Jan 2017
A simple guide needs to be written to walk a fairly new person thru setting up a Raspberry Pi and a RTL(2832) dongle to listen to the DMR repeater using DSD/DSDCC. I am not sure the Pi2 will have enough snot, but its worth a try. Everything will be done command line, non-gui as that is the most resource efficient way.

DSD is the digital speech decoder. It can decode D-Star, DMR and P25.

This guide will assume you have touched Linux before, but will attempt to be step by step. And later I will explain things in a updated version.

```
pi@raspberrypi ~ $ sudo raspi-config
# Choose option 1 to "Expand Filesystem" - Ensures that all of the SD card storage is available to the OS
# Choose Finish & reboot
pi@raspberrypi ~ $ sudo apt-get update
pi@raspberrypi ~ $ sudo apt-get upgrade
```

Blacklist the normal rtl driver:

```
pi@raspberrypi ~ $ cat <<EOF>no-rtl.conf
blacklist dvb_usb_rtl28xxu
blacklist rtl2832
blacklist rtl2830
EOF
pi@raspberrypi ~ $ sudo mv no-rtl.conf /etc/modprobe.d/

pi@raspberrypi ~ $ sudo apt-get install git git-core cmake
pi@raspberrypi ~ $ sudo apt-get install libusb-1.0-0-dev
pi@raspberrypi ~ $ sudo apt-get install build-essential sox
```

Install the modified RTL drivers and SDR tools:

```
pi@raspberrypi ~ $ git clone git://git.osmocom.org/rtl-sdr.git
pi@raspberrypi ~ $ cd rtl-sdr/
pi@raspberrypi ~/rtl-sdr $ mkdir build && cd build
pi@raspberrypi ~/rtl-sdr/build $ cmake ../ -DINSTALL_UDEV_RULES=ON
pi@raspberrypi ~/rtl-sdr/build $ make
pi@raspberrypi ~/rtl-sdr/build $ sudo make install
pi@raspberrypi ~/rtl-sdr/build $ sudo ldconfig
pi@raspberrypi ~/rtl-sdr/build $ cd ~
pi@raspberrypi ~ $ sudo cp ./rtl-sdr/rtl-sdr.rules /etc/udev/rules.d/
pi@raspberrypi ~ $ sudo reboot

pi@raspberrypi ~ $ rtl_test
```

```
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.

Info: This tool will continuously read from the device, and report if
samples get lost. If you observe no further output, everything is fine.

Reading samples in async mode...
```

Apparently you need the mbe library even if you use a AMBE dongle:

```
pi@raspberrypi:~/dsdcc/build $ cd ~
pi@raspberrypi ~ $ git clone https://github.com/szechyjs/mbelib.git
pi@raspberrypi ~ $ cd mbelib/
pi@raspberrypi:~/mbelib $ mkdir build && cd build
pi@raspberrypi:~/mbelib/build $ cmake ..
pi@raspberrypi:~/mbelib/build $ make
pi@raspberrypi:~/mbelib/build $ sudo make install
pi@raspberrypi:~/mbelib/build $ cd ~
```

Apparently you also need to build serialDV even if you don't use it:

```
pi@raspberrypi ~ $ git clone https://github.com/f4exb/serialDV.git
pi@raspberrypi ~ $ cd serialDV/
pi@raspberrypi:~/serialDV $ mkdir build && cd build
pi@raspberrypi:~/serialDV/build $ cmake ..
pi@raspberrypi:~/serialDV/build $ make
pi@raspberrypi:~/serialDV/build $ sudo make install
pi@raspberrypi:~/serialDV/build $ cd ~
```

Now install the new DSD (use the cmake "*-DUSE_MBELIB=ON*" option only if you need mbelib support, no AMBE dongle):

```
pi@raspberrypi ~ git clone https://github.com/f4exb/dsdcc.git
pi@raspberrypi ~ cd dsdcc/
pi@raspberrypi:~/dsdcc $ mkdir build && cd build
pi@raspberrypi:~/dsdcc/build $
pi@raspberrypi:~/dsdcc/build $ cmake -DUSE_MBELIB=ON ..
```

edit CMakeCache.txt in the build directory to reflect (per a reddit post):

```
LIBSERIALDV_INCLUDE_DIR:PATH=/usr/local/include/serialdv
LIBSERIALDV_LIBRARY:FILEPATH=/usr/local/lib/libserialdv.so
pi@raspberrypi:~/dsdcc/build $
```

```
pi@raspberrypi:~/dsdcc/build $ make
pi@raspberrypi:~/dsdcc/build $ sudo make install
-----------
```

Or install the old version of DSD:
First Dependencies

```
pi@raspberrypi ~ $ sudo apt-get install libsndfile1-dev fftw3-dev liblapack-dev portaudio19-dev
```

#Building itpp IT++ is a C++ library of mathematical, signal processing and communication
#classes and functions. Its main use is in simulation of communication systems and for
#performing research in the area of communications. It does a lot of the heavy lifting for dsd.

```
pi@raspberrypi ~ $ wget -O itpp-latest.tar.bz2 
http://sourceforge.net/projects/itpp/files/latest/download?source=files
pi@raspberrypi ~ $ tar xjf itpp-latest.tar.bz2
pi@raspberrypi ~ $ cd itpp-4.3.1/
pi@raspberrypi:~/itpp-4.3.1 $ mkdir build && cd build
#make -j tells it to use all the cores it can
#takes a while and may seem not so promising
pi@raspberrypi:~/itpp-4.3.1/build $ make -j
pi@raspberrypi:~/itpp-4.3.1/build $ sudo make install
pi@raspberrypi:~/itpp-4.3.1/build $ cd ~
```

The program

```
pi@raspberrypi ~ $ git clone https://github.com/szechyjs/dsd.git
pi@raspberrypi ~ $ cd dsd/
pi@raspberrypi:~/dsd $ mkdir build && cd build
pi@raspberrypi:~/dsd/build $ cmake ..
pi@raspberrypi:~/dsd/build $ make
pi@raspberrypi:~/dsd/build $ sudo make install
pi@raspberrypi:~/dsd/build $ sudo ldconfig /usr/local/lib
-----------
```

Now everything should be installed, and it's a matter of usage. -help will always tell you about command line options.

```
pi@raspberrypi:~ $ dsd -help
Digital Speech Decoder 1.7.0-dev (build:v1.6.0-86-g7ee04e5)
mbelib version 1.3.0

Usage:
  dsd [options]            Live scanner mode
  dsd [options] -r <files> Read/Play saved mbe data from file(s)
  dsd -h                   Show help

Display Options:
  -e            Show Frame Info and errorbars (default)
  -pe           Show P25 encryption sync bits
  -pl           Show P25 link control bits
  -ps          Show P25 status bits and low speed data
  -pt          Show P25 talkgroup info
  -q           Don't show Frame Info/errorbars
  -s           Datascope (disables other display options)
  -t           Show symbol timing during sync
  -v <num>     Frame information Verbosity
  -z <num>     Frame rate for datascope

Input/Output options:
  -i <device>  Audio input device (default is /dev/audio, - for piped stdin)
  -o <device>  Audio output device (default is /dev/audio)
  -d <dir>     Create mbe data files, use this directory
  -r <files>   Read/Play saved mbe data from file(s)
  -g <num>     Audio output gain (default = 0 = auto, disable = -1)
  -n           Do not send synthesized speech to audio output device
  -w <file>    Output synthesized speech to a .wav file
  -a           Display port audio devices

Scanner control options:
  -B <num>     Serial port baud rate (default=115200)
  -C <device>  Serial port for scanner control (default=/dev/ttyUSB0)
  -R <num>     Resume scan after <num> TDULC frames or any PDU or TSDU

Decoder options:
  -fa          Auto-detect frame type (default)
  -f1          Decode only P25 Phase 1
  -fd          Decode only D-STAR
  -fi          Decode only NXDN48* (6.25 kHz) / IDAS*
  -fn          Decode only NXDN96 (12.5 kHz)
  -fp          Decode only ProVoice*
  -fr          Decode only DMR/MOTOTRBO
  -fx          Decode only X2-TDMA
  -l           Disable DMR/MOTOTRBO and NXDN input filtering
  -ma          Auto-select modulation optimizations (default)
  -mc          Use only C4FM modulation optimizations
  -mg          Use only GFSK modulation optimizations
  -mq          Use only QPSK modulation optimizations
  -pu          Unmute Encrypted P25
  -u <num>     Unvoiced speech quality (default=3)
  -xx          Expect non-inverted X2-TDMA signal
  -xr          Expect inverted DMR/MOTOTRBO signal

  * denotes frame types that cannot be auto-detected.
```

```
Advanced decoder options:
  -A <num>      QPSK modulation auto detection threshold (default=26)
  -S <num>      Symbol buffer size for QPSK decision point tracking
                (default=36)
  -M <num>      Min/Max buffer size for QPSK decision point tracking
                (default=15)
```

```
pi@raspberrypi:~ $ dsdccx -help
DSDDstar::reset_header_strings
DSDDecoder::resetFrameSync: symbol 0 (0)
Digital Speech Decoder DSDcc

Usage:
  dsd [options] Live scanner mode
  dsd -h        Show help

Display Options:
  -e            Show Frame Info and errorbars (default)
  -pe           Show P25 encryption sync bits - not supported
  -pl           Show P25 link control bits - not supported
  -ps           Show P25 status bits and low speed data - not supported
  -pt           Show P25 talkgroup info - not supported
  -q            Don't show Frame Info/errorbars
  -t            Show symbol timing during sync
  -v <num>      Frame information Verbosity

Input/Output options:
  -i <device>   Audio input device (default is /dev/audio, - for piped stdin)
  -o <device>   Audio output device (default is /dev/audio, - for stdout)
  -g <num>      Audio output gain (default = 0 = auto, disable = -1)
  -U <num>      Audio output upsampling
                0: no upsampling (8k) default
                6: normal upsampling to 48k
                7: 7x upsampling to trade audio drops against bad audio quality
  -n            Do not send synthesized speech to audio output device
  -L <filename> Log messages to file with file name <filename>. Default is stderr
                If file name is invalid messages will go to stderr
  -D <device>   Use DVSI AMBE3000 based device for AMBE decoding (e.g. ThumbDV)
                You must have compiled with serialDV support (see Readme.md)
                Device name is the corresponding TTY USB device e.g /dev/ttyUSB0

Scanner control options:
  -R <num>      Resume scan after <num> TDULC frames or any PDU or TSDU
```

```
Decoder options:
  -d <num>      Set data rate:
     0          2400 bauds
     1          4800 bauds (default)
     2          9800 bauds
  -fa           Auto-detect frame type (default)
  -fr           Decode only DMR/MOTOTRBO
  -fd           Decode only D-STAR
  -fm           Decode only DPMR Tier 1 or 2 (6.25 kHz)
  -fy           Decode only YSF
  -fi           Decode only NXDN48 (6.25 kHz) / IDAS* - detection only
  -fn           Decode only NXDN96 (12.5 kHz)  - detection only
  -f1           Decode only P25 Phase 1 - not supported
  -fp           Decode only ProVoice - not supported
  -fx           Decode only X2-TDMA - not supported
  -T <num>      TDMA slots processed:
     0          none
     1          slot #1 (default) use this one for FDMA
     2          slot #2
     3          slots #1+2 mixed
  -l            Disable matched filter
  -pu           Unmute Encrypted P25 - not supported
  -u <num>      Unvoiced speech quality (default=3)

pi@raspberrypi:~ $
```

Usage
Using rtl_fm to listen to the Green Bay DMR repeater (442.83125) and piping that into the old or new DSD....

Old DSD:

```
rtl_fm -f 442.83125M -g 45 -s 16k | sox -t raw -r 16k -b 16 -c 1 -e signed-integer - -r 48k -t raw - | dsd -i - -o - | play -q -t s16 -r 8k -c 1 -
```

New DSD:

```
rtl_fm -f 442.83125M -M fm -g 100 -s 70K -r 48K -E dc - | dsdccx -fr -i - -o - -U 0 -g 50 | play -q -t s16 -r 8k -c 1 -
```

Observations:
There's lots of skipped/missing audio bits.
You need serialDV if you planned on just mbelib.
I get the "Error writing to output" repeatedly if not using the ambe3000 dongle.

The -D /dev/ttyUSB0 bit tells dsdcxx to use the AMBE3000 dongle:

```
rtl_fm -f 442.8312M -M fm -g 50 -s 70K -r 48K -E dc - | dsdccx -fr -i - -o - -D /dev/ttyUSB0 -g 50 |
aplay -f S16_LE -
```