OFDM Spectrum

Coronavirus version. In this lab the output is to a frequency sink instead of an SDR so that the lab can be done without test equipment. Version 2 - corrected number of screen captures required from 5 to 4.

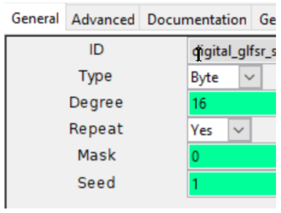
Introduction

In this lab you will create an OFDM transmitter using GNU Radio Companion ~~and an Analog Devices ADALM-Pluto SDR~~. You will then measure the effect of changing various parameters on the power spectrum of the OFDM signal.

Create an OFDM Transmitter

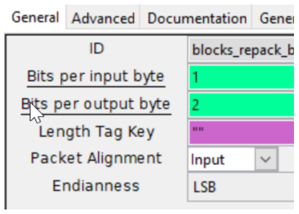
The OFDM transmitter is shown in Figure 1 and is composed of the following blocks:

- The block contains four **Variable** blocks to configure the flowgraph:
  - `samp_rate` the sample rate, initially set to 1 MHz
  - `fft_len` the OFDM symbol length in samples, initially 128
  - `ncarrier` the number of data subcarriers placed in each OFDM symbol, initially 32
  - `cp_len` the length of the cyclic prefix added to each OFDM symbol, initially 16
- the **GLFSR**[1] Source block generates random 0/1 values (bits). The period of the sequence is $2^{\text{Degree}} - 1$. In this case the sequence will repeat after 65535 bits.



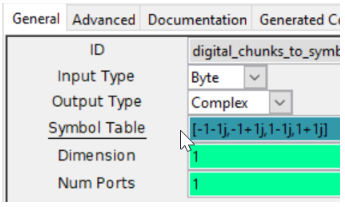
ID	digital_glfsr_s
Type	Byte
Degree	16
Repeat	Yes
Mask	0
Seed	1

- the **Repack Bits** block groups/ungroups bits. In this case it collects two one-bit samples and creates one two-bit value (i.e. value between 0 and 3).



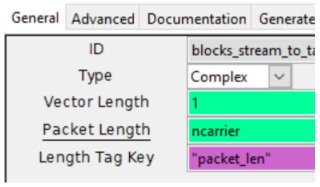
ID	blocks_repack_b
Bits per input byte	1
Bits per output byte	2
Length Tag Key	""
Packet Alignment	Input
Endianness	LSB

- the **Chunks to Symbols** block is configured to map 2-bit numbers into a complex value representing one of four QPSK constellation points. The array with the four possible complex values must be in Python syntax. Imaginary values are indicated with a `j` suffix.



ID	digital_chunks_to_symb
Input Type	Byte
Output Type	Complex
Symbol Table	[-1-1j,-1+1j,1-1j,1+1j]
Dimension	1
Num Ports	1

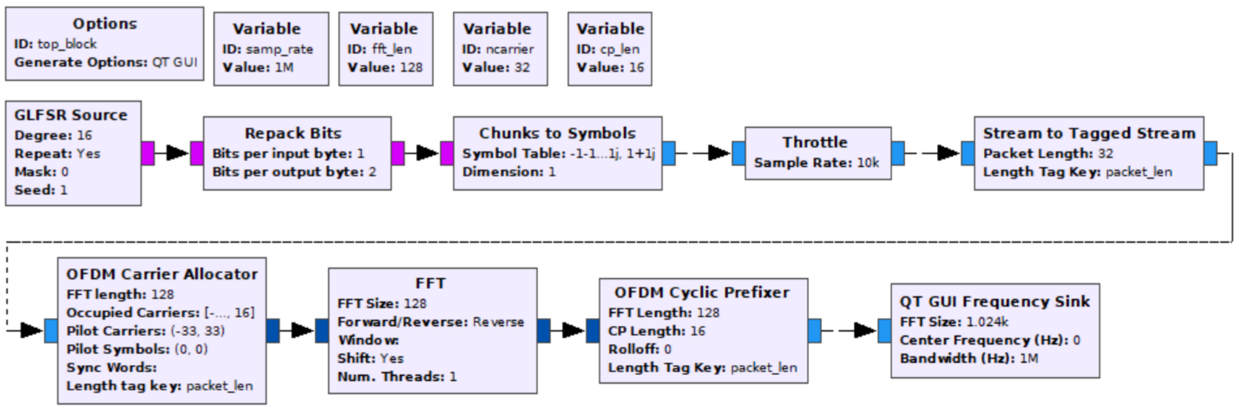
- the **Stream to Tagged Stream** block adds a "tag" object to the sample stream. Here the tag is being used indicate the length of a group of samples that should be processed together. In this case the number of subcarriers to be placed in each OFDM symbol (`ncarrier`) in placed in a tag labelled `packet_len`.



ID	blocks_stream_to_t
Type	Complex
Vector Length	1
Packet Length	ncarrier
Length Tag Key	"packet_len"

- the **Throttle** block reduces the real-time sampling rate. Since there is no actual sink or source to set the sampling rate, the flowgraph will process samples as quickly as it can. This uses up all of the CPU time and makes the computer unresponsive. A typical PC should be able to compute at a sample rate of 10 k samples per second as configured in the flow graph in Figure 1.

[1]Galois linear feedback shift register



Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 1M

Variable
ID: fft_len
Value: 128

Variable
ID: ncarrier
Value: 32

Variable
ID: cp_len
Value: 16

GLFSR Source
Degree: 16
Repeat: Yes
Mask: 0
Seed: 1

Repack Bits
Bits per input byte: 1
Bits per output byte: 2

Chunks to Symbols
Symbol Table: -1-1...1j, 1+1j
Dimension: 1

Throttle
Sample Rate: 10k

Stream to Tagged Stream
Packet Length: 32
Length Tag Key: packet_len

OFDM Carrier Allocator
FFT length: 128
Occupied Carriers: [-..., 16]
Pilot Carriers: (-33, 33)
Pilot Symbols: (0, 0)
Sync Words:
Length tag key: packet_len

FFT
FFT Size: 128
Forward/Reverse: Reverse
Window:
Shift: Yes
Num. Threads: 1

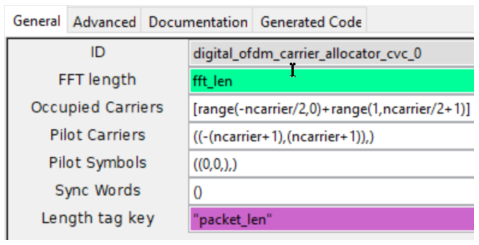
OFDM Cyclic Prefixer
FFT Length: 128
CP Length: 16
Rolloff: 0
Length Tag Key: packet_len

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 1M

Figure 1: GRC flowgraph for an OFDM transmitter.

- the **OFDM Carrier Allocator** block places the data and pilot values into the correct subcarriers locations in an array of `fft_len` values (with zeros elsewhere). The data values are placed on the subcarriers nearest the carrier but not on the carrier frequency itself (index 0).

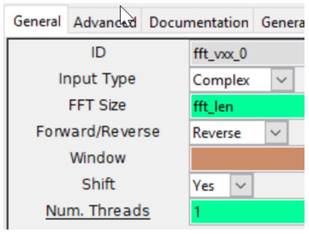
  Pilots subcarriers are not needed since we are not implementing a receiver. However, the block does not work properly without them so zero-value pilots are added on either side of the data subcarriers.



General | Advanced | Documentation | Generated Code

ID	digital_ofdm_carrier_allocator_cvc_0
FFT length	fft_len
Occupied Carriers	[range(-ncarrier/2,0)+range(1,ncarrier/2+1)]
Pilot Carriers	((-(ncarrier+1),(ncarrier+1)),)
Pilot Symbols	((0,0),)
Sync Words	0
Length tag key	"packet_len"

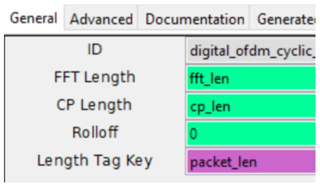
- the **FFT** block performs in inverse FFT on `fft_len` frequency-domain values (the complex subcarrier values) samples to generate `fft_len` complex time-domain samples



General | Advanced | Documentation | Genera

ID	fft_vxx_0
Input Type	Complex
FFT Size	fft_len
Forward/Reverse	Reverse
Window	
Shift	Yes
Num. Threads	1

- the **OFDM Cyclic Prefix** block adds a cyclic prefix of `cp_len` samples to each OFDM symbol



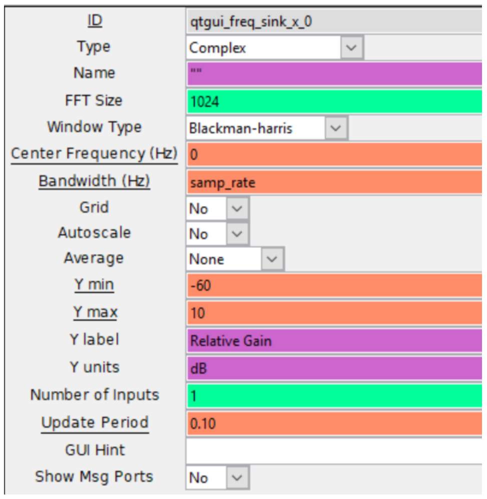
General | Advanced | Documentation | Generate

ID	digital_ofdm_cyclic_
FFT Length	fft_len
CP Length	cp_len
Rolloff	0
Length Tag Key	packet_len

- ~~the **Pluto SDR Sink** block represents the software that configures and writes samples to the SDR. The SDR's RF output is connected to the spectrum analyzer so you can measure the power spectrum of the OFDM signal.~~

  ~~If the Pluto SDR sink block is not available, follow the instructions at: `https://tcom.bcit.ca/` under the link "how to Add Pluto SDR Support" and click on the Reload Blocks (🔄) button.~~

- a Frequency Sink block that displays the baseband spectrum of the signal. The only change from default settings is the Y-axis range:

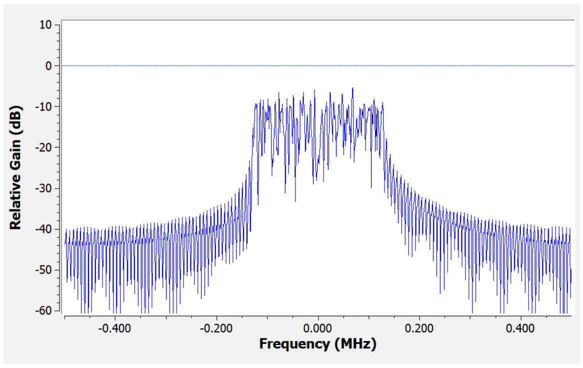


ID	qtgui_freq_sink_x_0
Type	Complex
Name	""
FFT Size	1024
Window Type	Blackman-harris
Center Frequency (Hz)	0
Bandwidth (Hz)	samp_rate
Grid	No
Autoscale	No
Average	None
Y min	-60
Y max	10
Y label	Relative Gain
Y units	dB
Number of Inputs	1
Update Period	0.10
GUI Hint	
Show Msg Ports	No

Measure the Power Spectrum

~~Connect the Pluto SDR's USB port (the one marked with the USB symbol) to a USB port on the PC. Connect the TX output of the Pluto SDR to the RF input of a spectrum analyzer using the supplied SMA-to-BNC cable:~~

Run the flowgraph. Ignore warnings about missing xterm or broken libusb. ~~This flowgraph has no GUI widgets so you will just see a blank window while it is running.~~ You will see the spectrum of the OFDM signal:



Change Parameters

Change each of the following parameters and observe the effect on the spectrum. To keep things simple, make the following changes relative to the first measurement rather than cumulative.

- change `fft_len` from 128 to 256
- change `ncarrier` from 32 to 64
- change `samp_rate` from 1 MHz to 2 MHz
- ~~change the spectrum analyzer span from 500 kHz to 5 MHz~~

In each case, try to predict the effect of the change. Capture the spectrum in each case. By the end of the lab you should have a screen capture of your flowgraph and ~~five spectrum analyzer capture~~ four frequency sink files for your report.

Lab Report

Your lab report, in addition to the usual identification information, should contain:

- a screen capture of your Gnu Radio Companion flow graph
- the ~~spectrum analyzer~~ frequency sink screen captures for the ~~five~~ four conditions listed above
- brief explanations of the shape of the initial spectrum and how each change affected the observed spectrum.