

Nauteff P-1 Chaînes de compilation

Emmanuel Gautier

9 novembre 2025

Résumé

Ce document contient les informations permettant la génération des codes du prototype Nauteff P-1 et des documentations.

Table des matières

1	Buts du document	2
2	Documents applicables et de référence	3
3	Terminologie	4
4	Environnement de développement	5
5	Outils de développement	6
5.1	Compilateur et éditeur de liens arm-none-eabi-gcc	6
5.1.1	Installation	6
5.2	EDI eclipse	6
5.3	openocd	7
6	Génération du projet	8

Chapitre 1

Buts du document

Ce document contient les informations permettant d'installer, de configurer et utiliser les outils de développement. Ces outils sont la chaîne de compilation croisée GCC, eclipse et openocd.

Chapitre 2

Documents applicables et de référence

Chapitre 3

Terminologie

Chapitre 4

Environnement de développement

Le développement est réalisé avec des PC fonctionnant sous Linux/debian. Au bureau il est réalisé avec les matériels suivant :

- un PC permettant de faire fonctionner eclipse et la chaîne de compilation ;
- un oscilloscope pour certaines mise au point ;
- occasionnellement un voltmètre ;
- petit outillage : pinces tournevis, pince à dénuder,...

Lors des essais en mer du prototype un petit PC consommant peu mais ne permettant pas d'utiliser eclipse permet de faire des modifications du logiciel et, en particulier, des ajustements de paramètres. Il embarque la chaîne de compilation et openocd. La compilation est réalisée avec la commande make.

OS	:	Debian Buster / amd64
eclipse	:	2021 mars
	:	
	:	
	:	

Chapitre 5

Outils de développement

5.1 Compilateur et éditeur de liens arm-none-eabi-gcc

5.1.1 Installation

L'installation est faite avec apt-get ou synaptic ou autre. Il faut les paquets suivants :

binutils-arm-none-eabi	assembleur, éditeur de liens et outils binaires GNU pour les processeurs ARM Cortex-A/R/M
gcc-arm-none-eabi	GCC cross compiler for ARM Cortex-A/R/M processors
gdb-arm-none-eabi	GNU debugger for ARM Cortex-A/R/M processors
libstdc++-arm-none-eabi-newlib	GNU Standard C++ Library v3 for ARM Cortex-A/R/M processors (newlib)
libnewlib-arm-none-eabi	C library and math library compiled for bare metal using Cortex A/R/M

5.2 EDI eclipse

l'EDI est Eclipse C++ téléchargeable sur le site eclipse.org.

Choisir "Eclipse IDE for Embedded C/C++ developers".

Plugins : TexLipse

Ajouter le plugin C/C++ GDB Hardware Debugging Developer Resources. Ce plugin est accessible a <http://download.eclipse.org/tools/cdt/releases/NN.MM>.
il faut utiliser le filtre gdb

Dans Eclipse marketplace demandes openocd et choisir GNU MCU Eclipse

5.3 openocd

Le logiciel qui permet d'accéder à la sonde est openocd version 0.9.0.

Chapitre 6

Génération du projet

<File>, <New project>, <C/C++ Project>, <C managed build>, <Debug> & <Release>,

Project Name : NauteffAutopilot, <ARM Cross GCC>

Chip Family	STM30x/31x
Flash size	64k
RAM size	12k
CCM size	4k
External clock	8000000
Content	Empty
Use system calls	Freestanding (no Posix system calls)
Trace output	ARM ITM (SWIO) pour essayer
Check some & most warnings	Oui
Enable -Werror	Non
Use -Og on debug	Oui
Use newlib nano	Oui
Exclude unused	Oui
Use link optimizations	Non
Include folder	inc
Source folder	src
System folder	system
CMSIS Library folder	cmsis
C library folder	newlib
linker script folder	ldscripts
Select configuration	Debug & release

Dans

<C/C++> <Build> <Tools settings>

ARM family	cortex-m4
Instruction set	Thumb
Float ABI	FP instructions (hard)

Créer les répertoires src et inc avec <File> <New Folder>

Dans la vue Project explorer sélectionner le projet puis <Project>, <Properties> <C/C++ Build>, <Settings> <Cross Settings> Prefix : arm-none-eabi- Path /usr/bin <Cross GCC Compiler> All options : "-O0 -g3 -Wall -c -fmessage-length=0"

Commande de compilation : arm-none-eabi-gcc -g3 -Wall -fno-math-errno -mcpu=cortex-m4 -mthumb -mfloat-abi=hard -fsingle-precision-constant -mfpv4-sp-d16 -specs=nosys.specs -specs=nano.specs -linclude -lm -lg -lc -Tnauteff.ld -Xlinker -Map -Xlinker nauteff.map