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-compilateur pour Raspberry Pi 1](#_Toc484012529)

[Sous Linux 1](#_Toc484012530)

[Sous Windows 1](#_Toc484012531)

[2. Installation de Eclipse CDT 2](#_Toc484012532)

[3. Trouver l’adresse IP de la Raspberry 2](#_Toc484012533)

[Sous Linux 2](#_Toc484012534)

[Sous Windows 2](#_Toc484012535)

[4. Configuration du projet Destijl 3](#_Toc484012536)

[Choix du workspace 3](#_Toc484012537)

[Création du projet 4](#_Toc484012538)

[Création du dépôt Git 5](#_Toc484012539)

[Configuration du dépôt 6](#_Toc484012540)

[Configuration du projet 8](#_Toc484012541)

[Compilation et débogage du programme 11](#_Toc484012542)

# Installation d’un cross-compilateur pour Raspberry Pi

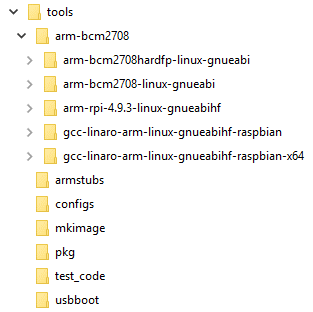
Afin de compiler une application pour Raspberry Pi sur un PC, il est nécessaire d’avoir un cross-compilateur. Suivant si vous êtes sur Linux ou Windows, la procédure d’installation ne sera pas la même.

## 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

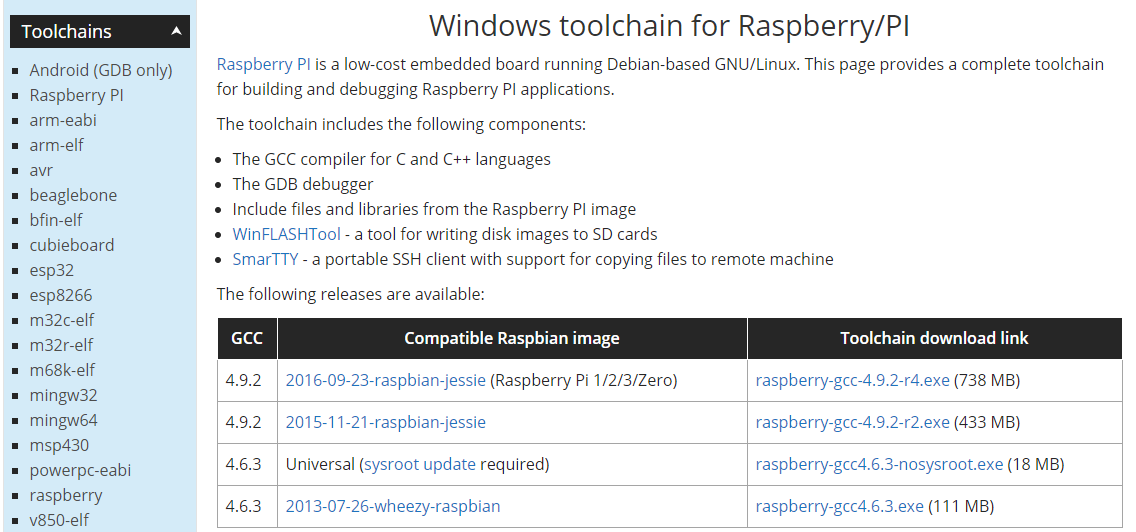
Remarque : pour un ordinateur 64bits, prenez plutôt le dossier « gcc-linaro-arm-linux-gnueabihf-raspbian-x64 »

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 :



# 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.

# 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)

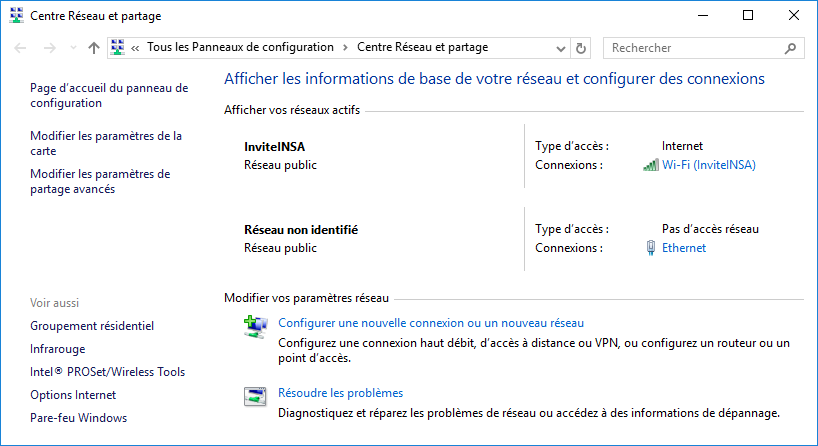
En fixant l’IP de la Raspberry, vous aurez aussi à fixer l’IP de la carte Ethernet de votre ordinateur comme indiqué ci-dessous (pensez à pinger la Raspberry Pi à la fin de la manipulation pour s’assurer que la configuration est correcte) :

## Sous Linux

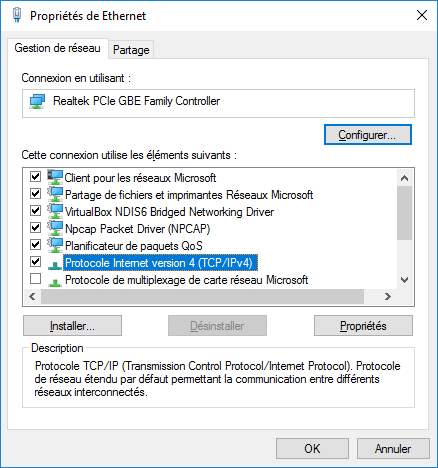
* Configurer l’IP avec ifconfig ou en modifiant le fichier « /etc/network/interfaces »

## 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 :

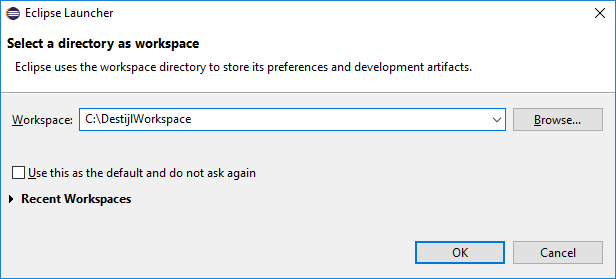


* Remplir les champs de la nouvelle fenêtre comme suit :
  + Cochez « Utiliser l’adresse IP suivante »
  + Remplir le champ adresse IP avec la valeur souhaitée (exemple : 192.168.0.1)
  + Remplir le champ masque de sous-réseau en fonction de votre adresse IP (exemple : pour une adresse en 192.168.X.X,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

# Configuration du projet Destijl

## Choix du workspace

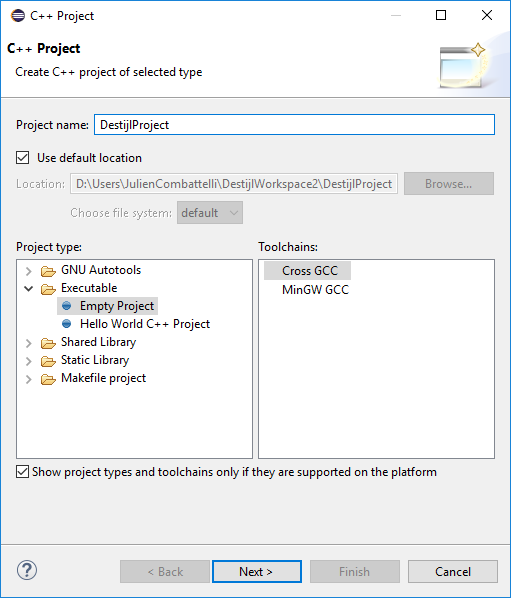
Au démarrage de Eclipse, sélectionnez le workspace de votre choix, c’est là que seront stockés vos projets (exemple : C:\DestijlWorkspace) comme illustré dans l’écran suivant :



## Création du projet

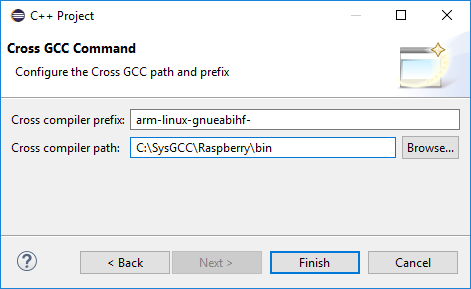
* Créez un nouveau projet C++ en cliquant sur File > New > C++ Project.
* Dans la fenêtre « C++ Project » :
* Saisissez le nom de votre choix (exemple : DestijlProject)
* Dans « Project type », choisissez « Empty Project »
* Dans « Toolchains » choisissez « Cross GCC »

Votre fenêtre devrait ressembler à ceci :

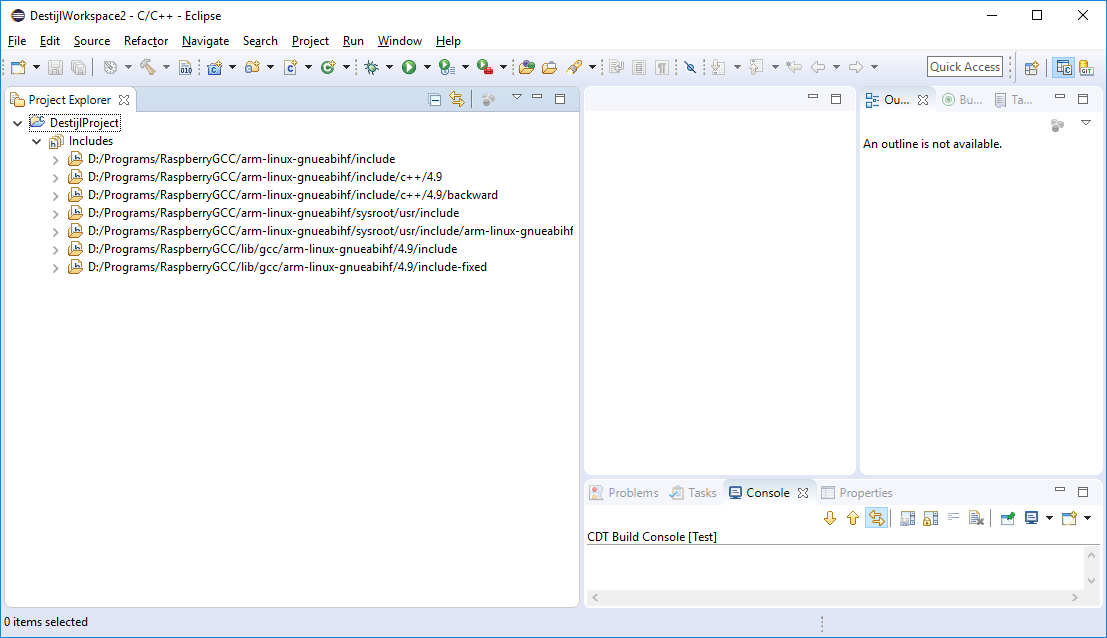


* Cliquez sur Next
* Dans la fenêtre « Select Configurations », ne changez rien et cliquez sur « Next »
* Dans la fenêtre « Cross GCC Command » :
* Dans le champs « Cross compiler prefix », mettre « arm-linux-gnueabihf- »
* Dans le champs « Cross compiler path », mettre le chemin vers le compilateur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

Voici un exemple :

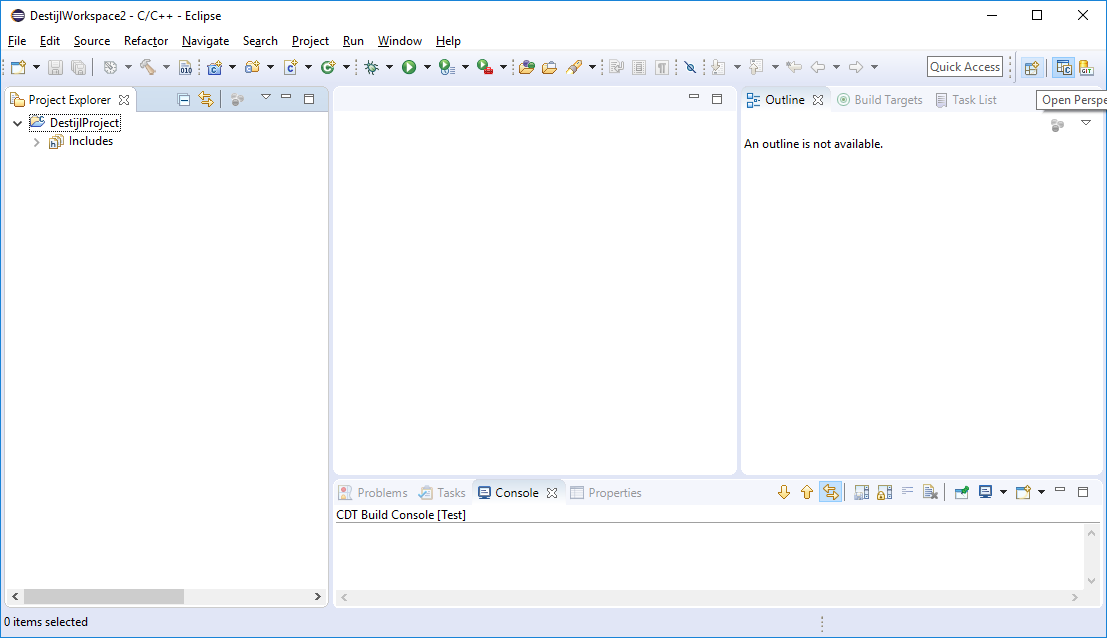


* Si le compilateur a bien été trouvé, vous devriez avoir les répertoires d’inclusion suivants (les chemins peuvent différer suivant où vous avez installé le compilateur) :

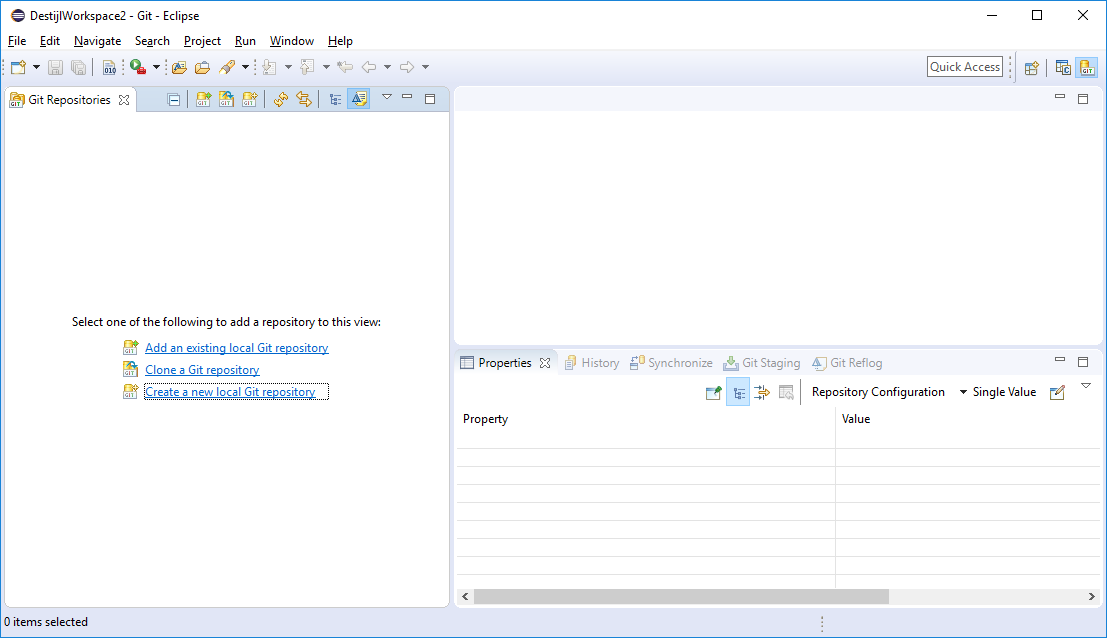


## Création du dépôt Git

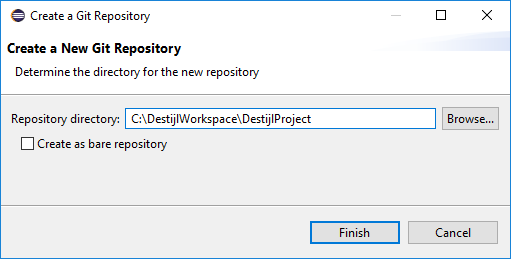
* Ouvrez ensuite la perspective Git. Pour cela, cliquez sur le bouton indiqué dans la capture suivante :



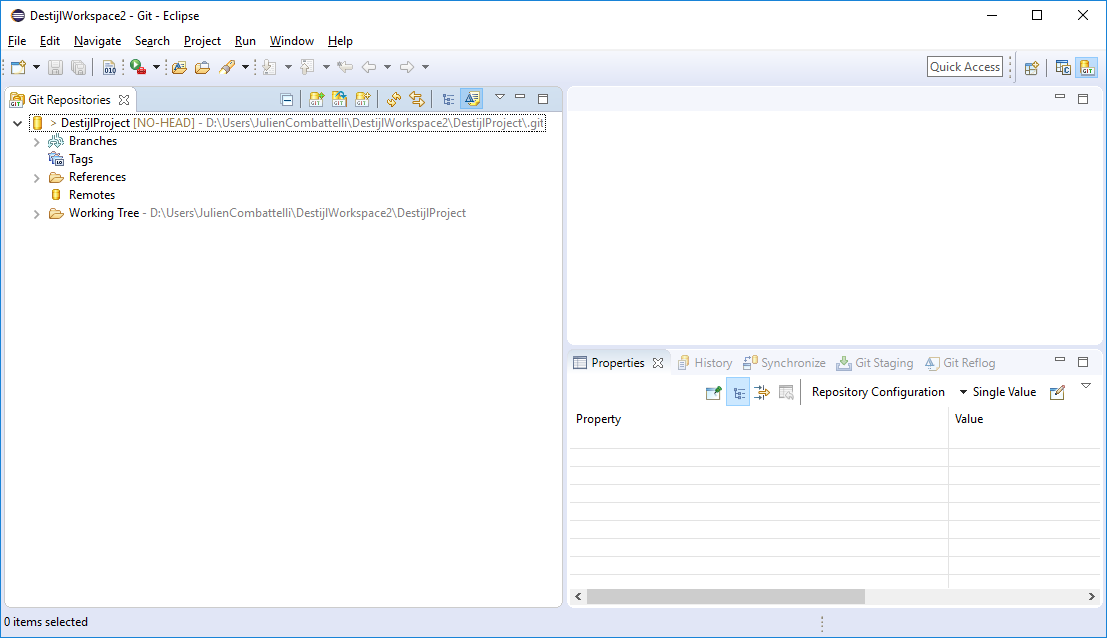
* Dans la fenêtre « Open Perspective », sélectionnez « Git »
* Dans l’onglet « Git Repositories », sélectionnez « Create a new local Git repository », comme illustré ci-dessous :



* 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.

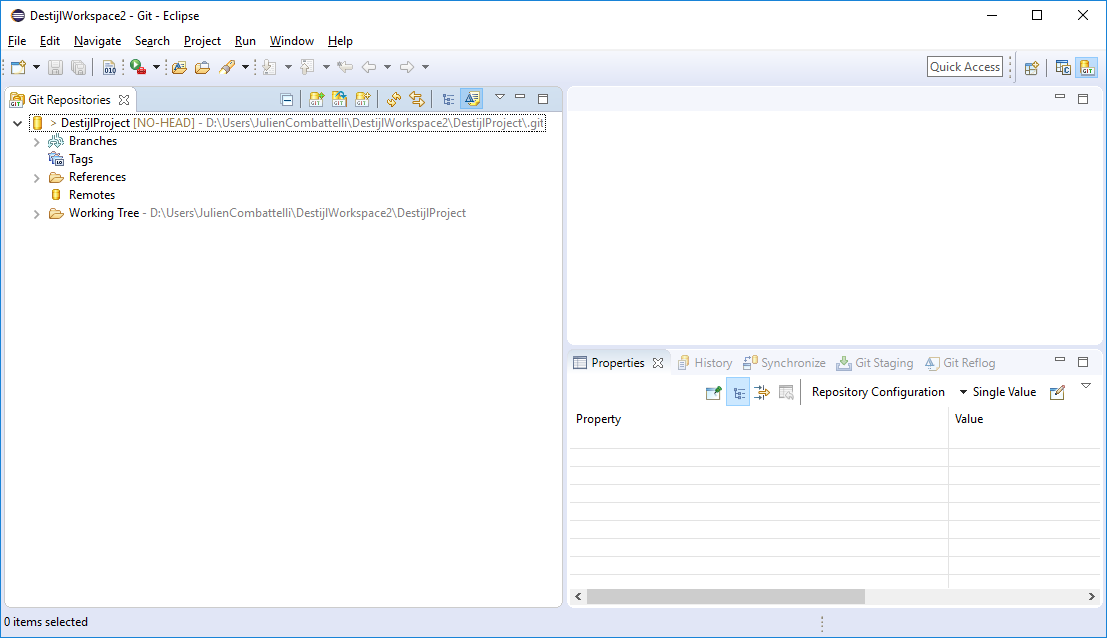


* Cliquez sur Finish
* Un nouveau dépôt devrait apparaitre dans Eclipse comme suit :

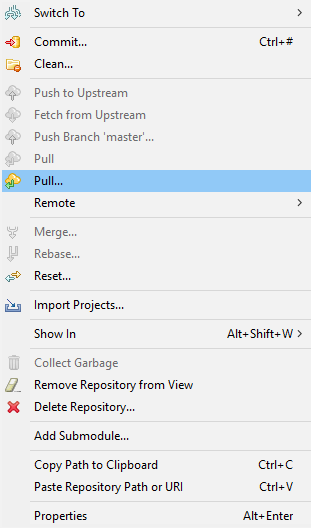


## Configuration du dépôt

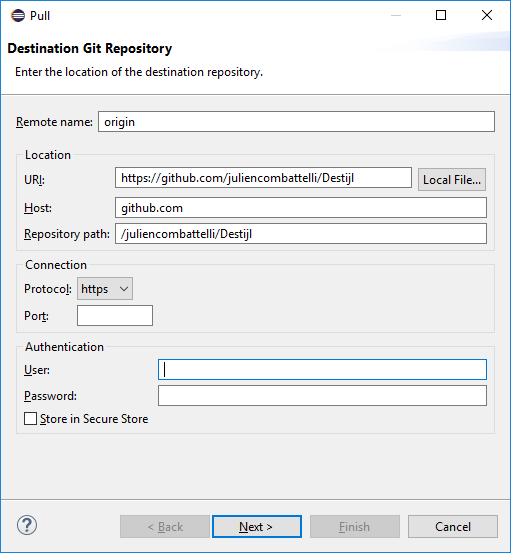
* Cliquez-droit sur le dépôt (encadré en vert ci-dessous) :



* Dans le menu qui apparait, 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) :



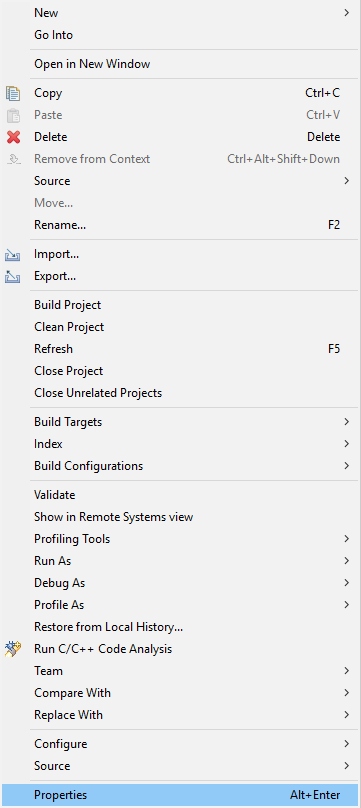
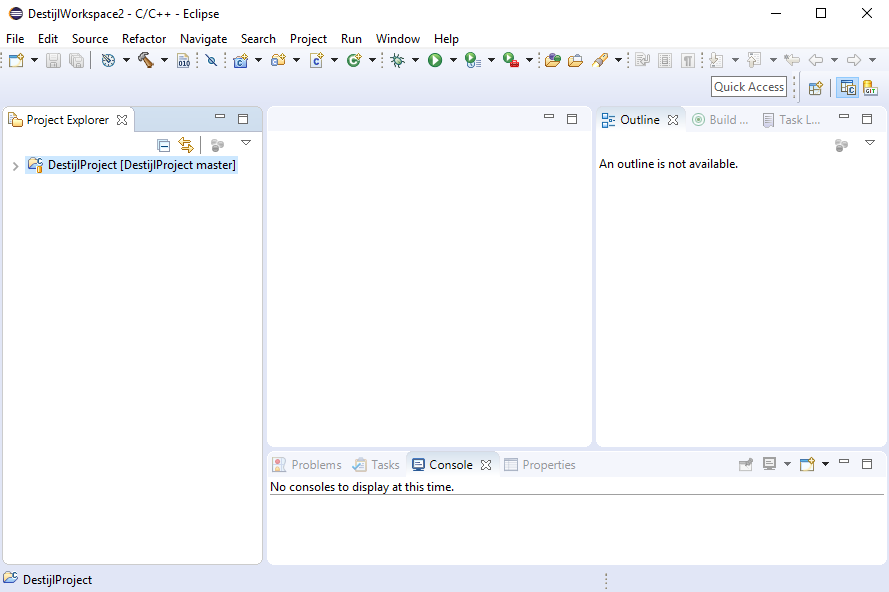
* Cliquez sur Next
* Dans la fenêtre « Pull », cliquez sur « New Remote » et saisissez à nouveau les champs comme ci-dessus (s’ils ne sont pas déjà remplis) puis cliquez sur Finish
* Dans le champ « Reference », saisissez « refs/heads/master » et cochez l’option « Configure upstream for push and pull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

## Configuration du projet

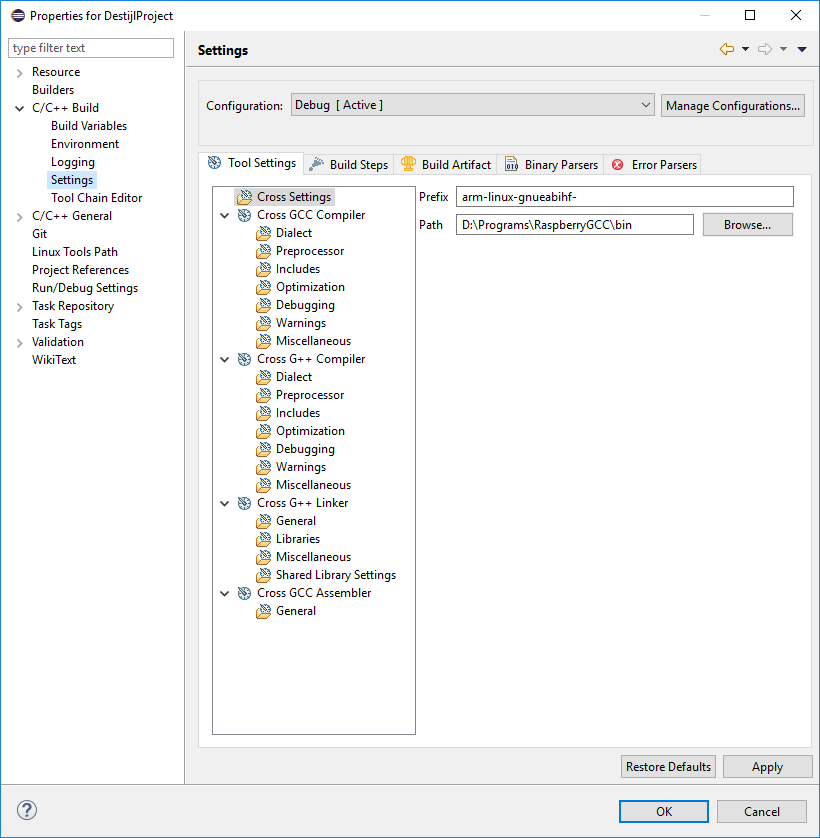
* Retournez sur la perspective C/C++ en cliquant sur le bouton encadré ci-dessous :



* 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}"

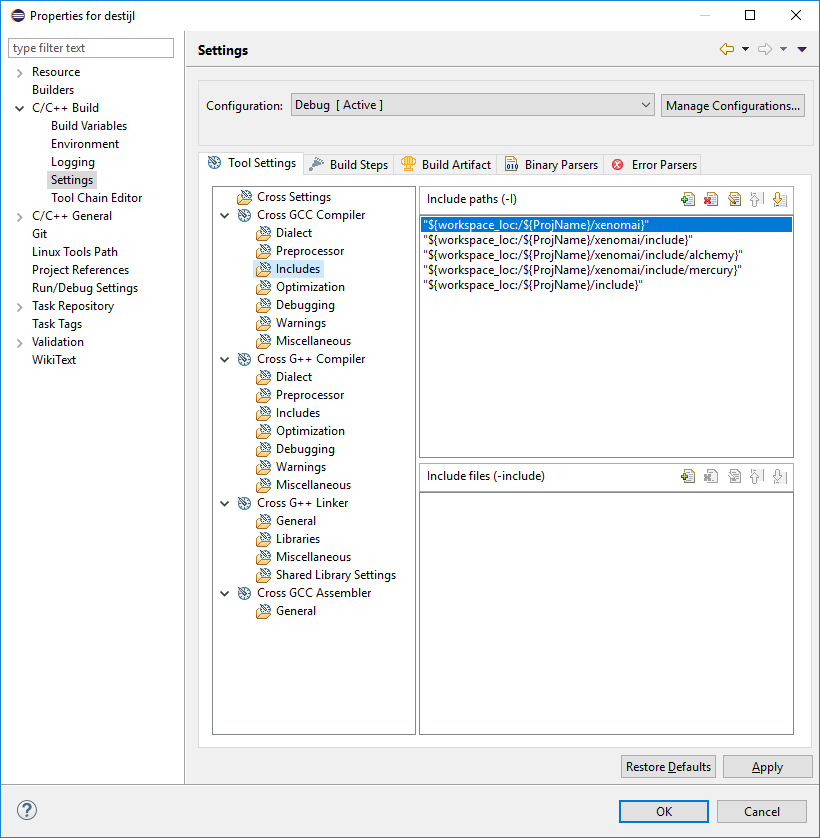
"${workspace\_loc:/${ProjName}/xenomai/include}"

"${workspace\_loc:/${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 « Librairy 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 :

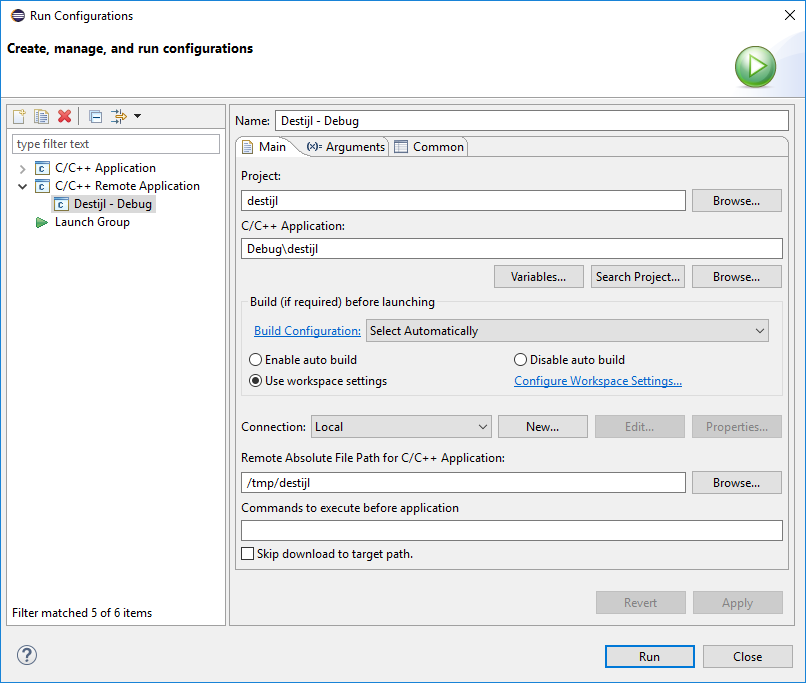
-Wl,-rpath="${workspace\_loc:/${ProjName}/lib}" -Wl,-rpath="${workspace\_loc:/${ProjName}/xenomai/lib}"

* Puis dans « Other objects » ajoutez :

"${workspace\_loc:/${ProjName}/xenomai/lib/xenomai/bootstrap.o}"

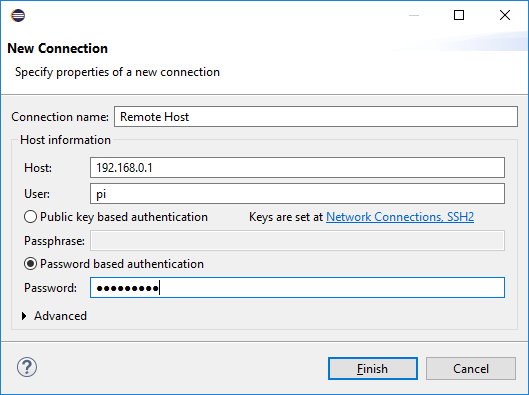
## Compilation et débogage du programme

* 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 :



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