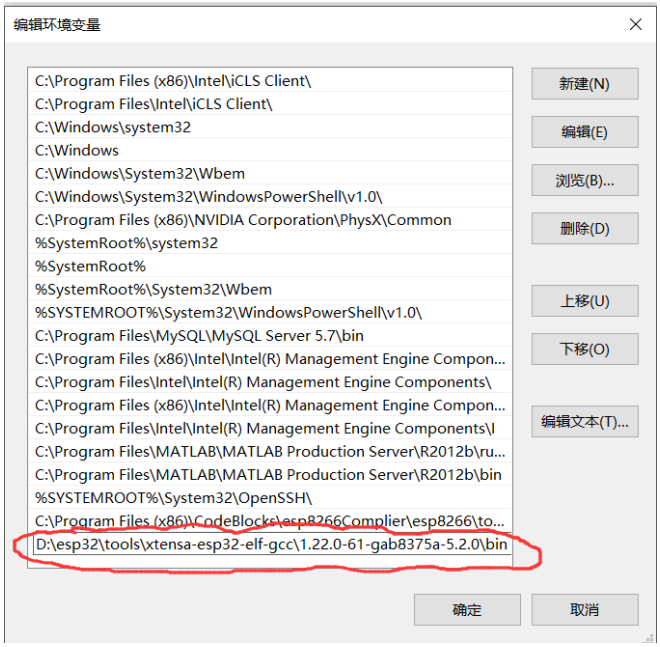
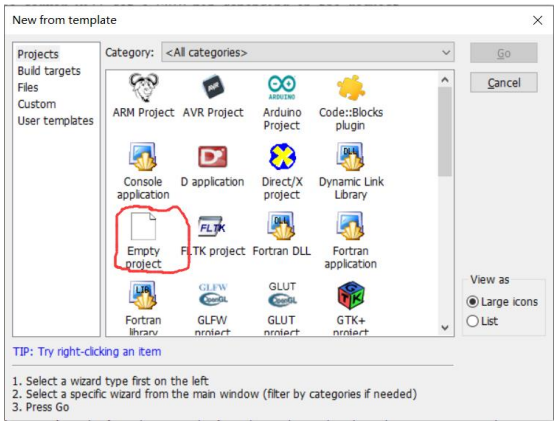
Mise en place de la chaine de developpement sous Codeblocks d'un module Wemos, basé sur un ESP WROOM 32

1 Configurer l'environnement de code::bolcks

1.1 Ajoutez le répertoire où se trouve le compilateur xtensa-esp32-elf-gcc à la variable d'environnement système



1.2 Créer un Empty project



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

ETH.cpp	2018/6/7 9:27	CPP 文件	8 KB
ETH.h	2018/6/7 9:27	H 文件	3 KB
WiFi.cpp	2017/6/15 14:25	CPP 文件	3 KB
WiFi.h	2017/6/15 14:25	H 文件	2 KB
WiFiAP.cpp	2018/6/11 15:04	CPP 文件	9 KB
WiFiAP.h	2017/6/15 14:26	H 文件	2 KB
WiFiClient.cpp	2018/6/11 15:04	CPP 文件	10 KB
WiFiClient.h	2017/6/15 14:26	H 文件	3 KB
WiFiGeneric.cpp	2018/6/11 15:05	CPP 文件	16 KB
WiFiGeneric.h	2018/6/11 15:05	H 文件	3 KB
WiFiMulti.cpp	2018/6/20 9:44	CPP 文件	9 KB
WiFiMulti.h	2018/6/11 15:06	H 文件	2 KB
WiFiScan.cpp	2018/6/12 16:42	CPP 文件	8 KB
WiFiScan.h	2017/6/15 14:25	H 文件	2 KB
WiFiServer.cpp	2018/6/11 15:06	CPP 文件	4 KB
WiFiServer.h	2017/6/15 14:26	H 文件	2 KB
WiFiSTA.cpp	2018/6/11 15:07	CPP 文件	17 KB
WiFiSTA.h	2018/6/11 15:07	H 文件	4 KB
WiFiType.h	2018/6/11 15:07	H 文件	2 KB
WiFiUdp.cpp	2017/6/15 14:25	CPP 文件	7 KB
WiFiUdp.h	2017/6/15 14:25	H 文件	3 KB

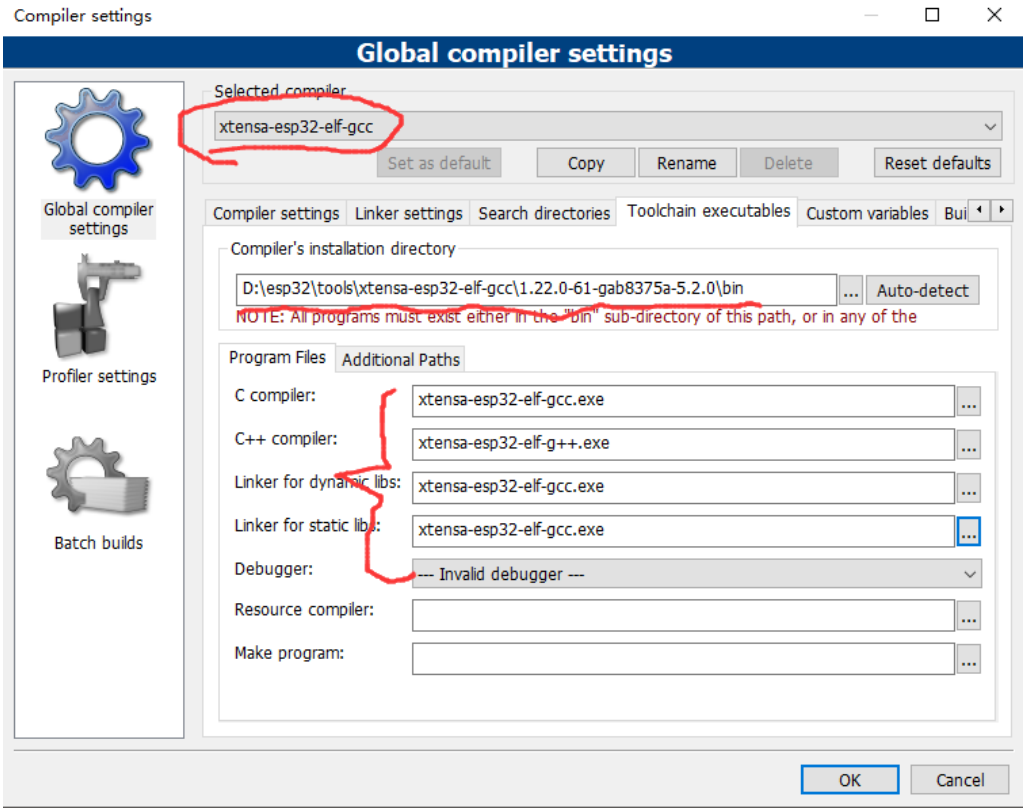
1.3 Configurer l'environnement de construction

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

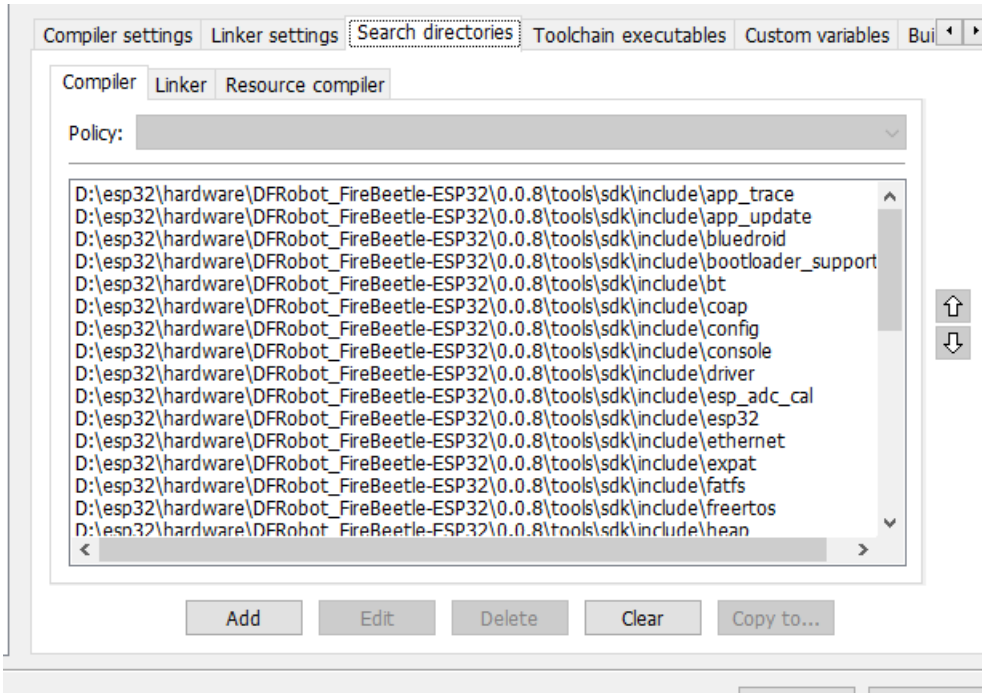


1.4 Définir le répertoire de recherche du fichier d'en-tête et le répertoire de recherche de l'éditeur de liens

Settings—>compiler-->Search directories

<Compiler>

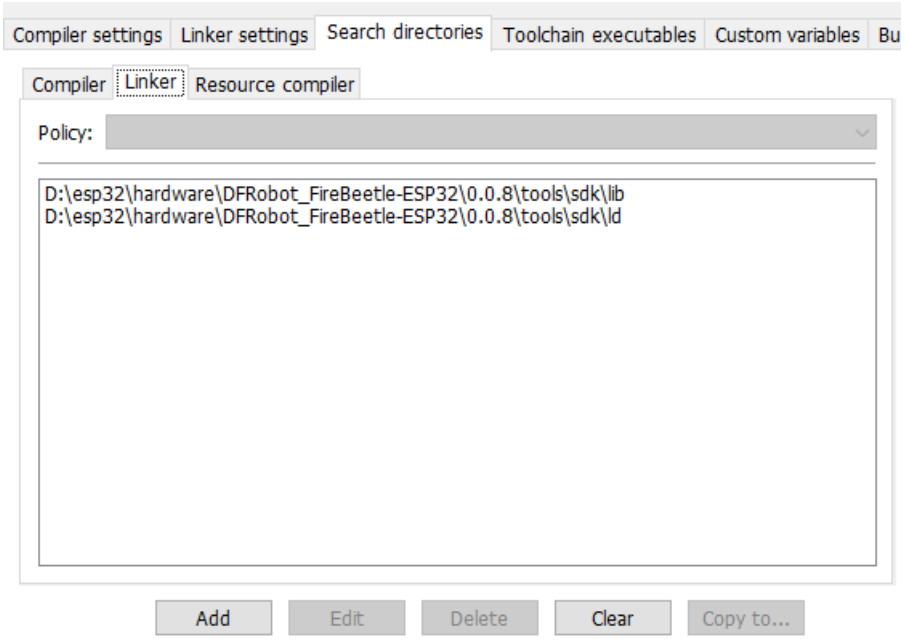
Ajoutez tous les fichiers du répertoire xxx \ esp32 \ hardware \ DFRobot_FireBeetle-ESP32 \ 0.0.8 \ tools \ sdk \ include



<Linker>

Xxx\esp32\hardware\DFRobot_FireBeetle-ESP32\0.0.8\tools\sdk\lib

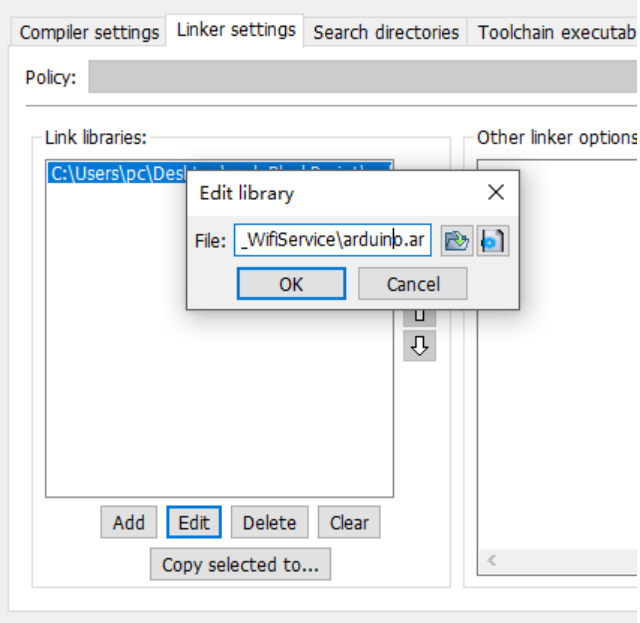
Xxx\esp32\hardware\DFRobot_FireBeetle-ESP32\0.0.8\tools\sdk\ld



1.5 Ajouter le chemin de la bibliothèque de liens

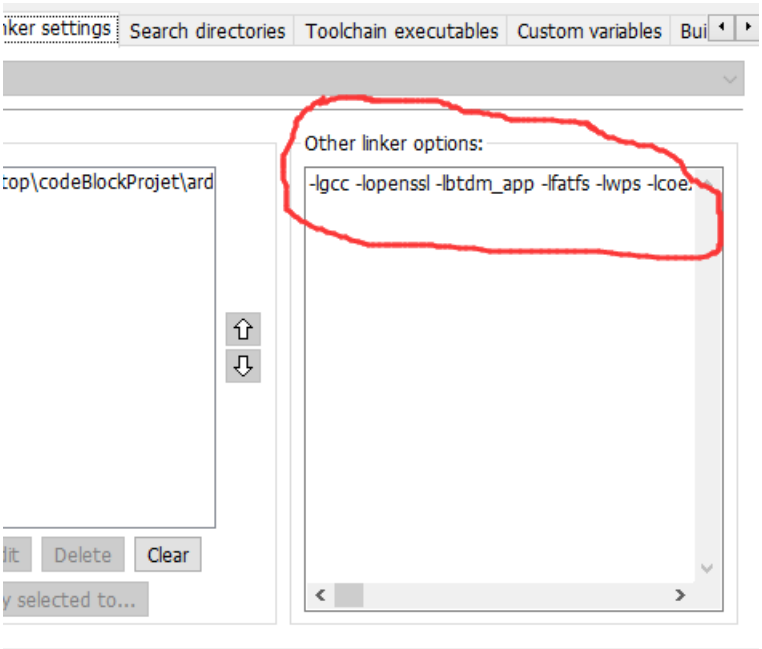
Settings—>compiler-->Linker settings

Mettez arduino.ar dans notre projet.



<Other linker options>

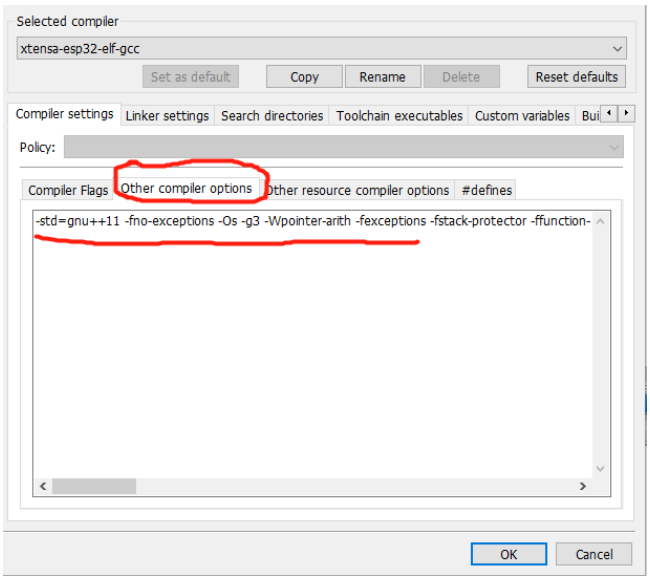
-lgcc -lopenssl -lbtdm_app -lfatfs -lwps -lcoexist -lwear_levelling -lhal -lnewlib -ldriver -lbootloader_support -lpp -lmesh -lsmartconfig -ljsmn -lwpa -lethernet -lphy -lapp_trace -lconsole -lulp -lwpa_supplicant -lfreertos -lbt -lmicro-ecc -lcxx -lxtensa-debug-module -lmdns -lvfs -lsoc -lcore -lsdmmc -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>

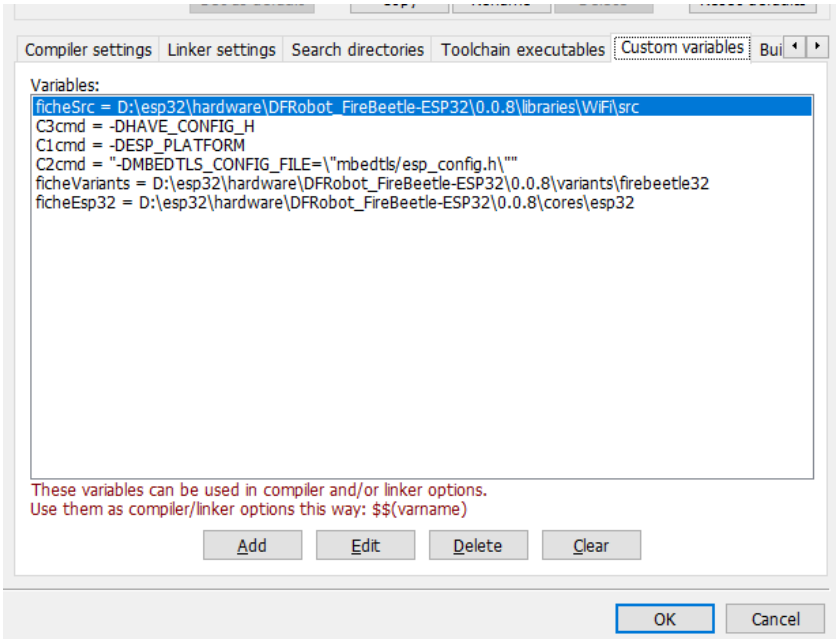
-std=gnu++11 -fno-exceptions -Os -g3 -Wpointer-arith -fexceptions -fstack-protector -ffunction-sections -fdata-sections -fstrict-volatile-bitfields -mlongcalls -nostdlib -w -Wno-error=unused-function -Wno-error=unused-but-set-variable -Wno-error=unused-variable -Wno-error=deprecated-declarations -Wno-unused-parameter -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



1.6 Ajouter des variables personnalisées

Settings—>compiler-->Custom variables

Ajouter les variables suivantes:

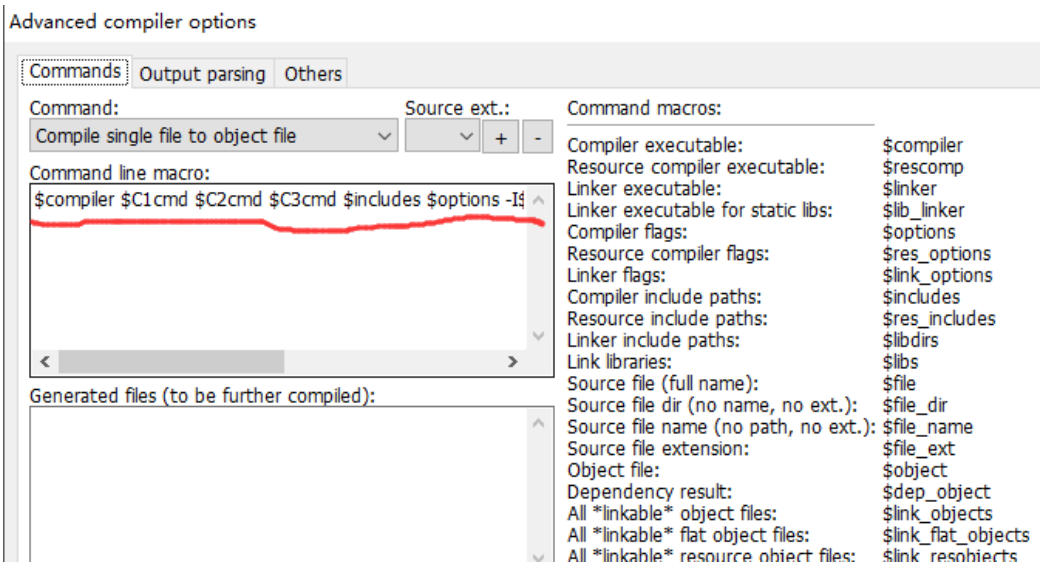


1.7 Modifier les instructions de compilation / lien

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

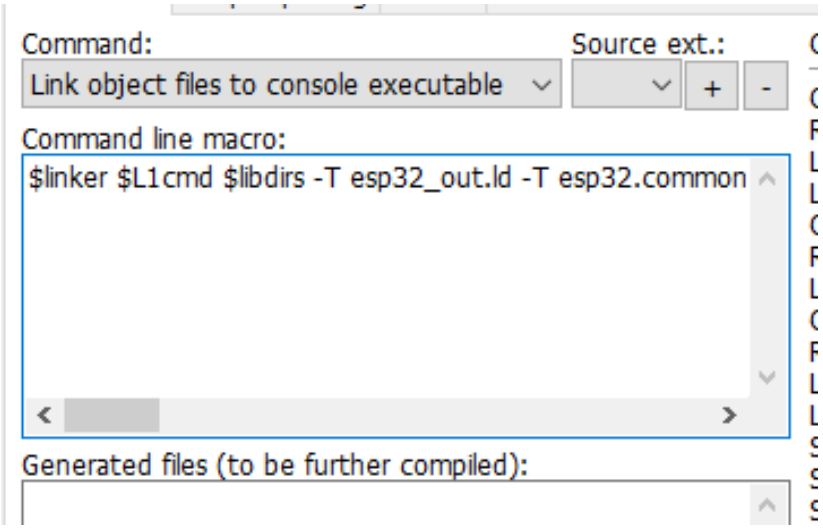
Commands->Compile single file to object file

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



Commands->Link object files to console executable

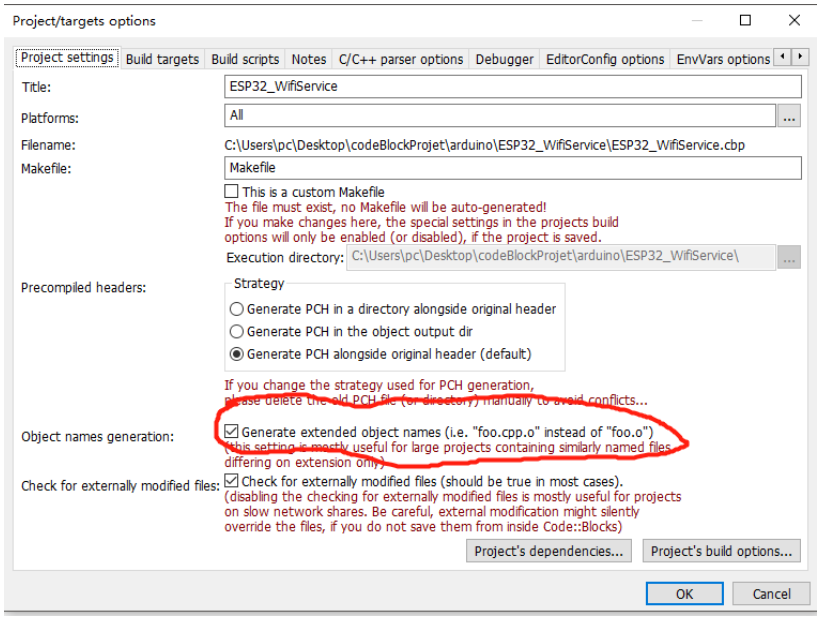
$linker $L1cmd $libdirs -T esp32_out.ld -T esp32.common.ld -T esp32.rom.ld -T esp32.peripherals.ld -T esp32.rom.spiram_incompatible_fns.ld -u ld_include_panic_highint_hdl -u call_user_start_cpu0 -Wl,--gc-sections -Wl,-static -Wl,--undefined=uxTopUsedPriority -u __cxa_guard_dummy -u __cxx_fatal_exception -Wl,--start-group $link_objects $link_resobjects $libs $link_options -o $exe_output



1.8 Définir les propriétés du projet

Project→properties

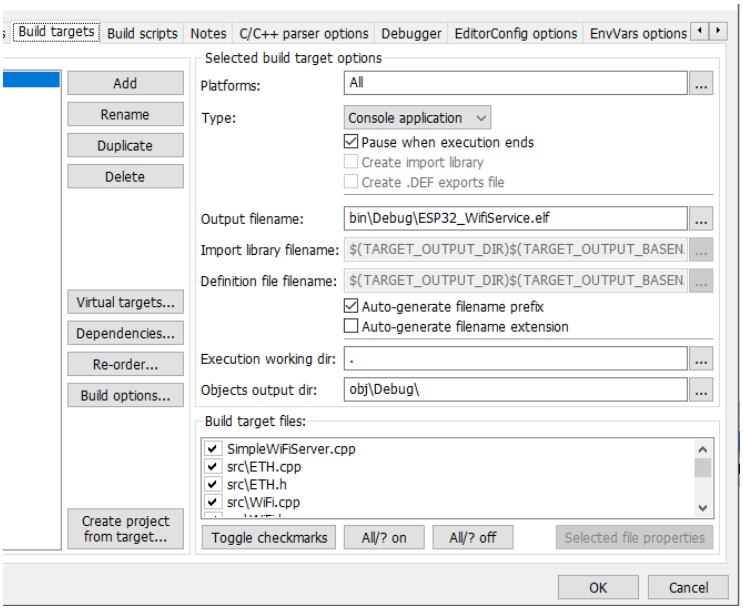
<Project settings>: Cocher Object names generation



<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

2 Démonstration de performance