

Mirics Limited.			
SDR API Linux Installation			
Applications			
Revision History			
Revision	Release Date:	Reason for Change:	Originator
1.0	15 th May 2014	Pre-Release 0.0.1	APC
1.1	3 rd August 2014	Update for 1.1 build	APC
1.2	28 th May 2015	Include .run details	APC
2.0	31 st May 2015	Rewrite to include apps	APC

Contents

1	Introduction	3
2	Overview	3
2.1	Before you start	3
3	API Installation	4
3.1	Background	4
3.2	udev	4
3.3	libusb	4
3.4	API Install.....	4
4	Gnu Radio	5
4.1	Background	5
4.2	Dependencies.....	5
4.3	Download and Build.....	5
5	gr-osmosdr.....	6
5.1	Background	6
5.2	Download and Build.....	6
6	GQRX	7
6.1	Background	7
6.2	Dependencies.....	7
6.3	Download and Build.....	7
6.4	Start GQRX	7
7	Building with the API.....	8
7.1	Compilation.....	8
7.2	API Library Usage	8
	Legal Information	9

1 Introduction

This document details the steps required for developers to use the Mirics SDR API on a Linux platform. Developers will also need to make sure that users systems are also correctly setup in order for the Mirics SDR API to work fully.

Please note that Ubuntu 14.04 64bit has been used for the development of this API. It may work on other Linux systems but it will have not been tested. If you would like to feedback your experience on other systems please send an email to sdr.support@mirics.com

2 Overview

These instructions are broken into several parts. Depending on what you are trying to do, you may either follow all or just some of the points listed.

The guide will go from a clean Linux install through the API install, and on to 3rd party applications.

In principle this should work on all Linux based systems, but systems such as Raspberry Pi may have processing limitations or compilation quirks that prevent this flow from running exactly as shown.

2.1 Before you start

Before starting it's a good idea to make sure your Linux install is up to date. There are a number of ways to do this, the way it's done under Ubuntu is...

```
sudo apt-get update  
sudo apt-get upgrade
```

Also, it's wise to get the building essentials installed as these are required by a number of the components installed in this guide. To do that...

```
sudo apt-get install build-essential
```

A reboot at this point will allow the upgrade to take effect.

3 API Installation

3.1 Background

The first thing is to get the Mirics SDR API installed. There are 2 dependencies, udev, which allows write access to the hardware and libusb, which is the USB control library.

3.2 udev

The udev system regulates access to attached USB devices. Whether the udev system is already installed or not, the udev-dev module needs to be installed here as is it a dependency to compile the libusb library later.

```
sudo apt-get install libudev-dev
```

The following line beginning with SUBSYSTEM needs to be added to the /etc/udev/rules.d/66-mirics.rules file...

```
e.g. sudo vi /etc/udev/rules.d/66-mirics.rules
```

```
SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", ATTRS{idVendor}=="1df7", ATTRS{idProduct}=="2500", MODE:="0666"
```

Once this file is in place you should reset the udev system by typing:

```
sudo service udev restart
```

Note: please make sure the device is not plugged in until the udev service has been restarted.

3.3 libusb

libusb is a C library that provides access to an attached USB device.

The libusb install can be downloaded from here:

<http://libusb.info>

Once downloaded, unzip the tarball and follow the installation instructions. The basic installation procedure is...

```
cd /to/where/libusb/tar/is/extracted
./configure
make
sudo make install
```

This will build libusb and install it in the /usr/local hierarchy.

3.4 API Install

Now the system is ready for the API to be installed. The Linux API installs will normally end in .run – these are self-contained extracting executables that need to be run as root...

```
chmod 755 filename.run
sudo ./filename.run
sudo ldconfig
```

The API is now ready to be used within your own applications or by other libraries or applications.

4 Gnu Radio

4.1 Background

Gnu Radio is required for the gr-osmosdr library to be installed. The gr-osmosdr library is required for applications such as Gnu Radio to communicate with the Mirics hardware. The website for Gnu Radio is <http://gnuradio.org>

4.2 Dependencies

Gnu Radio requires Git, boost, cmake, cppunit, libfftw3 and pulseaudio, to install those...

```
sudo apt-get install git
sudo apt-get install libboost-all-dev
sudo apt-get install cmake
sudo apt-get install libcppunit-dev
sudo apt-get install libfftw3-dev
sudo apt-get install osspd osspd-pulseaudio
```

4.3 Download and Build

To download and build Gnu Radio, use this sequence...

```
git clone --recursive http://git.gnuradio.org/git/gnuradio.git
cd gnuradio
mkdir build
cd build
cmake ../ -DENABLE_GR_RUNTIME=ON -DENABLE_GR_ANALOG=ON -DENABLE_GR_DIGITAL=ON -DENABLE_GR_BLOCKS=ON \
-DENABLE_GR_FILTER=ON -DENABLE_GR_FFT=ON -DENABLE_GR_AUDIO=ON -DENABLE_GR_IQBALANCE=ON
make && make test
sudo make install
sudo ldconfig
```

5 gr-osmosdr

5.1 Background

Currently the only official Mirics support in gr-osmosdr is via the SDRplay module. Any other gr-osmosdr module for the Mirics hardware is completely unofficial and is not supported by Mirics. For future updates and reference, the website for gr-osmosdr is <http://sdr.osmocom.org/trac/wiki/GrOsmoSDR>

5.2 Download and Build

There are some dependencies but they are taken care of in the Gnu Radio section above. To download and build the gr-osmosdr library...

```
git clone git://git.osmocom.org/gr-osmosdr
cd gr-osmosdr
mkdir build
cd build
cmake ../ -DENABLE_NONFREE=TRUE
make
sudo make install
sudo ldconfig
```

6 GQRX

6.1 Background

GQRX is an application base on Gnu Radio and enables the user to demodulate FM/AM signals via a GUI. The official GQRX website is <http://gqrx.dk/>

6.2 Dependencies

Some are listed above in the Gnu Radio section, the only additional one is qmake...

```
sudo apt-get install qmake
```

6.3 Download and Build

Follow these instructions to download and build GQRX...

```
git clone https://github.com/csete/gqrx.git gqrx.git
cd gqrx.git
mkdir build
cd build
qmake ..
make
sudo make install
```

6.4 Start GQRX

To start using GQRX type the following from the command line...

```
gqrx
```

7 Building with the API

7.1 Compilation

Here is a suggested simplified Makefile in order to successfully compile the SDR API into your application.

```
EXE = mir_test_app

CC = gcc -g
LD = gcc -g
LN = ln

CFLAGS = -fPIC -Wall
INCS = -I/usr/local/include
LINKFLAGS =
LIB_OBJS = main.o
all: $(EXE)
$(EXE): $(LIB_OBJS)
    @$(LD) $(LINKFLAGS) -o $(EXE) $(LIB_OBJS) -lpthread -lc -lm -ldl
%.o: %.c
    @echo "Compiling" $<
    @$(CC) $(CFLAGS) $(INCS) -c $<
```

7.2 API Library Usage

The API documentation has an example piece of code to give an idea of how the functions are to be used. A frequent question has been how to import the library. Here is an example just using the `mir_sdr_Init` function...

```
mir_sdr_Init_t mir_sdr_Init_fn = NULL;
void *ApiDll = NULL;
ApiDll = dlopen("/path/to/API/library.so", RTLD_LAZY);
if (ApiDll == NULL)
{
    printf("API failed to load\n");
    exit(1);
}
mir_sdr_Init_fn = (mir_sdr_Init_t) dlsym(ApiDll, "mir_sdr_Init");
if (mir_sdr_Init_fn == NULL)
{
    printf("Failed to map API functions\n");
    dlclose(ApiDll);
    exit(1);
}
mir_sdr_Init_fn = mir_sdr_Init;
err = mir_sdr_Init_fn(gr, fs, rf, bwType, ifType, &samplesPerPkt);
dlclose(ApiDll);
etc...
```


For more information contact:

Mirics Limited

info@mirics.com

www.mirics.com

Legal Information

Information in this document is provided solely to enable system and software implementers to use Mirics Ltd products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document. Mirics Ltd reserves the right to make changes without further notice to any of its products. Mirics Ltd makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Mirics Ltd assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Typical parameters that may be provided in Mirics Ltd data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters must be validated for each customer application by the buyer's technical experts. Mirics Ltd does not convey any license with the data herein. Mirics Ltd products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Mirics Ltd product could create a situation where personal injury or death may occur. Should Buyer purchase or use Mirics Ltd products for any such unintended or unauthorized application, Buyer shall indemnify and hold Mirics Ltd and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Mirics Ltd was negligent regarding the design or manufacture of the part. Mirics FlexiRF™, Mirics FlexiTV™ and Mirics™ are trademarks of Mirics Ltd