Xenomai 3 Installation

Step 1 : Install Ubuntu 18.04 (20.04 also can be enough)

Step 2 : Search your computer for the ‘Software Updater’ and make sure that everything is up to date.

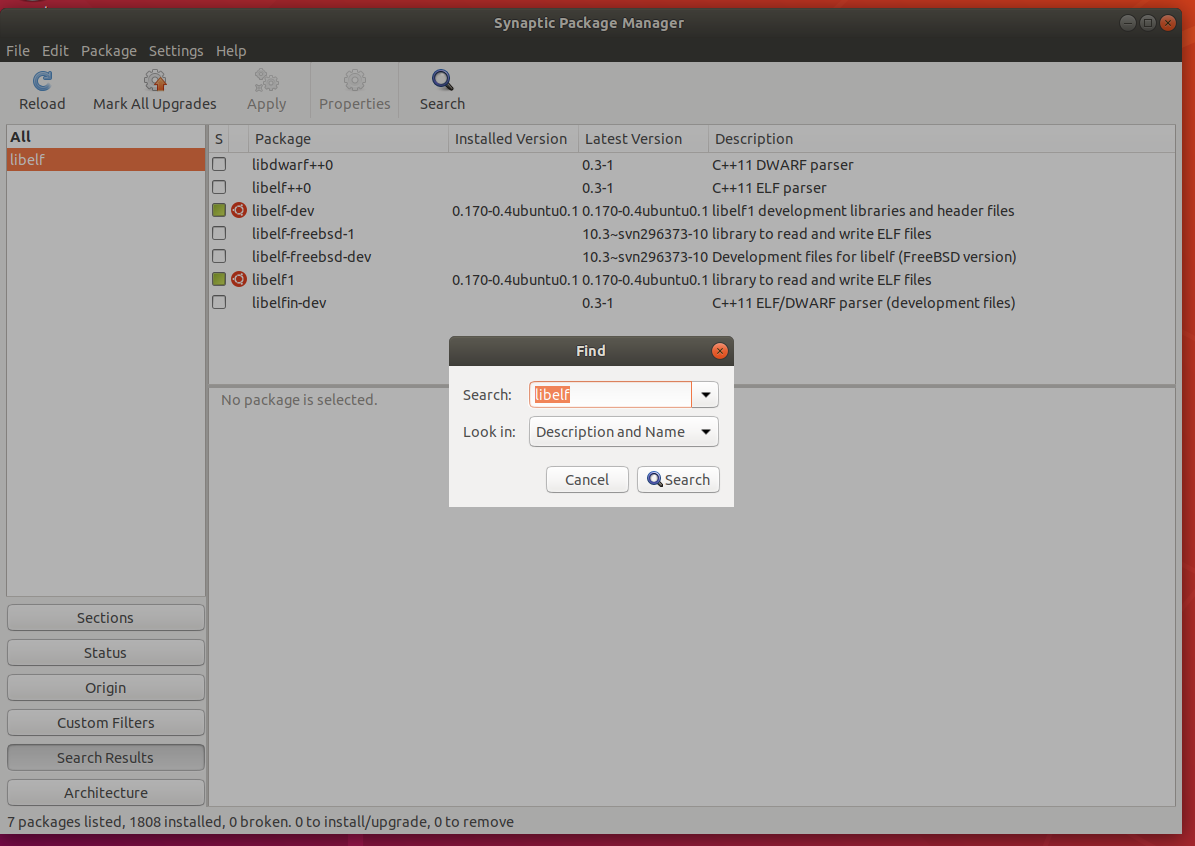
Step 3 : Open new terminal and run the command.

$sudo apt-get install devscripts debhelper dh-kpatches findutils kernel-package libncurses5-dev fakeroot zlib1g-dev dwarves flex bison libssl-dev

Step 4 : Open the ‘Ubuntu Software Center’ to install ‘Synaptic Package Manager’ then run the ‘Synaptic Package Manager’ to check off packages listed below. Then click ‘ Apply’

* libpopt-dev
* libglu1-mesa-dev
* libglade2-dev
* libelf-dev

You can easily find a specific library or package in ‘search’ and check off the box for the library or package as shown in Fig.

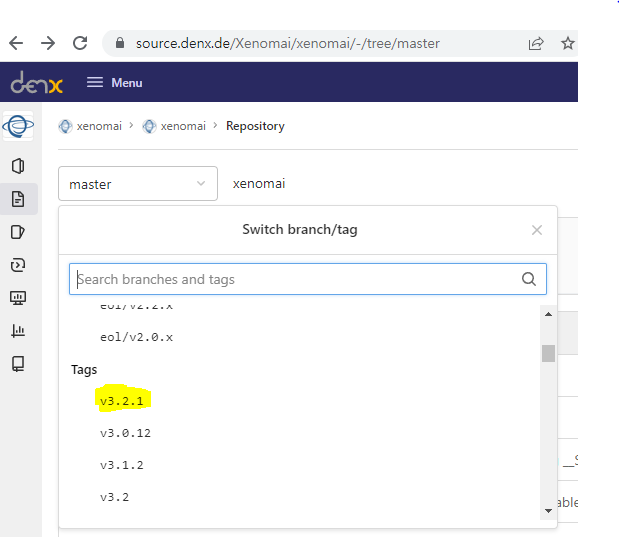


Step 5 : Download files listed below.

Linux kernel : linux 5.4.124 => [www.kernel.org/pub/linux/kernel/v5.x/](http://www.kernel.org/pub/linux/kernel/v5.x/)



Xenomai : xenomai 3.2.1 => https://source.denx.de/Xenomai/xenomai/-/tree/master



Ipipe patch : ipipe-5.4.124-x86-5 => xenomai.org/downloads/ipipe/v5.x

테이블이(가) 표시된 사진

자동 생성된 설명

Step 6 : In a terminal,

$sudo nautilus

Then, move all files downloaded at step 5 to /usr/src/ to extract them.

Step 7 : Modify superuser password and get superuser permission

In a terminal run the command.

$sudo passwd root

Set the password as ‘control’, Then

$su

Step 8 : Apply xenomai patch to linux kernel.

In the terminal,

$cd /usr/src/xenomai-v3.2.1

텍스트이(가) 표시된 사진

자동 생성된 설명

Then,

$./scripts/prepare-kernel.sh --linux=/usr/src/linux-5.4.124 --ipipe=/usr/src/ipipe-5.4.124-x86-5.patch --arch=x86\_64



Step 9 : Make configure file to build linux kernel

Move to ‘linux source directory’. Then,

$make gconfig

Then, Set the kernel option like this.

\* General setup

--> Local version - append to kernel release : -xenomai-3.2.1

--> Timers subsystem

--> High Resolution Timer Support (Enable)

\* Xenomai/cobalt

--> Sizes and static limits

--> Number of registry slots (512 --> 4096)

--> Size of system heap (Kb) (4096 --> 4096)

--> Size of private heap (Kb) (256 --> 256)

--> Size of shared heap (Kb) (256 --> 256)

--> Maximum number of POSIX timers per process (256 --> 512)

--> Drivers

--> RTnet

--> RTnet, TCP/IP socket interface (Enable)

--> Drivers

--> New intel(R) PRO/1000 PCIe (Enable)

--> Realtek 8169 (Enable)

--> Loopback (Enable)

--> Add-Ons

--> Real-Time Capturing Support (Enable)

​

\* Process type and features

--> Transparent Hugepage Support (Disable)

--> Allow for memory compaction (Disable)

--> Contiguous Memory Allocator (Disable)

--> Allow for memory compaction

--> page migration (Disable)

--> Multi-core scheduler support (Disable)

\* Device Drivers

--> Microsoft hyper-v guest support

--> Microsoft hyper-v client deivers (disable)

--> Staging drivers

--> Unisys SPAR driver support

--> Unisys visorbus driver (disable)

-->I2c Support

--> I2c Hardware Bus support

-->PC Speaker support (disable)

​

\* Power management and ACPI options

--> CPU Frequency scaling

--> CPU Frequency scaling (Disable)

--> ACPI (Advanced Configuration and Power Interface) Support

--> processor (Disable)

--> CPU Idle

--> CPU idle PM support (Disable)

Then, save it.

You can easily find the options using ‘Ctrl + F’ and you can use wheel scroll to select next searching things.

Step 10 : Remove System trusted keys

In the terminal, type this command to open .config file

$gedit .config

Then, change

CONFIG\_SYSTEM\_TRUSTED\_KEYS="debian/canonical-certs.pem"

to

CONFIG\_SYSTEM\_TRUSTED\_KEYS=""

Then, save it.

텍스트이(가) 표시된 사진

자동 생성된 설명-> 텍스트이(가) 표시된 사진

자동 생성된 설명

Step 11 : build kernel image

In the terminal,

$make bzImage modules -j4

$make modules\_install

$make install

Step 12 : Allow non-root users

$sudo addgroup xenomai --gid 1234

$sudo addgroup root xenomai

$sudo usermod -a -G xenomai $USER

Step 13 : Configure GRUB and reboot

$sudo gedit /etc/default/grub

Then edit the grub config :

GRUB\_DEFAULT="Advanced options for Ubuntu>Ubuntu, with Linux 5.4.124-xenomai-3.2.1"

#GRUB\_DEFAULT=saved

#GRUB\_SAVEDEFAULT=true

# Comment the following lines

#GRUB\_HIDDEN\_TIMEOUT=0

#GRUB\_HIDDEN\_TIMEOUT\_QUIET=true

GRUB\_TIMEOUT=5

GRUB\_CMDLINE\_LINUX\_DEFAULT="quiet splash xenomai.allowed\_group=1234"

GRUB\_CMDLINE\_LINUX=""

$sudo update-grub

$sudo reboot

Step 14 : Make sure that you are running the cobalt kernel :

In the terminal

$uname -a

*# Should return Linux 5.4.124-xenomai-3.2.1 #1 SMP Wed Sep 20 16:00:12 CEST 2017 x86\_64 x86\_64 x86\_64 GNU/Linux*

$dmesg | grep Xenomai

*# [ 1.417024] [Xenomai] scheduling class idle registered.*

*# [ 1.417025] [Xenomai] scheduling class rt registered.*

*# [ 1.417088] I-pipe: head domain Xenomai registered.*

*# [ 1.417704] [Xenomai] allowing access to group 1234*

*# [ 1.417726] [Xenomai] Cobalt v3.2.1*

텍스트이(가) 표시된 사진

자동 생성된 설명

Step 15 : build Xenomai Library

Go to Xenomai directory

$cd /usr/src/xenomai-3.2.1

Then,

$./script/bootstrap

$./configure --with-core=cobalt --enable-smp --enable--pshared --includedir=/usr/include/xenomai --libdir=/usr/lib

$make install

$sudo chmod 77 /dev/rtdm/\*

$reboot

* Install xenomai kernel and xenomai library complete.

PODO setup

Step 1 : PCAN driver installation

Go to this link,

<http://www.peak-system.com/fileadmin/media/linux/index.htm>

텍스트이(가) 표시된 사진

자동 생성된 설명

Download PCAN Driver Package and extract it.

Step 2 :

$make clean

$make NET=NO\_NETDEV\_SUPPORT

$sudo make install

$sudo modprobe pcan

You can use “lsmod” command to check pcan installed.

텍스트이(가) 표시된 사진

자동 생성된 설명

Step 3 : Qt installation

Ubuntu 18.04 can not support Qt creator version upper than 5

* Install Qt version 5.12.x

Download link : <https://download.qt.io/archive/qt/5.12/5.12.12/>

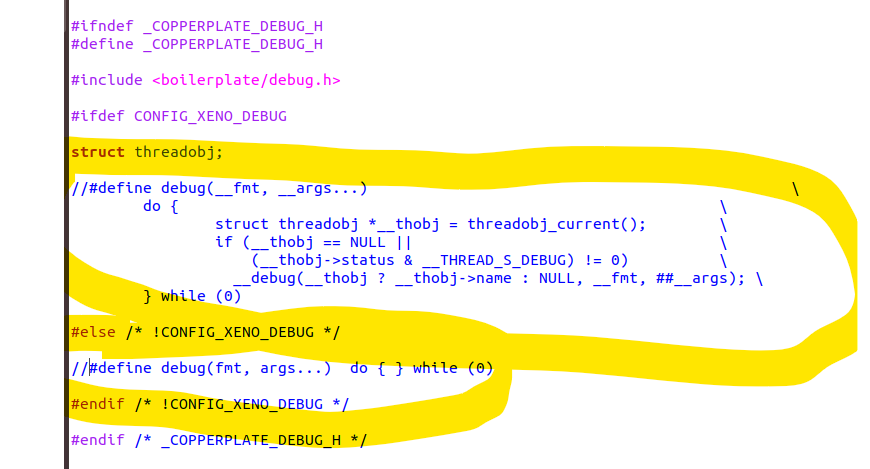
Download qt-opensource-linux-x64-5.12.12.run and running this file to install Qt and Qt creator

Step 4 : Modify xenomai3 debug.h

There is some error caused by xenomai debug.h

So you can modify /usr/include/xenomai/copperplate/debug.h like this.

텍스트, 스크린샷, 실내, 모니터이(가) 표시된 사진

자동 생성된 설명

Step 5 : Install spdlog

In PODO\_VR use spdlog library to log message.

Move to Download folder or Desktop folder or whatever you want.

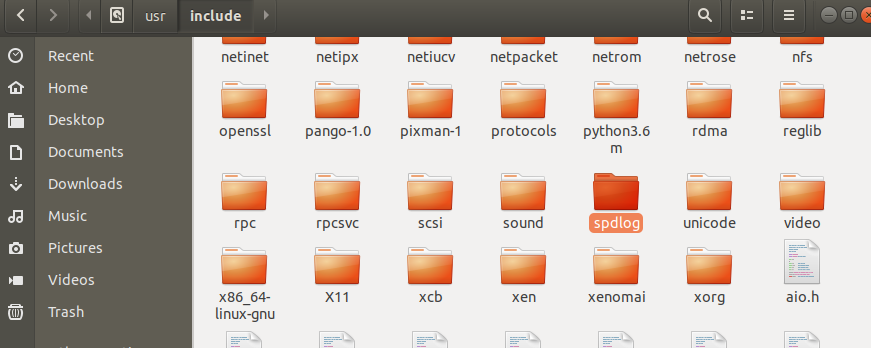
Download spdlog and build it.

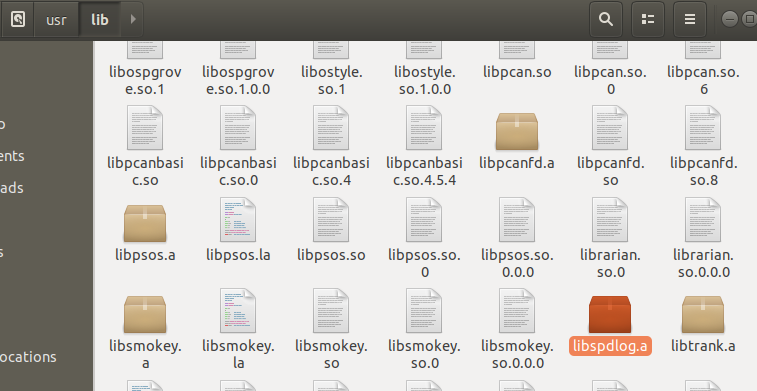
$ git clone https://github.com/gabime/spdlog.git

$ cd spdlog && mkdir build && cd build

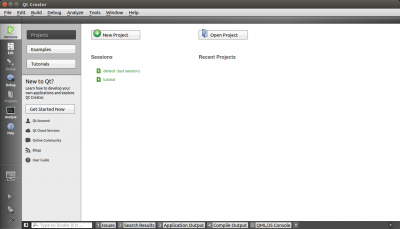
$ cmake .. && make -j

You have to move spdlog’s include folder to /usr/include and dynamic library file to /usr/lib

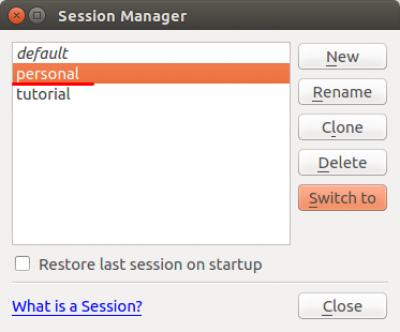


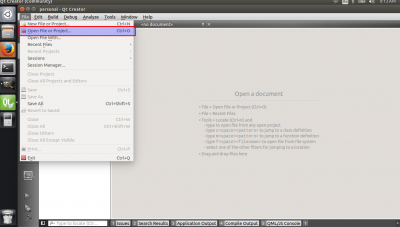


Step 6 : Setup PODO build environment.

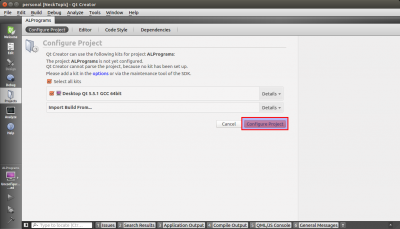
Once you have your personal copy open qt. Once in qt go to File→Session Manager. Once inside Session Manager create a new session, name it what ever you want. Then double click it inside the Session Manager to edit the session.  
[](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:podoinst3.png) [텍스트, 스크린샷, 모니터이(가) 표시된 사진

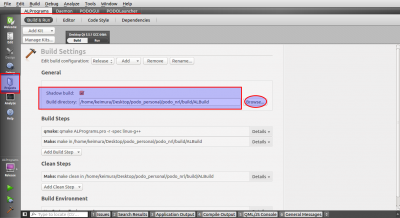
자동 생성된 설명](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:podoinst4.png) [텍스트이(가) 표시된 사진

자동 생성된 설명](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:podoinst5.png) [](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:podoinst6.png) [텍스트, 스크린샷, 모니터이(가) 표시된 사진

자동 생성된 설명](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:podoinst7.png)  
Now go to File→Open File or Project. Find your personal copy and go to [your personal copy]→podp\_nrl→src→ALPrograms. Once in find the .pro file and open it. If a pop up comes up click yes and then configure project in the next screen.  
[](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:podoinst8.png) [텍스트이(가) 표시된 사진

자동 생성된 설명](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:podoinst9.png) [텍스트이(가) 표시된 사진

자동 생성된 설명](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:podoinst10.png) [](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:podoinst11.png)  
Do this for Daemon, PODOGUI, and PODOLauncher as well.  
[텍스트이(가) 표시된 사진

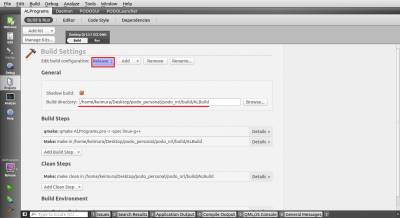
자동 생성된 설명](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:podoinst12.png)  
Once you have all of the projects open in your session go to the projects tab and change the build directory path. Click on the browse button in General and find your folder→podo\_nrl→build. Then find the appropriate folder for the project (ex for ALPrograms go into ALBuild) and click open.  
[](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podoinst101.png) [텍스트이(가) 표시된 사진

자동 생성된 설명](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podoinst102.png)  
Note: If you don't have the build folder in your file run the build.sh file inside your podo file from a terminal:

sudo chmod +x build.sh  
./build.sh

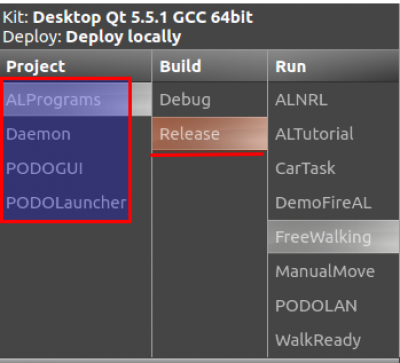
[](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:buildsh.png)[텍스트이(가) 표시된 사진

자동 생성된 설명](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:buildshterm.png)

Make sure that both Release and Debug have the same path. Repeat this for all the other projects, making sure that both Release and Debug have the same build directory. As shown in the second picture you may see a warning, you can ignore it.  
[](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podoinst103.png) [텍스트이(가) 표시된 사진

자동 생성된 설명](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podoinst104.png)

In the bottom left-hand corner of the QT window there is a computer icon, click it and make sure that all the projects are in release mode instead of Debug mode.  
[텍스트이(가) 표시된 사진

자동 생성된 설명](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podoinst105.png) [](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podoinst106.png)

Step 7 : Modify AL program make file(.pro file)

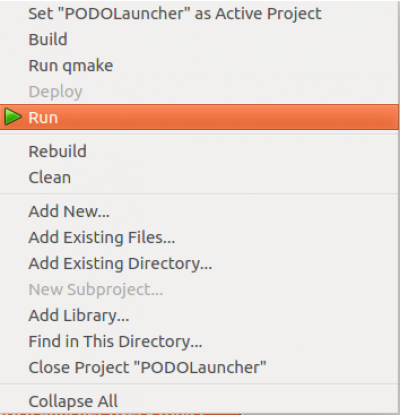
You have to add “-lmodechk” to “QMAKE\_LFLAGS” and “LIBS” every AL program’s pro file and Demon.pro file.

텍스트이(가) 표시된 사진

자동 생성된 설명

Once you have all of them in your session go to Build→Rebuild All.  
[텍스트이(가) 표시된 사진

자동 생성된 설명](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:podoinst13.png)  
Note: If you get an error about being unable to locate -lrbmodel just add the RBModel project and rebuild it byitself before rebuilding the others. This will create the rbmodel library the code is looking for. Don't forget to change the build location like the other projects for RBModel.

Now right click PODOLauncher in qt and run it.  
[](https://www.daslhub.org/unlv/wiki/lib/exe/detail.php?id=installing_xenomai3_podo3_and_qt&media=podo_inst:podoinst16.png)