

Eclipse

et

myRIO



en

C++

I. Fichiers nécessaires

Eclipse
http://www.ni.com/download/labview-real-time-module-2014/4846/en/

C Support for myRIO
http://www.ni.com/download/labview-myrio-toolkit-2018/7583/en/

NiRIO Drivers
http://www.ni.com/download/compactrio-device-drivers-january-2019/7833/en/

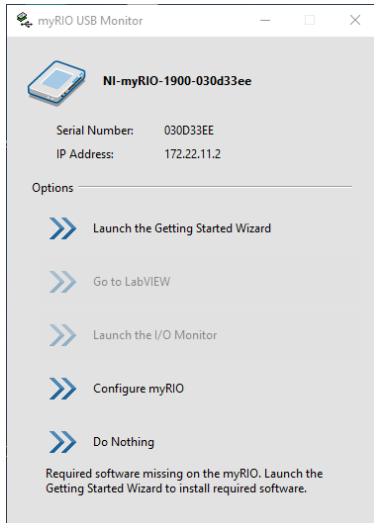
Java
https://www.java.com/fr/download/




II. Configuration du myRIO



Installer « NiRIO Driver » en laissant les options d'installation par défaut.
Redémarrer l'ordinateur quand proposé à la fin de l'installation.

Une fois l'installation terminée, connecter le myRIO à l'ordinateur. La fenêtre ci-dessous devrait apparaître.

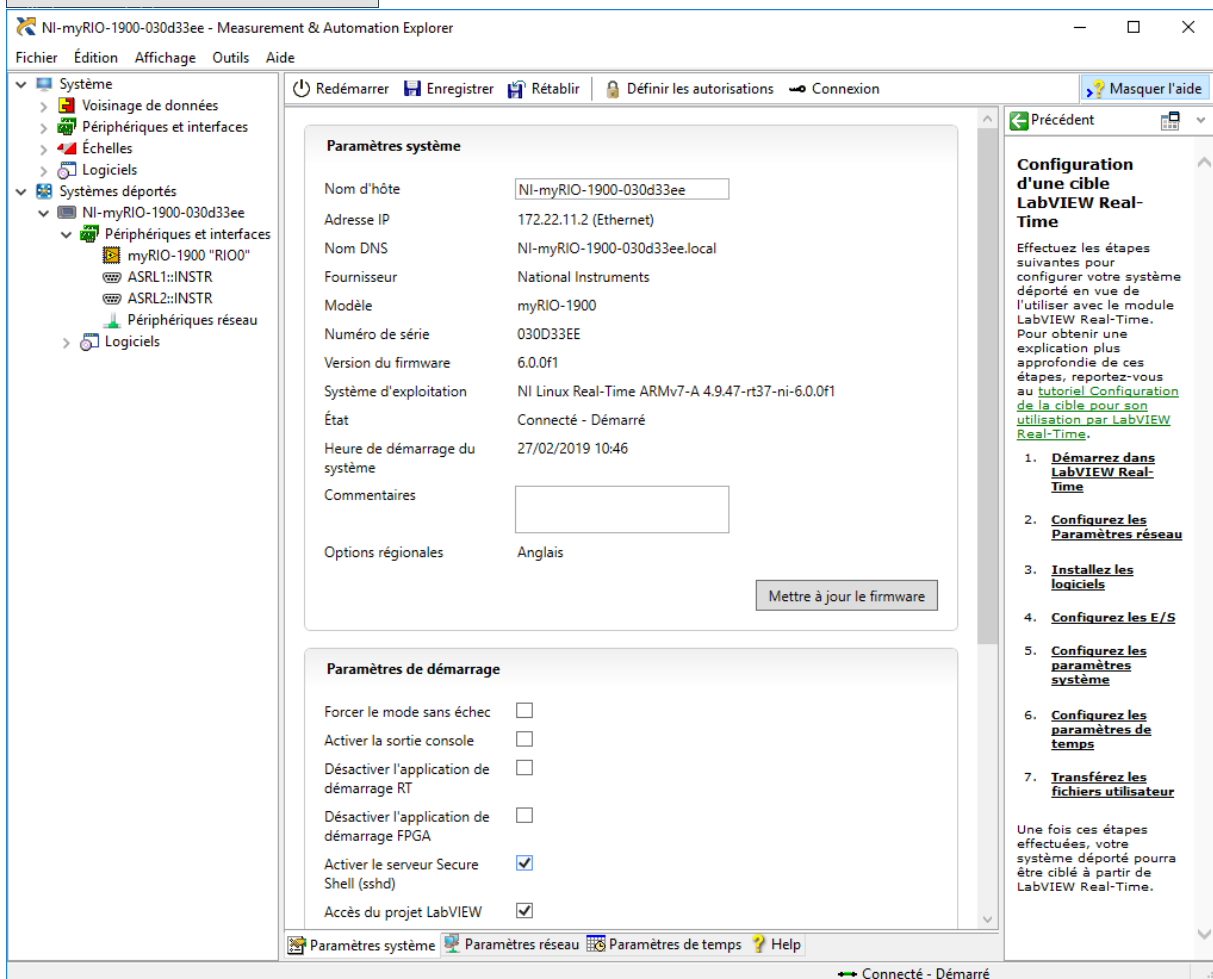
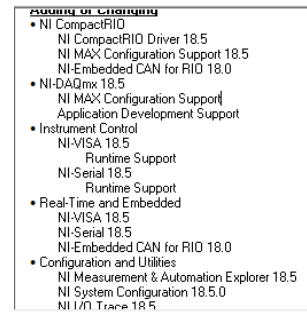


Choisir « Launch the Getting Started Wizard » et suivre les instructions. Le firmware sera flashé et mis à jour sur le myRIO.

Une fois l'installation terminée, lancer «  NI MAX » (un raccourci devrait se trouver dans le menu démarrer).

Étendre «  Systèmes déportés » et attendre la détection du myRIO. Le sélectionner, cocher « Activer le serveur Secure Shell (ssh) » puis cliquer sur «  Enregistrer ».

Fermer NI Max et myRIO USB Monitor.




III. Installation

Installer Java puis Eclipse.




Démarrer Eclipse (Démarrer/National Instruments/C & C++ Development Tools for NI Linux Real-Time 2014, Eclipse Edition).

Fermer l'onglet « Welcome » pour accéder à la vue projet.

Créer un projet  C/C++, nommé par exemple « APIForRIO ».



Choisir « Empty project » et « Cross GCC ». Cliquer sur « Next » deux fois.

Cross compiler prefix	arm-nilrt-linux-gnueabi-
Cross compiler path (sur ordinateur x64)	C:\Program Files (x86)\National Instruments\Eclipse\14.0\arm\sysroots\i686-nilrtsdk-mingw32\usr\bin\armv7a-vfp-neon-nilrt-linux-gnueabi

Créer un dossier «  src » dans le projet APIForRIO, puis y créer un fichier «  main.cpp » et un autre dossier : «  CAPI » (son chemin sera donc « APIForRIO/src/CAPI »).

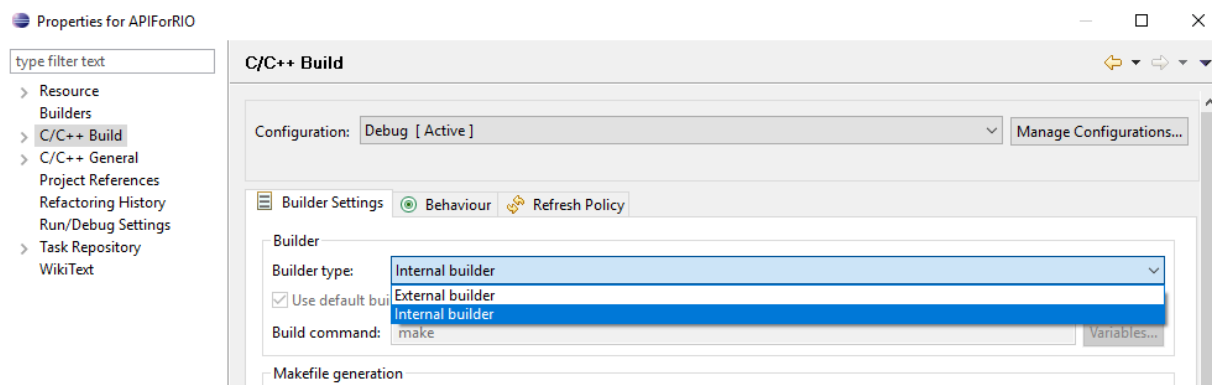
Extraire « C_Support_for_myRIO_v6.0 » dans un répertoire quelconque.

Copier tous les fichiers « .../C Support for myRIO/source/ » vers le dossier précédemment créé.

Clic droit sur «  CAPI », puis «  Refresh » pour actualiser les modifications du workspace.


Clic droit sur le nom du projet, puis « Properties ».

Dans « C/C++ Build », choisir « Internal builder » au lieu de « External builder ».



Étendre « C/C++ Build » et entrer dans « Settings ».

- Section «  Cross GCC Compiler »


Sous «  Defined symbols » : ajouter un symbole se nommant « MyRio_1900 ».


- Section «  Miscellaneous »

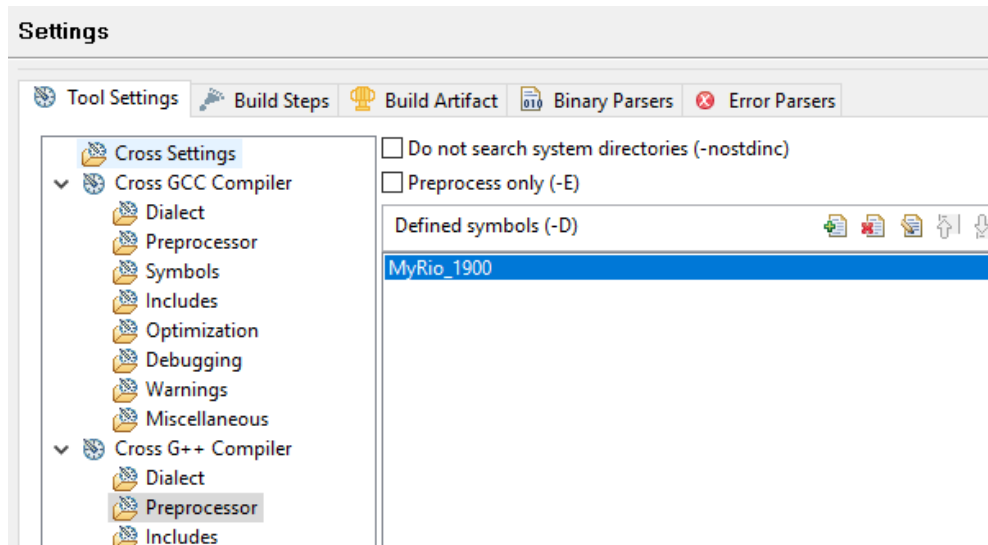
Dans le champ « Other flags », ajouter « -mfpu=vfpv3 -mfloat-abi=softfp » pour augmenter la précision en virgule flottante.

Other flags	-c -fmessage-length=0 -mfpu=vfpv3 -mfloat-abi=softfp
-------------	--



- Section «  Cross G++ Compiler »

Sous «  Preprocessor » : ajouter un symbole se nommant « MyRio_1900 ».



- Section «  Miscellaneous »

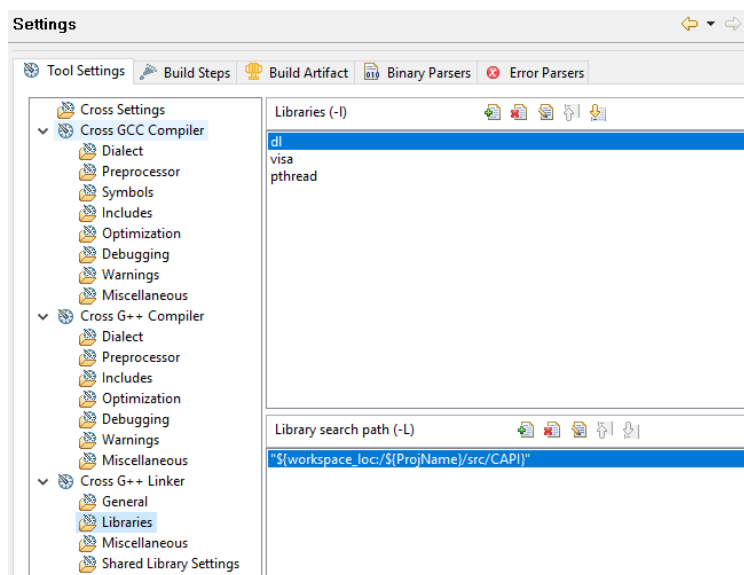
Dans le champ « Other flags », ajouter « -mfpv=vfpv3 -mfloat-abi=softfp » pour augmenter la précision en virgule flottante.

Other flags	-c -fmessage-length=0 -mfpv=vfpv3 -mfloat-abi=softfp
-------------	--

- Section «  Cross G++ Linker /  Libraries»

Ajouter les librairies et chemin de librairie suivants :

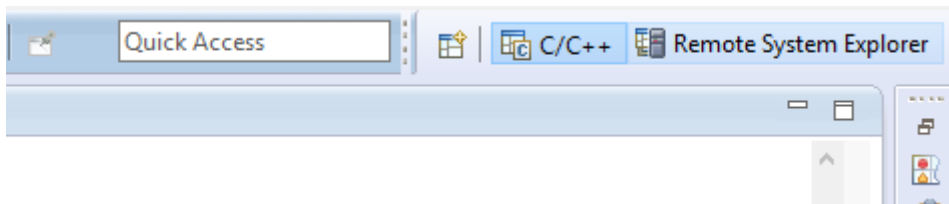
Libraries	dl
	visa
	pthread
Library search path	"\${workspace_loc}/\${ProjName}/src/CAP1"




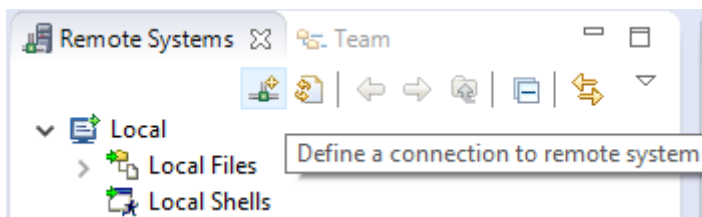
Vérifier la correspondance avec l'écran ci-contre, puis valider les nouvelles propriétés en cliquant sur OK.


IV. Connection au myRIO

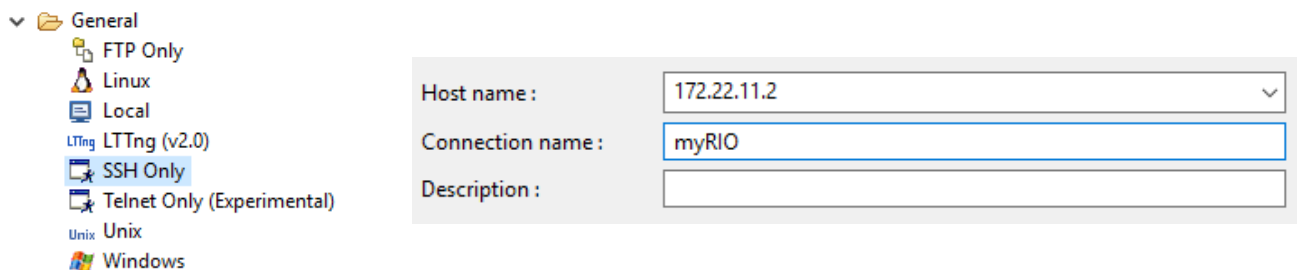
Sous Eclipse, choisir «  Remote System Explorer »





Choisir «  Define a connection to remote system »



Cliquer sur «  SSH Only », puis remplir les champs « Host name » et « Connection name » avec « 172.22.11.2 » et « myRIO ».





Ensuite, cliquer sur « Next », « Next », « Next » puis « Finish ».

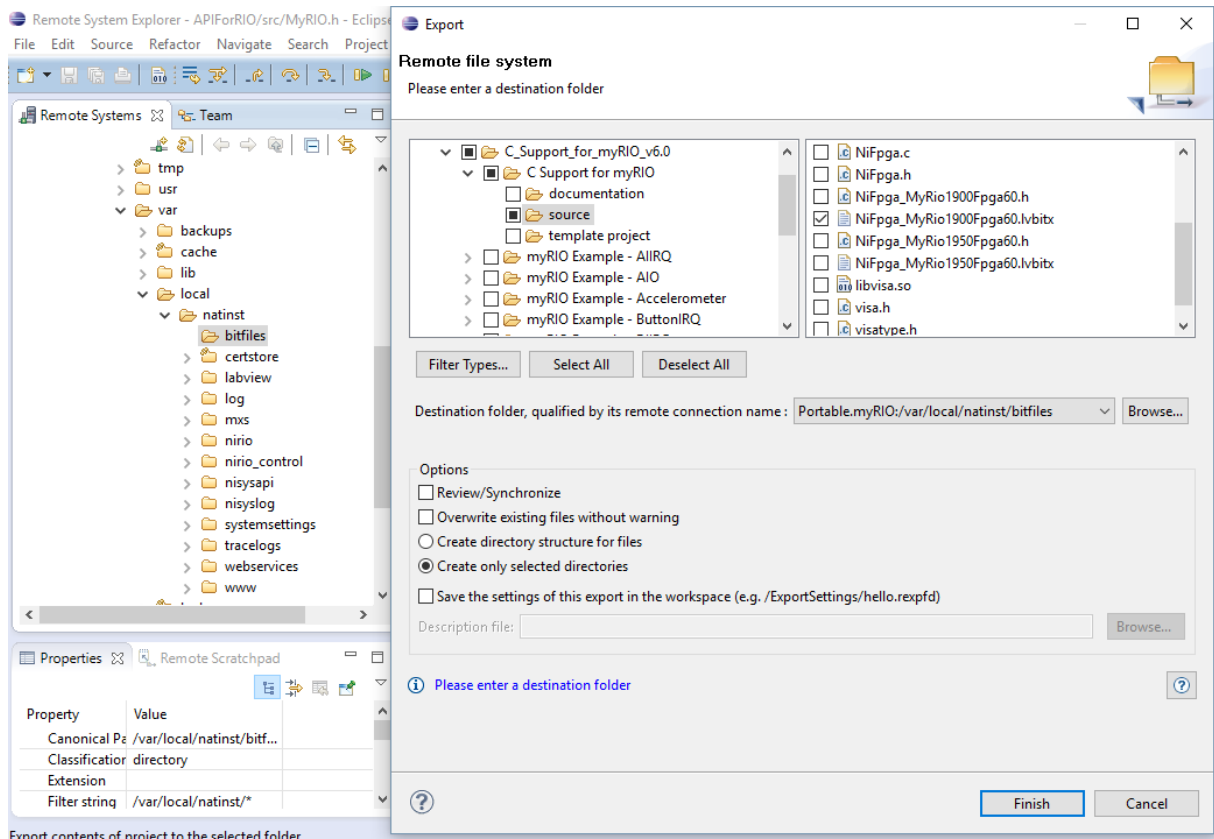
Une nouvelle connection est apparue. L'étendre, ouvrir «  Sftp Files » puis «  Root ».




Un nom d'utilisateur et un mot de passe sont demandés : entrer « admin » et laisser le champ du mot de passe vide. Valider.

Se rendre dans «  /var/local/natinst » et créer un dossier nommé « bitfiles ». Clic droit sur ce dossier, puis « Export from project... »

Se rendre, dans la fenêtre de gauche, dans le dossier « CAPI ». Cocher dans la fenêtre de droite le fichier «  NiFpga_MyRio1900Fpga60.lvbitx » puis cliquer sur « Finish ».

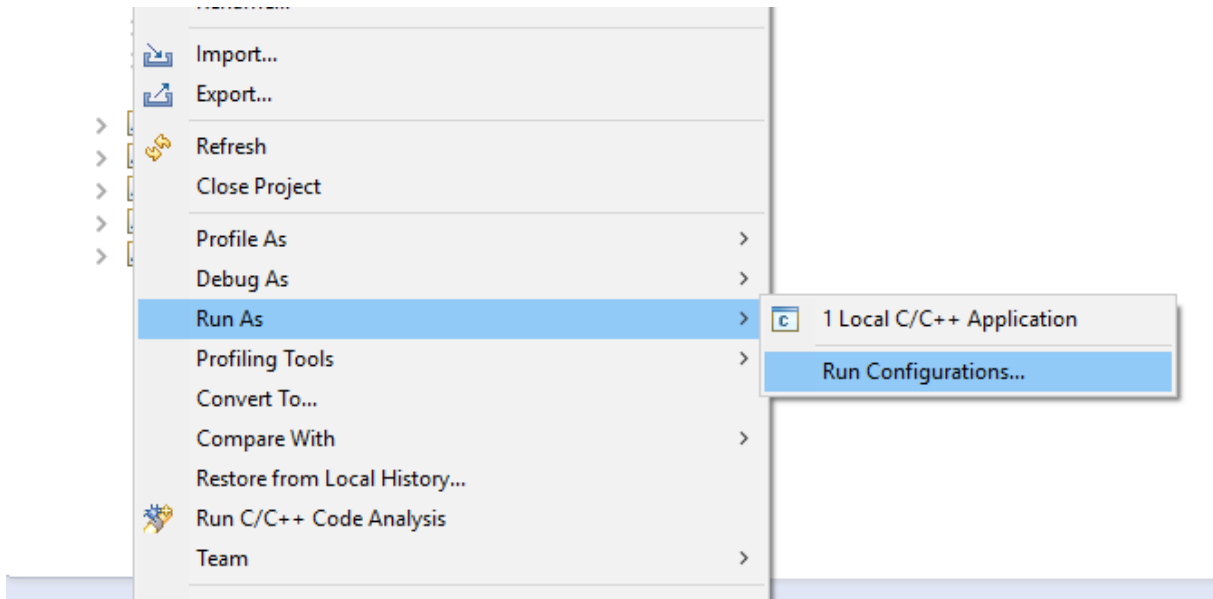


Le myRIO est maintenant configuré pour supporter des interactions en C.

Pour revenir en vue projet, sélectionner «  C/C++ » en haut à droite.

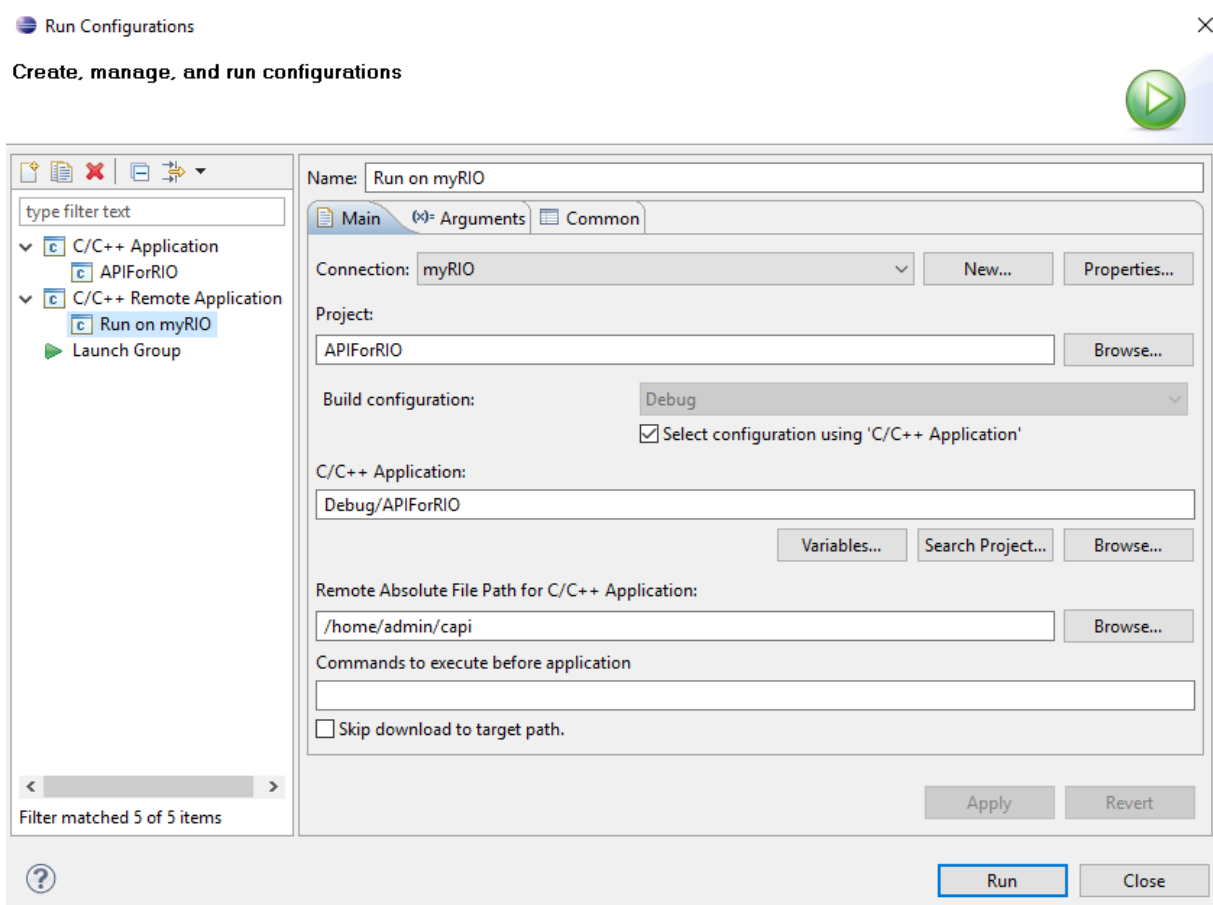
V. Configuration du téléversement

Clic droit sur le nom du projet, puis « Run As », puis « Run configurations... »





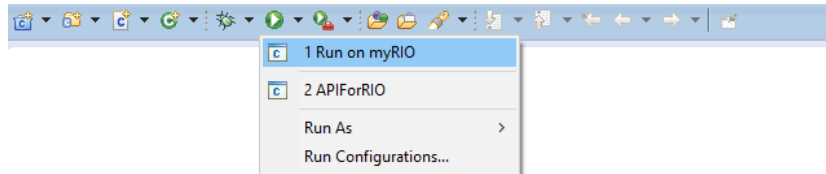
Clic droit sur « Remote Application », « New ».

Nommer la configuration (nous l'appelons « Run on myRIO »), choisir la connexion (myRIO précédemment connecté), le projet (APIForRIO), le chemin de l'application à envoyer et le chemin d'implantation distant. Vérifier la correspondance avec l'écran ci-contre :

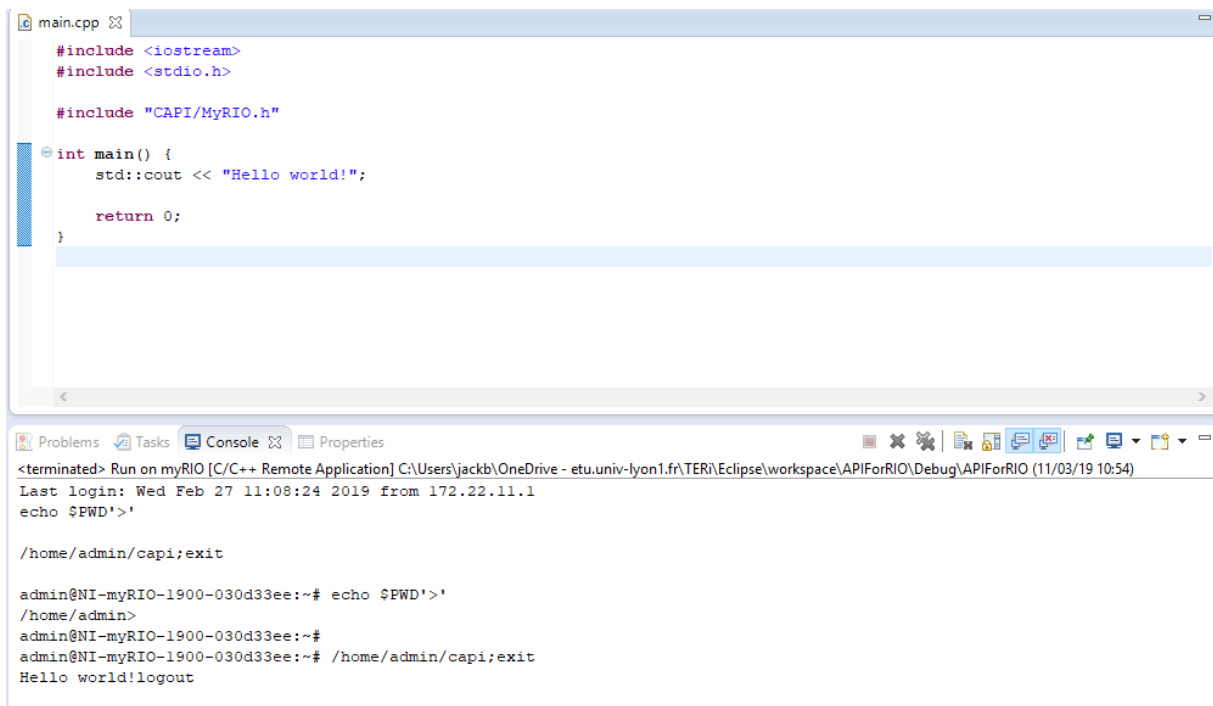


VI. Vérification du fonctionnement

Dans «  main.cpp », #inclure « CAPI/MyRIO.h » puis cliquer sur «  Run on myRIO » (la Run Configuration précédemment créée).



Si la configuration est correcte, la console devrait donner la sortie du myRIO.

A screenshot of the Eclipse IDE showing the console output. The console window is titled 'Console' and displays the following text:

```
<terminated> Run on myRIO [C/C++ Remote Application] C:\Users\jackb\OneDrive - etu.univ-lyon1.fr\TERA\Eclipse\workspace\APIForRIO\Debug\APIForRIO (11/03/19 10:54)
Last login: Wed Feb 27 11:08:24 2019 from 172.22.11.1
echo $PWD'>'

/home/admin/capi;exit

admin@NI-myRIO-1900-030d33ee:~# echo $PWD'>'
/home/admin>
admin@NI-myRIO-1900-030d33ee:~#
admin@NI-myRIO-1900-030d33ee:~# /home/admin/capi;exit
Hello world!logout
```

Pour vérifier le bon fonctionnement des registres, voici un programme allumant les LEDs du myRIO:

```
#include <iostream>
#include <stdio.h>

#include "CAPI/MyRIO.h"

extern NiFpga_Session myrio_session;

int main() {
    // ouvre une session myRIO
    NiFpga_Status status = MyRio_Open();
    if(MyRio_IsNotSuccess(status))
        std::cout << "MyRio not created : " << status << std::endl;

    // sélectionne les DigitalOutput LED 3:0, et envoie 0b1111 pour toutes les allumer
    NiFpga_MergeStatus(&status,
        NiFpga_WriteU8(myrio_session, DOLED30, 15));
    // attend un arrêt forcé
    while(true);
}
```





VII. Liens utiles pour aller plus loin

- **Personnalité FPGA par défaut**

http://zone.ni.com/reference/en-XX/help/373925B-01/myriohelp/myrio_fpga_personalities/

Documentation (registres associés...) :

<http://www.ni.com/product-documentation/14655/en/>

myRIO Shipping Personality 6.0 Reference

This document contains reference information about the myRIO shipping personality.

Contents

Introduction	4
Register Naming Convention	4
Peripheral Type	5
Channel Name	5
Property Names	6
System Control / Function Select	8
Host Synchronization Registers (SYS.x.RDY)	8
Function Select Registers (SYS.SELECTx)	9
myRIO Expansion Ports (MXP)	9
Mini Systems Port (MSP)	10
Onboard Device Registers	11
LEDs (DO.LED3:0)	11
Button (DI.BTN)	11
Accelerometer Value Registers (ACC.x.VAL)	12
AI/AO/Audio	13

- **Carte d'extension**

<https://learn.ni.com/teach/resources/808/robot-builder-s-guide-pitsco-tetrix-prime-for-ni-myrio>

Description des ports utilisés (moteurs, gyroscope...) de la page 38 à la page 44.

Motor Number (noted on motor board)	Pin Name (from left to right based on the image above)	Wire color (if using provided DC motor)	MXP Pin number	Name in software (based on MXP A)
1	Encoder B	Purple	22	A/ENC.B
1	Encoder A	Blue	18	A/ENC.A
1	Encoder Ground	Green	n/a	n/a
1	Encoder VCC	Brown	n/a	n/a
1	Motor +	Red	27 – PWM speed control, 15 – DIO direction control	A/PWM0 A/DIO2
	Motor -	Black		
2	Encoder B	Purple	22 (on opposite MXP port)	B/ENC.B
2	Encoder A	Blue	18 (on opposite MXP port)	B/ENC.A
2	Encoder Ground	Green	n/a	n/a

