

Set up QT Creator for Yocto

Rev 2.0 20150825



TechNexion

INNOVATORS OF TECHNOLOGY

Contents

1. Environment Setup.....	1
1.1 Build and Install Yocto with QT5.....	1
1.2 Download and install QT Creator.....	1
2. Set up QT Creator for Cross Compiling.....	1
2.1 Install QT toolchain.....	1
2.2 Configure QT Creator.....	2
2.3 Setup cross compiler in QT Creator.....	2
2.4 Setup qmake location in QT Creator.....	4
2.5 Setup kits in QT Creator.....	5
2.6 Test cross-compiling in QT Creator.....	5
3. Deploy Application into Target Board.....	7
3.1 Set up target board.....	7
3.2 Set up QT Creator for connecting the target board.....	8
4. Build and Deploy application in QT Creator.....	11

1. Environment Setup

The following environment setup is verified under ubuntu 12.04 with QT Creator 3.0.

1.1 Build and Install Yocto with QT5

For building Yocto “fsl-image-qt5 image”, please refer to the document “EDM_yocto_1.X_BSP_Build_Guide.pdt”.

1.2 Download and install QT Creator

Download QT Creator 3.4.2:

For 32 bit:

http://download.qt.io/official_releases/qtcreator/3.4/3.4.2/qt-creator-opensource-linux-x86-3.4.2.run

For 64 bit:

http://download.qt.io/official_releases/qtcreator/3.4/3.4.2/qt-creator-opensource-linux-x86_64-3.4.2.run

Install QT Creator 3.0:

```
# sudo chmod +x qt-creator-opensource-linux-x86_64-3.4.2.run
# ./qt-creator-opensource-linux-x86_64-3.4.2.run
```

Select the installation directory and follow the steps to complete installation.

2. Set up QT Creator for Cross Compiling

2.1 Install QT toolchain

Bitbake a poky toolchain

```
# DISPLAY=hdmi720p MACHINE=edm-fairy-imx6 source edm-setup-release.sh -b build-x11 -e x11
# bitbake meta-toolchain-qt5
```

Run the installation script located in “build-x11/tmp/deploy/sdk”

```
# sh poky-eglibc-x86_64-meta-toolchain-qt5-cortexa9hf-vfp-neon-toolchain-1.7.sh
```

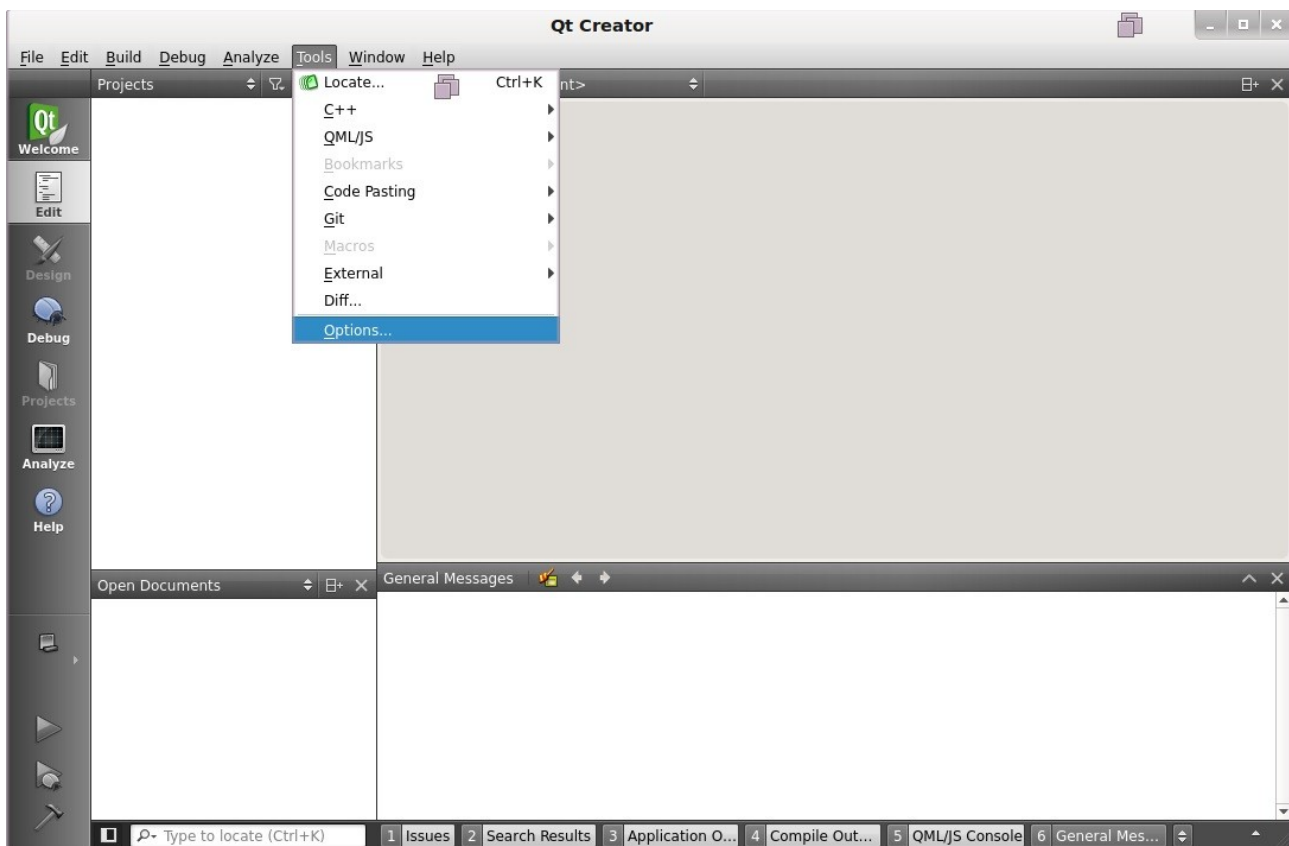
2.2 Configure QT Creator

Run qtcreator from the terminal after setting up environment.

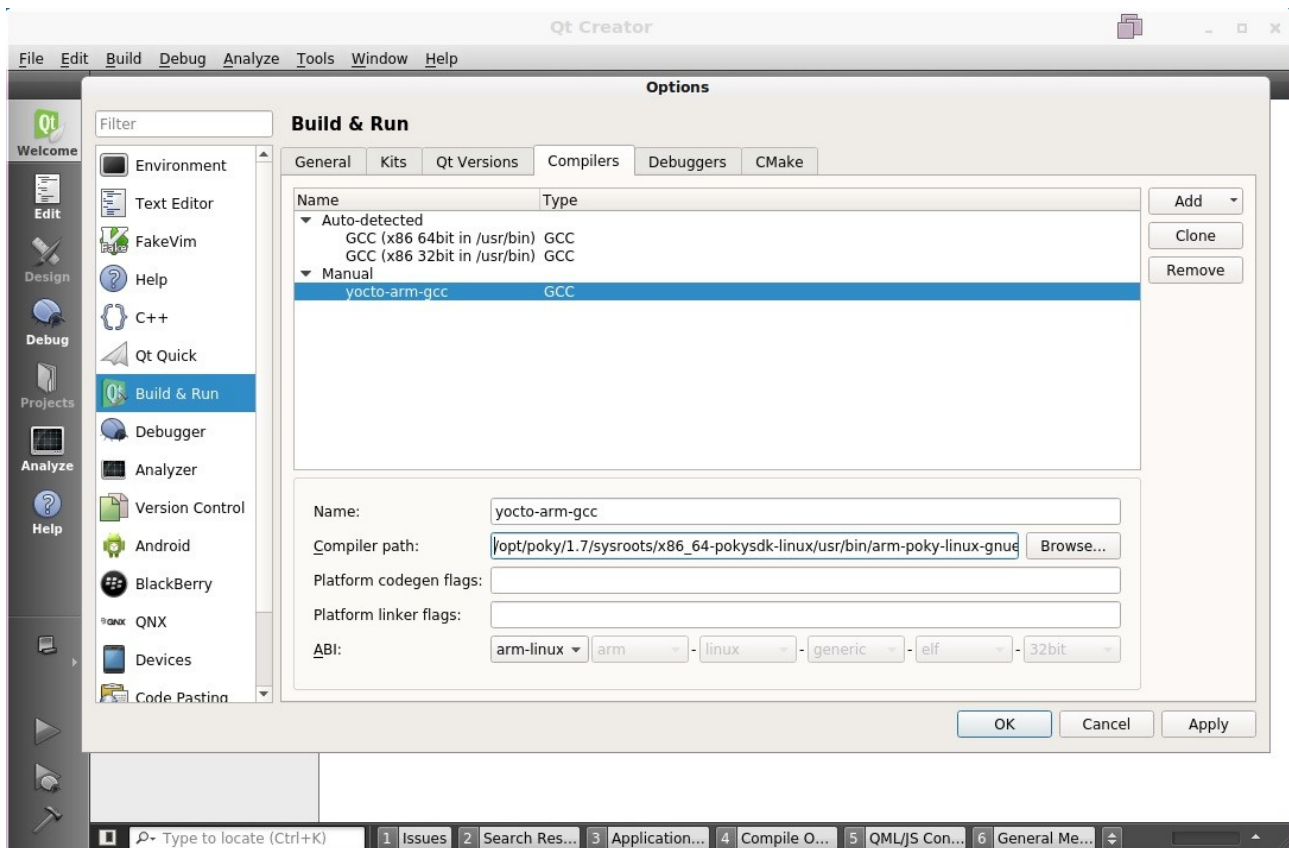
```
# source /opt/poky/1.7/environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi  
  
# ~/qtcreator-3.4.2/bin/qtcreator.sh
```

2.3 Setup cross compiler in QT Creator

Select **Tools > Options**



Select **Build & Run > Compilers > Add**



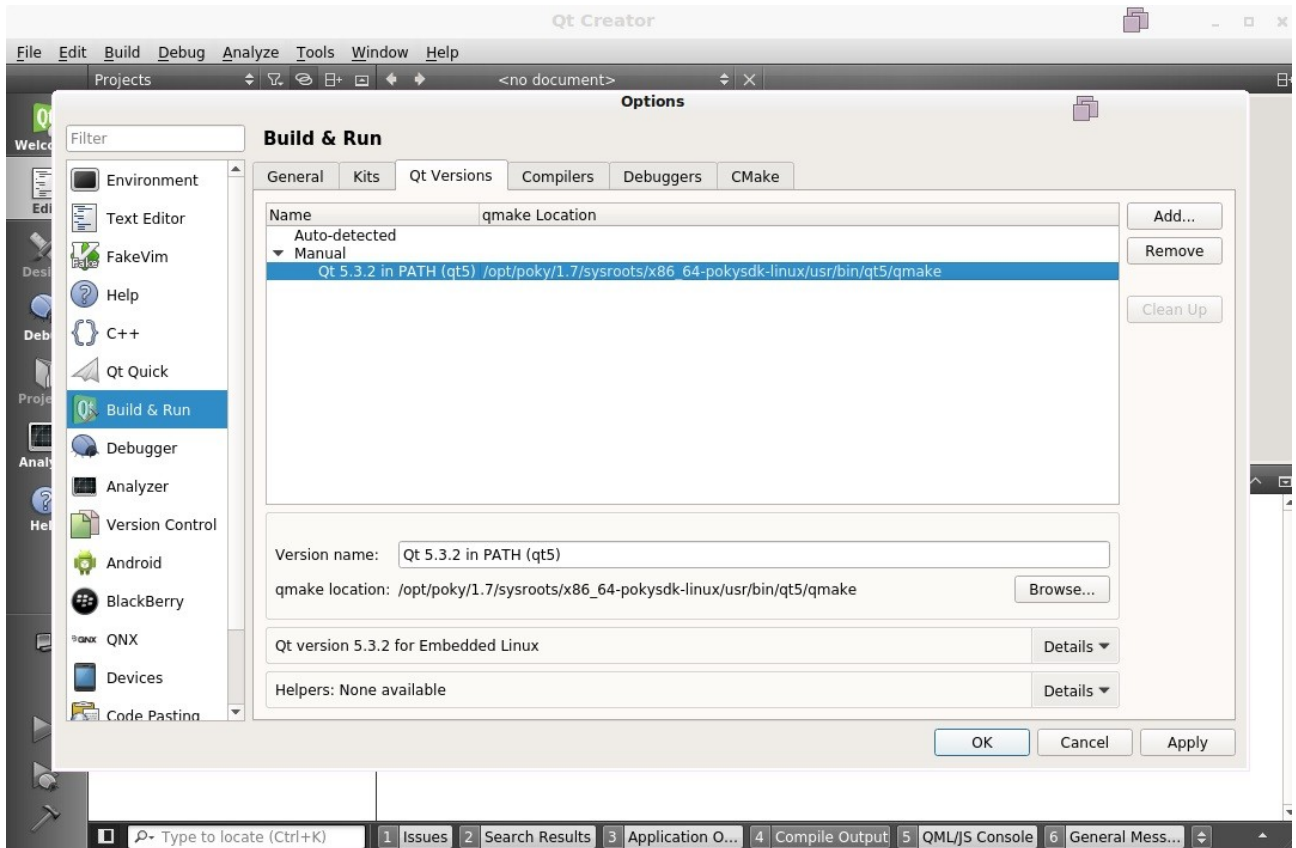
In the **Name** column: enter "**yocto-arm-gcc**"

In the **Compiler path** column: enter "**/opt/poky/1.7/sysroots/x86_64-pokysdk-linux/usr/bin/arm-poky-linux-gnueabi/arm-poky-linux-gnueabi-g++**"

Click "**Apply**".

2.4 Setup qmake location in QT Creator

Select **Build & Run > Qt Versions > Add**

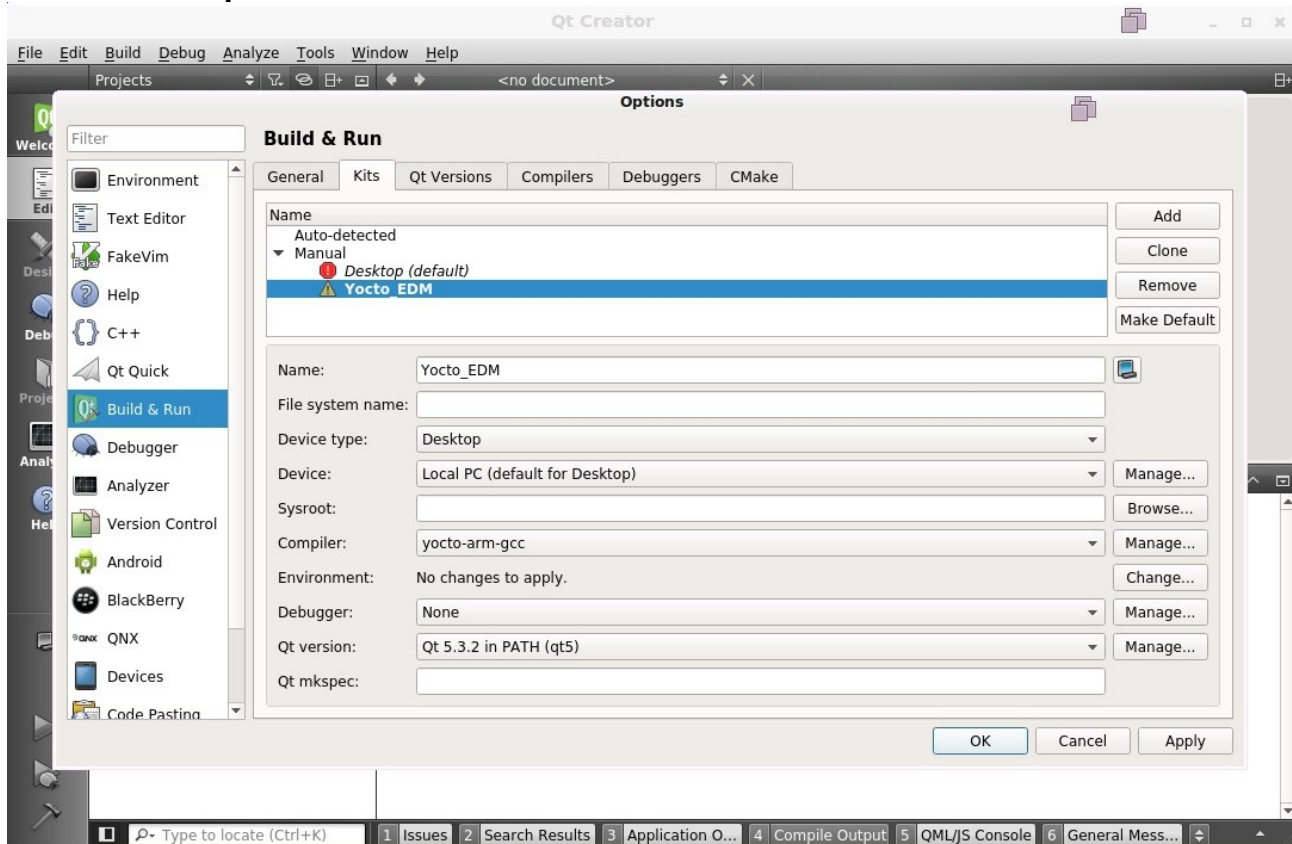


In the **qmake location** column: Select the path **`/opt/poky/1.7/sysroots/x86_64-pokysdk-linux/usr/bin/qt5/qmake`**

Click **"Apply"**.yo

2.5 Setup kits in QT Creator

Select **Tools > Options > Build & Run > Kits > Add**



In the **Name** column: enter "**Yocto_EDM**"

In the **Compiler** column: select "**yocto-arm-gcc**"

In the **QT version** column: select "**Qt 5.3.2 in PATH (qt5)**"

Click "**Apply**".

2.6 Test cross-compiling in QT Creator

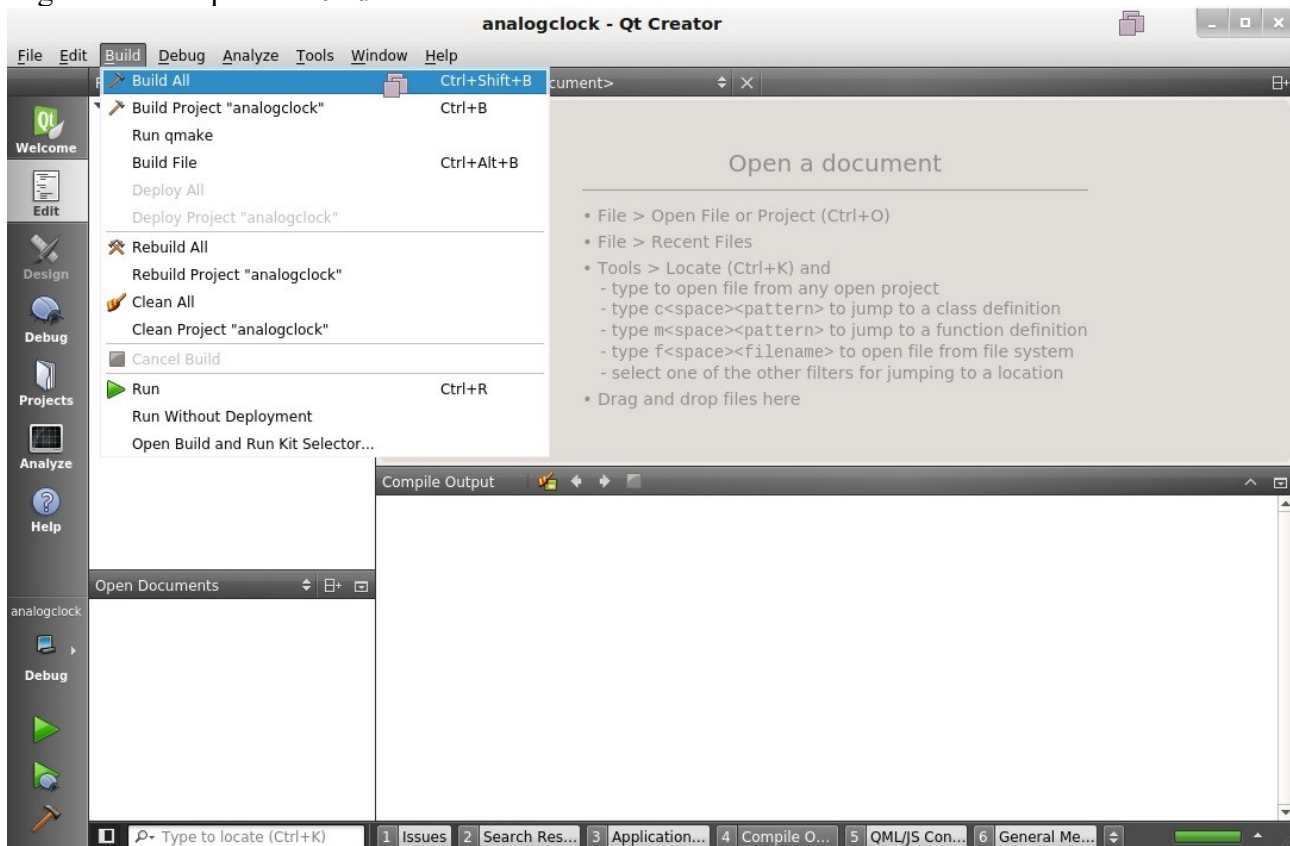
You can compile an example of QT to check if settings work.

Decompress the example:

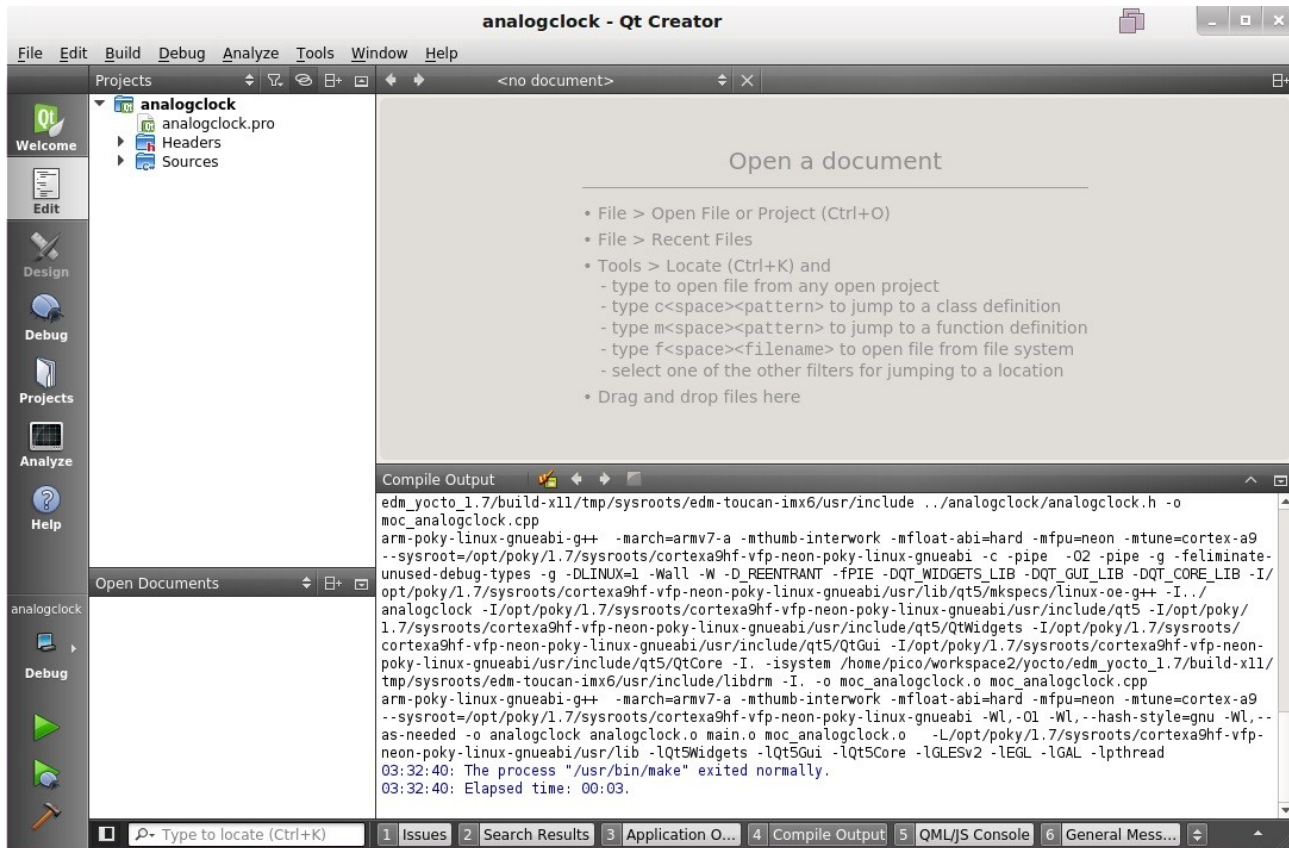
```
# tar zxvf analogclock.tar.gz
```

Select **Files > Open File or Project > "analogclock/analogclock.pro"**

Select "**project name**". The project name here is "**analogclock**".
Right click and press "**Build All**".



The following message in "Compile Output" tab shows completion of cross-compiling.



3. Deploy Application into Target Board

3.1 Set up target board

QT creator deploys the application via “SFTP”.

Check if “sftp server” is available on the target board.

```
root@edm-fairy-imx6:~# ls -al /usr/lib/openssh/sftp-server
-rwxr-xr-x 1 root root 67668 Aug 26 00:38 /usr/lib/openssh/sftp-server
```

If there is no “sftp server”, please install “**openssh-sftp-server**” package into Yocto rootfs.

Set up password for SFTP user.

```
rroot@edm-fairy-imx6:~# passwd root
Changing password for root
Enter the new password (minimum of 5 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Bad password: too short.
Warning: weak password (enter it again to use it anyway).
New password:
```

Re-enter new password:
passwd: password changed.

Set up static IP on target board.

```
root@edm-fairy-imx6:/usr/lib/connman/test# ./get-services
[ /net/connman/service/ethernet_001f7baa0123_cable ]
    IPv6.Configuration = { Method=off }
    AutoConnect = true
    Name = Wired
...
```

Note that [ethernet_001f7baa0123_cable](#) corresponds to your ethernet mac address. Then, run the set-nameservers script as follows:

```
root@edm-fairy-imx6:/usr/lib/connman/test# ./set-nameservers ethernet_001f7baa0123_cable
8.8.8.8 UR.FAV.DNS.SRV
```

Change your network connection to use static IP, and run the script set-ipv4-method as:

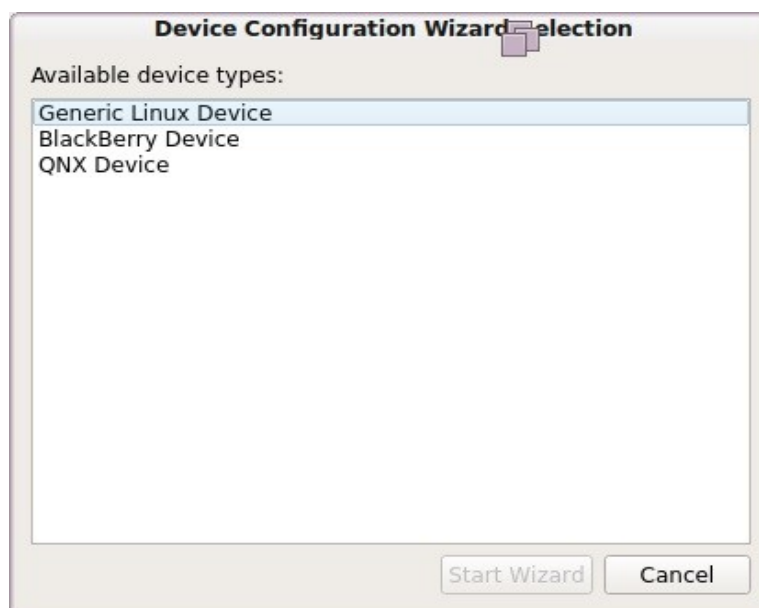
```
root@edm-fairy-imx6:/usr/lib/connman/test# ./set-ipv4-method ethernet_001f7baa0123_cable
manual UR.STA.TIC.IP 255.255.255.0 UR.GAT.EWY.IP
```

Reboot the board, the static ip will take effect.

3.2 Set up QT Creator for connecting the target board

Select **Tools > Options > Devices > Add**

Select device type as "Generic Linux Device"



TechNexion

In the **Name** column: enter "**EDM**"

In the **Authentication type** column: select "**Password**"

In the **Host name** column: enter "**IP address on target board**"

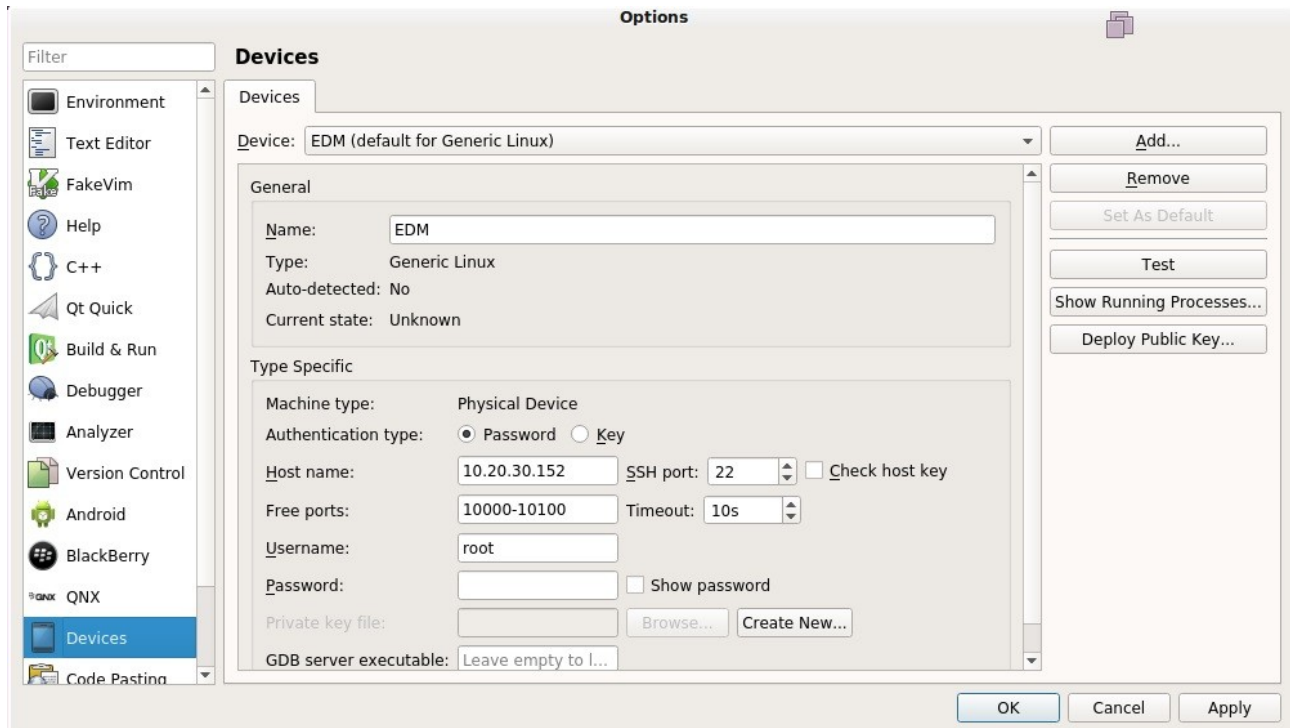
Your can use command "ifconfig" on EDM Fairy board to check the IP.

`root@edm-fairy-imx6:~# ifconfig`

In the **Username** column: enter "**root**"

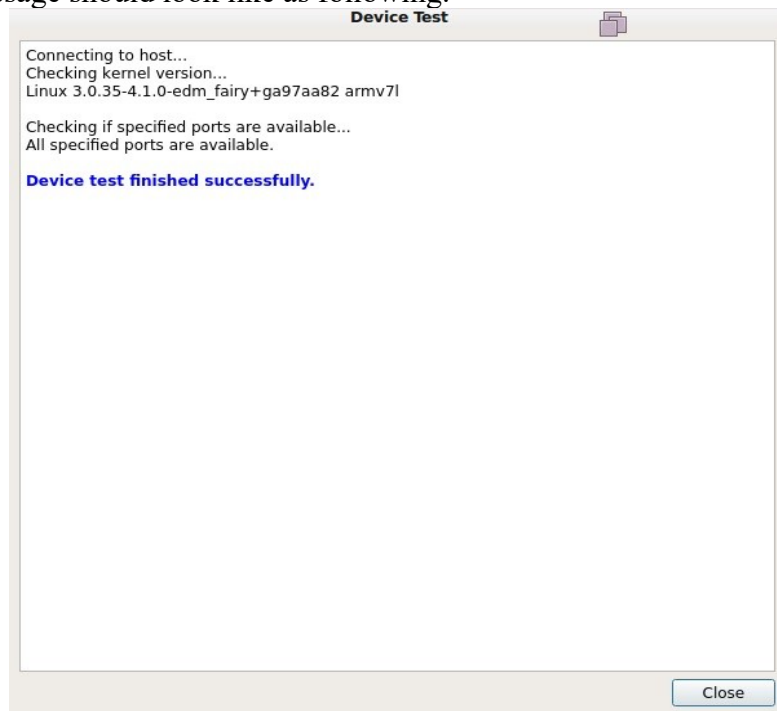
In the **Password** column: enter the password you set on the target board.

Click "**Apply**".



Select **Tools > Options > Devices > Test**

The successful message should look like as following.

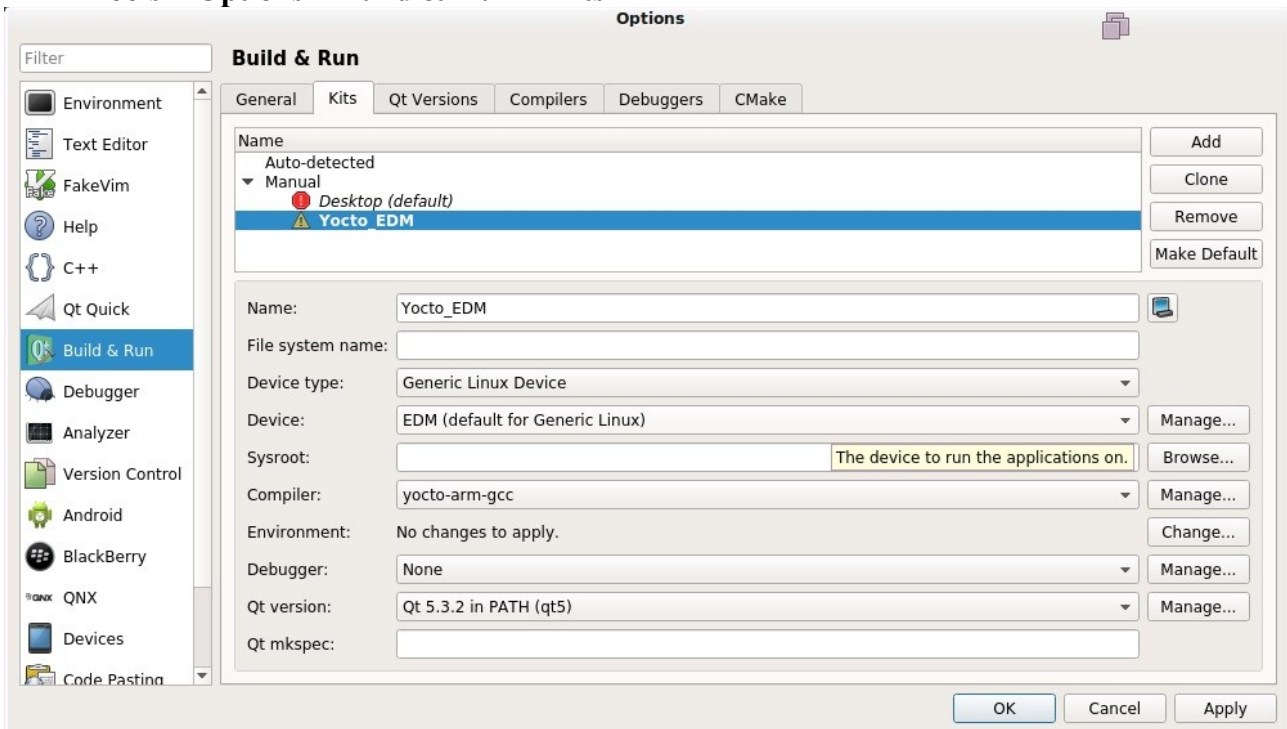


If test fails. You can use ssh to test the connnection between host PC and board to figure out the problems.

```
$ ssh -v root@{board_ip}
```

4. Build and Deploy application in QT Creator

Select **Tools > Options > Build & Run > Kits**

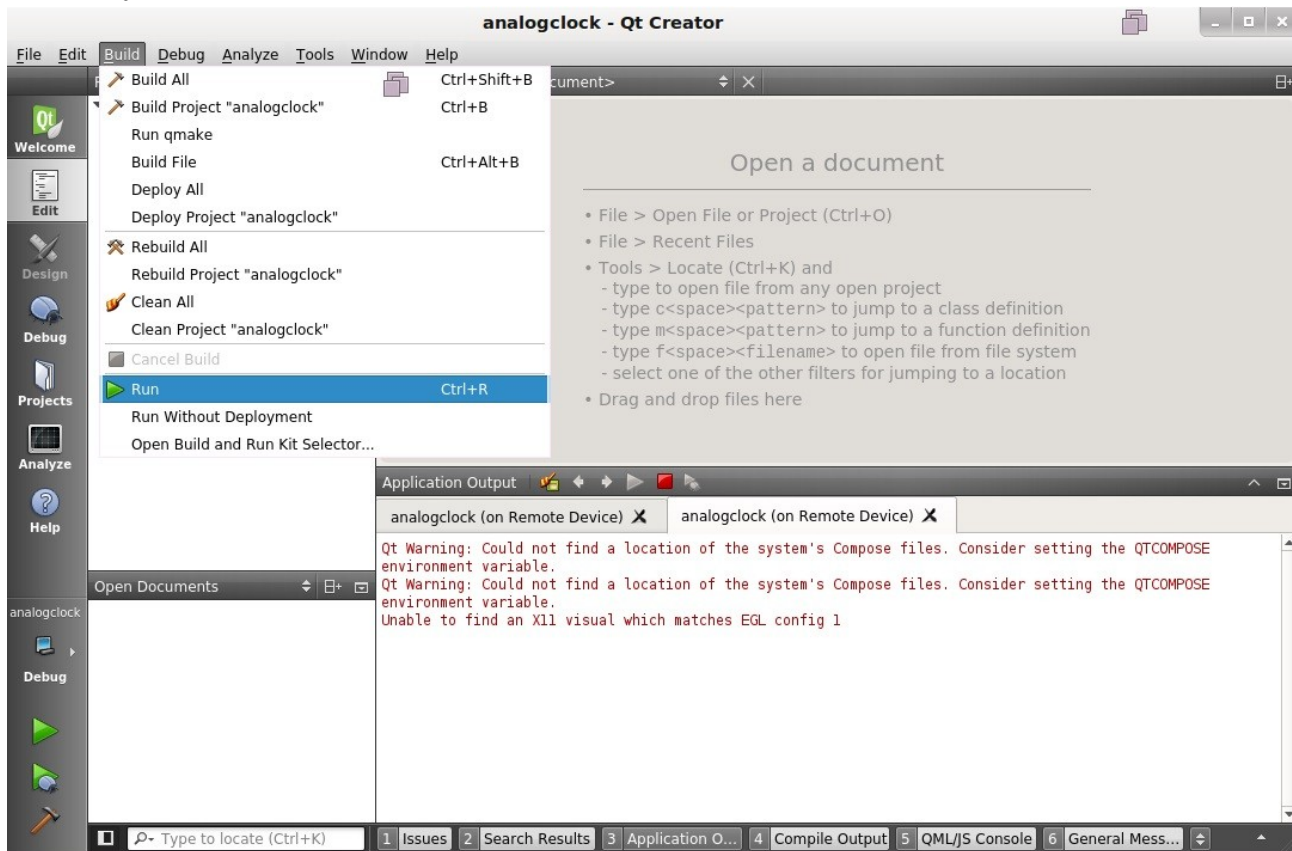


In the **Device type** column: select "**Generic Linux Device**"

In the **Device** column: select "**EDM**"

Click "**Apply**".

Back to the project view in QT Creator.
Press **"Run"**.



You will see the application runs on target board.

