Compilation and installation of MIRO and Matlab on Ubuntu 9.04 for UNBeatable-Sim

Carl Thibault

January 30, 2010

Abstract

This document was developed to assist with the installation of MIddle-ware for RObotics (MIRO) and Matlab. The instructions are presented to acquire and install the correct packages to enable the successful compilation of MIRO. If you are an accomplished UNIX user then this document may seem unnecessarily long, otherwise the specifics maybe helpful.

Carl Thibault is a M.Sc.E candidate in the Department of Electrical and Computer Engineering, Collaboration Based Robotics and Automation Group, University of New Brunswick, Fredericton, New Brunswick E3B 5A3, Canada. `Carl.T@unb.ca`

1 Introduction

This document is divided into several parts including installation of dependent packages, compiling ulmMiro, drdcMiro and Matlab. Matlab can be installed at any time since it is not dependent on Miro however the software developed by UNB is designed to cooperate with Matlab so it will be required for full operation. The other sections are dependent on each other and each section should should be complete before progressing to the next.

Greg Borten and Defense Research Development Canada Suffield provided created provided a wiki instruction set [1] for an older version of Ubuntu and a working example of the .miroProfile file. This information and email transaction was the backbone of porting the process to Ubuntu 9.04.

2 Obtaining and installing the operating system

Ubuntu is an open source Unix operating system based on the Debian operating system(OS). Ubuntu is known as the most user friendly in the family of open source systems and eases the burden of installing MIRO by offering pre-complied files or packages. These packages can be installed in ether a graphical or terminal application depending on user preference.

At the time of writing 9.04 was the latest stable version of Ubuntu that has proven to work with MIRO. The principles and packages described will port to later versions of Ubuntu however that was not tested by the author. One note if using video processing, Ubuntu 9.04 is a ”jaunty” version and required that opencv must be installed differently, this will be addressed later in the manual.

2.1 Finding older versions of Ubuntu

The Ubuntu home page will always display the latest version of Ubuntu. obtaining the 32bit desktop version of 9.04 can be found at this link.

`http://releases.ubuntu.com/9.04/`

2.2 Installation options

This OS can be installed in a a number of ways. If a temporary or test environment is required the OS can be installed in virtual machine software, such as Sun Virtual Box, where software emulates hardware within the host computer and operating system. A virtual machine would allow a Unix system to run on a windows machine like any other program. This however if not recommended when hight performance applications are required as the overhead of running multiple operating systems simultaneously can become evident when rendering graphics or running complex algorithms. Another advantage is that virtual machine to be exported to others in a working condition. A copy of a machine can be multiplied easily and distributed if required.

A USB key can also be generated to run or install operating systems as well. Using a program such as UNetbootin, available of both Windows and Linux, the application downloads a desired OS live image and loads it to a USB key. This key contains the full OS and can operate on a large verity of hardware. This can also be used to install the operating system and key files with out the need to burn a media disk.

`UNetbootin is available at http://unetbootin.sourceforge.net/`

3 Prerequisites

Several packages need to be added to your system before compiling MIRO that are not installed by default in Ubuntu.

Ubuntu offers two methods to install packages the most intuitive is the Synaptic Package Manager a visual user interface that lists the packages available and can be search to obtain the desired libraries or files, the other is the terminal command `sudo apt-get install PackageName` both are equivalent to the authors knowledge. When selecting a package the dependences will also be marked for installation and should be installed.

3.1 Subversion

Subversion is used to download and maintain source code from web repositories.

Using the manager or apt-get
install:

- subversion

3.2 Compliers

Ubuntu does include the C complier however the C++ complier is not default.

install:

- g++ and g++-4.3

3.3 Autotools

These packages assist in compiling source code.

install:

- Autotools-dev
- Automake with autoconf and m4 as dependencies
- Automake1.7 (is required by miro)
- libtool

3.4 QT

QT is used to create graphical user displays. At this time Miro does not support qt4 so qt3 must be installed.

Using the package manager install:

- qt3-dev-tool, the required library is libqt3-mt

3.5 libQGLViewer

Some utilities use this library and can be installed using the package manager however in order Miro to find them a symbolic link must be created due to a difference in naming convention between Ubuntu and fedora operating systems.

install:

- libqglviwer-dev and dependencies

Create a link on the installed library to accommodate the naming convention.

```
cd /usr/lib
sudo ln -s libqglviewer.so  libQGLViewer.so
```

3.6 libglut or freeglut

Freeglut is an OpenGL tool kit used to run OpenGL code for computer rendering and graphics.

install:

- freeglut3-dev

3.7 BOOST

Boost is a collection C++ libraries that programs can draw on to seed up development.

install:

- bjam
- ibboost1.35-dev and dependences

3.8 GSL

GSL is the GNU Scientific library for numerical analysis and is used by drdc code.

install:

- libgsl0-dev

3.9 Atlas and Blas

Atlas is a linear algebra library package used in localization and Blas is an optimized matrix calculator library. install:

- libatlas-base-dev

check to ensure libblas-dev is installed with the dependences

3.10 lapack and lapackpp

DRDC Miro requires these packages to be installed and increases the speed matrix manipulations in the computer. gfortran is a Fortran complier that on its own and can be used to compile program written in Fortran.

install:

- liblapack-dev
- gfortran

Lapackpp is unfortunately not available in the package manager at this time so must be compiled and installed from source code (src). The source can be obtained from the link below at the time of this wring 2.5.3 was the latest version and complied successfully.

```
http://sourceforge.net/projects/lapackpp/files/
```

After all the other packages are installed

untar the package ether using the interface of from the command line using

```
 tar xvfz file name.tar.gz
```

cd into the folder

```
./configure
make
sudo make install
```

3.11 OpenCV and FFMPEG

Opencv and ffmpeg are packages used for image and video processing and can be omitted if that function is not desired.

FFMPEG is available from the package manager and should not be installed with opencv since 9.04 deviates from the standards with their FFMPEG files. Obtain Opencv can be from source forge and the current version at the time was 2.0 and thought it takes time to compile, it does and 1.0 is also known to work with drdc code. The modification to the installation procedure is when configuration is called ffmpeg need to disabled.

install:

- ffmpeg

untar the file

```
 tar xvfz filename.tar.gz
 ./configure --without-ffmpeg
make
sudo make install
```

3.12 Fire wire IEEE1394 support

These libraries are used to support fire wire connections and were omitted from the install sine the hardware was not available but included for completeness of these instructions. To note both ulmMiro and drdcMiro include IEEE1394 as default so they will be disabled during the configuration at that time. Using the package manager

install:

- libraw1394-dev
- libavc1394-dev
- libdc1394-dev and libdc1394
- libdc1394-utils

4 ACE and TAO install

ACE (adaptive communication environment) and TAO are the real time implementations of CORBA (Common Object Request Broker Architecture)

4.1 Bash environment setup

Obtain the file ".miroProfile" and copy it to your home directory or it can be create it manually. This file exports the location of key components to the software being operated and complied. Using gedit or any other text editor change/assign the path to the Miro directories to match your current/future location of ulmMiro and drdcMiro. A common location for these files is the users home directory.

Since qt4 is the default QTlibrary add a line to export the location of the qt3 directory since the default for 9.04 is qt4 and that is not compatible with Miro that the time of writing.

Using the .miroProfle file several things are taken care of including locating and exporting the IP address. If creating file by hand you can obtained the IP address using the terminal command

```
 /sbin/ifconfig
```

An example of the key components required in the .miroProfile

```
export MIRO_ROOT=$HOME/ulmMiro
export DRDC_ROOT=$HOME/drdcMiro
export IP = 192.0.2.1

# Path to Miro and Drdc Libraries
export LD_LIBRARY_PATH=$DRDC_ROOT/lib$HOME/ulmMiro/lib:/usr/local/ipp_5.3.4.080/sharedlib/:
$LD_LIBRARY_PATH

# Path to qt3 dir
export QTDIR=/usr/share/qt3
```

Also using the text editor modify the .bashrc file located in your home directory to call .miroProfile by adding this line to the bottom of the file.

```
source $HOME/.miroProfile
```

To apply the changes you can source the bashrc file in the terminal or just close and reopen the terminal.

```
source ~/.bashrc
```

4.2 Install the required ACE and TAO packages

Use the package manager install the following packages
for ACE install:

- libace-dev and libace-5.6.3
- libacexml-dev and libacexml-5.6.3
- libace-htbp-5.6.3
- libace-ssl-5.6.3
- gperf-ace

for TAO install:

- libtao-dev and libtao-1.6.3
- libtao-orbsvcs-dev and libtao-orbsvcs-1.6.3
- tao-idl
- tao-naming

5 Installing ulmMiro

DRDC has build on the original ulmMiro and ulmMiro needs to be installed first for things to compile properly.

cd to the directory indicated by .miroProfile for ulmMiro.

Use svn to get a copy of the latest source code and save it to ulmMiro folder in the current directory

```
svn checkout https://svn.berlios.be/svnroot/repos/miro-middleware/trunk ulmMiro
```

This source code need to be boost strapped before being configured. In the configuration line you will notice a lot of features are disabled since we will not use those options to note especially is IEEE-1394 must me disabled if the Fire wire libraries are not installed. The others are special options for robots we do not have access to. They are omitted mainly to save time, hassle and space on the target machine since they will most likely will not be used. If that condition were to change compiling these packages at a later date is not an issue.

cd into the ulmMiro

```
./bootstrap

./configure --disable-Pioneer --disable-Sparrow99 --disable-BTTV --disable-IEEE1394
 --disable-QuickCam

make
```

During the `make` the makefile hung up on a few minor point most of them are not reverent to the current application and can be removed from the make file. For example

5.0.1 Error:

MainForm.h:28.26: error: idl/ChargerC.h: no such file or directory

5.0.2 Description

qtChargerC.h was not found where it was expected. This is correct since this is part of the Pioneer robot user interface that deals with the display indicating the status of the charging port. This idl file was not extracted to create ChargerC.h but qtcharger should have been disable but was not.

5.0.3 Workaround

comment out qtCharger from the utils/Makefile.am

After having worked through the make a recommendation is to run the `make clean` command be used to clean up any loose code that was generated in error.

Once `make clean` is complete, proceed to the command `make install` this may display an Error in that files in the dtd or the etc folder since they may already be on the hard drive. Theses can just be commented out in the makefile.am of ulmMiro if you experience that problem.

6 Installing DRDCMiro

A note on obtaining the drdcMiro code. The files I received were eventually saved on a windows machine. Windows and Unix use different ASCII codes for the newline command and the interpreter in Ubuntu 9.04 has displayed its dependency on the Unix format. The simplest fix I found was to obtain a zip file of the source code and then unzip the file in the terminal using an option in the unzip command "-a" . This option reformats the new line command to Unix for the whole folder. The complete syntax would be as follows

```
 unzip -a filename.zip
```

6.1 Installation

Move the unzipped folder to the location indicated in the .miroProfile

The permission on the booststrap file must be changed since unzip allocated the files a read-only by default. The terminal command to change the permissions is as follows

```
 chmod a=rwx bootstrap
```

Bootstrap the installation as with ulmMiro

```
./bootstrap
```

If this package or another creates trouble it can be removed from the installation as before an example of one that did not compile was obsAviod.

To removed obsAvoid from the installation and other packages as well remove obsAviod from drdc/src/autonomy MakeFile.am or wherever it resides in the code and remove from configure.ac in the drdcMiro folder.

Once the bootstrap is complete use configure without the fire wire libraries to prepare the installation and create the makefiles.

```
 ./configure --disable-IEEE1394
```

The final commands required to finish the installation of drdcMiro are as follows to complete the installation. The last step will be to put a copy of the Naming Service program to your home directory.

```
make

sudo make install
```

Copy Name Service to the users home directory from /usr/bin

```
cd usr/bin/
cp Naming_Service ~/
```

7 Installing Matlab R2009a

Matlab is a very versatile programming language but installing is a little different than previously done in this manual. Credit for the unintuitive commands goes to the Ubuntu community [2] and the Ubuntu help forms[3]

Make a new directory in /usr /local to hold Matlab once installed

```
sudo mkdir /usr/local/matlabR2009a
cd /usr/local/matlabR2009a
```

Start the install script file on the cd from your current location

```
sudo sh /Media/MATHWORKS_R209A/install
```

At UNB the install must be done manually without Internet so that option is selected in the pop up menu. Should the pop up menu not appear a library is probably not installed. Reference[2] is a good resource for unexpected errors, but noun were experienced in the authors installation.

following through with the direction, Install into the current directory and follow the directions. One special note for Ubuntu is be sure to include the symbolic link to /usr/local/bin during the installation. Initially they look grayed and will not be installed by default. There is a small check box that need to be clicked to enable that feature. Should that option no be available or does not work, the links can be applied manually as follows. The rest of the defaults appeared to work well.

```
sudo ln -s /usr/local/matlabR2009a/bin/matlab  /usr/local/matlab
sudo ln -s /usr/local/matlabR2009a/bin/mbuild  /usr/local/mbuild
sudo ln -s /usr/local/matlabR2009a/bin/mmc   /usr/local/mmc
sudo ln -s /usr/local/matlabR2009a/bin/mex   /usr/local/mex
```

The post installation script included in Matlab did not function as expected or at all. Currently the only way to start Matlab is to use the terminal command `matlab`. To create the Linux equivalent to a desktop shortcut the following commands can be applied

The following command will download and install the Matlab icon.png file

```
sudo wget http://upload.wikimedia.org/wikipedia/commons/2/21/Matlab_Logo.png -O
 /usr/share/icons/matlab.png
```

In Linux a shortcut to a program is called a luncher. Create a luncher by right clicking on the desktop and selecting "Creating a new launcher" the name is what will appear under the icon and the description is what displays when the mouse hovers over it. These can be anything desired, the key command to start Matlab on the desktop is as follows,

```
matlab -desktop
```

.

The icon picture should load after the command is entered in the correct field.

Everything should be installed properly now and a desktop shortcut or launcher to Matlab should exist.

8 References

1. AISS Wiki,"Ubuntu 8.10 i386", DRDC Suffeld, last updated April 29,2009,retrieved Aug 27, 2009.
2. Ubuntu Community,"MATLAB",[Online document], Jan 2010, [2010 Jan 28], Available https://help.ubuntu.com/community/MATLAB
3. Ubuntu forms,"creating shortcut for matlab",[Online document], Aug 2009 [2010 Jan 28], Available http://ubuntuforums.org/showthread.php?t=1141457