

User Guide for the WyJen OTG Mini RTL2832U/R820T SDR USB DVB-T, DAB, FM, TV Tuner Support MPEG-2, MPEG-4 H.264



Technology for the Wireless Generation

Table of Contents

Introduction	2
Applications	3
Other Compatible WyJen Modules	3
Known Compatible Smart Phones and Tablets.....	3
WyJen SDR Specifications and General Information.....	4
Model Number	4
Features	4
Product Specification.....	4
System Required	5
Package Includes	5
HW and SW Support.....	5
Payment.....	5
Shipping	5
Customer Service:.....	6
Packaging Details.....	7
Part Numbering	7
Free Downloadable Compatible Software Applications.....	8
http://sdr.osmocom.org/trac/wiki/rtl-sdr	8
Ubuntu Linux and Mac Software.....	9
Installation Method for Ubuntu Users.....	9
Install Gqrx	9
Step 1. Setup build environment.....	9
Step 2. Install GNURadio	9

<i>Step 3. Install rtl-sdr</i>	9
<i>Step 4. Test rtl-sdr</i>	10
<i>Step 5. Install gr-osmosdr</i>	11
<i>Step 6. Install gqrx</i>	12
<i>Install dl-fldigi</i>	12
<i>Listening for a Payload</i>	12
<i>Decoding the Payload</i>	13
<i>Hint</i>	13

Introduction

The WyJen Technologies SDR implements the RTL2832U/R820T chipset and is designed to work with multiple devices including Linux / Windows / PCs / MACs /Androids / iPhones. It is a high quality and reliable Software Defined Radio sold at a very reasonable price. USB dongles implementing the Realtek RTL2832U can be used as an inexpensive SDR because the chip can be configured to send the raw I/Q samples to the host (Linux/Windows/MAC PC, iphone, Android), which is officially used for DVB-T/DAB+/FM demodulation. There are also many downloadable online USA applications such as FM radio, ADS-B, etc., that work with this high quality USB RTL2832/R820T dongle. Also, other applications such as DVB-T work in much of Europe, Russia, Germany, Japan, Denmark, Nauru, United Arab Emirates, North Africa, South Africa, etc., on this USB RTL2832U/R820T Dongle.

At the university level this type of technology can become an invaluable research tool as it allows for the development of the SW prototype before any actual HW development.

Also, If the user prefers, this device can be used in an end product because of the PCB mounting holes and the USB interface can be connected directly to a host processor. Interested parties may contact WyJen Technologies for questions.

This product is compatible to the NooElect brand.



Applications

1. FM: both narrow band and wide band. The former is used on two way radio systems such as emergency services and private radio networks (like couriers and taxis) and UHF CB. The latter is the usual broadcast FM which most people are familiar used in their home or vehicle. Aircraft, boats and ships also use narrow band FM which can be listened in on using RTLSDR. The SDR# software can receive both narrowband and wideband FM and the latter which includes FM stereo.
2. ADS-B. The ADS-B application software can be downloaded from www.rf-connect.com/sdr-software.
3. AM: Most AM transmissions are below the bottom frequency of RTL SDR dongles. You will need a translator to get these frequency bands.
4. Upper/Lower Sideband (USB/LSB). See AM above.
5. CW: Continuous wave for morse code enthusiasts.
6. With GNURadio you can receive and demodulate digital modes such as pagers (POCSAG), ADS-B (aircraft positions), AIS (ship positions), AP25 and TETRA (digital trunk radio) and many others.
7. Satellite reception including receiving ham transmissions from the International Space Station are possible as seen by screen shots using RTLSDR and a 2.5m dish to track the carrier signal on deep space robots such as Voyager and the Mars missions.

Other Compatible WyJen Modules

The WyJen SDR works with the following WyJen wireless modules:

Manufacturer/Distributor	Part Number	Description	Sales
WyJen Technologies Incorporated	GTR315M-M13	315MHz ASK transceiver, raw data output, easy configuration	1-800- 490-4165

Some of the targeted applications of the GTR315M_M13 transceiver are:

- Remote controlled garage door openers
- Remote controlled robotics
- Remote keyless entry systems
- Remote controlled toys
- Wireless sensors
- Wireless computer peripherals
- Wireless security systems
- Wireless game consoles
- Wireless pressure monitoring systems

Known Compatible Smart Phones and Tablets

The list of the smart phones and tablets that are known to be USB OTG compatible with the WyJen SDR is provided below:

Acer™ Iconia™ Tab	Archos™ G9	HTC™ One	Samsung Galaxy Tab 7.7
Samsung™ Galaxy™S 2/3/4	Notion Ink™Adam™	Pantech™ Vega R3™	Samsung Galaxy Tab 2 7.0
Samsung Galaxy Nexus™	Motorola XOOM™	Samsung Samsung Ativ™ Tab	Samsung Galaxy Tab 2 10.1
LG™ Optimus G™	Toshiba Excite 10™	Samsung Galaxy Note	Asus™VivoTab RT™
Google™ Nexus 7™	Sony™ Tablet S	Samsung Galaxy Note 10.1	Huawei MediaPad
Motorola™ Droid™RAZR™	Sony Xperia™	Samsung Galaxy Tab 7.0	Sony Tablet
Samsung Galaxy Tab	Huawei™Ascend D/D1™	TrekStor SurfTab	

Note: This list only represents phones and tablets that has been tested thus far.

WyJen SDR Specifications and General Information

Model Number

WRX000-5V-WIQU05C-SDR-A

Features

1. On The Go (OTG) for android, iphones, etc., where applications are available
2. Watch and record digital terrestrial TV on PC or Laptop
3. Full DVB-T Set Top box bandwidth reception (6/7/8 MHz)
4. Still image snapshots
5. Time-shifting
6. Scheduled recording
7. Digital TV recording and playback as DVD quality
8. Support multi-picture display
9. Support EPG (Electronic Program Guide)
10. Support Teletext
11. Supports Wide band FM and Narrow Band FM functions
12. Support Microsoft DAB drivers
13. Remote control direct site at a distance of 40 meters
14. Support Windows 2000/XP/Vista/WIN7/MAC
15. Support both MPEG-2, MPEG-4(H.264) encoding

Product Specification

1. USB 2.0 Interface
2. Terrestrial Antenna Input, female MCX connector
3. Dimensions (LxWxH): 63.5mm*27mm*12mm
4. Weight: 30g

System Required

1. CPU: Pentium IV 1.8 GHz or above
2. Direct X9.0c or above
3. Memory: 128MB
4. 500MB of available hard disk space or above
5. CD-ROM Drive
6. USB 2.0 slot

Package Includes

1. Male MCX antenna (3 ft.)
2. R820T/RTL2832U Dongle, receives 24MHz - 1850MHz

HW and SW Support

For SW issues, we will provide a response within 24 hours

For HW issues, we will provide a response within 24 hours. For HW issues send email to technicalsupport@wyjen.com

Payment

We accept Paypal, West Union, TT. All major credit cards are accepted through secure payment processor ESCROW. Payment should be made within 3 days after order.



Shipping

USA: We arrange standard shipping (registered mail) thru the USA Postal Service (USPS). Registered mail (post office): Normal arrival at destination is 18-29 days. The following is an

estimated time table by registered mail:

Country	Delivery Time					
	Days	5-9	10-18	19-27	28-35	>35
United States	Rate(time arrived)	5%	62%	28%	5%	Refund/Resend
United Kingdom	Rate(time arrived)	5%	63%	27%	5%	Refund/Resend
Australia	Rate(time arrived)	6%	62%	27%	5%	Refund/Resend
France	Rate(time arrived)	4%	39%	42%	15%	Refund/Resend
Germany	Rate(time arrived)	6%	32%	42%	20%	Refund/Resend
Canada	Rate(time arrived)	0	51%	44%	5%	Refund/Resend
Spain	Rate(time arrived)	4%	48%	38%	10%	Refund/Resend
Italy	Rate(time arrived)	2%	19%	40%	39%	Post Claims
Other countries	Normally arrived at 14-29 days, Developing country will take longer time					

Worldwide Shipping: All goods are shipped to you direct from the USA. Shipping costs are based on the current rates of carriers such as UPS, FedEx, DHL, T, etc. Do you have a better way? We will, if you choose, use your preferred air or ocean freight forwarding service to deliver your merchandise to your desired location.

Drop Shipping: What if you don't want to deal with the headaches of handling goods, warehouse space, shipping and product returns? Then drop shipping is the answer. WyJen allows drop-ship orders for all registered customers with no extra charges.

Customer Service:

1. We are a direct manufacturer as well as service oriented organization which you can expect reliable and professional service. Our priority is your satisfaction.
2. Our dedicated sales managers take care of each order from start to finish. If you need assistance or have a question, please do not hesitate to ask. There are no bad questions.
3. Whether you need 10 items or 5,000, your order will be met with the same attention to quality and service.
4. We take your (and our) reputation seriously. We believe in building trust with all our clients.

5. We are ready to collaborate with you in all aspects of your business in order to achieve your business objectives.

Packaging Details

Unit Type: Kit

Package Weight: 0.15kg

Package Size: Padded package

Part Numbering

The part numbering scheme for the [WyJen \(Wyrless Jeneration\)](#) Technologies wireless modules is explained below.

WAABBBC-XX-YYYZWWT-VVV-U

W	WyJen Technologies Inc.
AA	TX stands for Transmitter RX stands for Receiver TR stands for Transmitter and Receiver
BBB	3 digit number representing carrier frequency 2 digit number representing carrier frequency with decimal point represented by D 000 24MHz to 1.805GHz
C	M stands for mega Hertz G stands for Giga Hertz
XX	5V stands for 5 volt technology 3V stands for 3.3V technology 18 stands for 1.8V technology 12 stands for 1.2V technology
YYY	A00 stands for ASK modulation F00 stands for FSK modulation P00 stands for PSK modulation Q00 stands for QAM modulation AF0 stands for ASK and FSK modulation AFP stands for ASK, FSK, and PSK modulation

AM0 stands for amplitude modulation
AFM stands for ASK and FM modulation
FFM stands for FSK and FM modulation
WIQ stands for Wideband frequency range and I/Q data output

- Z** S stands for SIP package
D stands for DIP package
M stands for surface mount package
U stands for USB jack interface
- WW** 2 digit number representing number of input, output, and power pins in the package
- T** A stands for 1.125 inches width by 0.75 inches height
B stands for 2 inches width by 0.75 inches height
C stands for LxWxH = 63.5mm*27mm*12mm
- VVV** 3 character series code
SDR – Software Defined Radio
- U** 1 character feature code
A SDR with ADS-B antenna

Notes:

1. T Field: For SIP package width is side with connector
2. BBB and C Fields are used together: Some example values are
315M
433M
868M
915M
2D5G
5D8G

Free Downloadable Compatible Software Applications

Many other exiting SW applications can run on the WyJen SDR. These software applications and their downloadable sites are provided next.

<http://sdr.osmocom.org/trac/wiki/rtl-sdr>

A list of applications that may be download are provided here:

1. [SDR GUI](#)
2. [SDR GUI](#)
3. [ADS-B RX](#)
4. [AIS RX](#)
5. [GPS RX \(Realtime!\) - Documentation](#)
6. http://www.cel.kit.edu/simulink_rtl_sdr.php

<https://www.cgran.org/wiki/gr-air-modes>

Ubuntu Linux and Mac Software

The following installation steps were derived from:

https://www.jeroenijhof.nl/wiki/index.php?title=Software-Defined_Radio_on_Ubuntu

For your convenience, I have included them here with some extra info.

Installation Method for Ubuntu Users

For Linux there are plenty of SDR related software available but my favorite one is [Gqrx](#).

To get Gqrx working with RTL-SDR is not hard but it will take some time...

Install Gqrx

Step 1. Setup build environment

```
mkdir SDR
cd SDR
```

Some packages will need to be installed to compile everything.

```
sudo apt-get install cmake swig libcppunit-dev python-cheetah doxygen \
libboost-all-dev python-sphinx fftw3-dev python-numpy libqwt-dev \
libusb++-dev libusb-dev libusb-1.0-0-dev libgsl0-dev python-wxgtk2.8 \
qtcreator
```

Step 2. Install GNURadio

Clone the repo and enter its new directory.

```
git clone http://git.gnuradio.org/git/gnuradio.git
cd gnuradio
```

Now make a build directory, enter it, and create the build environment.

```
mkdir build
cd build
cmake ../
```

Now build and install GNURadio

```
make
sudo make install
sudo ldconfig
```

Step 3. Install rtl-sdr

Now go back to the SDR directory and clone the rtl-sdr repo

```
cd ../../
```

```
git clone git://git.osmocom.org/rtl-sdr.git
```

Now enter the directory, create the build directory, and create the rtl-sdr build environment.

```
cd rtl-sdr/  
mkdir build  
cd build  
  
--- if user is not root type the following ----  
cmake ../ -DINSTALL_UDEV_RULES=ON  
  
--- else if user is root type the following ----  
cmake ../
```

Time to build and install rtl-sdr

```
make  
sudo make install  
sudo make install-udev-rules  
sudo ldconfig
```

Step 4. Test rtl-sdr

Just to make sure rtl-sdr works with your dongle, try to run the command **rtl_test**, and check the output

rtl_test -t

```
Found 1 device(s):  
 0: Generic, RTL2832U, SN:77771111153705700  
  
Using device 0: Generic RTL2832U  
Found Rafael Micro R820T tuner  
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7  
16.6 19.7 20.7 22.9 25.4 28.0 29.7 32.8 33.8 36.4 37.2 38.6 40.2 42.1  
43.4 43.9 44.5 48.0 49.6  
Sampling at 2048000 S/s.  
No E4000 tuner found ...
```

Step 4a. If step 4 works you should be able to listen to a FM station

For example, if in Jacksonville, FL, enter the following **rtl_fm** command:

```
rtl_fm -f 97.9e6 -M wbfm -s 200000 -r 48000 - | aplay -r 48k -f S16_LE
```

Found 1 device(s):

0: Generic, RTL2832U, SN: 77771111153705700

Using device 0: Generic RTL2832U

Found Rafael Micro R820T tuner

Tuner gain set to automatic.

Tuned to 98216000 Hz.

Oversampling input by: 6x.

Oversampling output by: 1x.

Buffer size: 6.83ms

Sampling at 1200000 S/s.

Output at 200000 Hz.

Playing raw data 'stdin' : Signed 16 bit Little Endian, Rate 48000 Hz, Mono

Jacksonville Christian Stations

```
rtl_fm -f 90.9e6 -M wbfm -s 200000 -r 48000 - | aplay -r 48k -f S16_LE
```

Others in Jacksonville, FL

```
rtl_fm -f 95.1e6 -M wbfm -s 200000 -r 48000 - | aplay -r 48k -f S16_LE
```

```
rtl_fm -f 96.9e6 -M wbfm -s 200000 -r 48000 - | aplay -r 48k -f S16_LE
```

```
rtl_fm -f 102.9e6 -M wbfm -s 200000 -r 48000 - | aplay -r 48k -f S16_LE
```

For your city, google local FM stations and try tuning to them using the above rtl_fm command.

Step 5. Install gr-osmosdr

Now go back to the SDR directory and clone the gr-osmosdr repo

```
cd ../../  
git clone git://git.osmocom.org/gr-osmosdr
```

Now enter the directory, create the build directory, and create the gr-osmosdr build environment.

```
cd gr-osmosdr/  
mkdir build  
cd build/  
cmake ../ -Wno-dev
```

Time to build and install gr-osmosdr

```
make
sudo make install
sudo ldconfig
```

Step 6. Install gqrx

Now go back to the SDR directory and clone the gqrx repo

```
cd ../../
git clone https://github.com/csete/gqrx.git
```

Now enter the directory and load the project in qtcreator

```
cd gqrx
qtcreator ./gqrx.pro
```

To build the package click Build->Build All. After the build finishes plug in your SDR and go into the build directory to execute gqrx.

```
cd ../gqrx-build-desktop-Qt_4_8_1_in_PATH__System__Release/
./gqrx
```

You should be up and running now.

Install dl-fldigi

[DI Fldigi](#) is an adapted version of the excellent free FLdigi soundcard decoding software. It takes the audio from your radio, decodes the balloon's signal, and then sends the telemetry it's found over the internet to a server running habitat, which plots the payloads position on to the [SpaceNear map](#).

```
sudo apt-add-repository ppa:simrunbasuita/dl-fldigi
sudo apt-get update
sudo apt-get install dl-fldigi
```

Listening for a Payload

So now that we have gqrx and dl-fldigi we can finally start receiving and decoding RTTY! Make sure you have your DVB-T dongle connected to the computer before starting gqrx ;)

```
./gqrx
```

Select your dongle from the list

.

Push the power button, fill in the frequency and make sure that Filter is Normal and Mode is USB. You should see a spike around the frequency of your payload

Just click in front of the spike and you will see a red vertical line appear.
Move this red vertical line until you see two spikes in the Audio Window like below.
At this point you should be hearing the RTTY from your payload

Decoding the Payload

When gqrx is running and you can still hear the RTTY from your payload it's time to decode the RTTY.

dl-fldigi will use the default analog audio input channel so the only thing you need to do is capture the audio from gqrx.

Audio capturing can be done in multiple ways and you need to decide which works best.

·Use alsamixer

```
Type F4 and you will go to the capture channels. Select the channel
named "Capture" and press the space bar. A comment "L - CAPTURE - R"
written in red should appear.
```

·Use gnome-alsamixer

```
Just select the box named "Rec." in the channel named "Capture".
```

·Use a jack-jack plug

```
You can always use a jack to jack plug and redirect the audio from the
headphone output to the microphone input. Just make sure your microphone
input is not muted.
```

Now start dl-fldigi

```
dl-fldigi --hab
```

As you can see the RTTY is being decoded and is perfect readable.

Hint

My payload is sending with 75 baud and a 350hz shift so if yours isn't and I highly doubt you need to adjust these settings in dl-fldigi.

Just click on RTTY in the left corner below