

Options de compilation pour un projet STM32 de base

1. Liste d'options communes	1
-mcpu	1
-mthumb	1
-mfloat-abi	1
-mfpv	1
-fmessage-length	2
-fsigned-char	2
-ffunction-sections	2
-fdata-sections	2
-fno-move-loop-invariants	2
-Wall	3
-Wextra	3
-O0	3
-g3	3
2. Liste d'options pour g++	4
-std	4
-fno-exceptions	4
-fno-rtti	4
-fno-use-cxa-atexit	4
-fno-threadsafe-statics	4
3. Liste d'options pour gcc	5
-std	5
-ffreestanding	5
4. Liste d'options pour le linker	6
-nostartfiles	6
-Xlinker	6
-gc-sections	6
-L	6
-Wl	6
-specs	7
Annexes	8
A. Avertissements activés par -Wall	8
B. Avertissements activés par -Wextra	10

1. Liste d'options communes

-mcpu

Spécifie le nom du processeur ARM visé. GCC utilise ce paramètre pour déterminer l'architecture(-march) et le type(-mtune) du processeur à des fins d'optimisation.

"-mcpu=cortex-m4" onfigure le processeur comme étant un Cortex-M4.

-mthumb

Détermine le jeu d'instructions utilisé entre ARM(-marm) et Thumb(-mthumb).

-mfloat-abi

Détermine l'interface binaire-programme pour les nombres à point flottant.

Les valeurs possibles sont : "soft", "softfp" et "hard".

Ex : "-mfloat-abi=hard"

-mfpv

Détermine le périphérique de traitement des nombres à point flottant.

Les valeurs possibles sont : "vfpv2", "vfpv3", "vfpv3-fp16", "vfpv3-d16", "vfpv3-d16-fp16", "vfpv3xd", "vfpv3xd-fp16", "neon-vfpv3", "neon-fp16", "vfpv4", "vfpv4-d16", "fpv4-sp-d16", "neon-vfpv4", "fpv5-d16", "fpv5-sp-d16", "fp-armv8", "neon-fp-armv8" et "crypto-neon-fp-armv8".

Ex : "-mfpv=fpv4-sp-d16" pour les cortex-m4.

-fmessage-length

"-fmessage-length=*n*"

Formate les messages d'erreur pour former des lignes de *n* caractères. Si *n*=0 alors les messages seront affichés sur une seule ligne.

-fsigned-char

Permet au type char d'être signé.

-ffunction-sections

Option d'optimisation qui place chaque fonction dans sa propre section dans le fichier de sortie. Le nom de la fonction détermine le nom de la section.

Des problèmes de debug peuvent survenir si cette option est utilisée en même temps que l'option -g.

-fdata-sections

Option d'optimisation qui place chaque objet de données dans sa propre section dans le fichier de sortie. Le nom de l'objet de données détermine le nom de la section.

Des problèmes de debug peuvent survenir si cette option est utilisée en même temps que l'option -g.

-fno-move-loop-invariants

Option qui prévient l'optimisation des variables sans variation dans les boucles. Cette option facilite le debug.

Cette option est ajoutée automatiquement au profil d'optimisation -Og.

-Wall

Option qui active les avertissements jugés discutables et qui sont facilement corrigés ou évités. Voir la fin du document pour la liste des avertissements activés.

-Wextra

Option qui active des avertissements supplémentaires non ajoutés par "-Wall". Voir la fin du document pour la liste des avertissements activés.

-O0

Profil d'optimisation qui réduit le temps de compilation et ne fait aucune optimisation.

-g3

-g^{lvl}

Spécifie le niveau de détail dans les informations disponibles lors du debug. Le niveau 3 inclut des informations supplémentaires comme la définitions des macros.

Les valeurs possible de "lvl" vont de 0 à 3, 3 étant le niveau le plus détaillé.

2. Liste d'options pour g++

-std

Détermine la version du standard pour le langage de programmation.

Les standards sont disponibles dans la version de base ou la version en GNU du dialecte.

EX : "c++98", "c++11", "c++14", "gnu++98", "gnu++11" ou "gnu++14".

-fno-exceptions

Désactive la gestion des exceptions. Les exceptions sont, par défaut, activées en C++ et désactivées en C.

-fno-rtti

Désactive la génération des "run-time type information". Cette option affecte l'utilisation de `dynamic_cast` et de l'opérateur `typeid`.

-fno-use-cxa-atexit

Spécifie l'utilisation de la fonction `atexit` plutôt que `__cxa_atexit` pour enregistrer le destructeur des objets statiques. Cette option va à l'encontre du standard C++ concernant l'ordre d'appel des destructeurs des objets statiques. (Voir <http://mentorembedded.github.io/cxx-abi/abi.html#dso-dtor>)

-fno-threadsafe-statics

Désactive l'ajout de code pour rendre l'initialisation de variables locales statiques "thread-safe".

3. Liste d'options pour gcc

-std

Détermine la version du standard pour le langage de programmation.

Les standards sont disponibles dans la version de base ou la version en GNU du dialecte.

EX : "c99", "c11", "gnu99" ou "gnu11".

-ffreestanding

Spécifie que la librairie standard peut être inexistante dans le système visé et que le point de départ du programme n'est pas nécessairement la fonction `main()`. Équivalent à `"-fno-hosted"`.

4. Liste d'options pour le linker

-nostartfiles

Prévient l'utilisation des fichiers de démarrage standards par l'éditeur de liens.

-Xlinker

-Xlinker option

Permet de passer une option à l'éditeur de liens qui n'est pas reconnue par GCC.

-gc-sections

Active le récupérateur de mémoire(garbage collector) pour omettre les sections inutilisées.

Utilisation : `-Xlinker -gc-sections`

-L

-Lsearchdir

Ajoute le chemin d'accès *searchdir* à la recherche des librairies et des "linker scripts".

-Wl

-Wl,option

Permet de passer une option à l'éditeur de liens. Si l'option contient des virgules, l'option sera séparée en plusieurs options. Il est possible d'utiliser cette fonctionnalité pour passer des paramètres à l'option.

Ex : `"-Wl,-Map,output.map"` passera l'option `"-Map output.map"` à l'éditeur de liens.

-specs

`-specs=file.specs`

GCC effectue son travail en invoquant une séquence de programmes dans un ordre précis. Ce comportement est contrôlé par les "spec strings". Les "spec strings" par défaut peuvent être modifiées grâce à la commande `-specs` en spécifiant un nouveau fichier ".specs".

Annexes

A. Avertissements activés par -Wall

- Waddress
- Warray-bounds=1 (only with -O2)
- Wbool-compare
- Wbool-operation
- Wc++11-compat -Wc++14-compat
- Wchar-subscripts
- Wcomment
- Wduplicate-decl-specifier (C and Objective-C only)
- Wenum-compare (in C/ObjC; this is on by default in C++)
- Wformat
- Wint-in-bool-context
- Wimplicit (C and Objective-C only)
- Wimplicit-int (C and Objective-C only)
- Wimplicit-function-declaration (C and Objective-C only)
- Winit-self (only for C++)
- Wlogical-not-parentheses
- Wmain (only for C/ObjC and unless -ffreestanding)
- Wmaybe-uninitialized
- Wmemset-elt-size
- Wmemset-transposed-args
- Wmisleading-indentation (only for C/C++)
- Wmissing-braces (only for C/ObjC)
- Wnarrowing (only for C++)
- Wnonnull
- Wnonnull-compare
- Wopenmp-simd
- Wparentheses
- Wpointer-sign
- Wreorder
- Wreturn-type
- Wsequence-point
- Wsign-compare (only in C++)
- Wsizeof-pointer-memaccess
- Wstrict-aliasing

- Wstrict-overflow=1
- Wswitch
- Wtautological-compare
- Wtrigraphs
- Wuninitialized
- Wunknown-pragmas
- Wunused-function
- Wunused-label
- Wunused-value
- Wunused-variable
- Wvolatile-register-var

B. Avertissements activés par -Wextra

- Wclobbered
- Wempty-body
- Wignored-qualifiers
- Wimplicit-fallthrough=3
- Wmissing-field-initializers
- Wmissing-parameter-type (C only)
- Wold-style-declaration (C only)
- Woverride-init
- Wsign-compare (C only)
- Wtype-limits
- Wuninitialized
- Wshift-negative-value (in C++03 and in C99 and newer)
- Wunused-parameter (only with -Wunused or -Wall)
- Wunused-but-set-parameter (only with -Wunused or -Wall)