt -lmicro-ecc -lcxx -lxtensa-debug-module -lmdns -lvfs -lsoc -lcore -lsdmmc -lcoap -ltcpip\_adapter -lc\_nano -lrtc -lspi\_flash -lwpa2 -lesp32 -lapp\_update -lnghttp -lspiffs -lespnow -lnvs\_flash -lesp\_adc\_cal -llog -lexpat -lm -lc -lheap -lmbedtls -llwip -lnet80211 -lpthread -ljson -lstdc++ -Wl,--end-group -Wl,-EL



Settings—>compiler-->Compiler Setting

<Other compiler options>

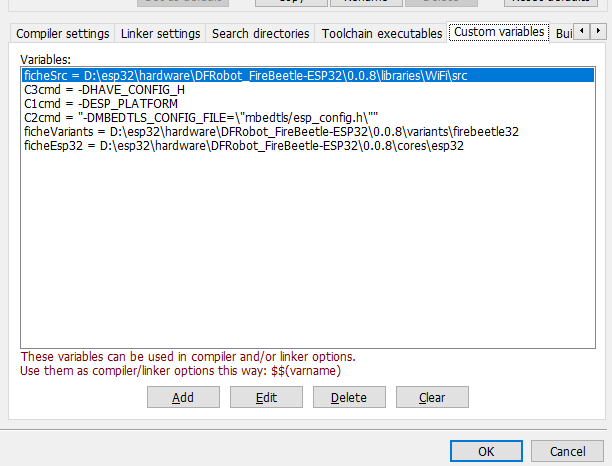
-std=gnu++11 -fno-exceptions -Os -g3 -Wpointer-arith -fexceptions -fstack-protector -ffunction-sections -fdata-sections -fstrict-volatile-bitfields -mlongcalls -nostdlib -w -Wno-error=unused-function -Wno-error=unused-but-set-variable -Wno-error=unused-variable -Wno-error=deprecated-declarations -Wno-unused-parameter -Wno-sign-compare -fno-rtti -MMD -c -DF\_CPU=240000000L -DARDUINO=10807 -DARDUINO\_ESP32\_DEV -DARDUINO\_ARCH\_DFROBOT\_FIREBEETLE-ESP32 "-DARDUINO\_BOARD=\"ESP32\_DEV\"" "-DARDUINO\_VARIANT=\"firebeetle32\"" -DESP32 -DCORE\_DEBUG\_LEVEL=0



* 1. **Ajouter des variables personnalisées**

Settings—>compiler-->Custom variables

Ajouter les variables suivantes:

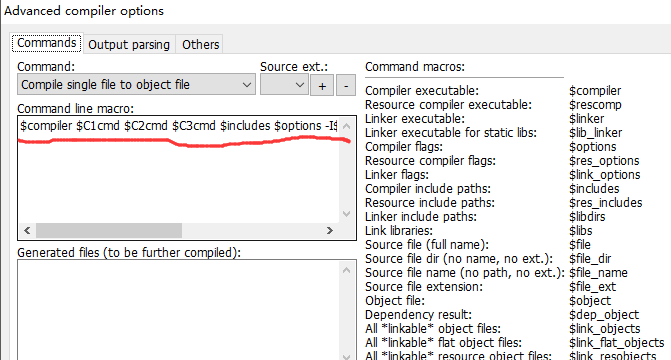


**1.7 Modifier les instructions de compilation / lien**

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

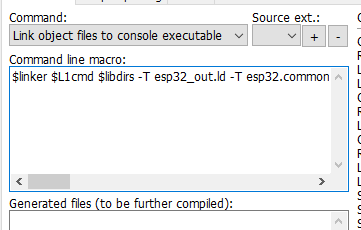
Commands->Compile single file to object file

$compiler $C1cmd $C2cmd $C3cmd $includes $options -I$ficheEsp32 -I$ficheVariants -I$ficheSrc $file -o $object



Commands->Link object files to console executable

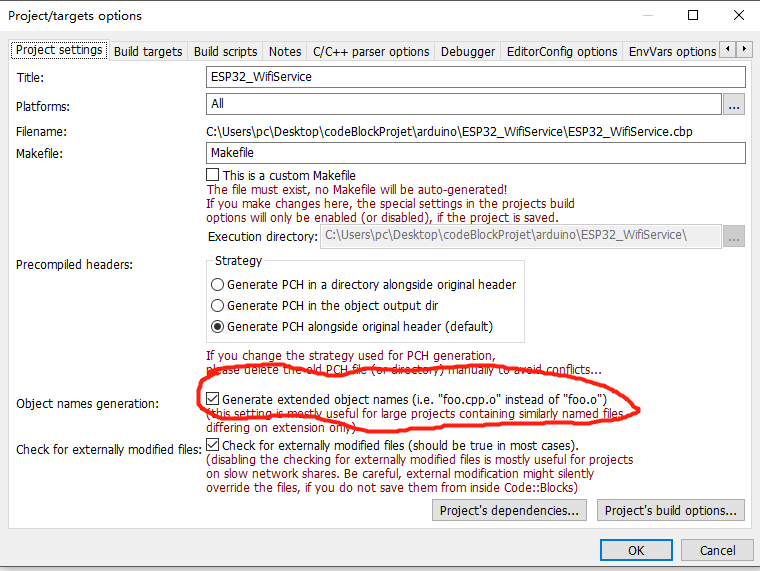
$linker $L1cmd $libdirs -T esp32\_out.ld -T esp32.common.ld -T esp32.rom.ld -T esp32.peripherals.ld -T esp32.rom.spiram\_incompatible\_fns.ld -u ld\_include\_panic\_highint\_hdl -u call\_user\_start\_cpu0 -Wl,--gc-sections -Wl,-static -Wl,--undefined=uxTopUsedPriority -u \_\_cxa\_guard\_dummy -u \_\_cxx\_fatal\_exception -Wl,--start-group $link\_objects $link\_resobjects $libs $link\_options -o $exe\_output



**1.8 Définir les propriétés du projet**

Project🡪properties

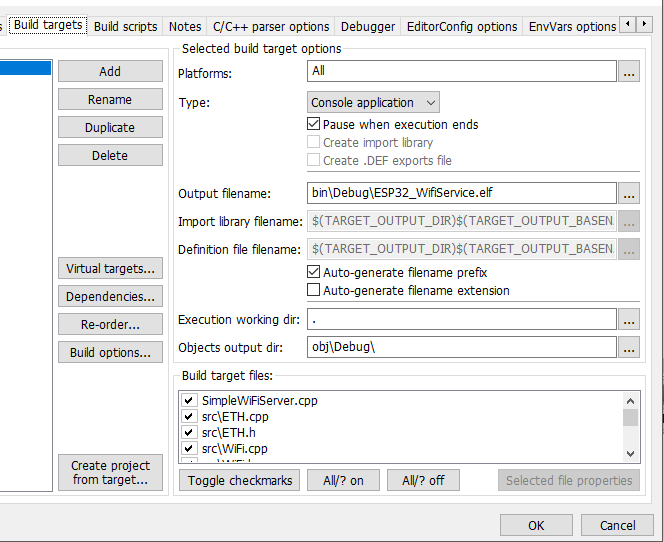
<Project settings>: Cocher Object names generation



<Build tragets>:

Output filename: Changer le fichier avec le suffixe elf.

Décocher Auto-generate filename extension.



Project🡪Build options

<Pre/post build steps>

Post-build steps:

gen\_esp32part.exe -q D:\esp32\hardware\DFRobot\_FireBeetle-ESP32\0.0.8\tools\partitions\default.csv output.partitions.bin

esptool.exe --chip esp32 elf2image --flash\_mode dio --flash\_freq 80m --flash\_size 4MB -o output.bin $exe\_output

1. **Démonstration de performance**

