

Projet De Stijl

Plateforme pour robots mobiles

Guide de prise en main d'Eclipse pour programmer la Raspberry Pi

Version : 31 mai 2017

Julia Cohen
Julien Combattelli
Solène Dothée
Guillaume Sarthou

Table des matières

1.	Installation d'un cross-compileur pour Raspberry Pi.....	1
	Sous Linux.....	1
	Sous Windows.....	1
2.	Installation de Eclipse CDT.....	2
3.	Trouver l'adresse IP de la Raspberry	2
	Sous Linux.....	2
	Sous Windows.....	2
4.	Configuration du projet Destijl	3
	Choix du workspace	3
	Création du projet	4
	Création du dépôt Git.....	5
	Configuration du dépôt	6
	Configuration du projet.....	8
	Compilation et débogage du programme	11

1. Installation d'un cross-compileur pour Raspberry Pi

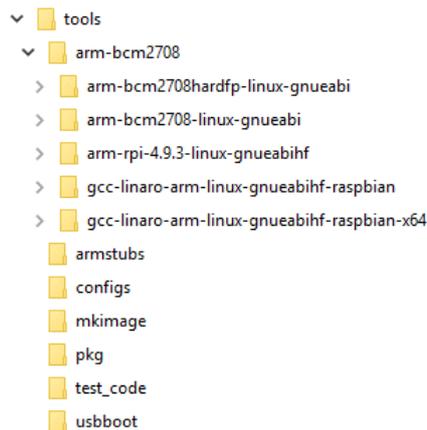
Afin de compiler une application pour Raspberry Pi sur un PC, il est nécessaire d'avoir un cross-compileur. Suivant si vous êtes sur Linux ou Windows, la procédure d'installation ne sera pas la même.

Sous Linux

Dans un terminal, exécutez la commande suivante :

```
git clone https://github.com/raspberrypi/tools.git
```

L'architecture du dossier est la suivante :



Vous pouvez copier le dossier « tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian » là où vous le souhaitez

Remarque : pour un ordinateur 64bits, prenez plutôt le dossier « gcc-linaro-arm-linux-gnueabihf-raspbian-x64 »

Les exécutables du compilateur se trouvent dans « gcc-linaro-arm-linux-gnueabihf-raspbian/bin »

Sous Windows

Allez sur <http://gnutoolchains.com/raspberry/>

Téléchargez et installez raspberry-gcc-4.9.2-r4.exe comme indiqué sur l'écran ci-dessous :

- Toolchains ▲
- Android (GDB only)
- Raspberry Pi
- arm-eabi
- arm-elf
- avr
- beaglebone
- bfin-elf
- cubieboard
- esp32
- esp8266
- m32c-elf
- m32r-elf
- m68k-elf
- mingw32
- mingw64
- msp430
- powerpc-eabi
- raspberry
- v850-elf

Windows toolchain for Raspberry/PI

Raspberry PI is a low-cost embedded board running Debian-based GNU/Linux. This page provides a complete toolchain for building and debugging Raspberry PI applications.

The toolchain includes the following components:

- The GCC compiler for C and C++ languages
- The GDB debugger
- Include files and libraries from the Raspberry PI image
- WinFLASHTool - a tool for writing disk images to SD cards
- SmartTTY - a portable SSH client with support for copying files to remote machine

The following releases are available:

GCC	Compatible Raspbian image	Toolchain download link
4.9.2	2016-09-23-raspbian-jessie (Raspberry Pi 1/2/3/Zero)	raspberry-gcc-4.9.2-r4.exe (738 MB)
4.9.2	2015-11-21-raspbian-jessie	raspberry-gcc-4.9.2-r2.exe (433 MB)
4.6.3	Universal (sysroot update required)	raspberry-gcc4.6.3-nosysroot.exe (18 MB)
4.6.3	2013-07-26-wheezy-raspbian	raspberry-gcc4.6.3.exe (111 MB)

2. Installation de Eclipse CDT

Eclipse est téléchargeable à l'adresse <https://www.eclipse.org/downloads/eclipse-packages/>.

Choisissez le package « Eclipse IDE for C/C++ Developers » et téléchargez la version adaptée à votre système d'exploitation.

Il est conseillé de prendre Eclipse Mars ou une version plus récente, pour bénéficier du plugin Git intégré.

3. Trouver l'adresse IP de la Raspberry

Le moyen le plus simple pour trouver son adresse et de la fixer :

- Prenez la carte SD de la Raspberry et insérer-la dans un ordinateur
- Sur la carte SD, dans le fichier cmdline.txt, ajouter ip=<adresseIP> à la fin de la ligne en remplaçant <adresseIP> par l'adresse désirée (exemple : 192.168.0.2)

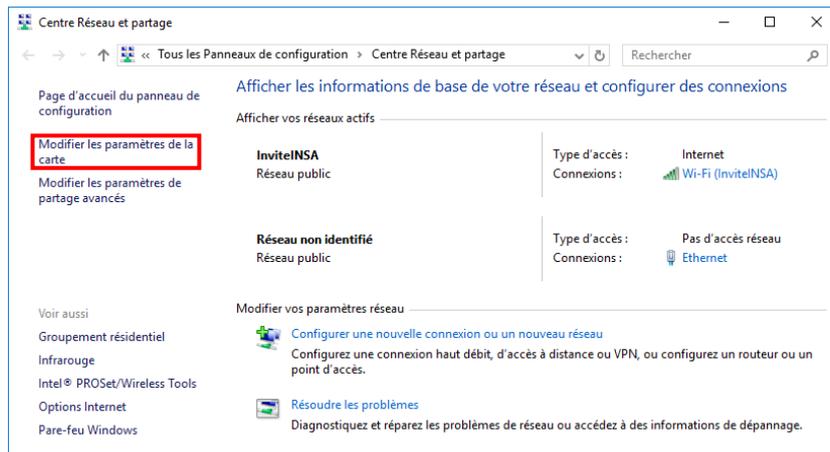
En fixant l'IP de la Raspberry, vous aurez aussi à fixer l'IP de la carte Ethernet de votre ordinateur comme indiqué ci-dessous (pensez à pinger la Raspberry Pi à la fin de la manipulation pour s'assurer que la configuration est correcte) :

Sous Linux

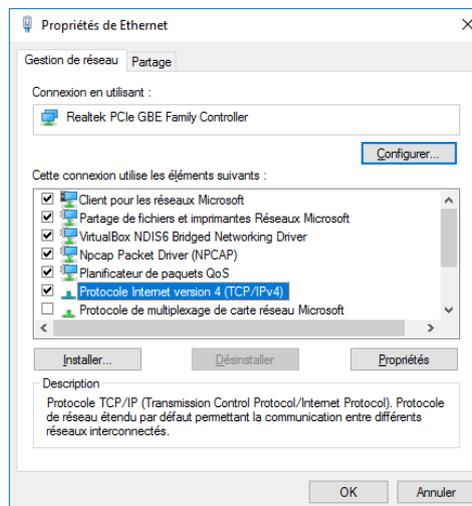
- Configurer l'IP avec `ifconfig` ou en modifiant le fichier « `/etc/network/interfaces` »

Sous Windows

- Dans le panneau de configuration, ouvrez le Centre réseau et partage
- Cliquez sur « Modifier les paramètres de la carte » :



- Dans la liste des cartes réseaux, cliquez-droit sur votre carte Ethernet, puis cliquez sur « Propriétés »
- Dans la liste des propriétés, double-cliquez sur « Protocole Internet Version 4 », comme ci-dessous :

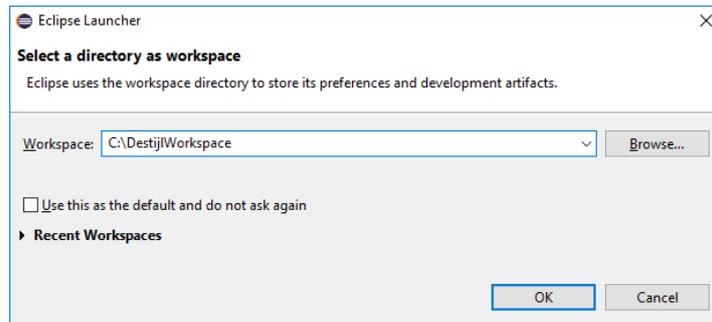


- Remplir les champs de la nouvelle fenêtre comme suit :
 - Cochez « Utiliser l'adresse IP suivante »
 - Remplir le champ adresse IP avec la valeur souhaitée (exemple : 192.168.0.1)
 - Remplir le champ masque de sous-réseau en fonction de votre adresse IP (exemple : pour une adresse en 192.168.X.X, mettre 255.255.255.0)
 - Appuyez sur Ok

4. Configuration du projet Destijl

Choix du workspace

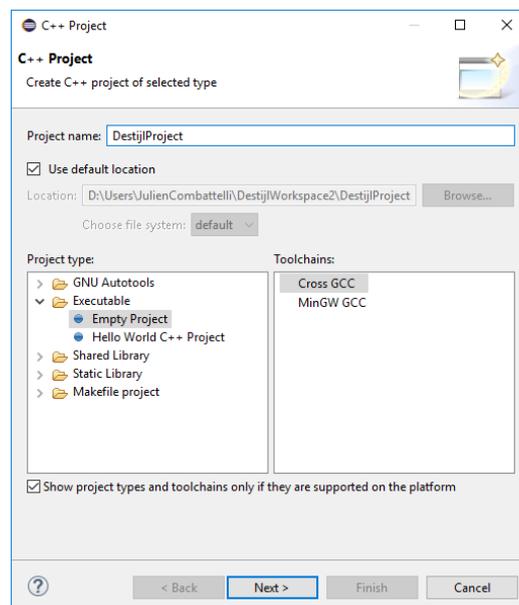
Au démarrage de Eclipse, sélectionnez le workspace de votre choix, c'est là que seront stockés vos projets (exemple : C:\DestijlWorkspace) comme illustré dans l'écran suivant :



Création du projet

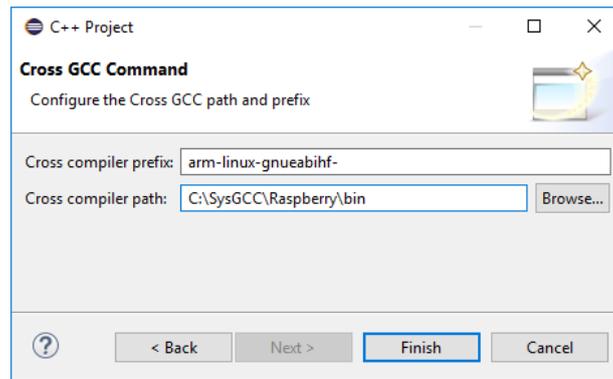
- Créez un nouveau projet C++ en cliquant sur File > New > C++ Project.
- Dans la fenêtre « C++ Project » :
 - Saisissez le nom de votre choix (exemple : DestijProject)
 - Dans « Project type », choisissez « Empty Project »
 - Dans « Toolchains » choisissez « Cross GCC »

Votre fenêtre devrait ressembler à ceci :

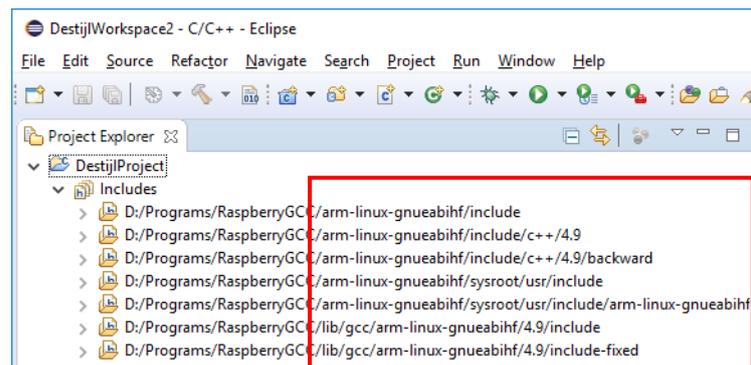


- Cliquez sur Next
- Dans la fenêtre « Select Configurations », ne changez rien et cliquez sur « Next »
- Dans la fenêtre « Cross GCC Command » :
 - Dans le champs « Cross compiler prefix », mettre « arm-linux-gnueabi- »
 - Dans le champs « Cross compiler path », mettre le chemin vers le compilateur que vous avez installé précédemment

Voici un exemple :

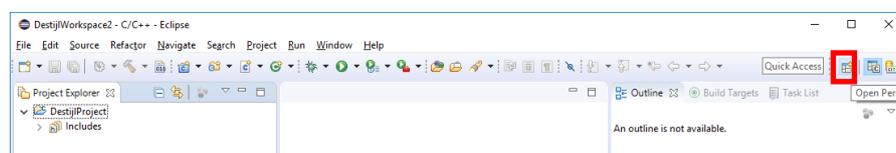


- Si le compilateur a bien été trouvé, vous devriez avoir les répertoires d'inclusion suivants (les chemins peuvent différer suivant où vous avez installé le compilateur) :

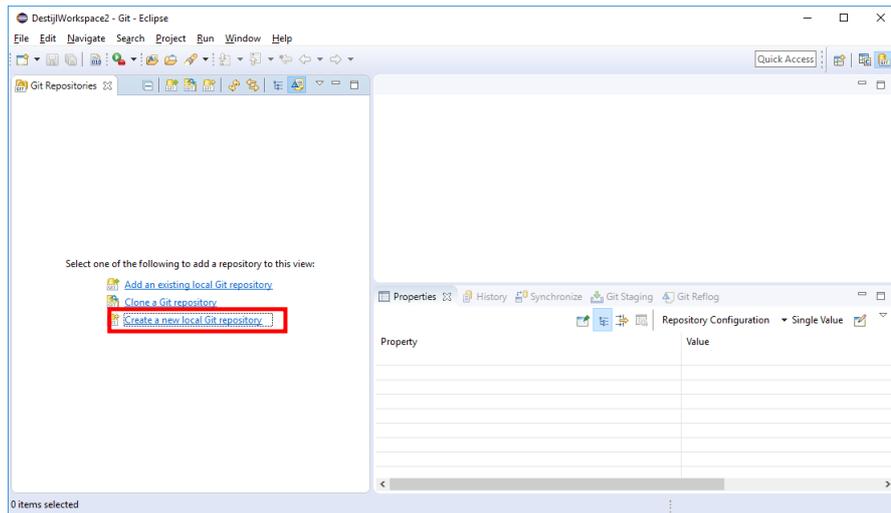


Création du dépôt Git

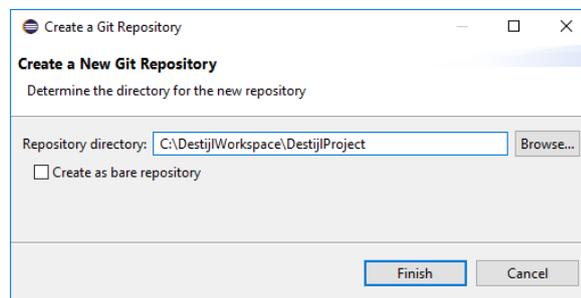
- Ouvrez ensuite la perspective Git. Pour cela, cliquez sur le bouton indiqué dans la capture suivante :



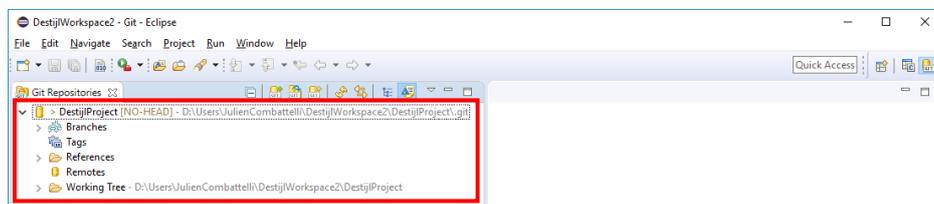
- Dans la fenêtre « Open Perspective », sélectionnez « Git »
- Dans l'onglet « Git Repositories », sélectionnez « Create a new local Git repository », comme illustré ci-dessous :



- Dans le champ « Repository directory », renseignez le chemin vers le projet que vous avez créé (exemple : C:\DestijlWorkspace\DestijlProject), si un avertissement « Directory [...] is not empty » est signalé, ignorez-le.

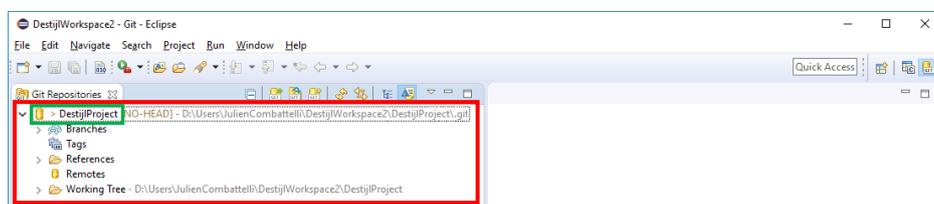


- Cliquez sur Finish
- Un nouveau dépôt devrait apparaître dans Eclipse comme suit :

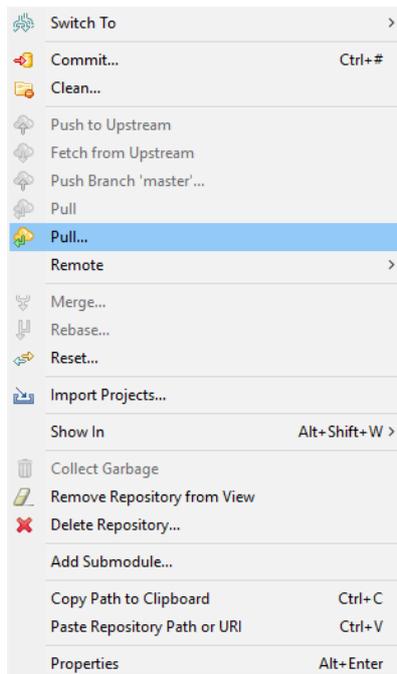


Configuration du dépôt

- Cliquez-droit sur le dépôt (encadré en vert ci-dessous) :



- Dans le menu qui apparaît, cliquez sur « Pull... »



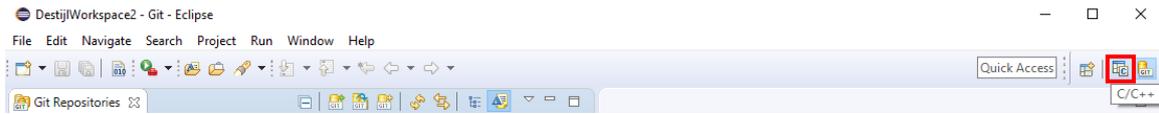
- Dans la fenêtre qui apparait, remplissez les éléments comme dans la fenêtre suivante (les champs de la section « Authentication » ne sont pas obligatoire) :

A screenshot of a 'Pull' dialog box. The title bar says 'Pull'. The main heading is 'Destination Git Repository' with the instruction 'Enter the location of the destination repository.' Below this, there are several sections: 'Remote name:' with a text box containing 'origin'; 'Location' with 'URI:' (https://github.com/juliencombattelli/Destijl), 'Host:' (github.com), and 'Repository path:' (/juliencombattelli/Destijl); 'Connection' with 'Protocol:' (https) and 'Port:'; and 'Authentication' with 'User:' and 'Password:' text boxes, and a checkbox for 'Store in Secure Store'. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

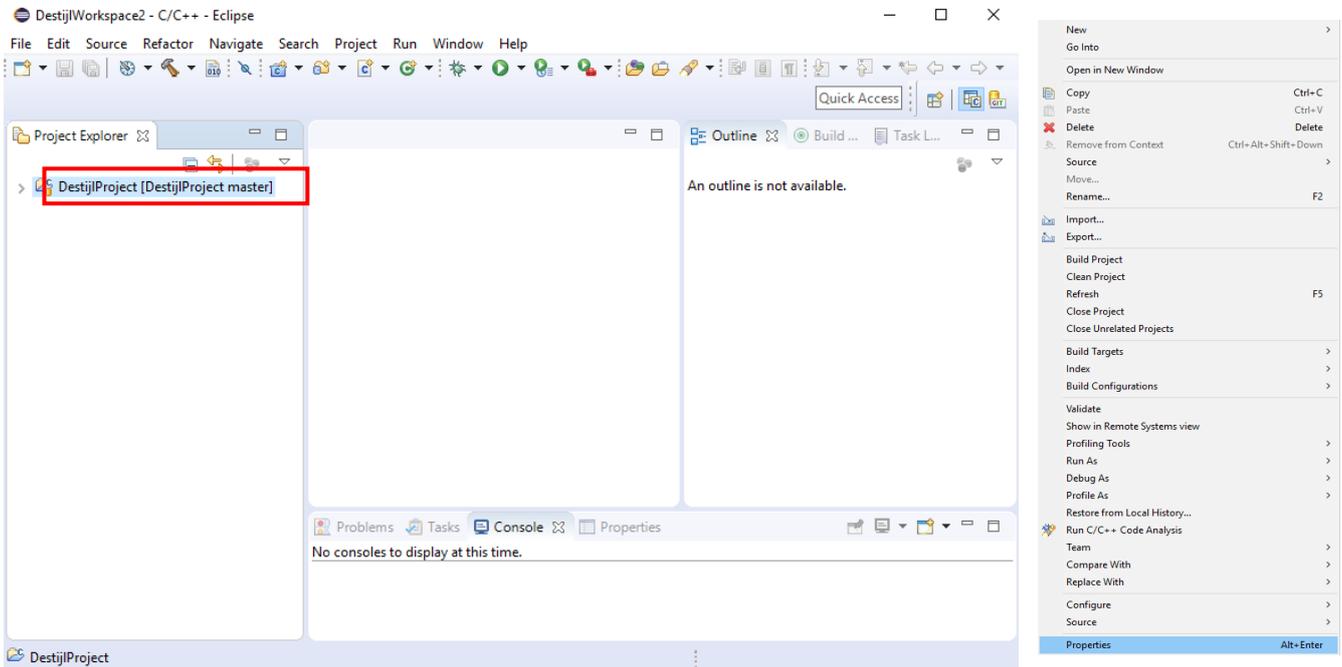
- Cliquez sur Next
- Dans la fenêtre « Pull », cliquez sur « New Remote » et saisissez à nouveau les champs comme ci-dessus (s'ils ne sont pas déjà remplis) puis cliquez sur Finish
- Dans le champ « Reference », saisissez « refs/heads/master » et cochez l'option « Configure upstream for push and pull »

Configuration du projet

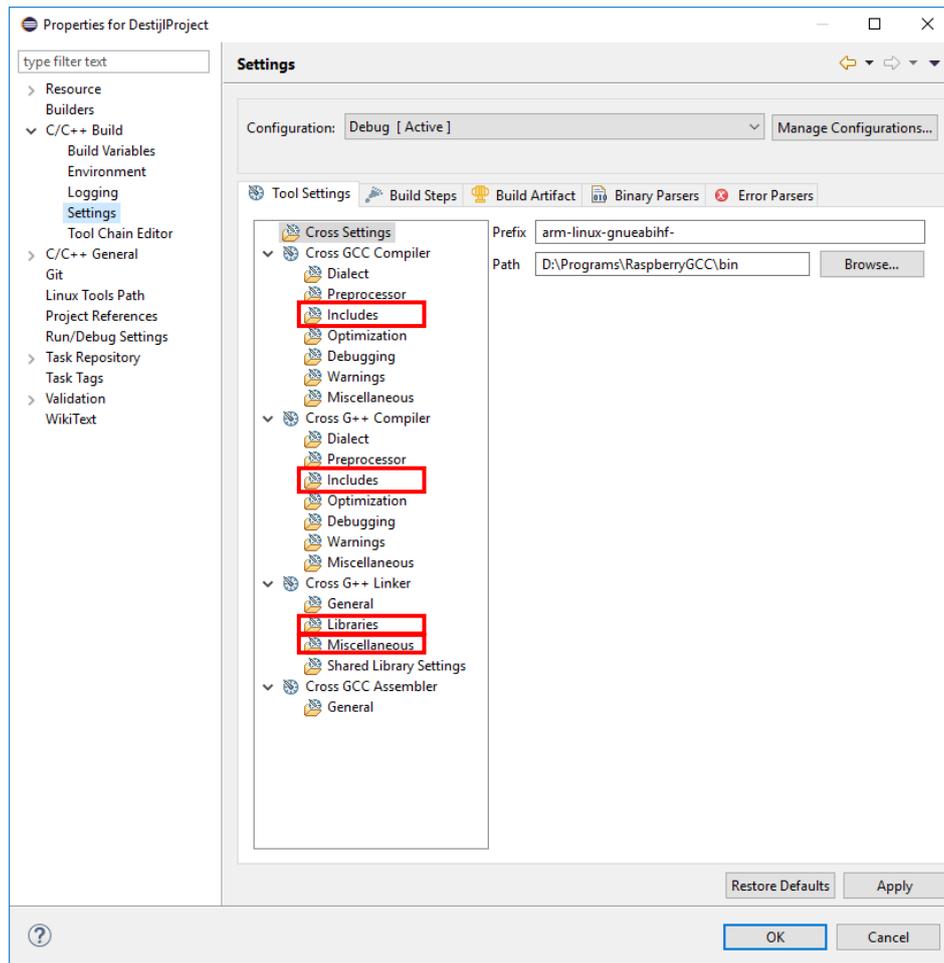
- Retournez sur la perspective C/C++ en cliquant sur le bouton encadré ci-dessous :



- Cliquez-droit sur le projet et allez dans « Properties », comme illustré ci-dessous :



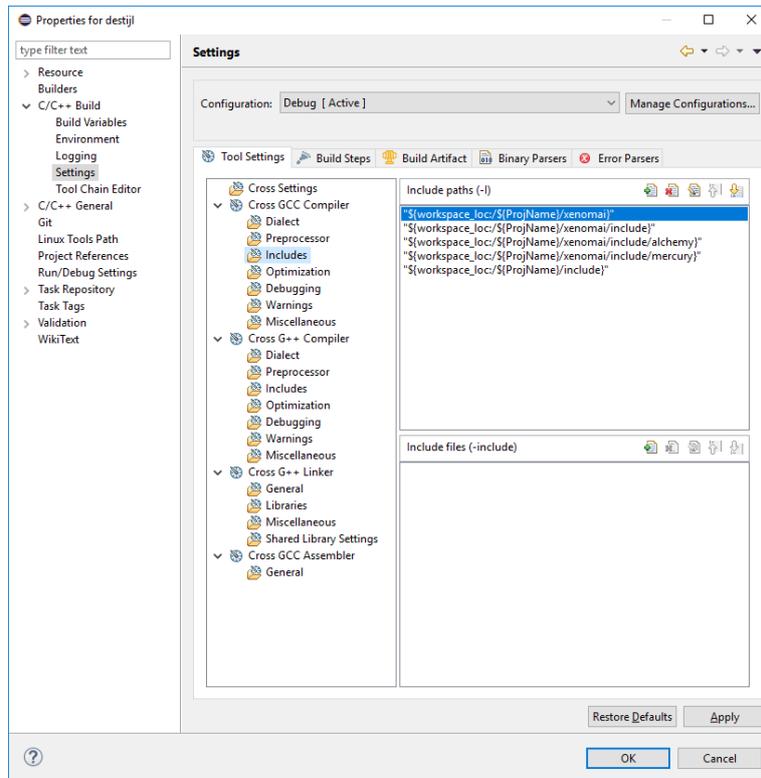
- Allez dans « C/C++ Build > Settings », puis dans l'onglet « Tool settings » (les pages à modifier sont encadrées en rouge):



- Allez dans « Cross GCC Compiler > Includes » et copiez-collez les lignes suivantes dans « Include paths (-I) » :

```
"${workspace_loc}/${ProjName}/xenomai"
"${workspace_loc}/${ProjName}/xenomai/include"
"${workspace_loc}/${ProjName}/xenomai/include/alchemy"
"${workspace_loc}/${ProjName}/xenomai/include/mercury"
"${workspace_loc}/${ProjName}/include"
```

Vous devriez avoir une fenêtre similaire :



- Effectuez la même manipulation pour « Cross G++ Compiler > Includes »

- Dans « Cross G++ Linker > Librairies » :

- Dans « Librairies (-l) », copiez-collez les lignes suivantes :

```

opencv_core
pthread
opencv_highgui
opencv_imgproc
opencv_imgcodecs
raspicam_cv
copperplate
mercury
alchemy

```

- Puis dans « Library search path (-L) » copiez-collez les lignes suivantes :

```

"${workspace_loc}/${ProjName}/lib"
"${workspace_loc}/${ProjName}/xenomai/lib"

```

- Dans « Cross G++ Linker > Miscellaneous » :

- Dans le champ « Linker flags », copiez :

```

-Wl,-rpath="${workspace_loc}/${ProjName}/lib" -Wl,-rpath="${workspace_loc}/${ProjName}/xenomai/lib"

```

- Puis dans « Other objects » ajoutez :

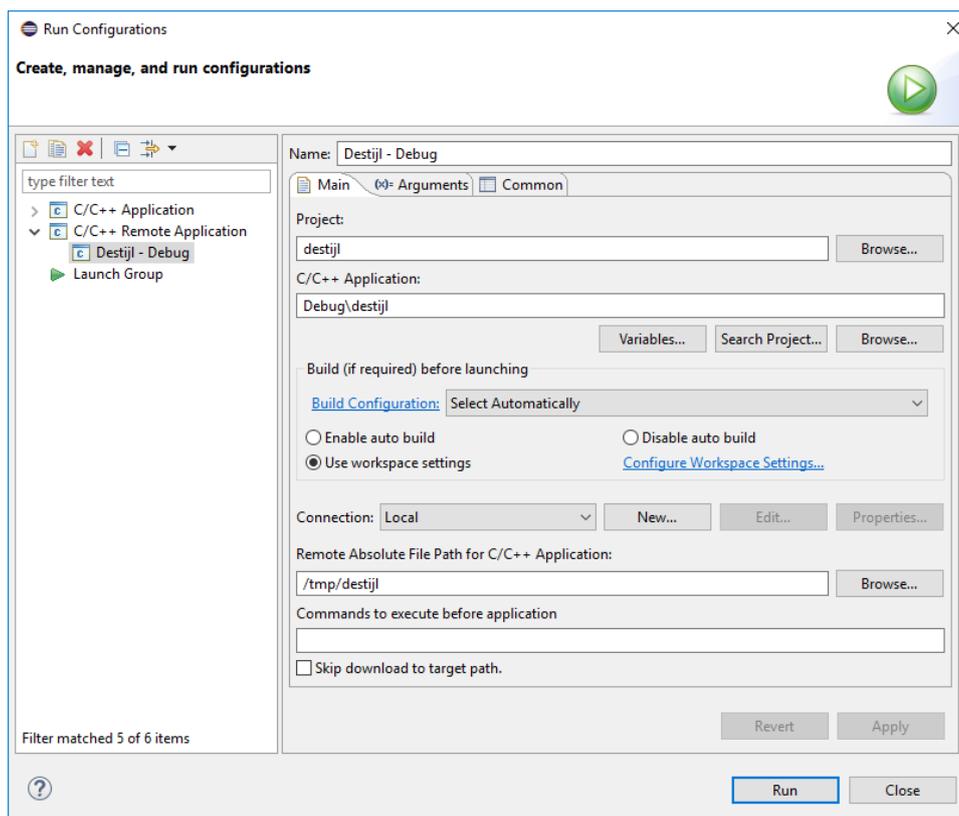
```

"${workspace_loc}/${ProjName}/xenomai/lib/xenomai/bootstrap.o"

```

Compilation et débogage du programme

- Pour compiler le projet, appuyez sur le marteau dans la barre d'outils.
- Dans la barre de menu, cliquez sur Run puis sur « Run Configuration » (ou « Debug Configuration » si vous souhaitez déboguer le programme).
- Remplissez la fenêtre comme suit :
 - Dans le champs « Project », mettre le nom de votre projet
 - Dans « C/C++ Application », mettre l'exécutable avec le chemin pour y accéder
 - Dans « Remote Absolute File Path », mettre où sera situé l'exécutable sur la RPi (si vous n'êtes pas connecté en super utilisateur, cette espace doit être accessible à un utilisateur normal)



- Dans cette même fenêtre, cliquez sur « New... » pour créer une nouvelle connexion
- Choisissez une connexion SSH et appuyez sur Ok
- Dans la fenêtre « New Connection » :
 - Dans le champs « Host », mettre l'adresse IP de la Raspberry Pi
 - Dans « User », mettre le nom de la session sur laquelle se connecter (exemple : « pi » ou « root »)
 - Cocher l'option « Password based authentication »
 - Dans le champs password, mettre le mot de passe de la session (exemple : « raspberry » pour la session pi)

Vous devriez avoir une fenêtre similaire :

New Connection
Specify properties of a new connection

Connection name: Remote Host

Host information

Host: 192.168.0.1

User: pi

Public key based authentication Keys are set at [Network Connections, SSH2](#)

Passphrase:

Password based authentication

Password: ●●●●●●●●

▶ Advanced

Finish Cancel

- Appuyer sur « Run » pour lancer le téléchargement de l'exécutable sur la Raspberry et exécuter le programme.