

*Embedded based teaching pendant optimized for industrial robots.*

# **CG4300-DMC**

# **Software Development**

# **Environment Guide**

## **(R1) Version**

**DAINCUBE Corp.**  
ARM Cortex-A9 Linux system

**FORM 170703F – 2019.04.03**



### **CG4300-DMC**

### **Software Development Environment Guide**

DAINCUBE Corp.  
Web: [www.daincube.com](http://www.daincube.com)  
E-mail: [sales@daincube.com](mailto:sales@daincube.com)  
Tel: 82-32-329-9783~4  
Fax: 82-32-329-9785

#401-701, Bucheon TechnoPark 4-Danji,  
655 Pyeongcheon-ro, Wonmi-gu, Bucheon-Si,  
Gyeonggi-Do, Republic of Korea

**Copyright © 2005–2019 Daincube**  
**All rights reserved.**  
**Printed in the Republic of Korea**

# Preface

## Copyright notice

Copyright © 2005–2019 Daincube. All rights reserved.

Copying of this document, and giving it to others and the use or communication of the Contents thereof, is forbidden without express authority. Offenders are liable to the payment of damages.

All rights are reserved in the event of the grant of a patent or the registration of a utility model or design.

## Important information

**This documentation is intended for qualified audience only.**

**The product described herein is not an end user product.**

**It was developed and manufactured for further processing by trained personnel.**

**This product is protected by the "End-User License Agreement" (EULA).**

## Disclaimer

Although this document has been generated with the utmost care no warranty or liability for correctness or suitability for any particular purpose is implied. The information in this document is provided "as is" and is subject to change without notice.

## Trademarks

All used product names, logos or trademarks are property of their respective owners.

## Product support

DAINCUBE Corp.

Web: [www.daincube.com](http://www.daincube.com)

E - MAIL: [support@daincube.com](mailto:support@daincube.com)

# Safety precautions

Be sure to observe all of the following safety precautions.

Strict observance of these warning and caution indications are a MUST for preventing accidents, which could result in bodily injury and substantial property damage. Make sure you fully understand all definitions of these terms and related symbols given below, before you proceed to the manual.

# Symbols

The following symbols may be used in this specification:



## Warning

Warnings indicate conditions that, if not observed, can cause personal injury.



## Caution

Cautions warn the user about how to prevent damage to hardware or loss of data.



## Note

Notes call attention to important information that should be observed.

## Revision History

[illegible]

# Contents

<b>1. How to build a development environment.....</b>	<b>5</b>
1.1. Host Requirements .....	5
1.1.1. Ubuntu 14.04.5 32bit .....	5
1.2. Ubuntu settings .....	5
1.2.1. Ubuntu login .....	5
1.2.2. Ubuntu terminal .....	6
1.2.3. Host package installation .....	6
1.2.4. Samba server setup (If necessary) .....	7
1.2.5. ssh server setup (If necessary) .....	7
1.2.6. tftp server setup (If necessary) .....	7
1.2.7. nfs server setup (If necessary) .....	7
1.3. Build tool setup .....	8
1.3.1. Qt Creator setup .....	8
1.3.2. Qt Creator 3.5.1 setup .....	8
1.4. Qt GUI application project .....	13
1.4.1. Debugging a Qt GUI application .....	13
1.4.2. Create a Qt GUI application (Hello world) .....	18
1.4.3. Xenomai real-time project .....	23
<b>2. EtherCAT Example .....</b>	<b>25</b>
2.1. CG4300-DMC and Servo Driver Connection .....	25
2.2. EtherCAT Sample application description and usage .....	27
2.2.1. EtherCAT Master & sample application execution .....	27
2.2.2. Terminating the EtherCAT Master & sample application .....	28
2.2.3. EtherCAT sample application argument .....	28
2.2.4. EtherCAT sample application command .....	28
2.2.5. EtherCAT sample application build .....	28
2.3. Qt EtherCAT GUI example .....	28
2.3.1. CG4300-DMC and servo driver connection .....	28
2.3.2. Running the EtherCAT Master & Qt GUI sample application .....	28
<b>3. FAQ .....</b>	<b>31</b>
3.1. Default password .....	31

# 1. How to build a development environment

## 1.1. Host Requirements

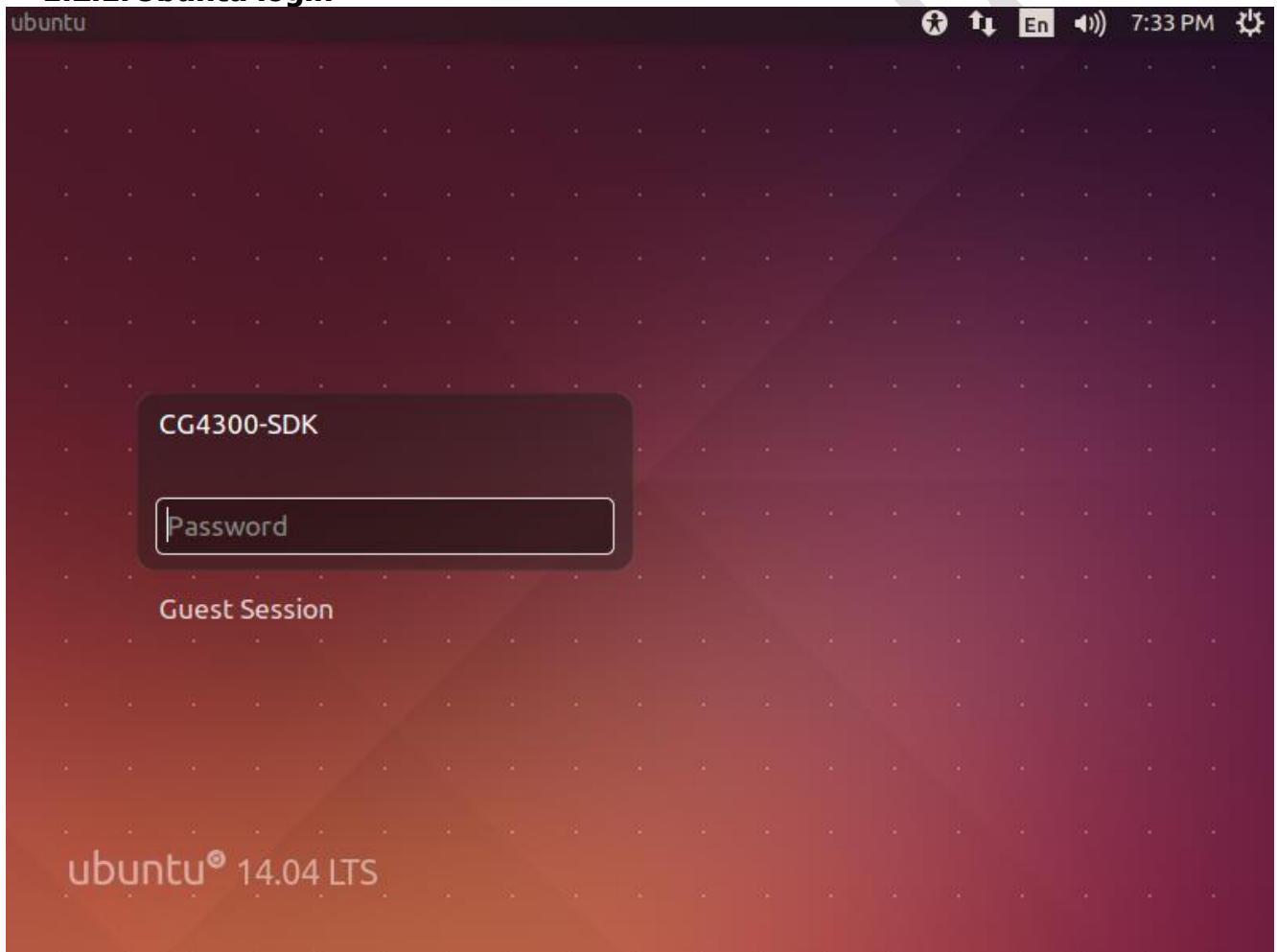
### 1.1.1. Ubuntu 14.04.5 32bit

1. Describes building a development environment based on Ubuntu 14.04.5.

2. In Ubuntu 14.04 or later, development environment was not built. If you want to proceed to a version other than 14.04, you may encounter a lot of problems and many additional settings may be needed to fix it. There are many different versions and versions of Linux distributions, and we can not support all of them, so you will have to solve the ones that are not covered in this manual yourself. Please note. **Also, in using vmware image provided, vmware license is not provided by DainCube Co., Ltd. We are not responsible for any problems caused by license.**

## 1.2. Ubuntu settings

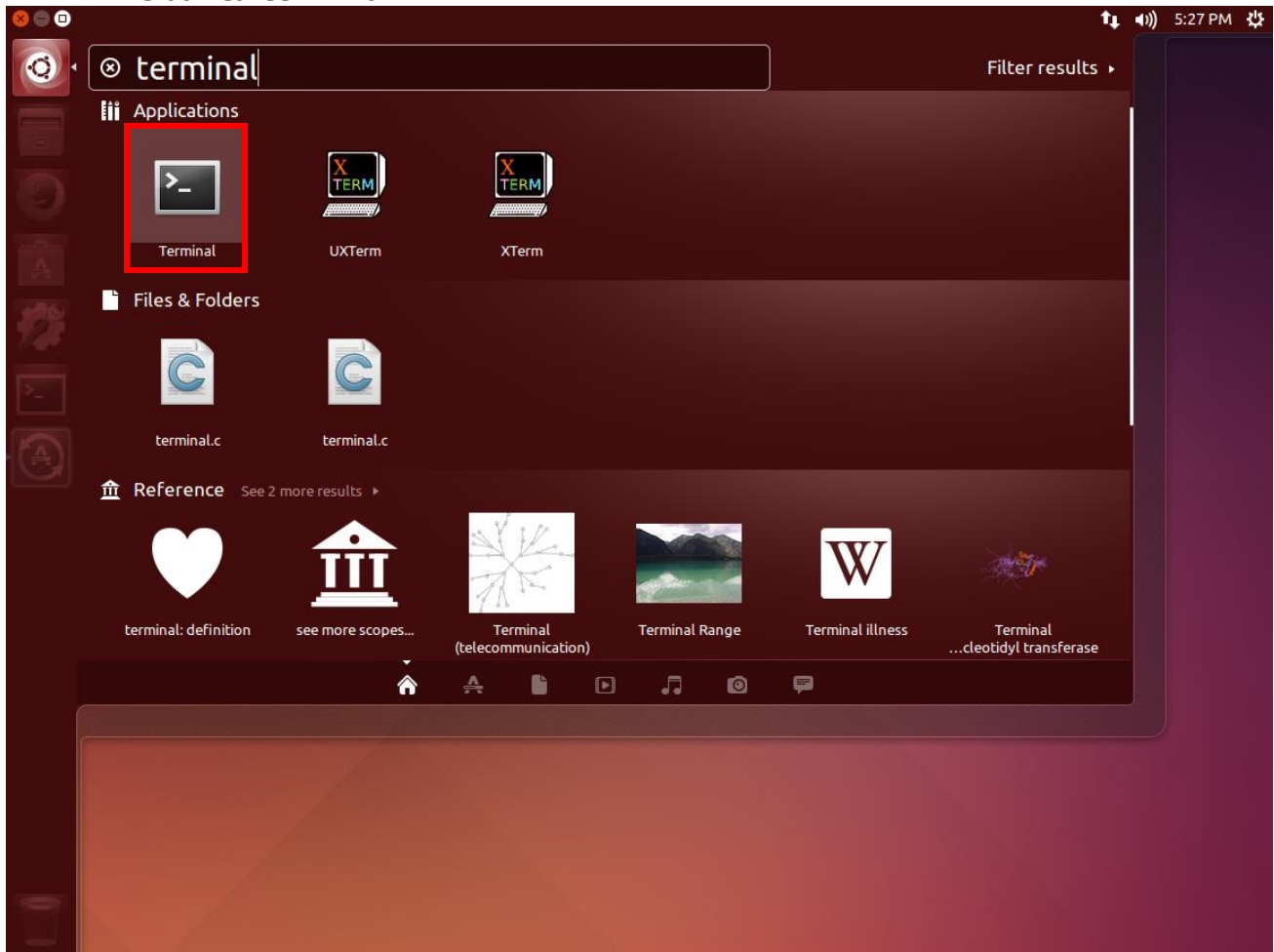
### 1.2.1. Ubuntu login



Log in to the already installed CG4300-SDK Ubuntu. Password is "daincube".

For administrator accounts, the password is the same as "daincube".

### 1.2.2. Ubuntu terminal



In order to enter the command line in Ubuntu, you need to run Terminal.

### 1.2.3. Host package installation

After installing Ubuntu, install the utilities and libraries by typing the following command at the console. These are the programs you need to develop. (However, vmware I have provided is already installed.)

```
$sudo apt-get install asciidoc autoconf automake bison build-essential  
$sudo apt-get install ccache chrpath coreutils curl cvs desktop-file-utils  
$sudo apt-get install diffstat docbook-utils dpkg g++ gawk gcc gcc-multilib  
$sudo apt-get install gettext git-core groff help2man intltool  
$sudo apt-get install libdbus-glib-1-dev libfreetype6-dev  
$sudo apt-get install libgl1-mesa-dev libglu1-mesa-dev libgtk2.0-dev liblzo2-dev  
$sudo apt-get install liborbit2-dev libsdl1.2-dev libtool lzop  
$sudo apt-get install m4 make mercurial ncurses-dev python-pysqlite2  
$sudo apt-get install python-xcbgen rpm scrollkeeper sed socat subversion  
$sudo apt-get install tcl texi2html texinfo  
$sudo apt-get install unzip uuid-dev wget xsltproc xterm zlib1g zlib1g-dev
```

#### 1.2.4. Samba server setup (If necessary)

If Samba (file sharing utility) is installed, development becomes easier. If necessary, install and set up as below.

```
$sudo ufw disable
```

```
$sudo apt-get install samba
```

```
$sudo vi /etc/samba/smb.conf
```

```
[/etc/samba/smb.conf]  
[global]  
    hosts allow = 192.168. 192.168. 127.  
  
##### Authentication #####  
    guest ok = yes  
    guest account = root  
  
[home]  
    path = /home  
    writeable = yes  
    guest ok = yes
```

```
$sudo service samba restart
```

```
$sudo shutdown -r now
```

#### 1.2.5. ssh server setup (If necessary)

You can also connect to ssh client.

```
$sudo apt-get install openssh-server
```

```
$sudo gedit /etc/ssh/sshd_config
```

```
$sudo /etc/init.d/ssh restart
```

#### 1.2.6. tftp server setup (If necessary)

Please install it if necessary.

```
$sudo apt-get install tftpd-hpa
```

```
$sudo vim /etc/default/tftpd-hpa
```

```
$sudo service tftpd-hpa restart
```

#### 1.2.7. nfs server setup (If necessary)

Please install it if necessary.

```
$sudo apt-get install nfs-common nfs-kernel-server portmap
```

```
$sudo vi /etc/exports
```

```
$sudo /etc/init.d/nfs-kernel-server restart
```

```
$sudo /etc/init.d/portmap restart
```

## 1.3. Build tool setup

### 1.3.1. Qt Creator setup

The vmware I have provided has Qt4.8.7 with the build completed. It is built with the agreement of LGPL V2 agreement. If you do not agree, you have to build and install Qt4.8.7 directly.

※ **If you use Qt 4.8.7 which is build without consent, you will notice that there is no license problem that can occur with DainCube Co.,Ltd.**

If you agree, unzip Qt4.8.7.tar.gz in / home / daincube / Desktop to the / home / core folder.

```
$tar -xvf Qt4.8.7.tar.gz -C /home/core/
```

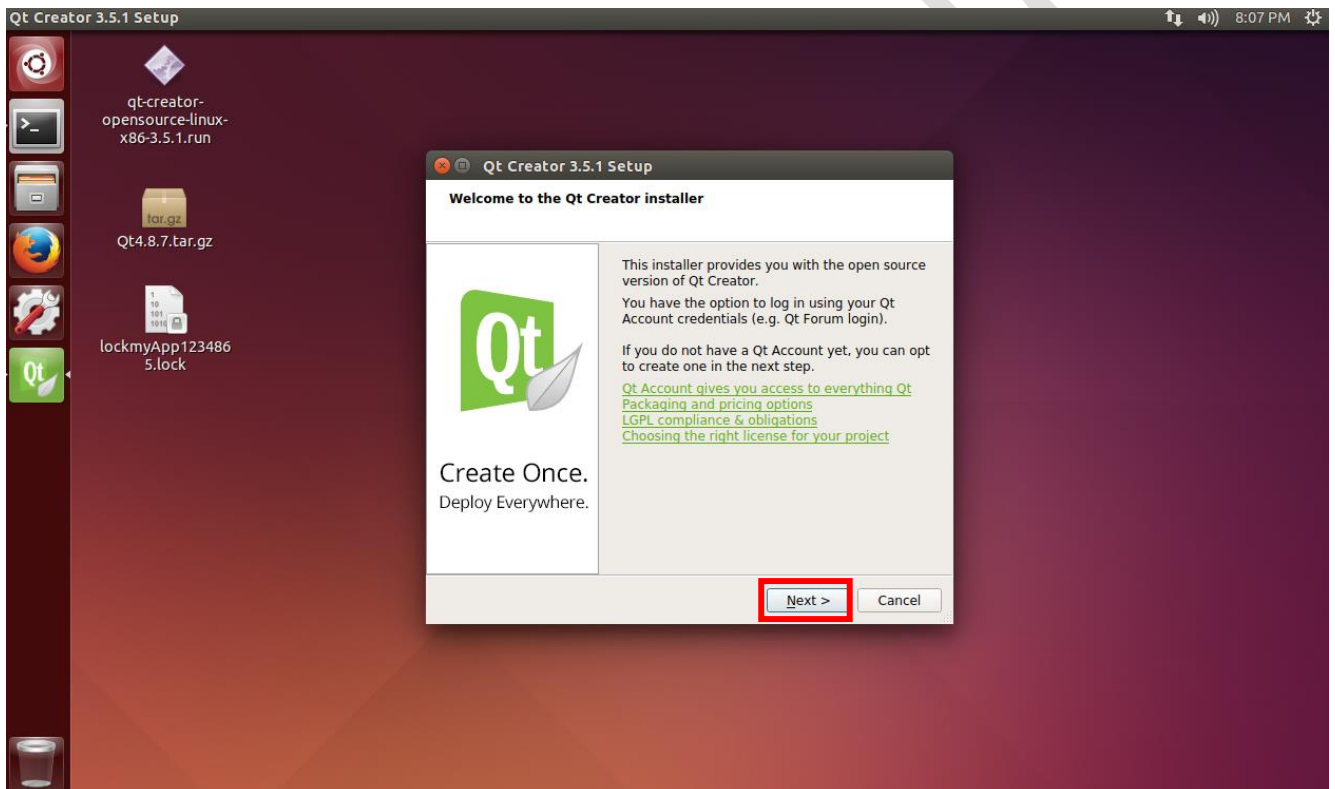
### 1.3.2. Qt Creator 3.5.1 setup

Install qt-creator-opensource-linux-x86-3.5.1.run located in /home/daincube/Desktop.

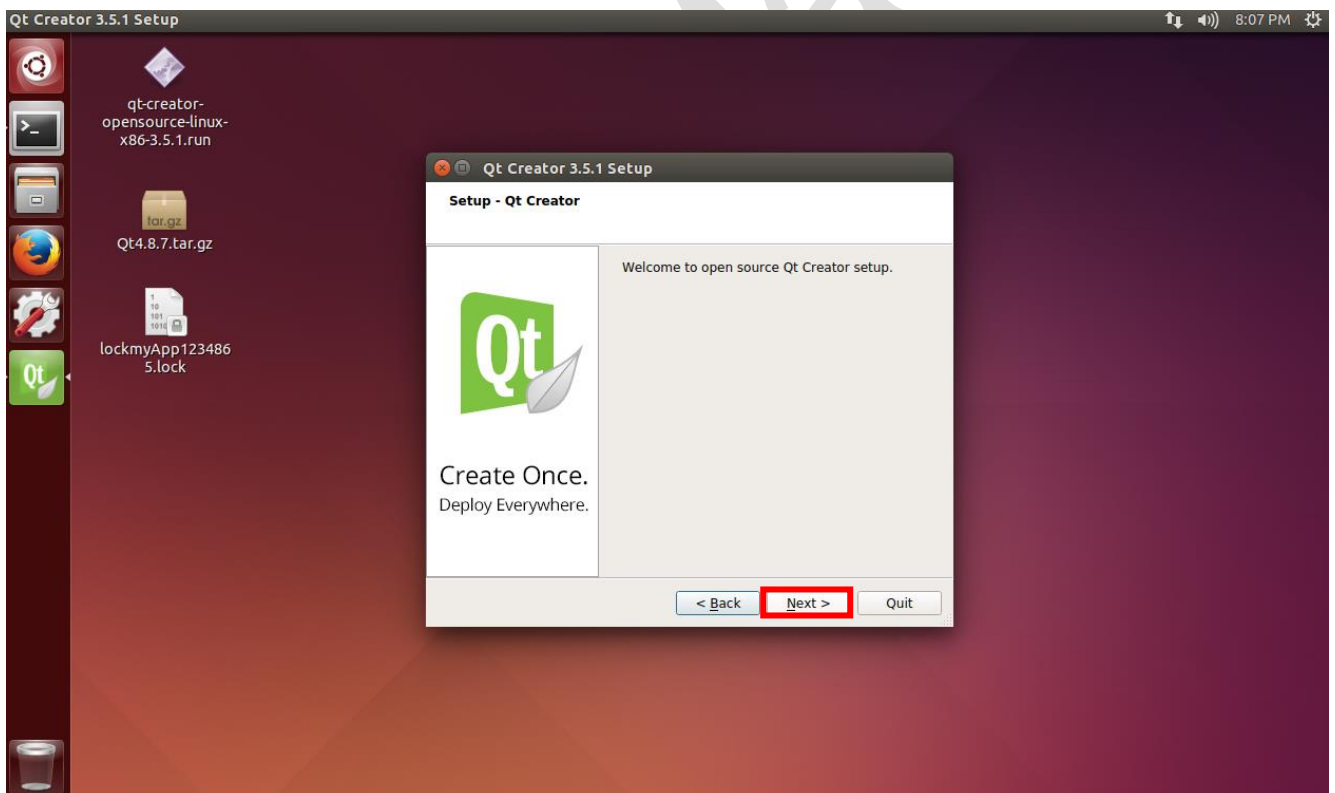
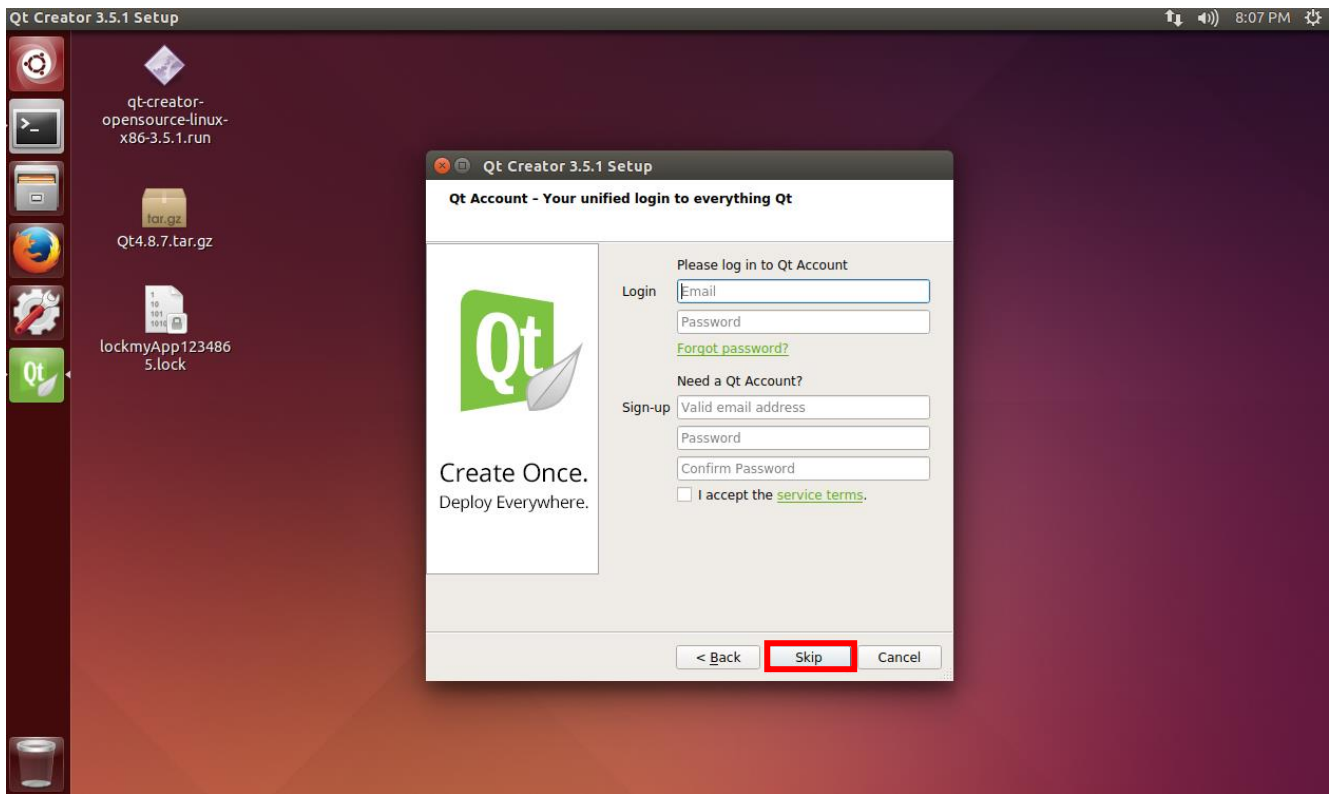
After giving permission to the installation file, execute the following command.

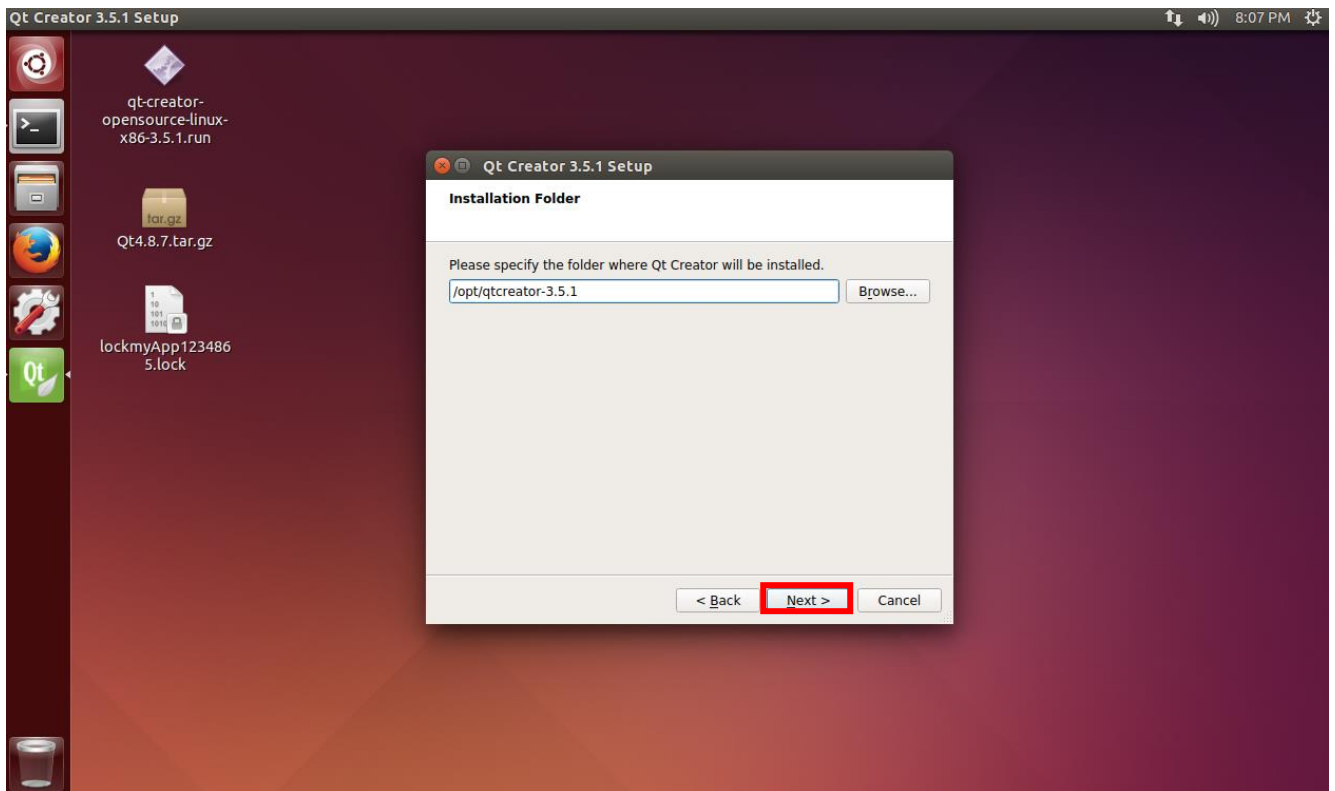
```
$chmod a+x /qt-creator-opensource-linux-x86-3.5.1.run
```

```
$/qt-creator-opensource-linux-x86-3.5.1.run
```

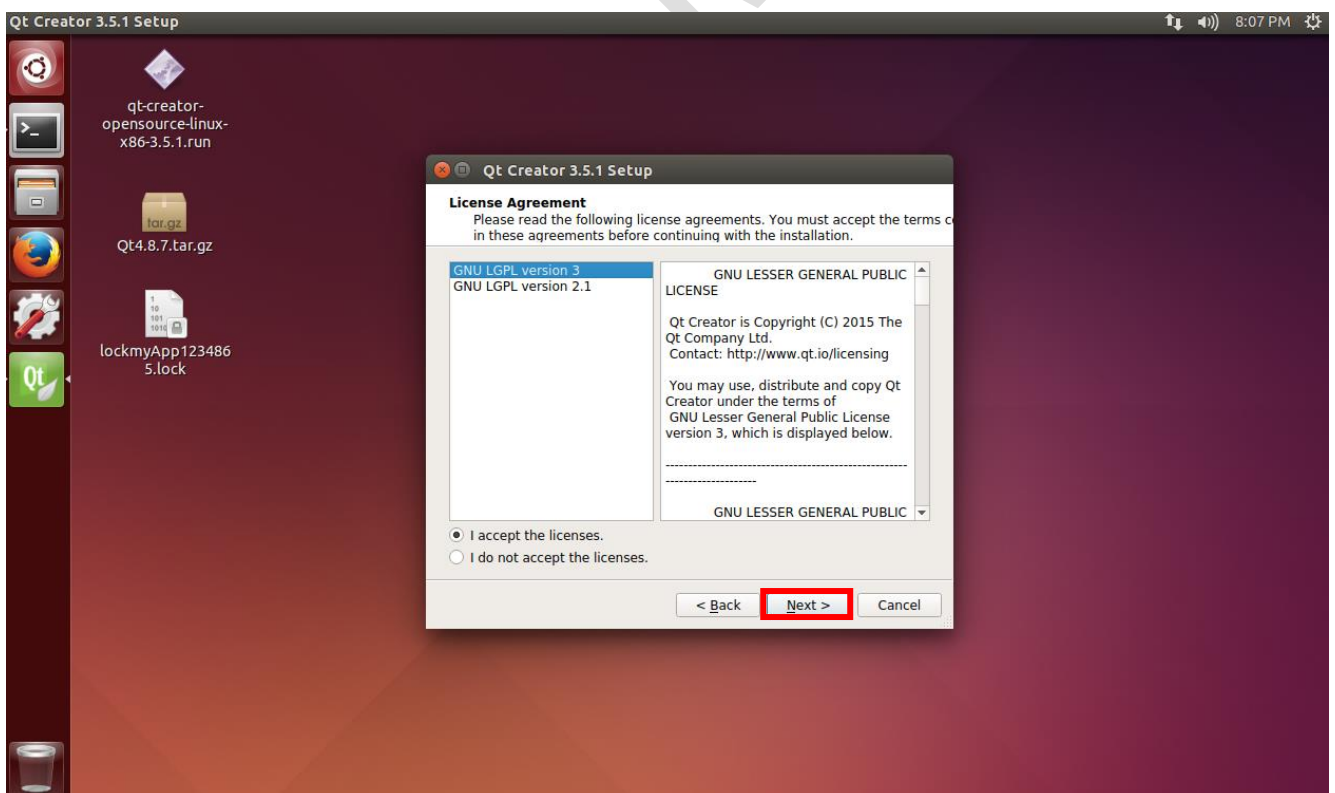


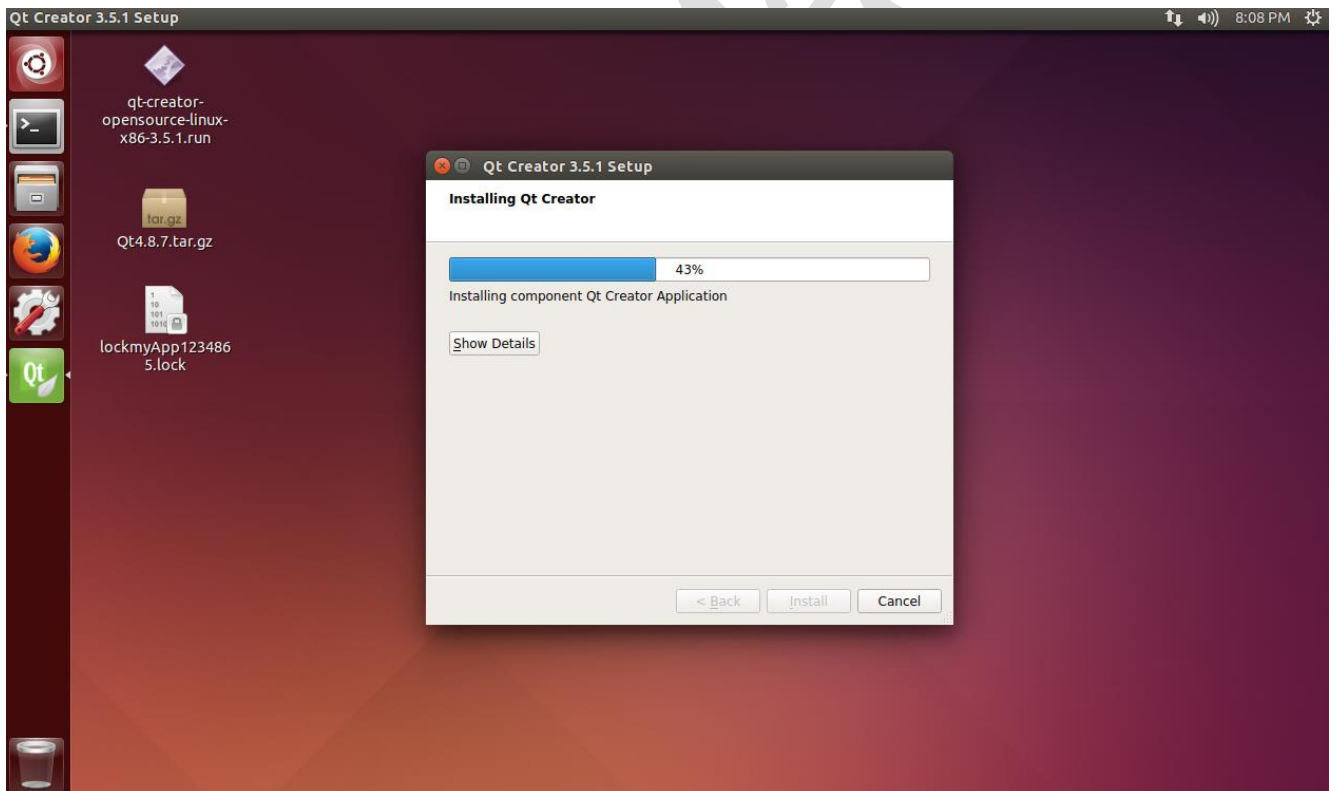
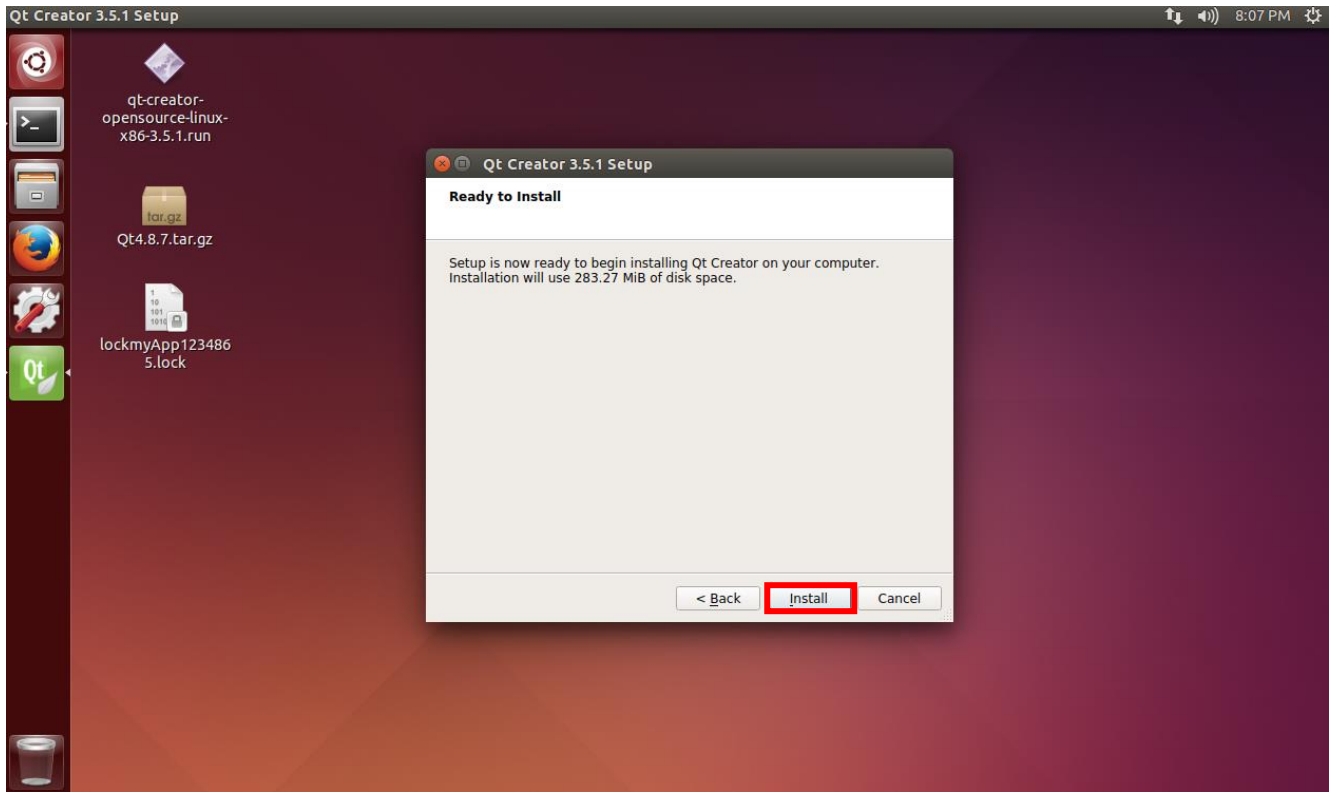


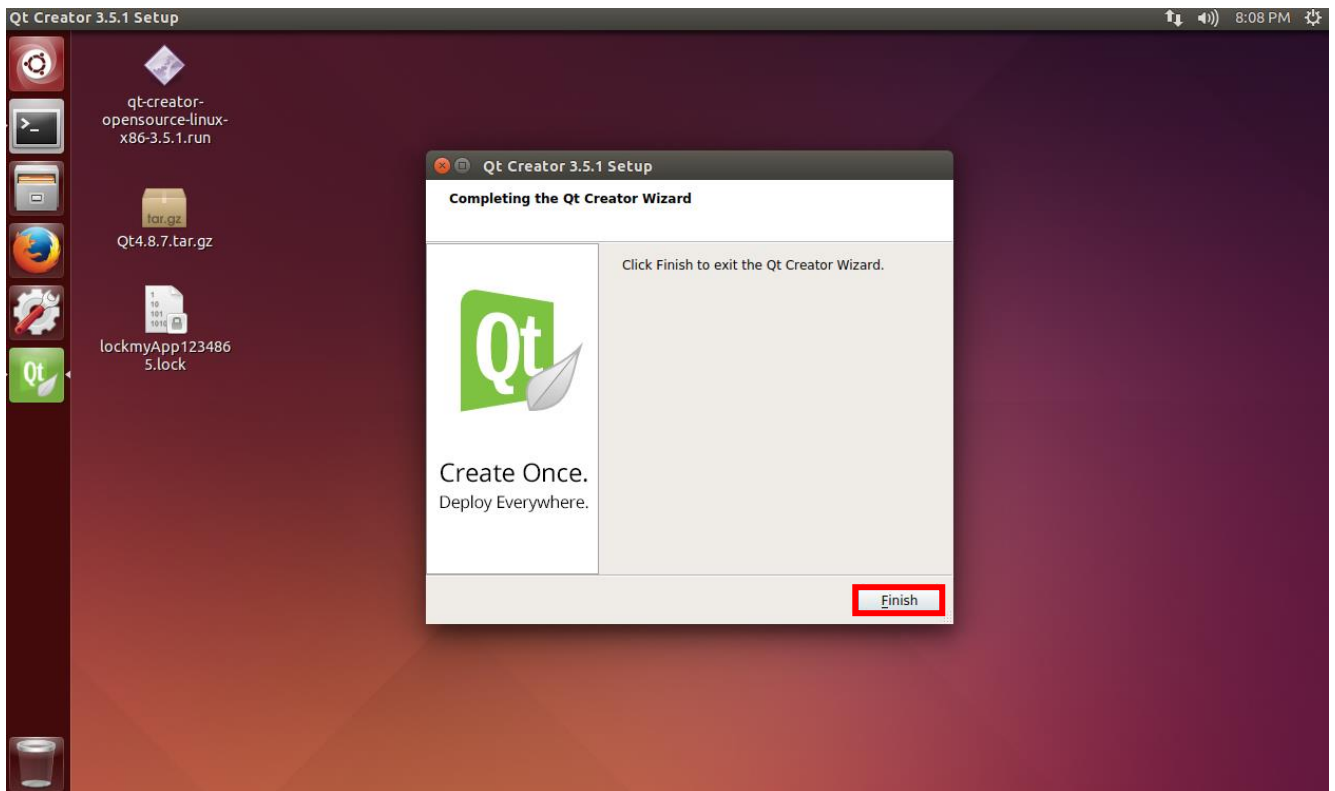




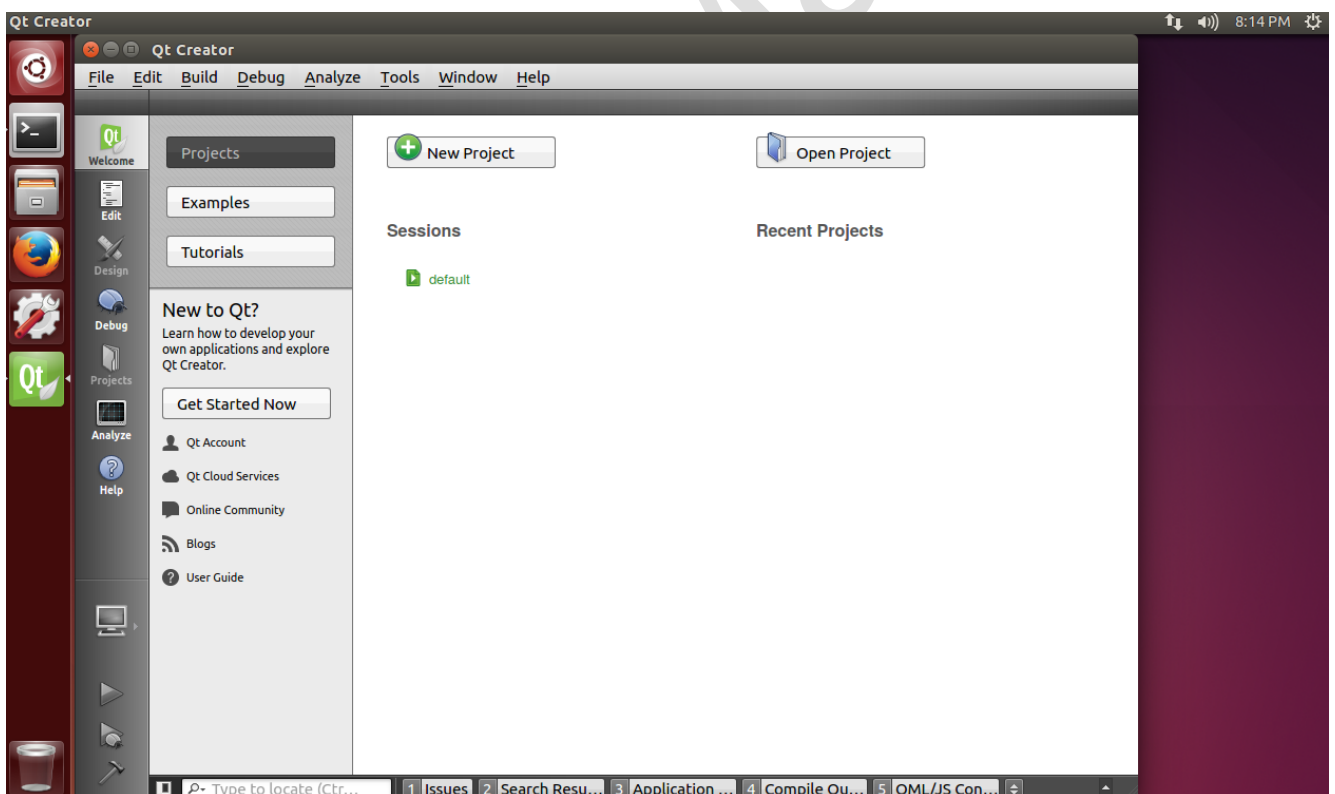
✂ You must agree to the License Agreement when installing. This manual guides you how to install under the assumption that you agree to LGPL v2, LGPL v3. In case of licensing problems, Daincube Co., Ltd. will not be liable.







Execute Qt Creator 3.5.1.

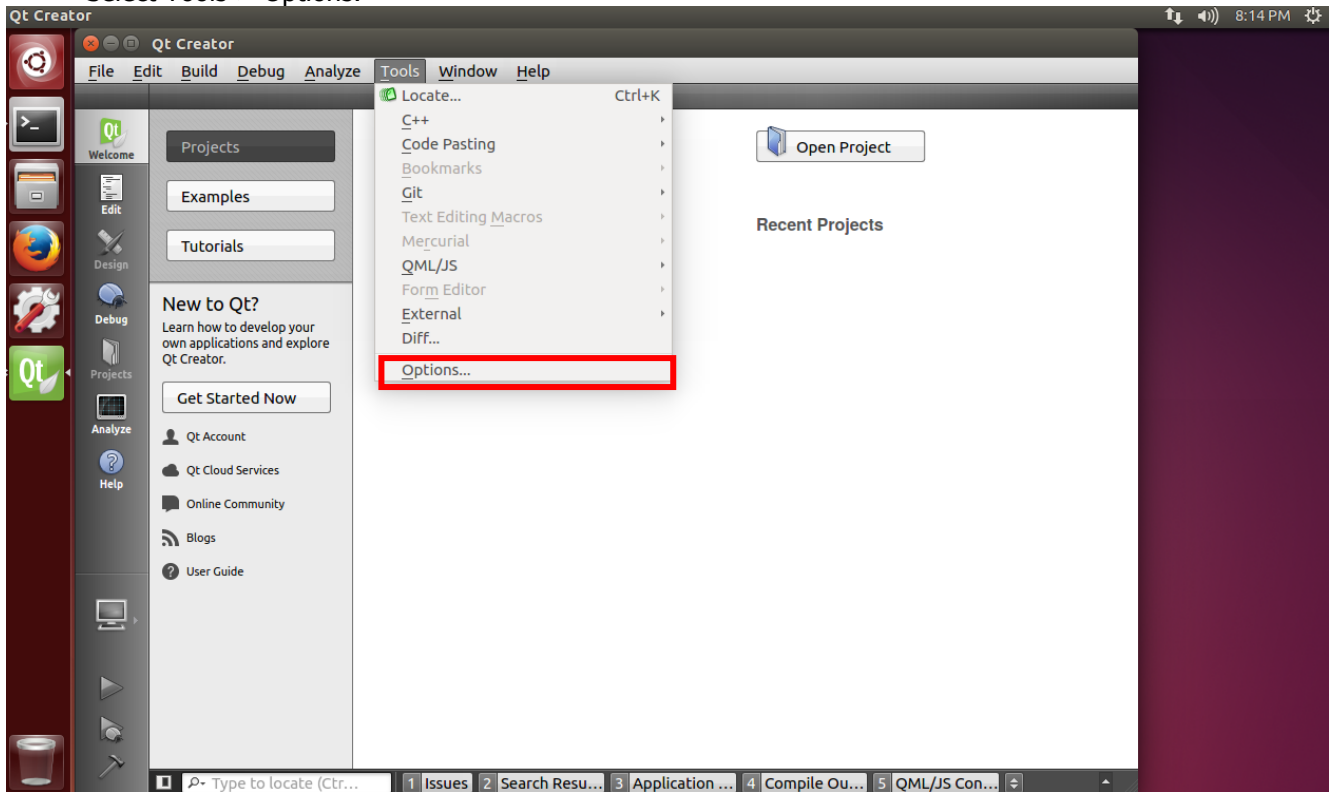


## 1.4. Qt GUI application project

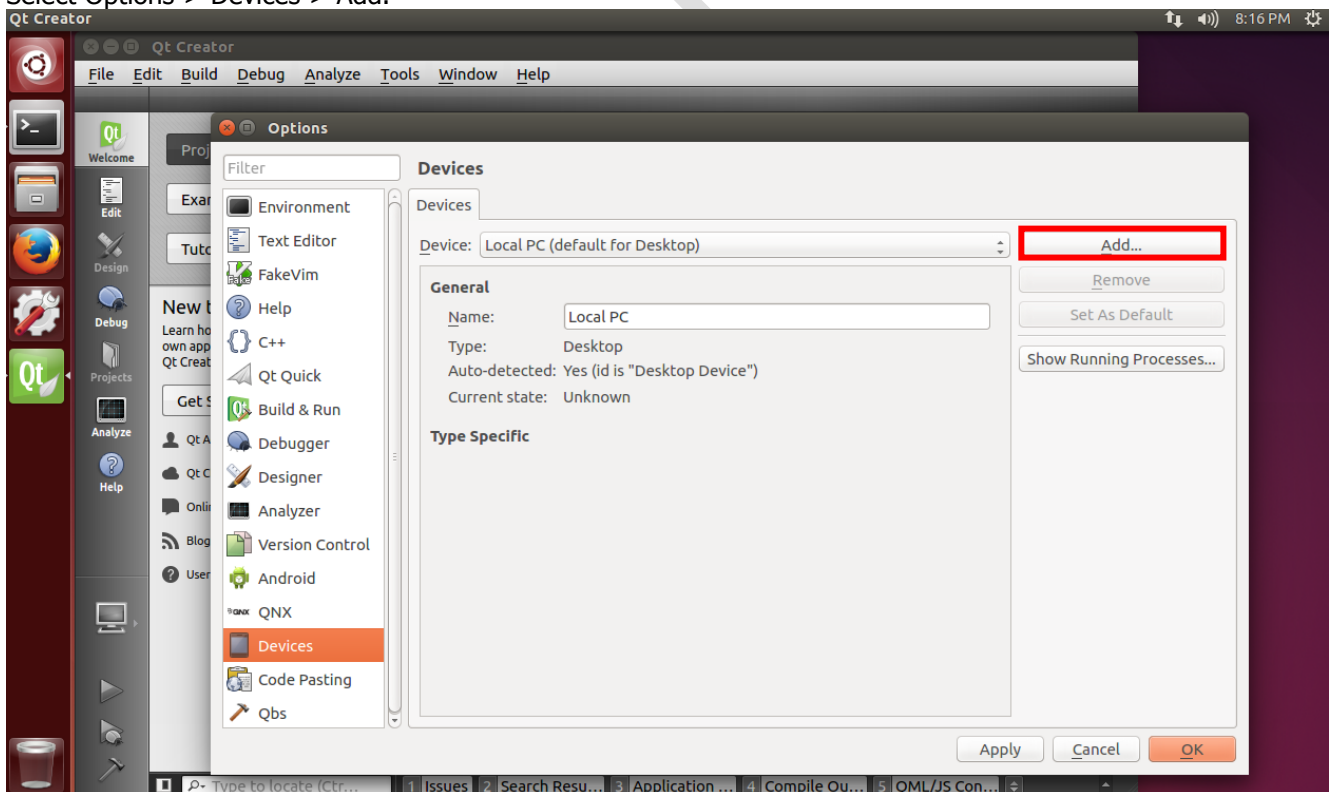
### 1.4.1. Debugging a Qt GUI application

Some settings are required for debugging as shown below.

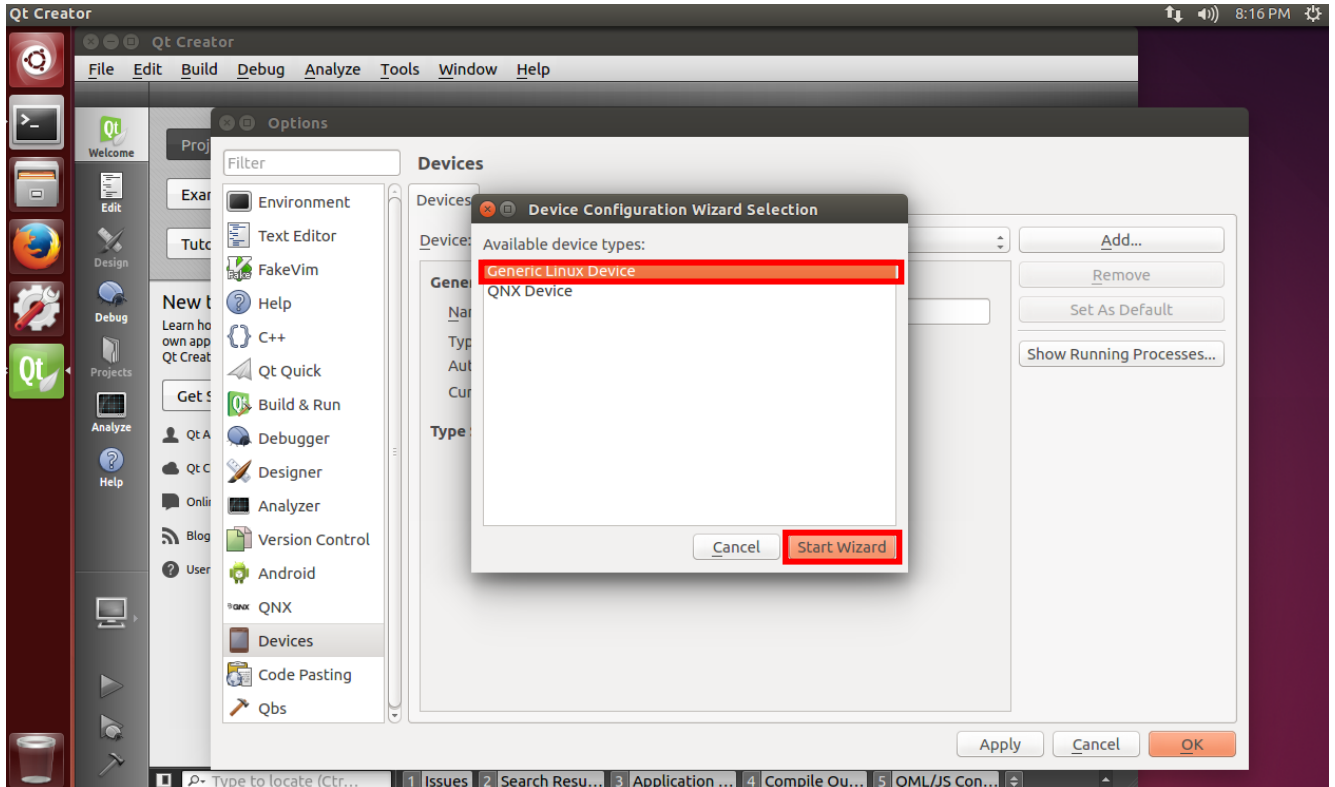
Select Tools > Options.



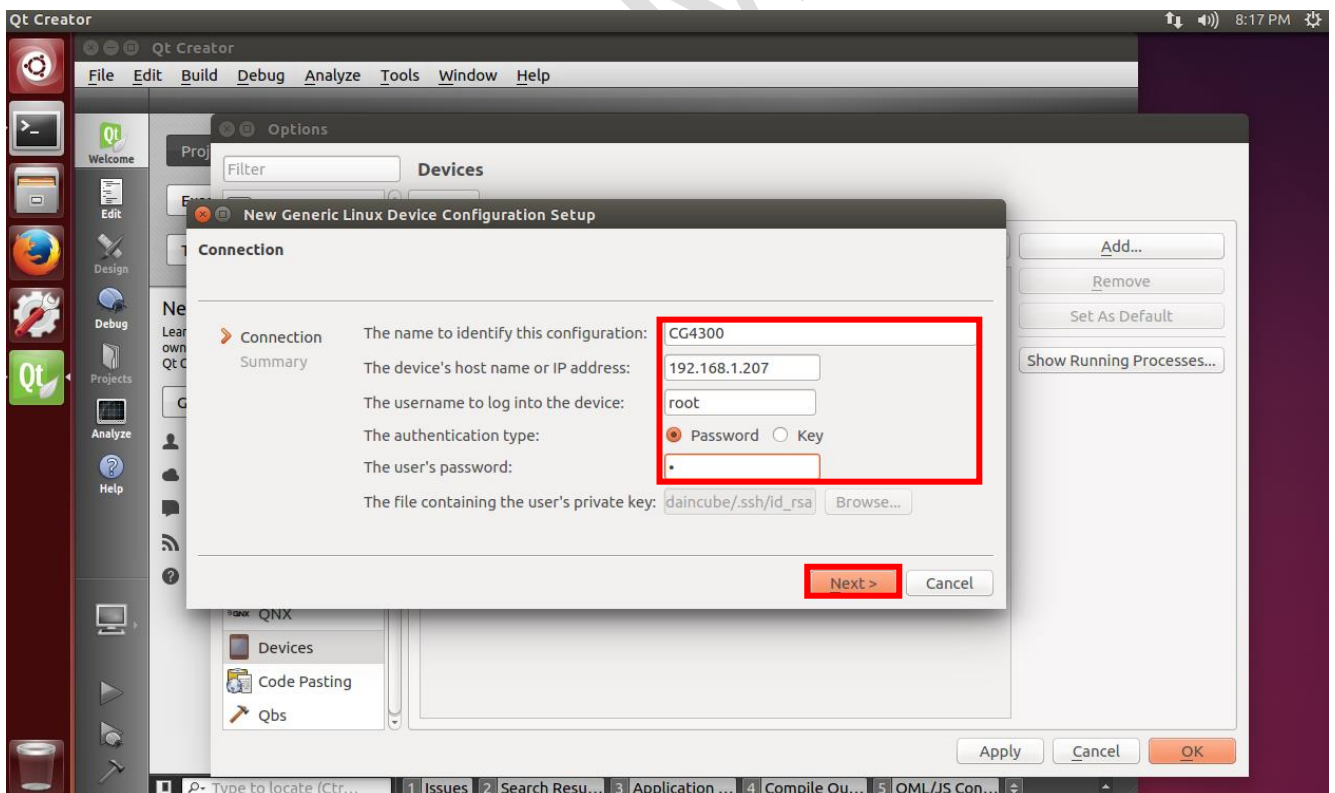
Select Options > Devices > Add.



Select Generic Linux Device and Click Start Wizard.

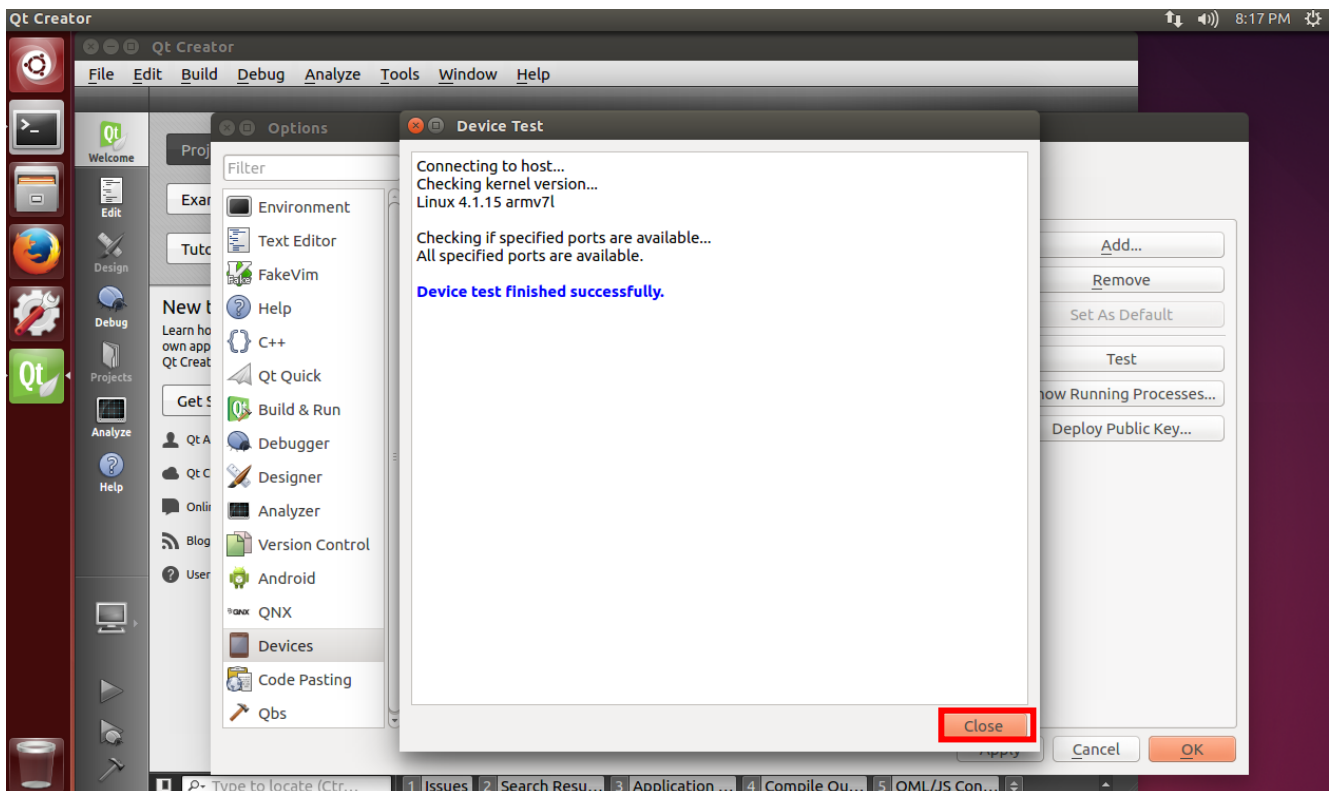


Input the device information. Default ip is "192.168.1.207" and password is "1". After completing the entry, click Next.

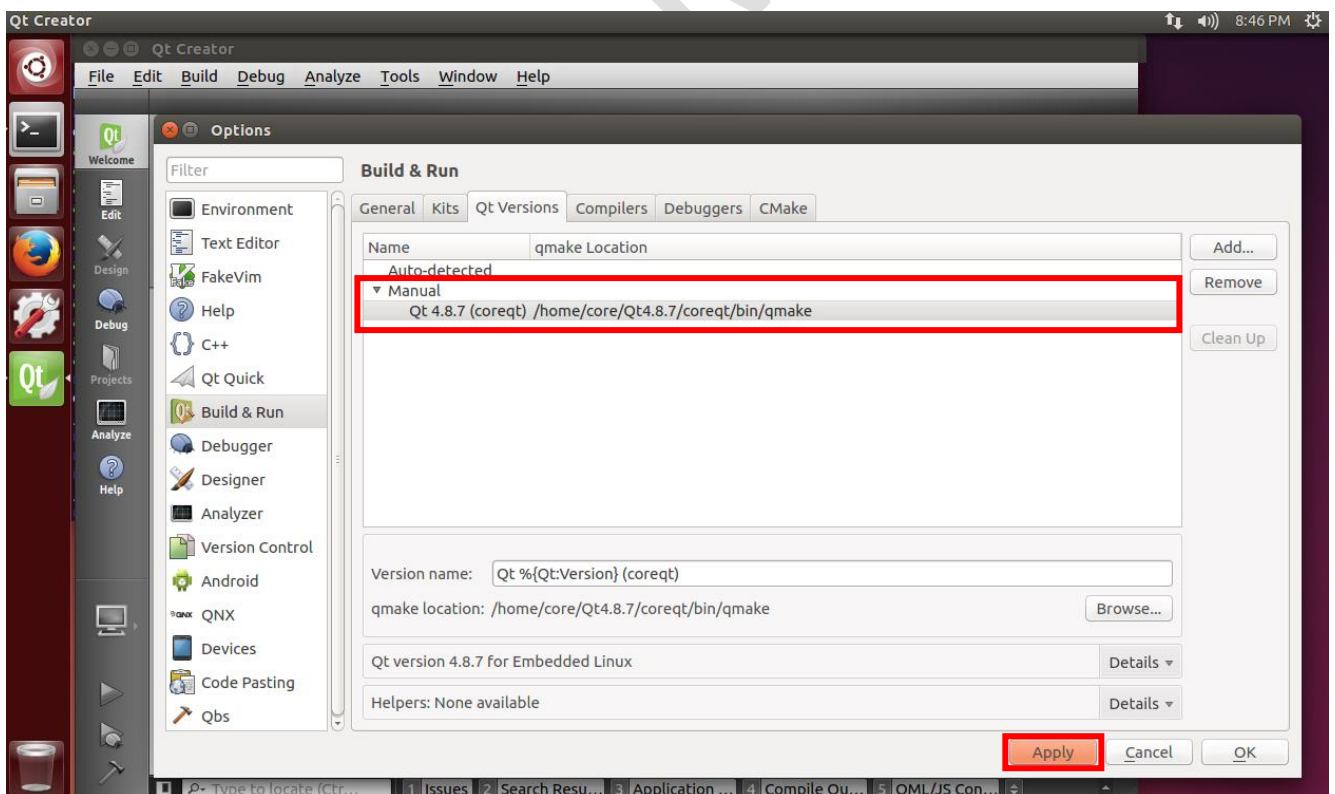




Test connection with Device. When you are finished successfully, click Close.

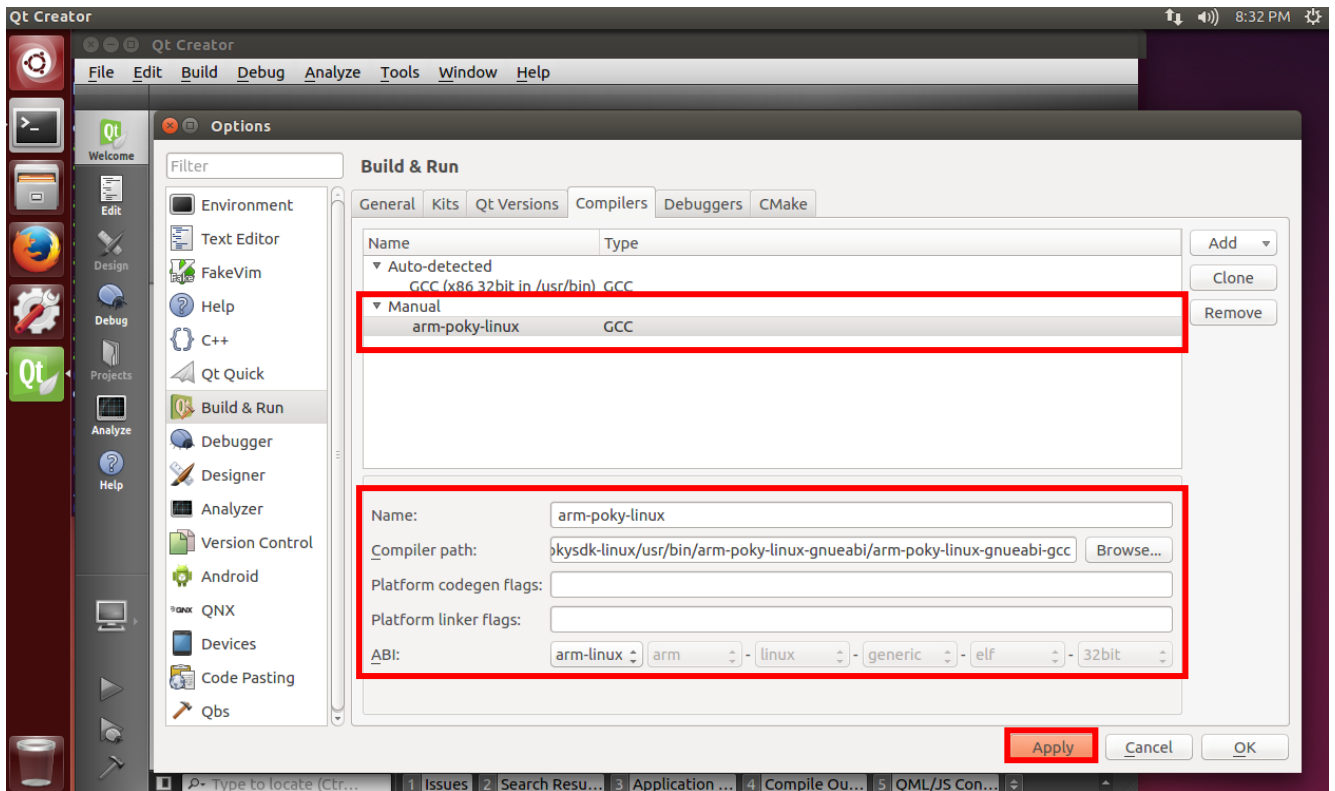


In Option> Build & Run> Qt Versions> Add, specify the location of qmake.  
(Path : /home/core/Qt4.8.7/coreqt/bin)



Under Options> Build & Run> Compilers> Add, select GCC, specify the compiler name and location, and click Apply.

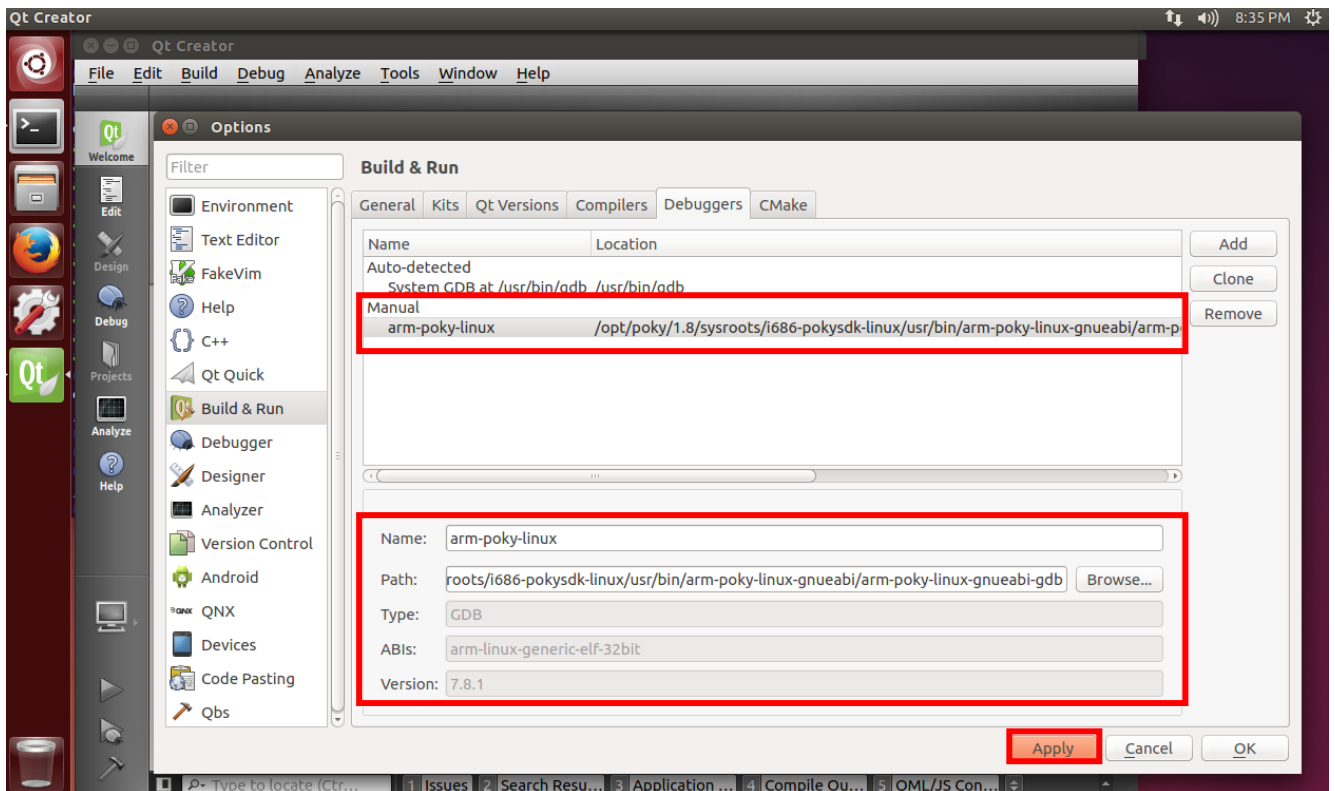
(Path : /opt/poky/1.8/sysroots/i686-pokysdk-linux/usr/bin/arm-poky-linux-gnueabi/arm-poky-linux-gnueabi-gcc)



Select Debugger under Options> Build & Run> Debuggers> Add, specify a name and location, and click Apply.

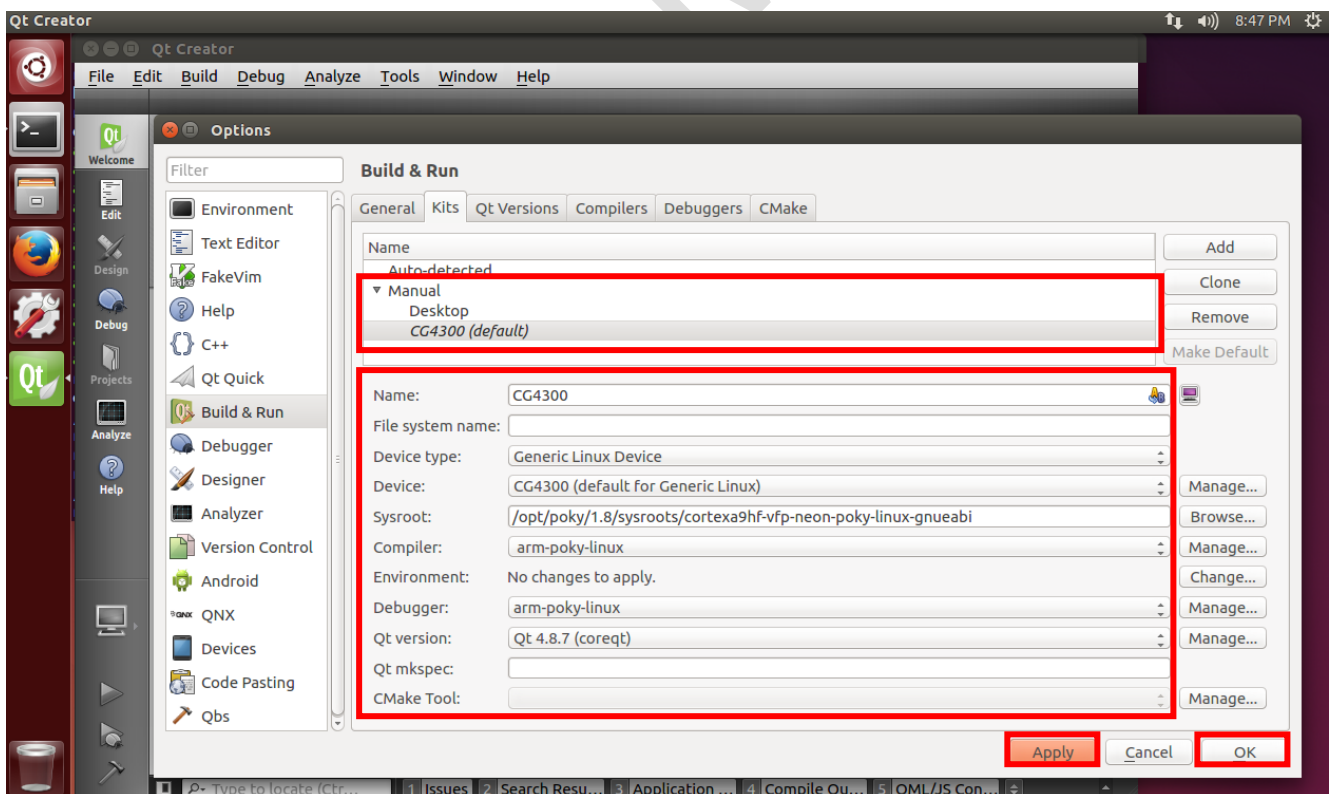
(Path : /opt/poky/1.8/sysroots/i686-pokysdk-linux/usr/bin/arm-poky-linux-gnueabi/arm-poky-linux-gnueabi-gdb)





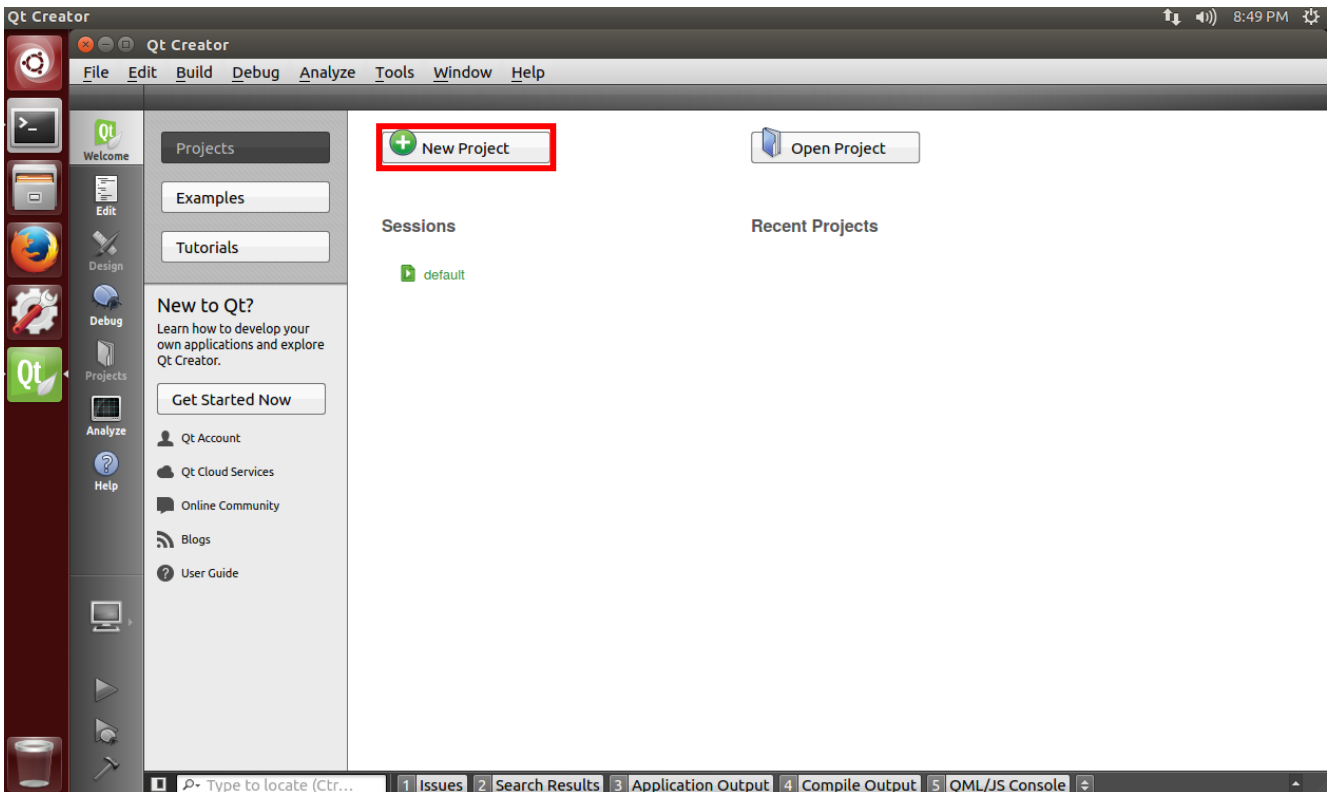
In Options> Build & Run> Kits> Add, select the name and sysroot path, Qt version, Compiler and Debugger that you set up above, click Make Default, and click Apply, OK.

(sysroot Path : /opt/poky/1.8/sysroots/cortexa9hf-vfp-neon-poky-linux-gnueabi)

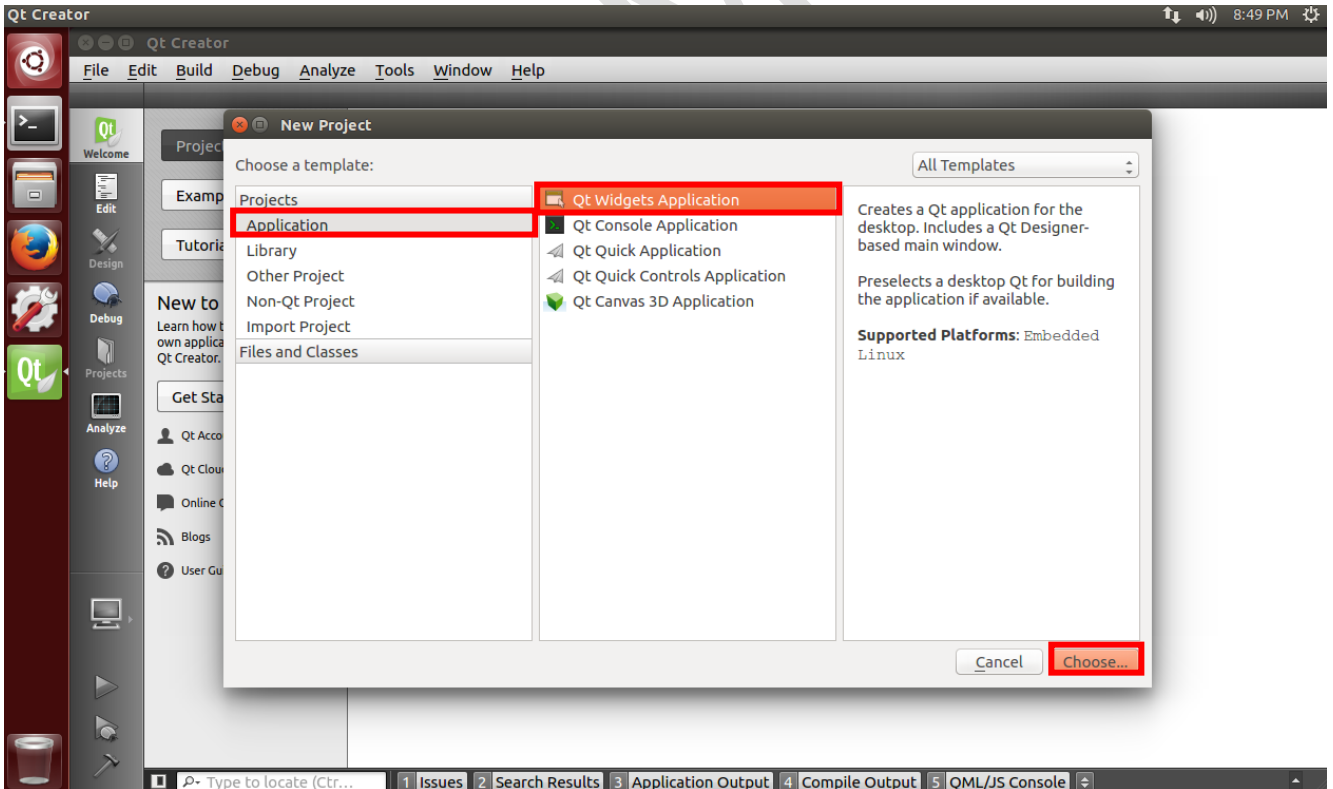


### 1.4.2. Create a Qt GUI application (Hello world)

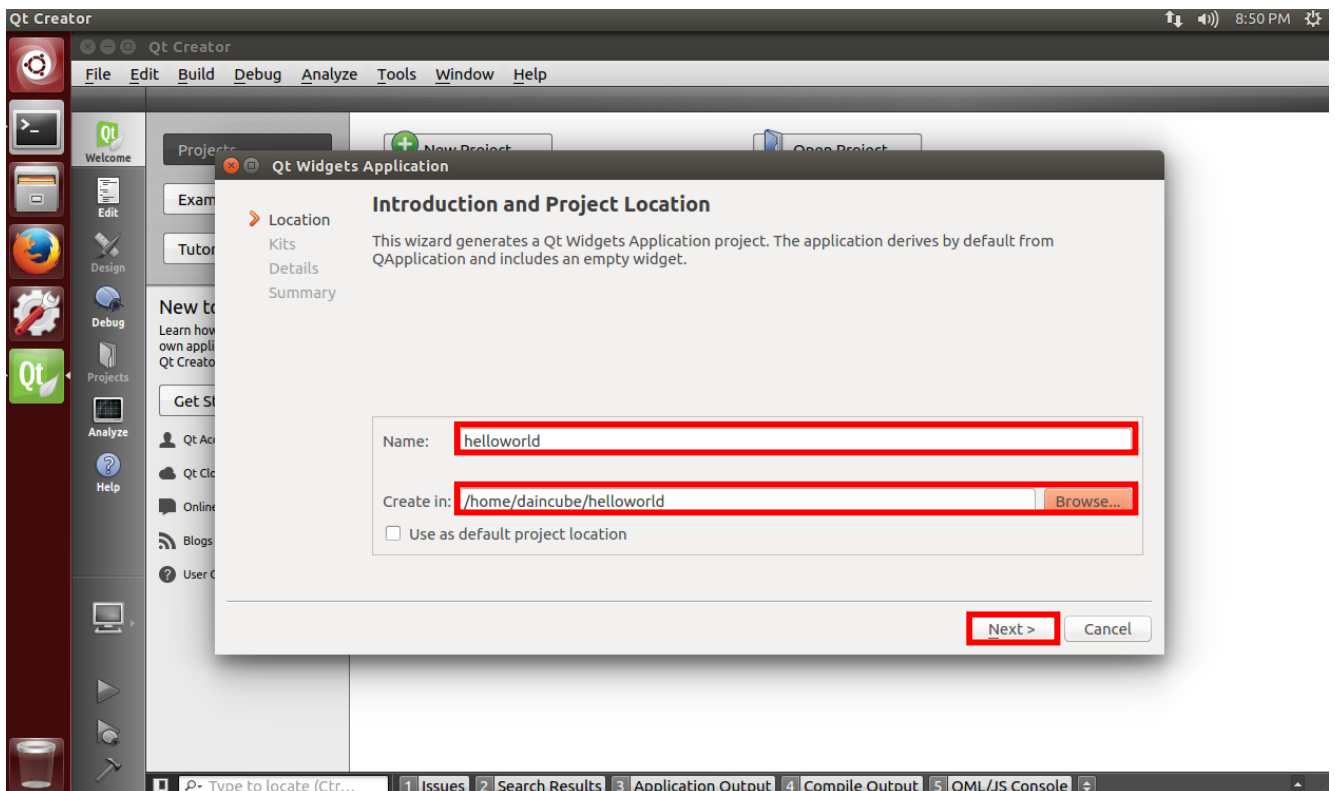
Select File > New Project and create project.



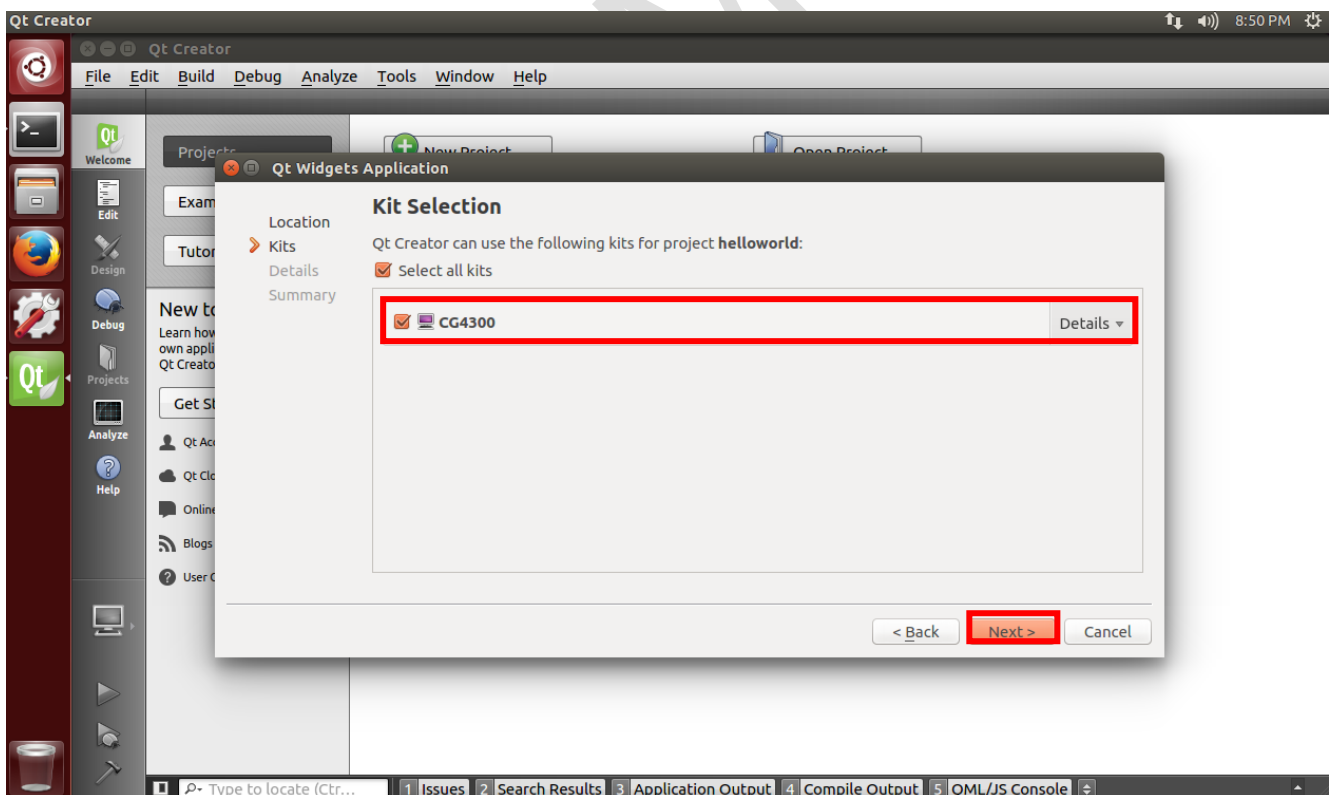
Select Projects > Application > Qt widgets Application and click Choose.



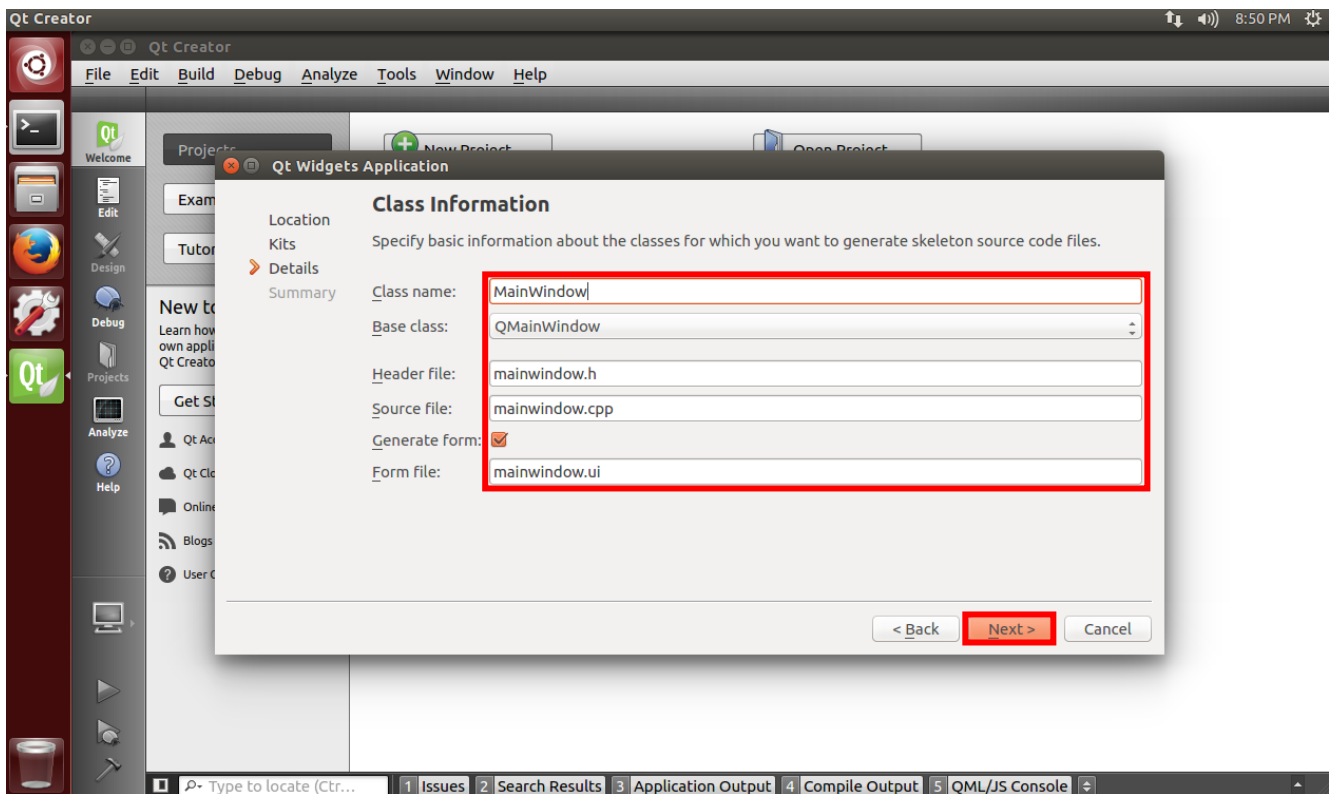
Input project name and path, click Next.



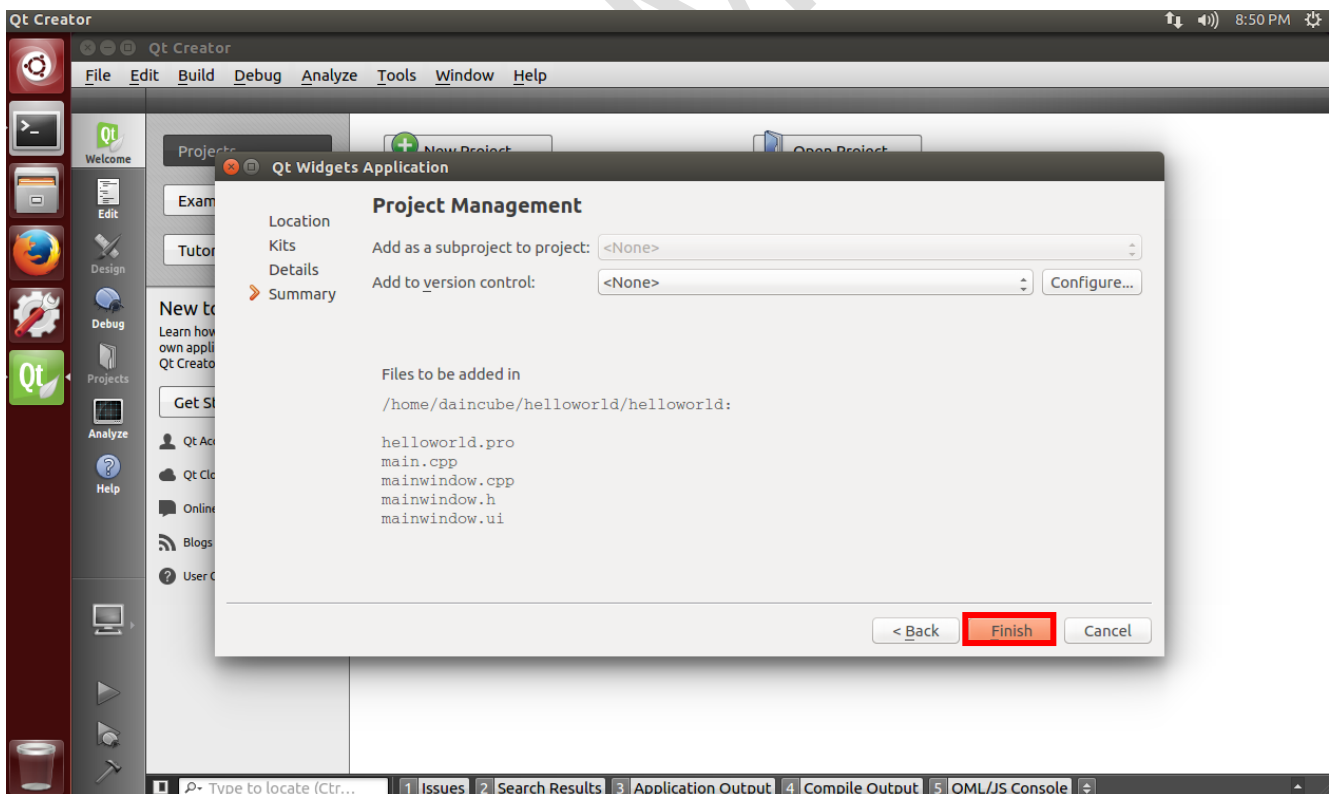
Select CG4300 in "Kits Selection".



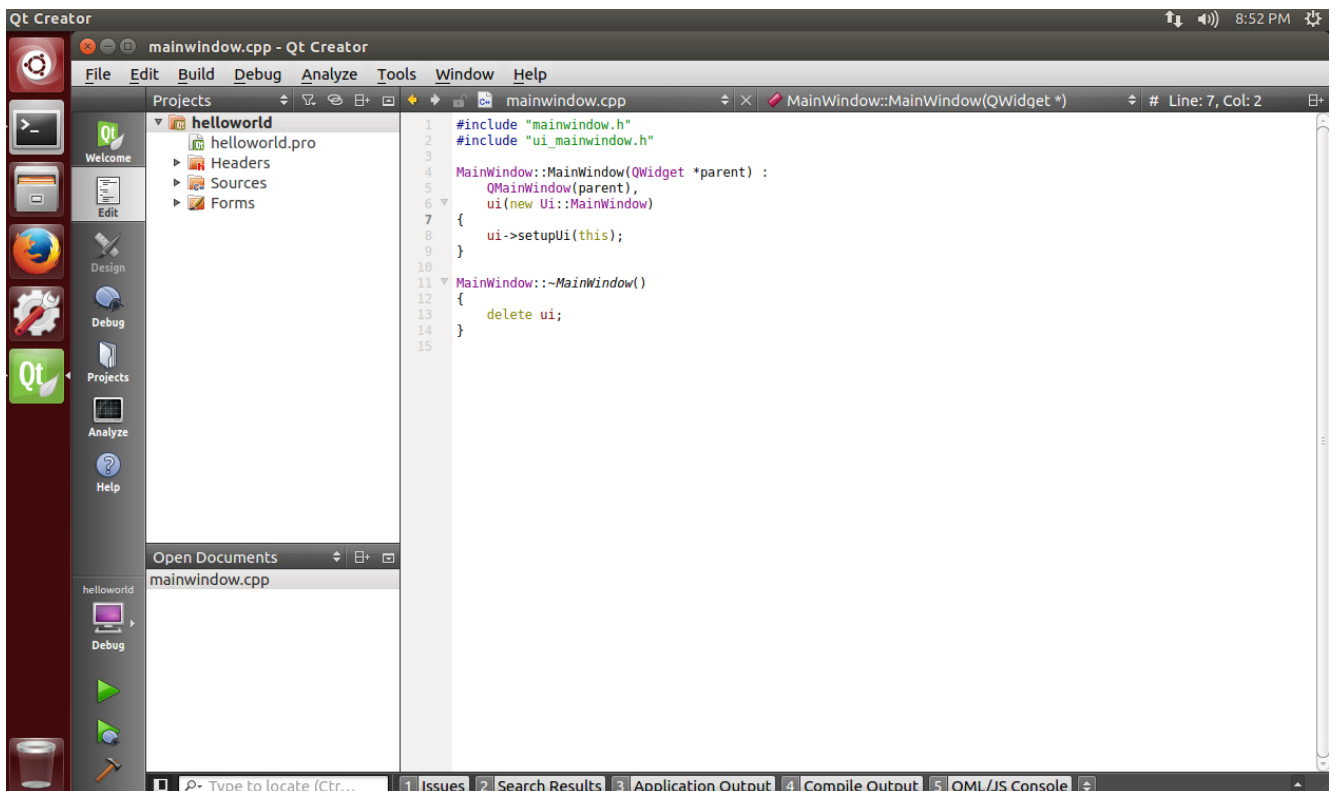
Setting for name in "Class information".



Setting for version control tool (SVN, GIT...). If set for version control tool, click Finish.

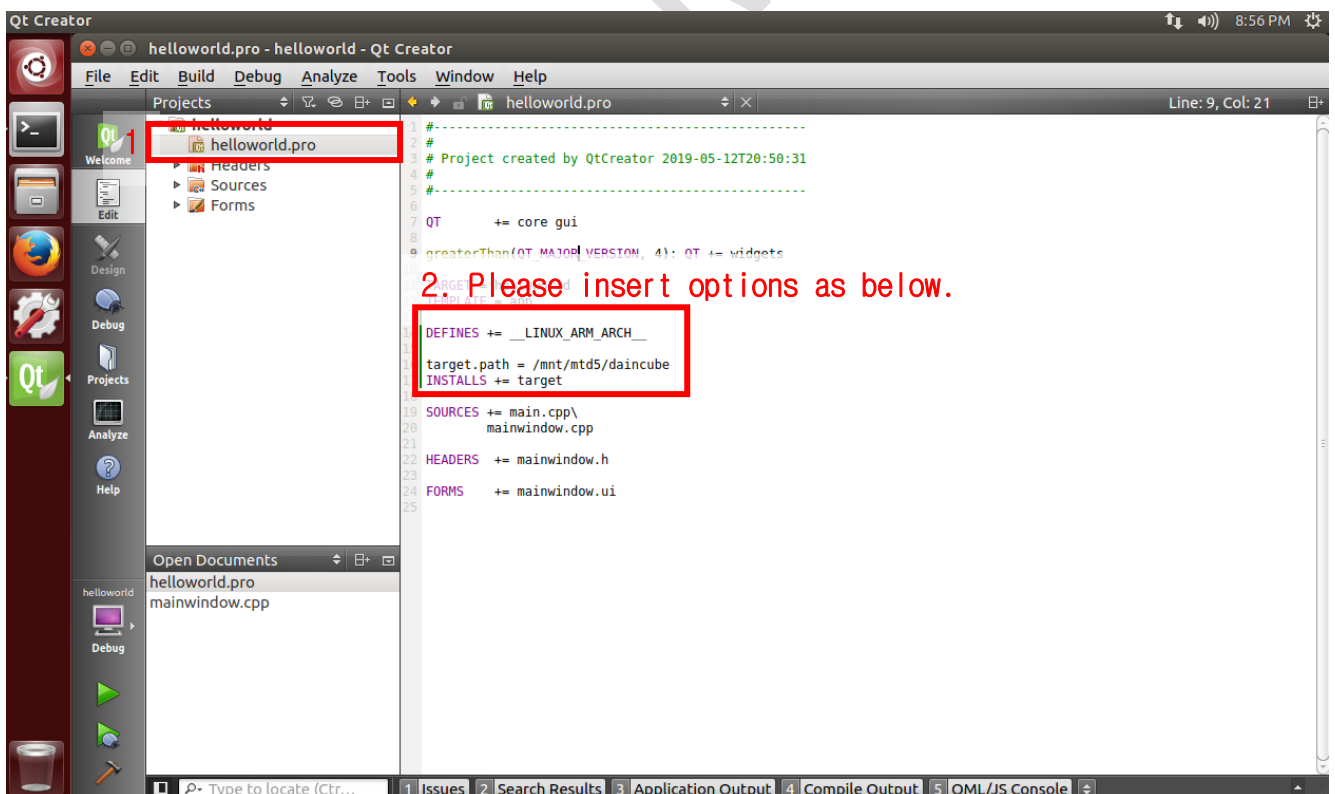


Complete to projects.



Setting for Cross-compiler path.

target.path is internal path in CG4300 board.



*\$vi helloworld.pro*

**[helloworld.pro]**

DEFINES += \_\_LINUX\_ARM\_ARCH\_\_

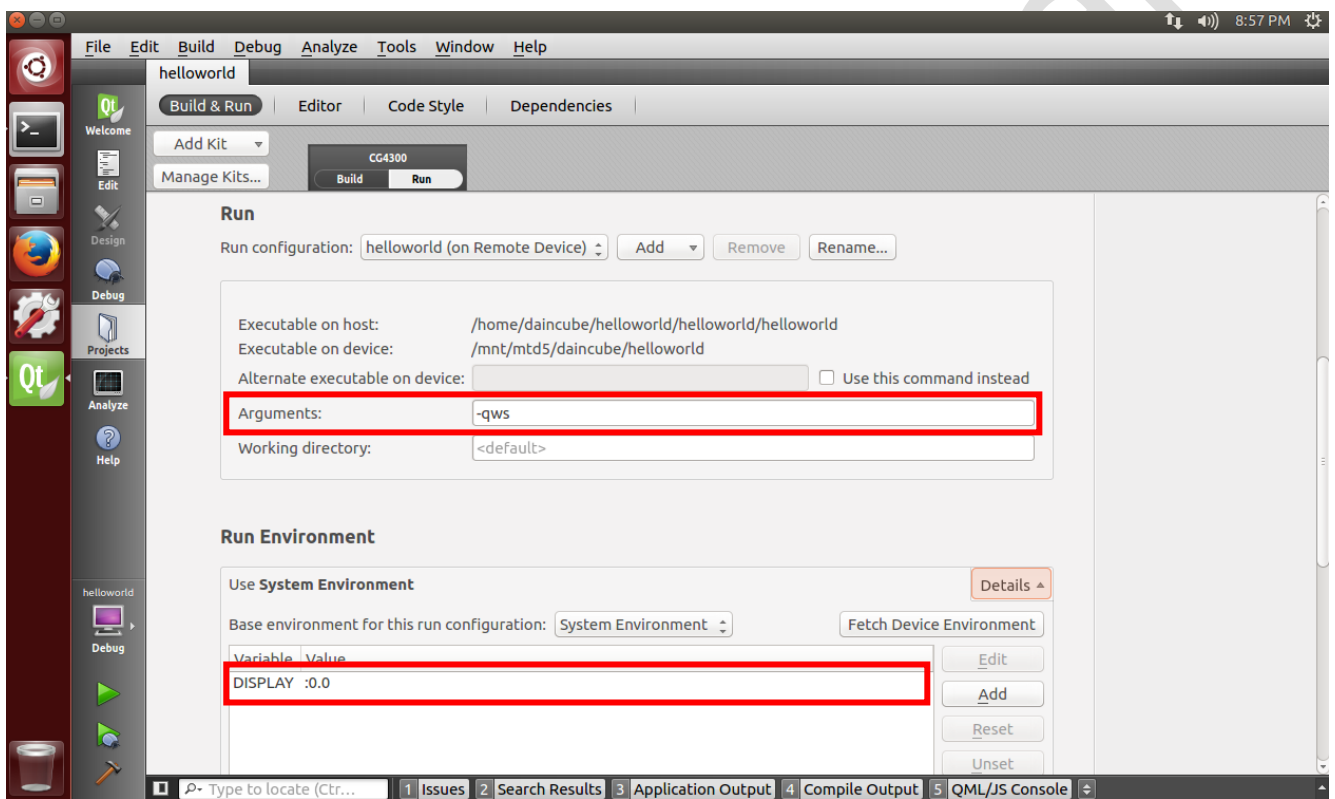
target.path = /mnt/mtd5/daincube

INSTALLS += target

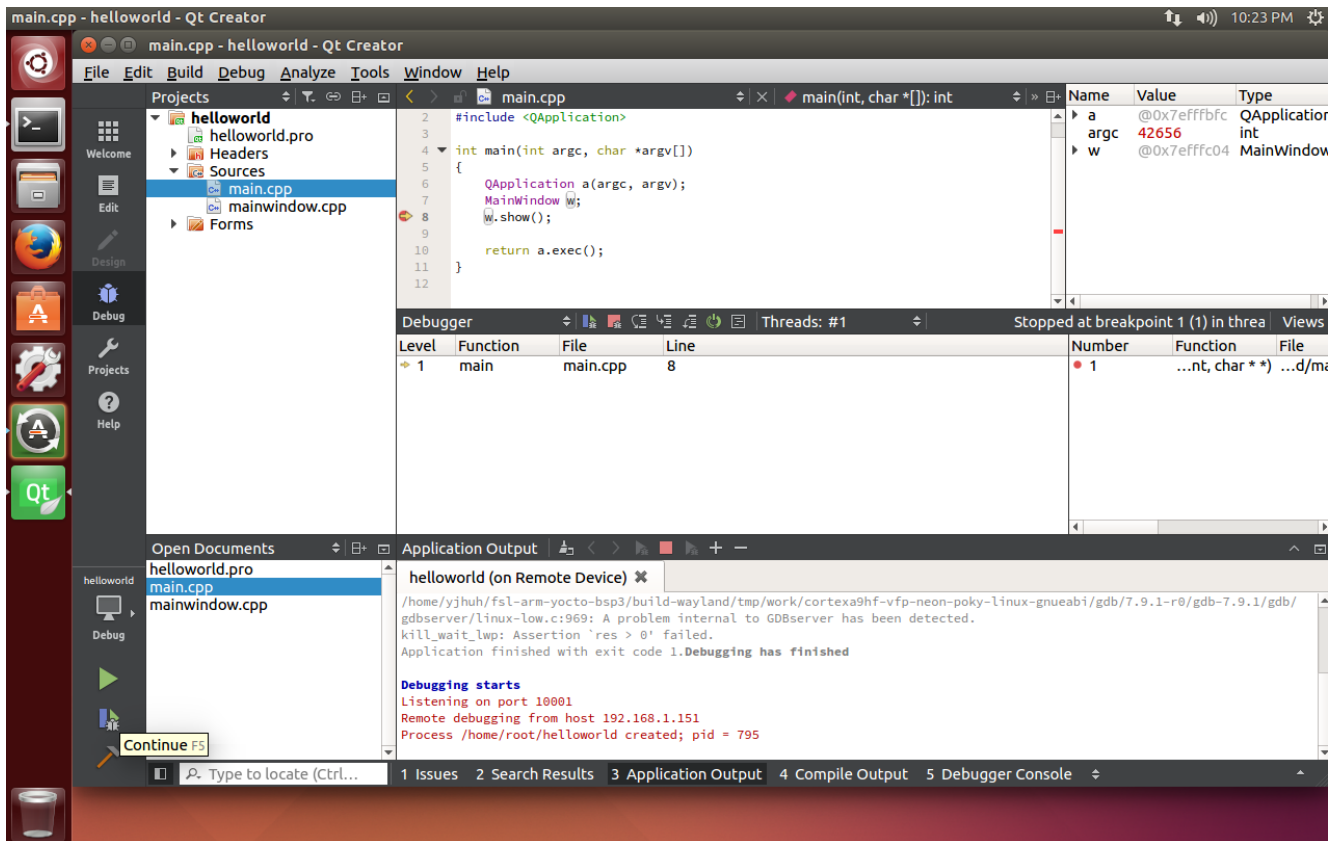
Add for string in helloworld.pro.

You can build and debug remotely. (When CG4300 is selected in Kit selection)

Before debugging, you need to set Argument and Run environment in Qt.

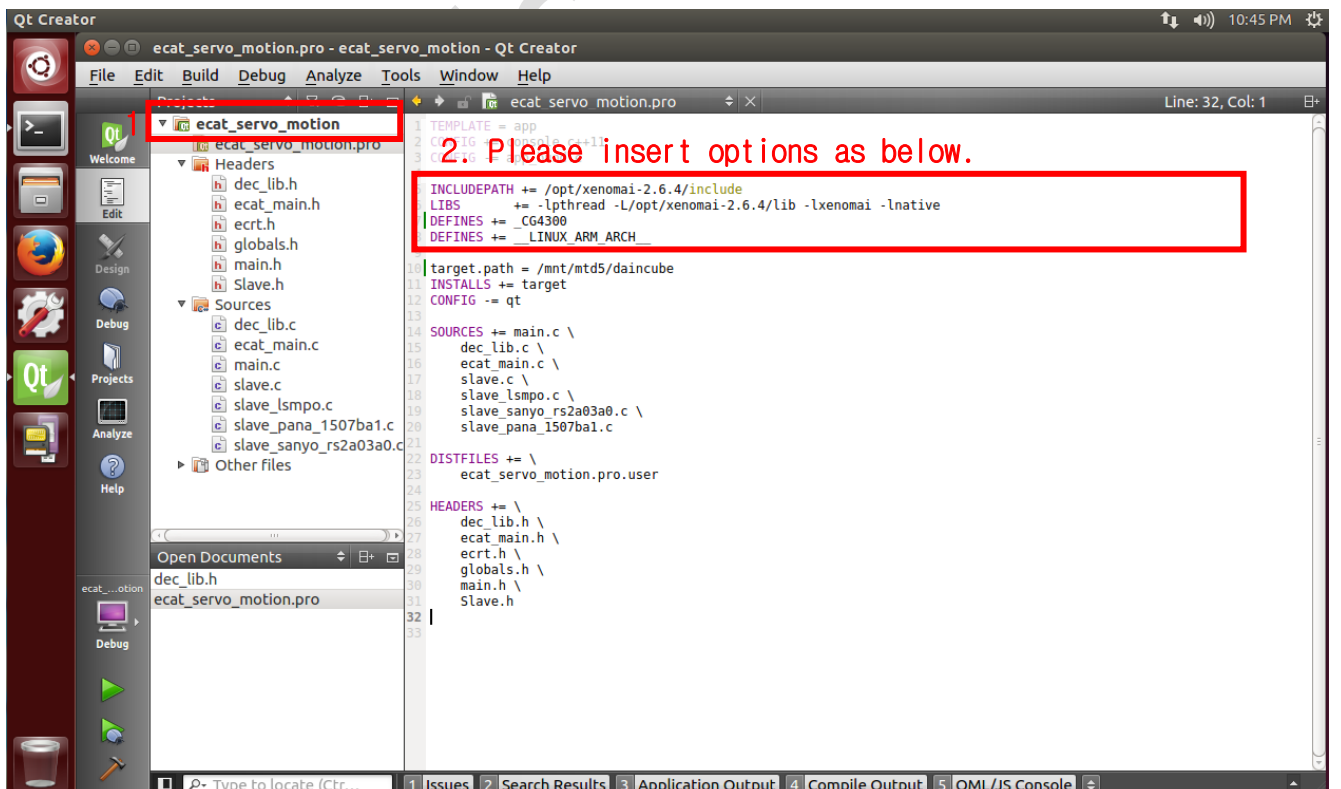


This picture is debugging CG4300 remotely.



### 1.4.3. Xenomai real-time project

To use the Xenomai real-time api, you need to specify the path to the Xenomai library and include file in the project. This example program (ecat\_servo\_motion) is not a Qt application. However, you can create it as a plain C project and manage it with Qt Creator.



Add the following string to the Qt project file (ecat\_servo\_motion.pro) using the Xenomai api.

```
[ecat_servo_motion.pro]
```

```
INCLUDEPATH += /opt/xenomai-2.6.4/include
```

```
LIBS      += -lpthread -L/opt/xenomai-2.6.4/lib -lxenomai -lnative
```

```
DEFINES += __LINUX_ARM_ARCH__
```

```
target.path = /mnt/mtd5/daincube
```

```
INSTALLS += target
```

User's Manual

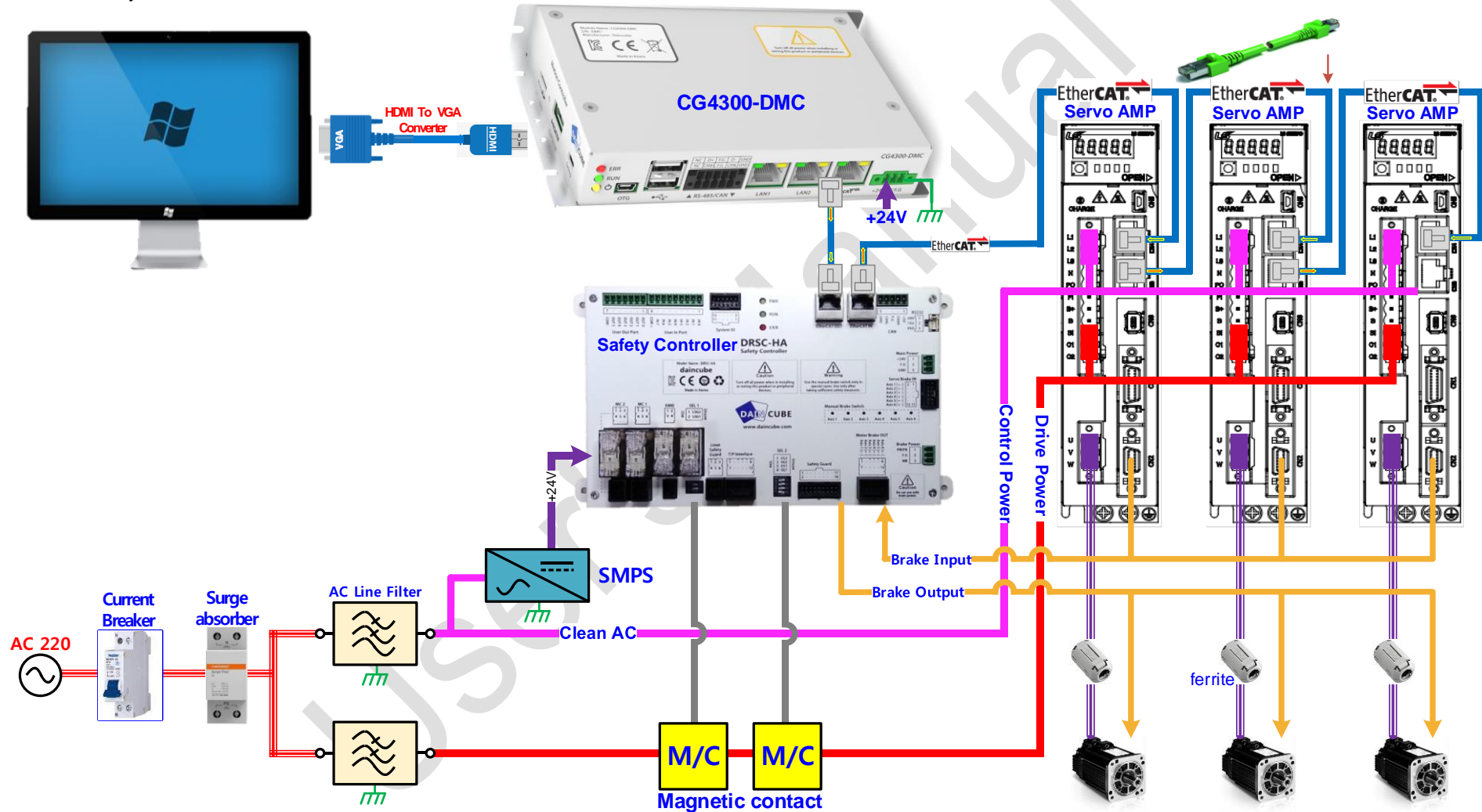


## 2. EtherCAT Example

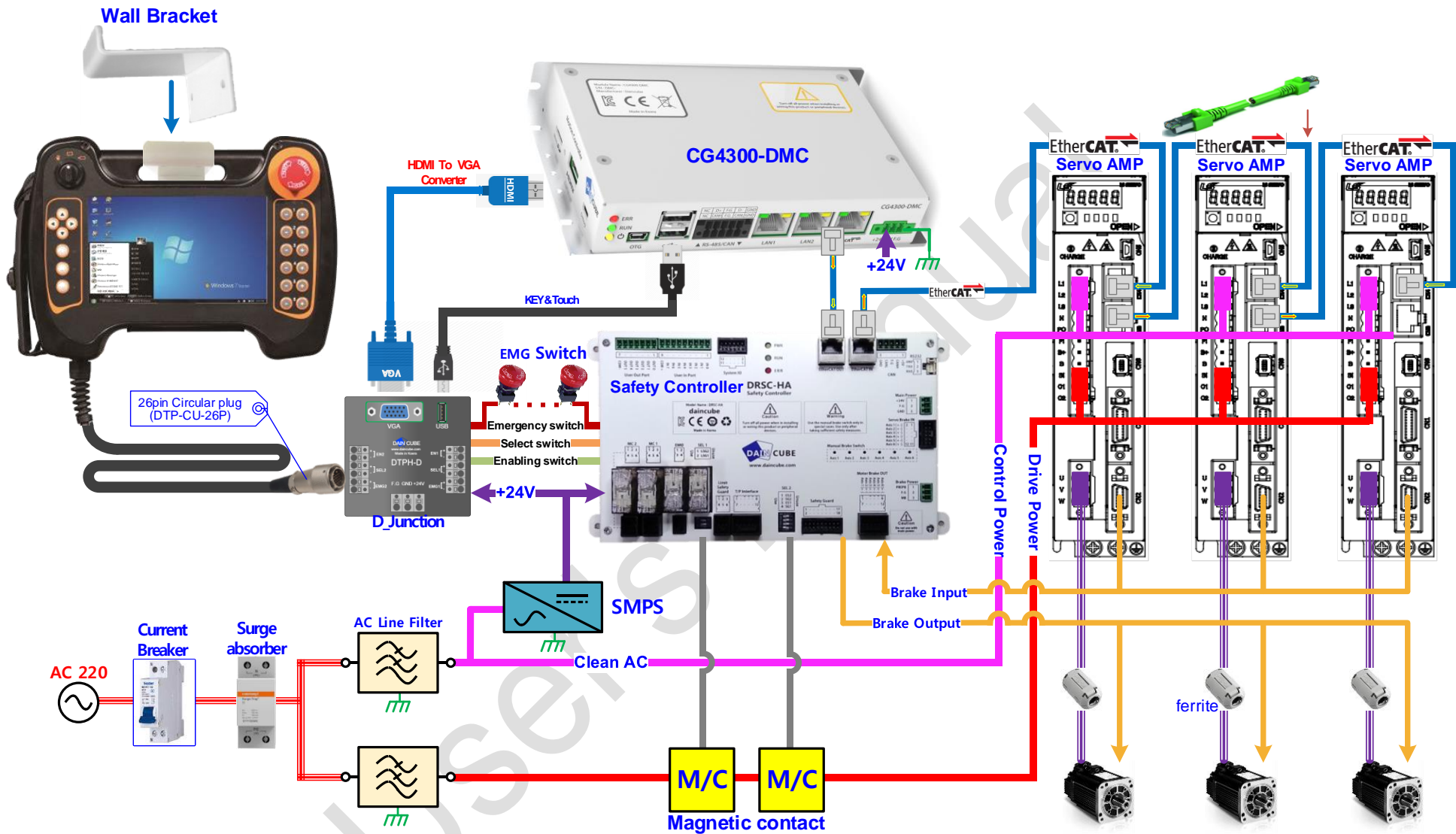
This section describes how to use the example program to control the servo drive and motor and how to build it.

### 2.1. CG4300-DMC and Servo Driver Connection

There are two ways to connect the CG4300-DMC to the monitor via the HDMI cable or to the DTP7H-D. This manual is based on how to connect to the monitor.



[Picture 7-1] Monitor and CG4300-DMC connection diagram



[Picture 7-2] DTP7H-D and CG4300-DMC connection diagram

## 2.2. EtherCAT Sample application description and usage

### 2.2.1. EtherCAT Master & sample application execution

After booting by connecting to CG4300-DMC as above 6.1, power on the CG4300-DMC, and check the "start.master" and "stop.master" files as below by typing the following command on the CG4300-DMC Console terminal I will.

[start.master script]

```
[root@imx6qsabresd/]# vi /ecat/start.master

/etc/init.d/ethercat start          -> EtherCAT Master Start
sleep 3
insmod /ecat/ec_ddcxen.ko          -> Add EtherCAT Master Module

export LD_LIBRARY_PATH=/usr/xenomai/lib
export DISPLAY=:0

/home/daincube/ecat_servo_motion -c 1000 -v l -s 30 -t f -a 1
```

[stop.master script]

```
[root@imx6qsabresd/]# vi /ecat/stop.master

rmmod ec_ddcxen                  -> Remove EtherCAT Master Module
sleep 1
/etc/init.d/ethercat stop          -> Start EtherCAT Master
```

At the CG4300-DMC Console terminal, enter the following commands to run EtherCAT Master and QT Sample Application.

```
[root@CG3300 /]#
[root@CG3300 /]# /ecat/start.master
Shutting down EtherCAT master 1.5.2 EtherCAT: Master module cleaned up.
done
Starting EtherCAT master 1.5.2 EtherCAT: Master driver 1.5.2 2eff7c993a63
EtherCAT 0: Registering RTDM device EtherCAT.
EtherCAT: 1 master waiting for devices.
IPv6: ADDRCONF(NETDEV_UP): sit0: link is not ready
ec_fec 2188000.ethernet: Unbalanced pm_runtime_enable!
... skip ...

EtherCAT WARNING 0-2: Slave did not sync after 5000 ms.
EtherCAT 0: Slave states on main device: OP.
ec-app: Slave[0] Set SERVO_ON
ec-app: Slave[1] Set SERVO_ON
ec-app: Slave[2] Set SERVO_ON
ec-app: [slv_inx:0] = target_val:1886989899
ec-app: [slv_inx:1] = target_val:-990230200
ec-app: [slv_inx:2] = target_val:1053377192
ec-app: ServoRun ..
Command :
```

When this is done, the motor will start and you can control the servo driver by entering Command at the CG4300-DMC Console terminal.

### 2.2.2. Terminating the EtherCAT Master & sample application

When executing EtherCAT Sample Application, enter the following command at the CG4300-DMC Console terminal to terminate the Sample Application and then enter the command again to terminate the EtherCAT Master.

Command:  
Command: **exit**  
EtherCAT Stop  
ec-app: dec\_close\_domain\_data (0)

### 2.2.3. EtherCAT sample application argument

When you run EtherCAT Sample Application, you need to input and execute arguments for the servo driver.

Ex) ./ecat\_servo\_motion -c 1000 -a 2 -v l -s 25 -t f

Option	Argument	Range	unit	Default
Cycle Time	-c	1000 ~ 16000	us	1000
Servo Axis	-a	1 ~ 6	Axis	2
Servo Vendor	-v	s: sanyo o: omron l: ls mecapiion e: elmo	-	l: ls mecapiion
Run Speed	-s	1 ~ 100	-	25
Run Type	-t	f: forward r: reverse t: task	-	f: foward

### 2.2.4. EtherCAT sample application command

You can control the servo driver by entering additional commands while running the EtherCAT Sample Application.

Option	Command	Operate
Servo ON	on	Servo driver status is Servo ON status.
Servo OFF	off	Servo driver status becomes Servo OFF status.
Servo Run	run	Servo driver status becomes Servo Run status and motor starts.
Servo Stop	stop	Servo driver status becomes Servo Stop status and motor stops.
Exit	exit	Exit the application.

### 2.2.5. EtherCAT sample application build

Extract the file "CG4300-DMC\_SDK >> Example >> EtherCAT\_Example >>" ecat\_serv\_motion\_qt.tar.bz2 "to the desired location on the host PC.

[How to unzip]

```
HOST_PC # mkdir /Example
HOST_PC # tar -xvf /home/daincube/ecat_servo_motion_qt.tar.bz2
```

Please refer to 6.4.3 and build from Qt.

## 2.3. Qt EtherCAT GUI example

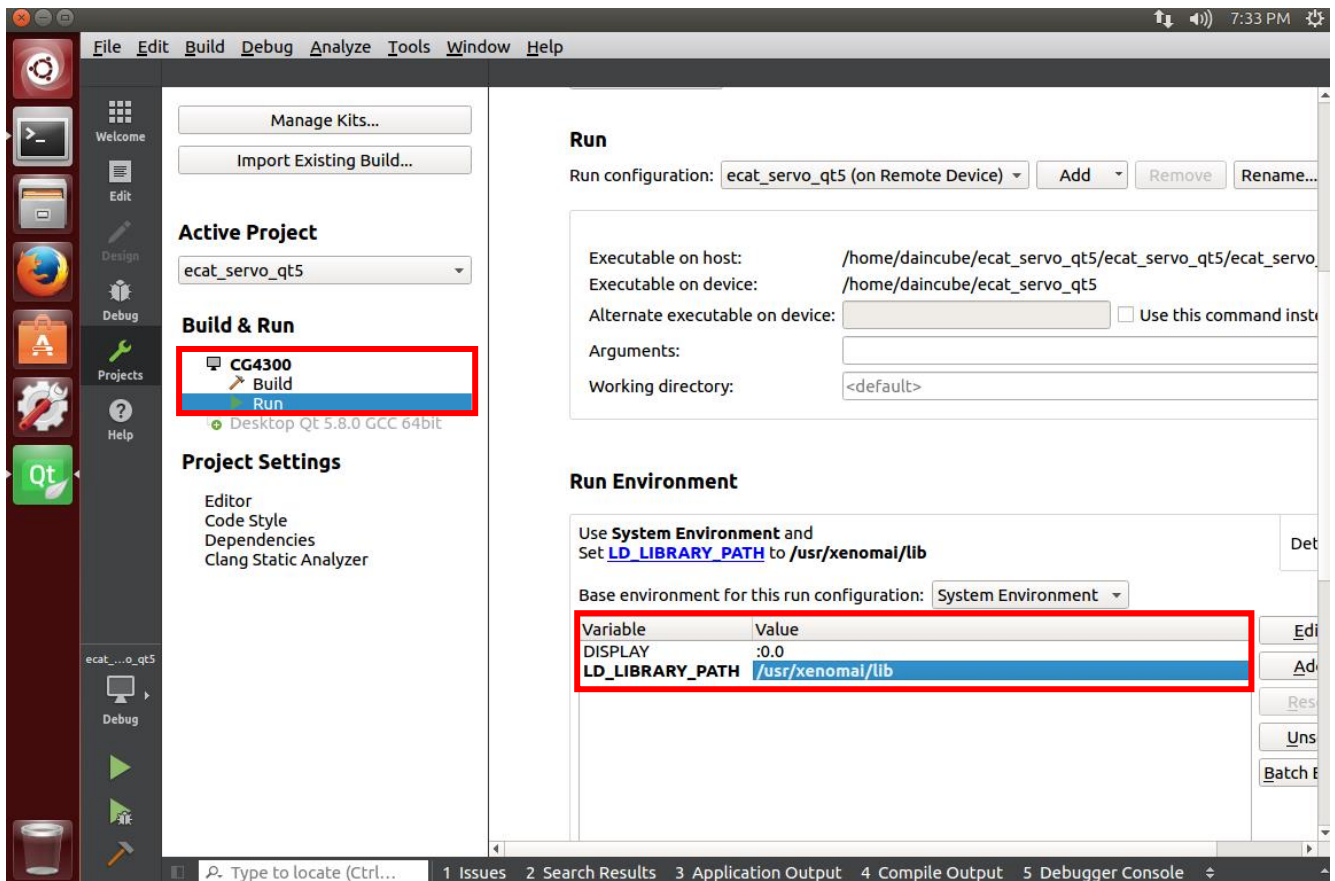
This section explains how to use the QT example program and how to build the motor by controlling the servo driver with EtherCAT master of CG4300-DMC.

### 2.3.1. CG4300-DMC and servo driver connection

2.1. Refer to the item and connect CG4300-DMC to Servo Driver.

### 2.3.2. Running the EtherCAT Master & Qt GUI sample application

When debugging the Qt EtherCAT GUI, you need to set up a Run environment.

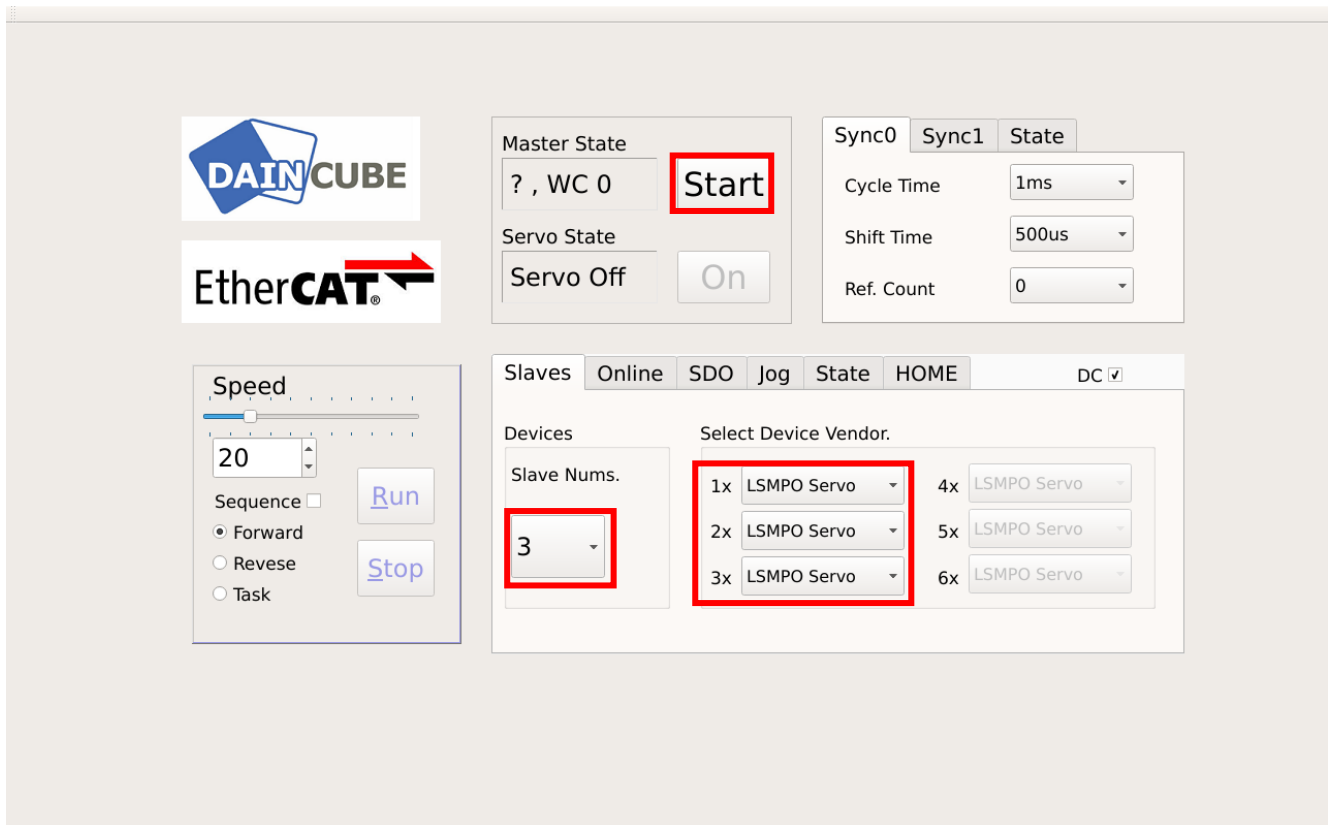


#### Qt EtherCAT GUI Execute

When QT Application is executed, the following screen appears on the CG4300-DMC screen.

Set Cycle Timer, Slave Nums, Device vendor, Speed to Slave Device.

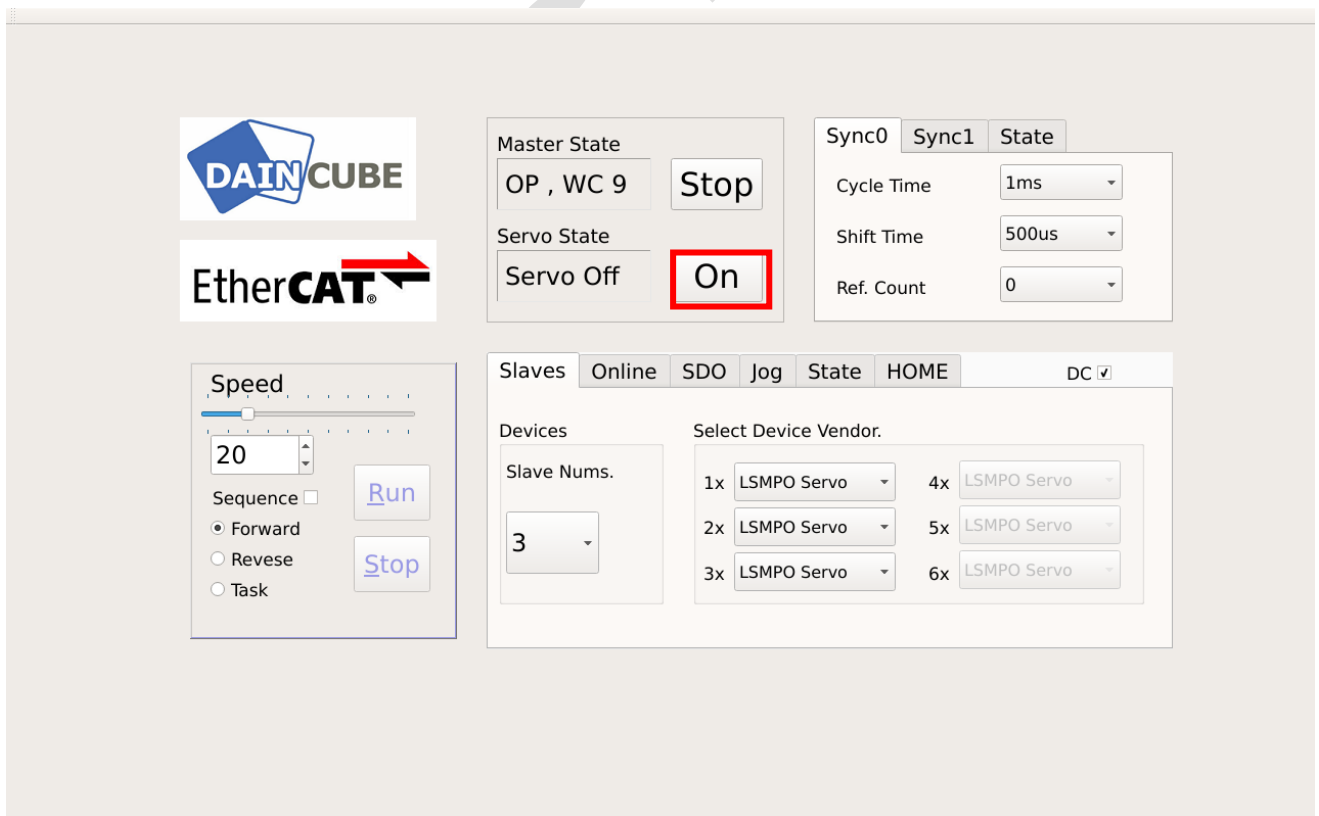
When you are finished, click the "Start" button to start EtherCAT.



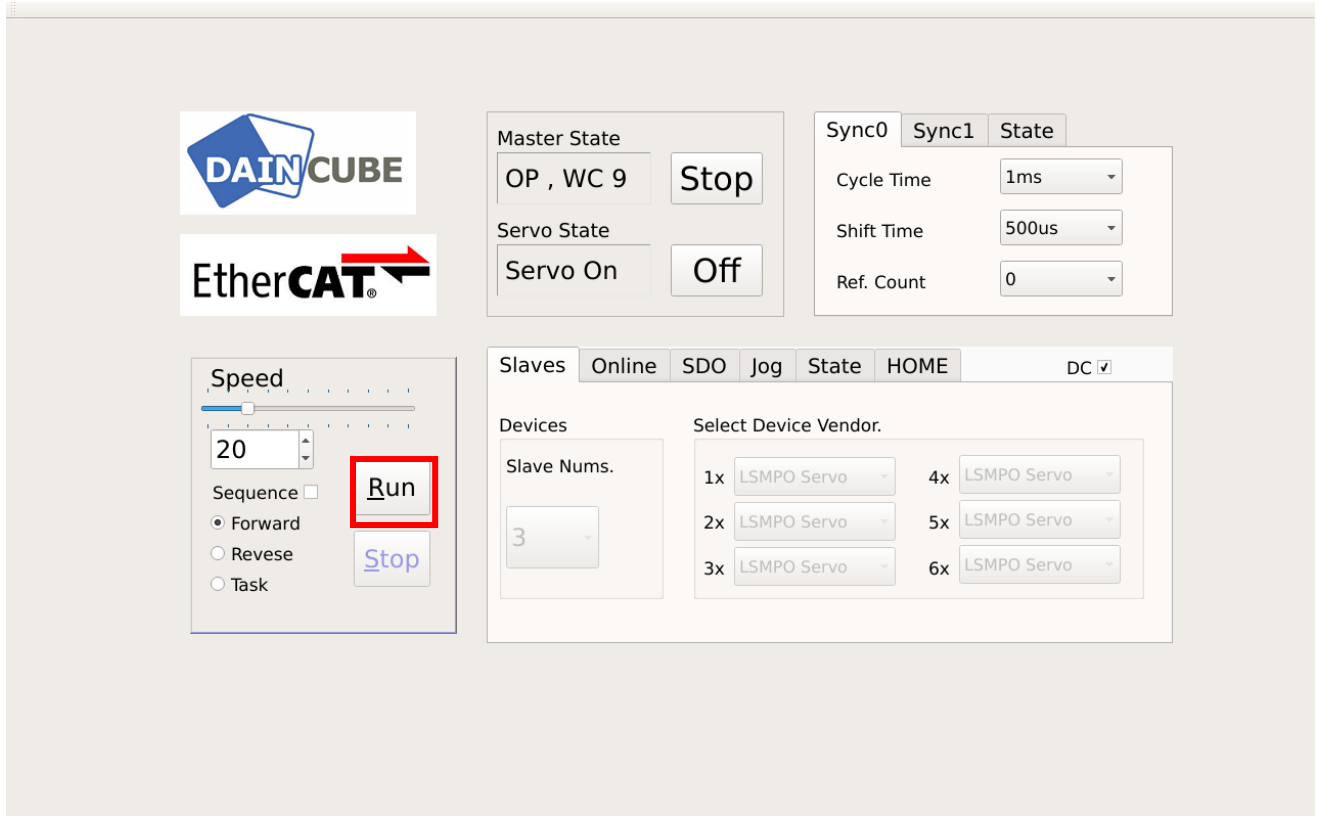
The Slave Device status is displayed in the EtherCAT State window and you can see the State change. EtherCAT State changes to Pre-Op mode. Wait until OP mode changes.

If the slave state is changed to "OP" mode, the slave device is ready. Click the "On" button to activate the Servo Driver.

When Master State is OP, press the On button to turn Servo On.



Press the Run button to turn the motor.



### 3. FAQ

#### 3.1. Default password

Target board (CG4300)

ID : root

PW : daincube