[[sx1276]]

Semtech SX1276 M17

SX1276 (https://web.archive.org/web/20220726075750/https://www.semtech.com/products/wireless-rf/lora-core/sx1276) supports M17 transmit (receiver work is in progress). Transmit tests have been performed using a PJRC Teensy 4.1 (https://web.archive.org/web/20220726075750/https://www.pjrc.com/store/teensy41.html) paired with an Adafruit RFM96 433MHz SX1276 radio module (https://web.archive.org/web/20220726075750/https://www.adafruit.com/product/3073). A static M17 baseband modulation file was constructed using PC based tools. This file was used to modulate the SX1276 generating a true M17 physical layer transmission.

The key is placing the SX1276 in FSK mode and writing the modulation data to the low byte of the deviation register *RegFdevLsb* (0x05) at a rate of 48kHz. While the base frequency of the SX1276 can only be changed during certain device states/transitions, the deviation register can be updated in real time.

Teensy 4.1 to Adafruit RFM96 radio module Wiring

A PJRC Teensy 4.1 (https://web.archive.org/web/20220726075750/https://www.pjrc.com/store/teensy41.html) is connected to an Adafruit RFM96 433MHz SX1276 radio module (https://web.archive.org/web/20220726075750/https://www.adafruit.com/product/3073)

Teensy Pin	Radio Module Pin	Function
3.3V	VIN	3.3V
GND	GND	Ground
10	CS	SPI Chip Select
13	SCK	SPI Clock
11	MOSI	SPI MOSI
12	MISO	SPI MISO
33	RST	SX1276 Reset
36	G0	SX1276 DIO0
35	G1	SX1276 DIO1
34	G2	SX1276 DIO2

For the receive testing, the Teensy 4.1 needs to be connected to a LAN () (assumed IP class C addressing, requires DHCP) via ethernet using an Ethernet Kit (https://web.archive.org/web/20220726075750/https://www.pjrc.com/store/ethernet_kit.html). S

How to perform the transmit test

The following assumes you have some experience (and tools) creating an M17 bitstream. See Transmitting M17 with rpitx to build tools for a Raspberry Pi. Specifically, follow the instructions from Set Up Your Environment through Compile and Install m17-cxx-demod.

Record an audio message

Audacity (https://web.archive.org/web/20220726075750/https://www.audacityteam.org/) is used to record an audio message (i.e. KA1PQK Testing). Export the file as WAV, Mono, 8kHz, 16bit PCM (ka1pqk_out.wav)

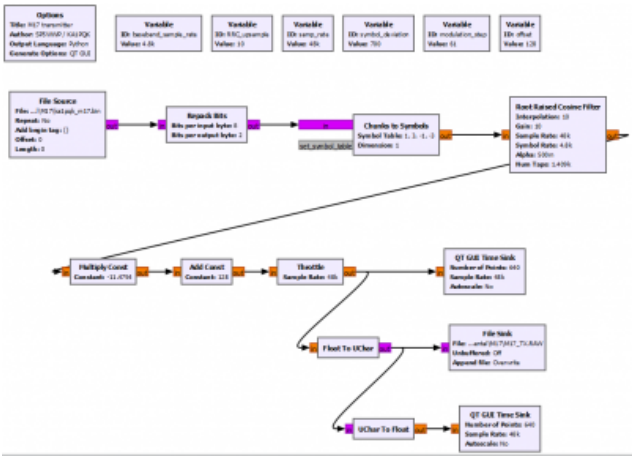
Generate the M17 bitstream file from a WAV audio file

```
sox ka1pqk_out.wav -t raw - | m17-mod -S KA1PQK -b > ka1pqk_m17.bin
```

Note: m17-mod could be used in baseband mode with the correct parameters to combine the next two sections as one operation.

- sox plays the WAV file and outputs raw PCM data
- m17-mod generates M17 bitstream frames (-b option)
  - packed 2 bits per symbol, 4 symbols per byte, symbol rate of 4.8kHz
  - callsign is added (-S option)
- frames are saved (ka1pqk_m17.bin)

Create the modulated baseband file from the bitstream file



gnuRadio (https://web.archive.org/web/20220726075750/https://www.gnuradio.org/) flowgraph can be found here (https://web.archive.org/web/20220726075750/https://github.com/robojay/M17_SX1276_Testing/blob/main/gnuRadio/M17-To-Raw.grc).

- reads M17 bitstream from the file (ka1pqk_m17.bin)
- unpacks the symbols from the bytes (2 bits per symbol, 4 symbols per byte)
- maps 2 bit values to float values (1, 3, -1, -3)
- performs Root Raised Cosine filtering (also interpolates from 4.8kHz symbol rate to 48kHz modulation rate)
- adjusts values for frequency deviation of the SX1276
- converts floating point value to unsigned byte
- saves 48kHz modulation byte stream (M17_TX.RAW)

M17_TX.RAW is saved on a micro SD card

Transmit

The micro SD card goes into the Teensy

Teensy Firmware Description

- SX1276 is reset, then minimally configured for FSK operation
- 48kHz periodic interrupt routine
  - reads the current modulation byte
  - sets the low byte of deviation register (RegFdevLsb 0x05) with the modulation byte
  - requests a new modulation byte
- loop() function
  - reads modulation bytes from the SD card (M17_TX.RAW) when requested
  - replays file from the beginning when the end is found

Note: This is an extremely brute force approach with no buffering, etc. It was the bare minimum to perform the test. Fortunately the Teensy 4.1 is an amazingly fast processor and very few samples were dropped.

Teensy Firmware for this test can be found here (https://web.archive.org/web/20220726075750/https://github.com/robojay/M17_SX1276_Testing/tree/main/Teensy41_SX1276_Tx_Test).

Receive

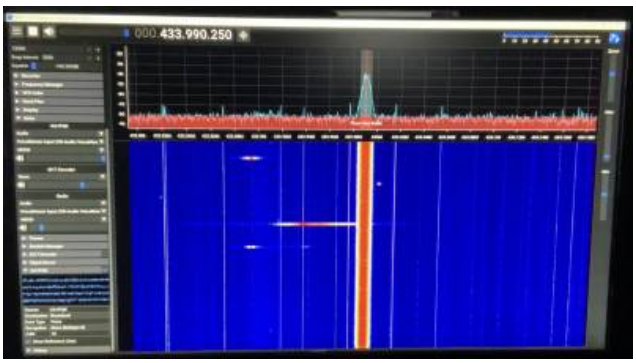
SDR++ (https://web.archive.org/web/20220726075750/https://github.com/AlexandreRouma/SDRPlusPlus) with the M17 decoder is connected to an Airspy R2 (https://web.archive.org/web/20220726075750/https://airspy.com/airspy-r2/)

- M17 decoder is enabled
- fine tune while observing the incoming signal versus the M17 reference lines
- full M17 decoding will happen when everything is aligned properly

Note: If the received signal spacing does not match the reference line spacing, adjustment of the frequency deviation parameters in the gnuRadio flowgraph may be required

Transmit Results

SX1276 transmission at approximately 434MHz decoded with SDR++ (click image for video)



How to perform the receive test

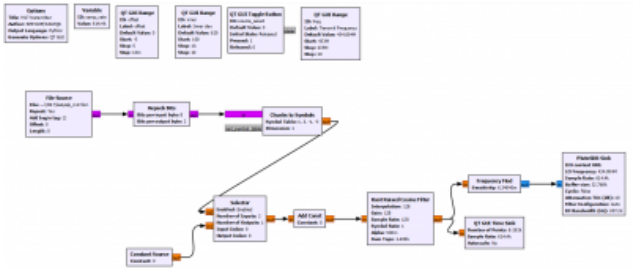
The following assumes you have some experience (and tools) creating an M17 bitstream. See Transmitting M17 with rpitx to build tools for a Raspberry Pi. Specifically, follow the instructions from Set Up Your Environment through Compile and Install m17-cxx-demod.

This also requires an RF source transmitting M17. For this test, an Analog Devices ADALM-Pluto SDR (https://web.archive.org/web/20220726075750/https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html#eb-overview) was used along with gnuRadio (https://web.archive.org/web/20220726075750/https://www.gnuradio.org/) (version 3.10 and higher include Pluto support in the main package).

As this is a very basic test setup where the transmitter and receiver frequencies are not going to be perfect, the transmitter frequency will need to be adjusted until acceptable results are obtained by the receiver.

Transmit a test message

Create an M17 bitstream file as described **here** and **here**. (ka1pqk_m17.bin)



The Pluto SDR can use the gnuRadio (https://web.archive.org/web/20220726075750/https://www.gnuradio.org/) flowgraph to transmit here (https://web.archive.org/web/20220726075750/https://github.com/robojay/M17_SX1276_Testing/blob/main/gnuRadio/M17-Pluto-Tx.grc).

Baseband Receive

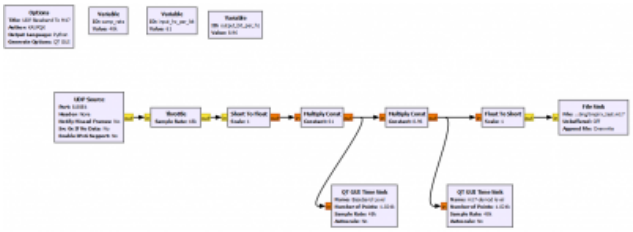
The firmware can be configured to stream either raw or scaled deviation values. Set the rawDeviation constant appropriately.

Teensy Firmware Description

- SX1276 is reset, then minimally configured for FSK operation
- Ethernet is configured
- 48kHz periodic interrupt routine
  - reads deviation from the frequency offset registers (RegFeiMsb 0x1d and RegFeiLsb 0x1e)
  - scales the deviation
  - places raw values or scaled deviation into buffer
  - flags when buffer is full
- loop() function
  - when buffer is full, broadcasts UDP packet
  - (may be toggled on/off by hitting a key in the terminal window)

Teensy Firmware for this test can be found here (https://web.archive.org/web/20220726075750/https://github.com/robojay/M17_SX1276_Testing/tree/main/Teensy41_SX1276_Rx_Test).

Baseband Demodulation (raw values)



gnuRadio (https://web.archive.org/web/20220726075750/https://www.gnuradio.org/) flowgraph to demodulate UDP stream here (https://web.archive.org/web/20220726075750/https://github.com/robojay/M17_SX1276_Testing/blob/main/gnuRadio/UDP-Baseband-To-M17.grc).

- listens for UDP baseband stream
- scales raw values
- saves scaled 48kHz baseband stream (rx_test.m17)

Demodulate the baseband file using m17-demo

```
cat rx_test.m17 | m17-demod -l -v | play -b 16 -r 8000 -c1 -t s16 -
```

Note: It is extremely useful to generate a constant zero deviation from the transmitter, activate the Teensy receive streaming, and monitor the deviation values using the UDP-Baseband-To-M17 gnuRadio flowgraph. If the zero deviation transmission is not showing up as zero in the time sink windows, adjust the transmitter frequency appropriately. This is bleeding edge…

Baseband Demodulation (scaled values)

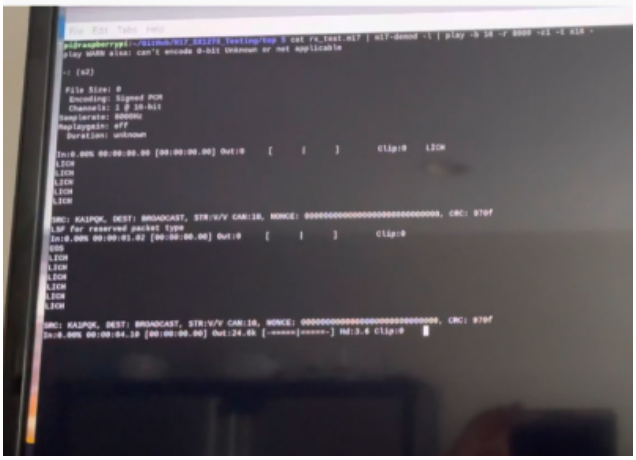
Scaled deviation values can be directly piped into m17-demod.

```
nc -u -l 8888 -k | m17-demod -l -v | play -b 16 -r 8000 -c1 -t s16 -
```

Note: Adjustment of the transmitter frequency will likely be required. Performing the zero deviation check/adjustment from the note in the prior section will get things in the right ballpark. This is bleeding edge…

Receive Results

SX1276 reception (click image for video)



sx1276.txt Last modified: 2022/02/28 17:24 by ka1pqk

M17 Project Wiki

Bleeding edge Ham radio!