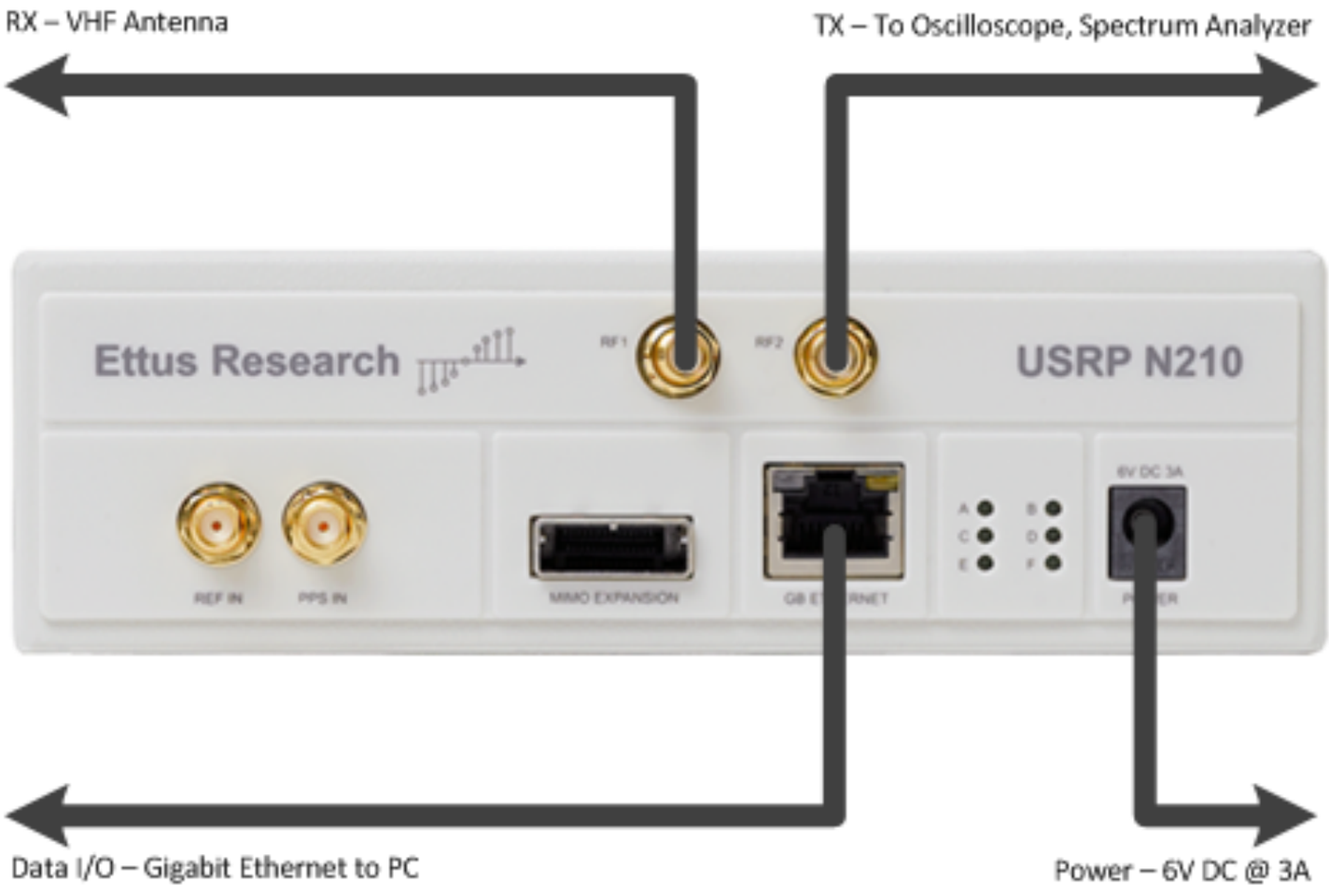
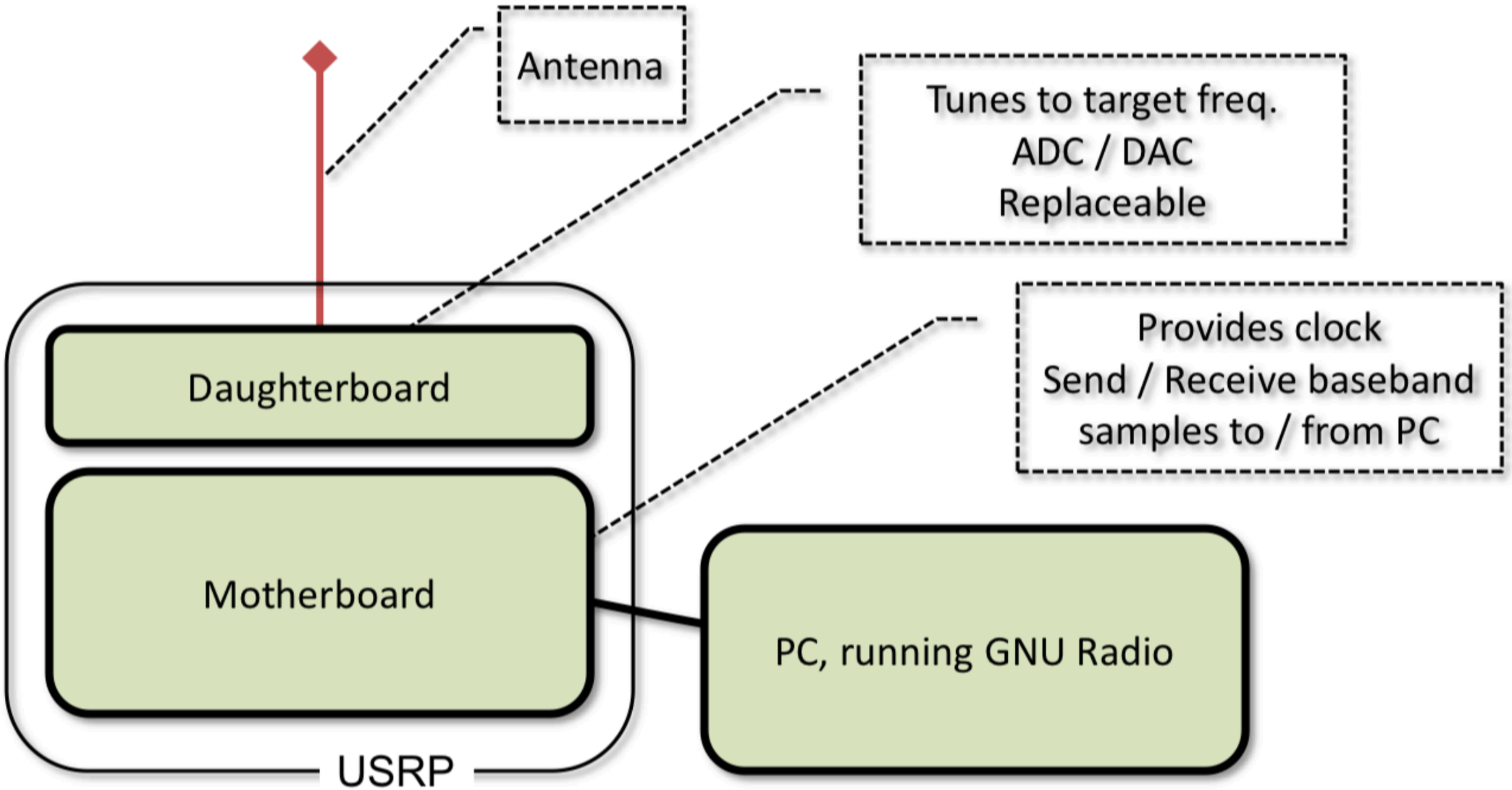
USRP and GNU Radio

Universal Software Radio Peripheral (USRP)

- Used for generating and receiving signals
- Efficient communication system prototyping



USRP structure



Setup USRP Part 1

- Connect the USRP to the computer
- Power up the USRP
- Open the terminal and run the following commands
  - sudo service network-manager stop
  - sudo ifconfig ethX 192.168.X.X
    - ✓ The first X is a unique number corresponding to a specific interface. You can use "ifconfig" cmd to check the number.
    - ✓ The second X is according to your USRP device IP
    - ✓ Set the third X whatever you want (except the USRP's IP)

Setup USRP Part 11

- Install UHD (universal hardware driver)
  - http://www.ettus.com/kb/category/software- documentation/installation
  - Source install
- Install GNU radio
  - Sudo apt-get install gnuradio-companion

Some useful functions

- **uhd_find_devices**
  - Devices attached to your system can be discovered

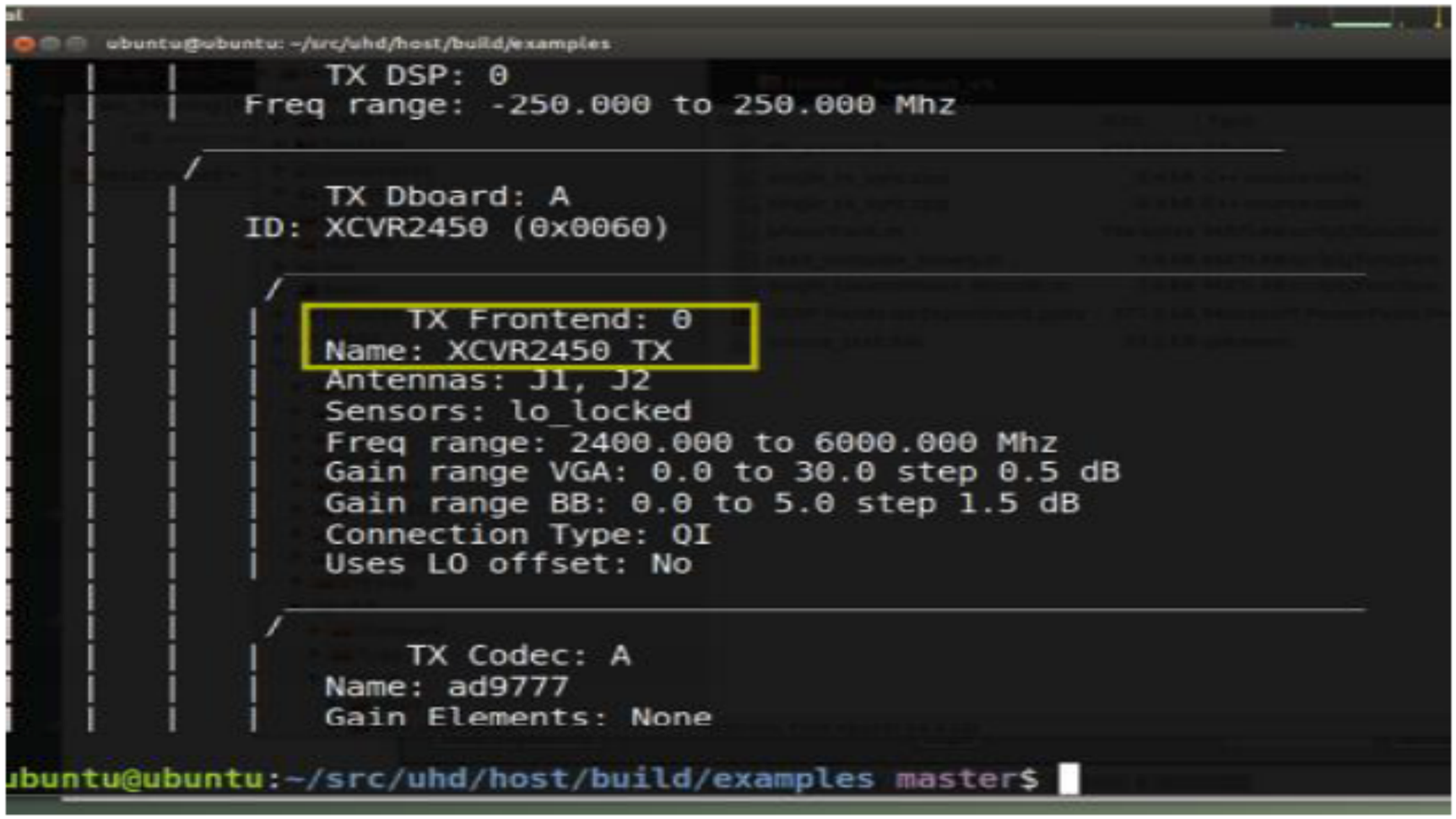
```
ubuntu@ubuntu:~/src/uhd/host/build/examples master$ uhd_find_devices
linux; GNU C++ version 4.7.2; Boost_104900; UHD_003.005.002-59-g638a41bb

--------------------------------------------------
-- UHD Device 0
--------------------------------------------------
Device Address:
    type: usrp2
    addr: 192.168.91.9
    name:
    serial: F42257
```

  - If no device is found, make sure you have connected the Ethernet cable, powered up the USRP and choose the right interface configuration

Some useful functions

- **uhd_usrp_probe**
  - See the properties of devices attached to your system

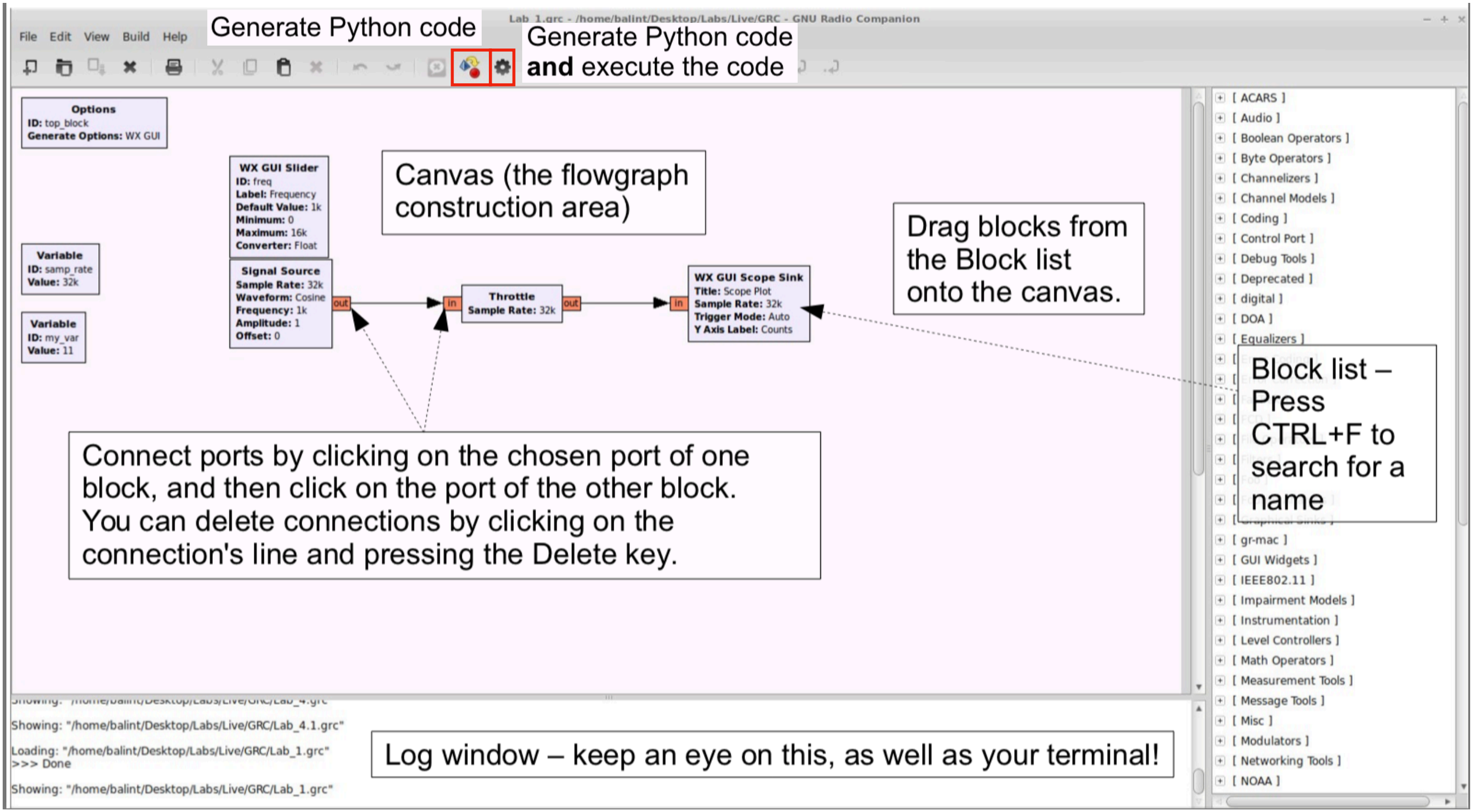


GNU Radio

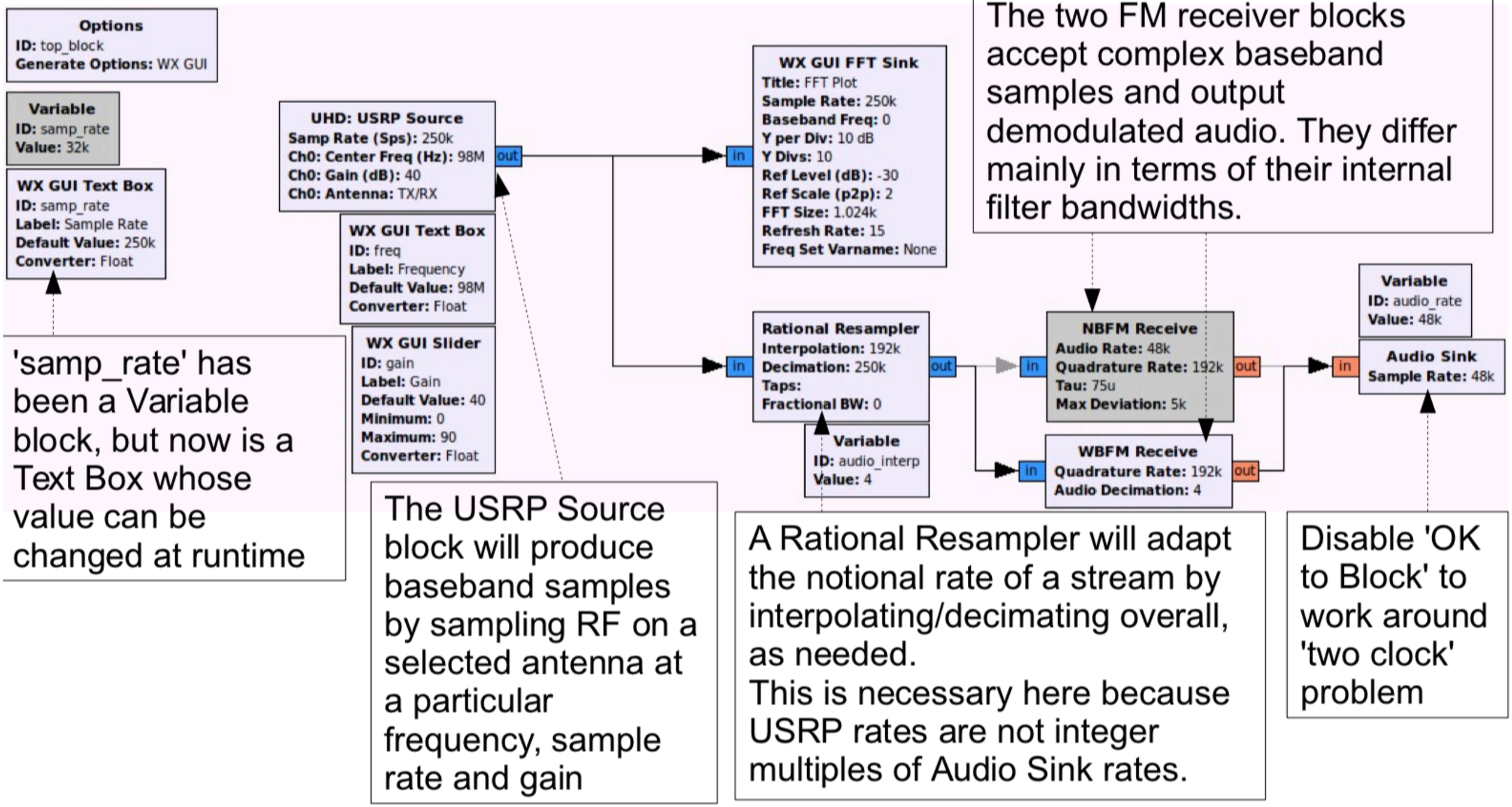
- An easy way to control USRP
- Open GNU Radio Companion:
  - Open a Terminal/Console/Command Prompt
  - Run ‘gnuradio-companion’

Reference: https://files.ettus.com/tutorials/labs/Lab_1-5.pdf

Overview


Generate Python code
Generate Python code and execute the code
File Edit View Build Help
Lab_1.grc - /home/balint/Desktop/Labs/Live/GRC - GNU Radio Companion
Options
ID: top_block
Generate Options: WX GUI
WX GUI Slider
ID: freq
Label: Frequency
Default Value: 1k
Minimum: 0
Maximum: 16k
Converter: Float
Canvas (the flowgraph construction area)
Variable
ID: samp_rate
Value: 32k
Variable
ID: my_var
Value: 11
Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0
out
in
Throttle
Sample Rate: 32k
out
in
WX GUI Scope Sink
Title: Scope Plot
Sample Rate: 32k
Trigger Mode: Auto
Y Axis Label: Counts
Drag blocks from the Block list onto the canvas.
[ACARS]
[Audio]
[Boolean Operators]
[Byte Operators]
[Channelizers]
[Channel Models]
[Coding]
[Control Port]
[Debug Tools]
[Deprecated]
[digital]
[DOA]
[Equalizers]
[gr-mac]
[GUI Widgets]
[IEEE802.11]
[Impairment Models]
[Instrumentation]
[Level Controllers]
[Math Operators]
[Measurement Tools]
[Message Tools]
[Misc]
[Modulators]
[Networking Tools]
[NOAA]
Block list – Press CTRL+F to search for a name
Connect ports by clicking on the chosen port of one block, and then click on the port of the other block. You can delete connections by clicking on the connection's line and pressing the Delete key.
Showing: "/home/balint/Desktop/Labs/Live/GRC/Lab_4.1.grc"
Loading: "/home/balint/Desktop/Labs/Live/GRC/Lab_1.grc"
>>> Done
Showing: "/home/balint/Desktop/Labs/Live/GRC/Lab_1.grc"
Log window – keep an eye on this, as well as your terminal!

UHD blocks


Options
ID: top_block
Generate Options: WX GUI
Variable
ID: samp_rate
Value: 32k
WX GUI Text Box
ID: samp_rate
Label: Sample Rate
Default Value: 250k
Converter: Float
'samp_rate' has been a Variable block, but now is a Text Box whose value can be changed at runtime
UHD: USRP Source
Samp Rate (Sps): 250k
Ch0: Center Freq (Hz): 98M
Ch0: Gain (dB): 40
Ch0: Antenna: TX/RX
out
WX GUI Text Box
ID: freq
Label: Frequency
Default Value: 98M
Converter: Float
WX GUI Slider
ID: gain
Label: Gain
Default Value: 40
Minimum: 0
Maximum: 90
Converter: Float
The USRP Source block will produce baseband samples by sampling RF on a selected antenna at a particular frequency, sample rate and gain
WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 250k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): -30
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None
in
The two FM receiver blocks accept complex baseband samples and output demodulated audio. They differ mainly in terms of their internal filter bandwidths.
Rational Resampler
Interpolation: 192k
Decimation: 250k
Taps:
Fractional BW: 0
in
out
Variable
ID: audio_interp
Value: 4
A Rational Resampler will adapt the notional rate of a stream by interpolating/decimating overall, as needed.
This is necessary here because USRP rates are not integer multiples of Audio Sink rates.
NBFM Receive
Audio Rate: 48k
Quadrature Rate: 192k
Tau: 75u
Max Deviation: 5k
WBFM Receive
Quadrature Rate: 192k
Audio Decimation: 4
Variable
ID: audio_rate
Value: 48k
Audio Sink
Sample Rate: 48k
Disable 'OK to Block' to work around 'two clock' problem

Tip: usually all parameters can be left as they are (except for sample rate, frequency, gain and antenna).

Mapping from physical (USRP) channel index to logical (GRC port) channel index (zero-based). Leave as the empty list '[]' for the default linear mapping.

Sets the number of output ports and duplicates the channel-specific parameters accordingly.

Tip: To be certain about any of the possble parameter values, consult the online documentation for your device and/or daughterboard. You can also run 'uhd_usrp_probe' in a terminal for hardware specs. Watch your console during runtime for any warning messages from UHD regarding invalid settings!

WX GUI Text Box
ID: samp_rate
Label: Sample Rate
Default Value: 250k
Converter: Float

Ch0: Center Freq (Hz): 98M
Ch0: Gain (dB): 40
Ch0: Antenna: TX/RX

WX GUI Text Box
ID: freq
Label: Frequency
Default Value: 98M
Converter: Float

Maximum: 90

Properties: UHD: USRP Source

General | Advanced | Documentation

ID	uhd_usrp_source_0	
Output Type	Complex float32	Sample type on output port
Wire Format	Automatic	Sample type from USRP
Stream args		RX streamer options
Stream channels	[]	
Device Addr		Same as UHD device args
Sync	don't sync	
Clock Rate (Hz)	Default	
Num Mboards	1	These will be covered later
Mb0: Clock Source	Default	
Mb0: Time Source	Default	
Mb0: Subdev Spec		Selects a 'side', e.g. A:A or A:B
Num Channels	1	
Samp Rate (Sps)	samp_rate	Valid range depends on hardware
Ch0: Center Freq (Hz)	freq	Valid range depends on hardware
Ch0: Gain (dB)	gain	Valid range depends on hardware
Ch0: Antenna	'TX/RX'	Usually 'TX/RX' or 'RX2'
Ch0: Bandwidth (Hz)	0	Usually 0

VLC CamCom Implementation

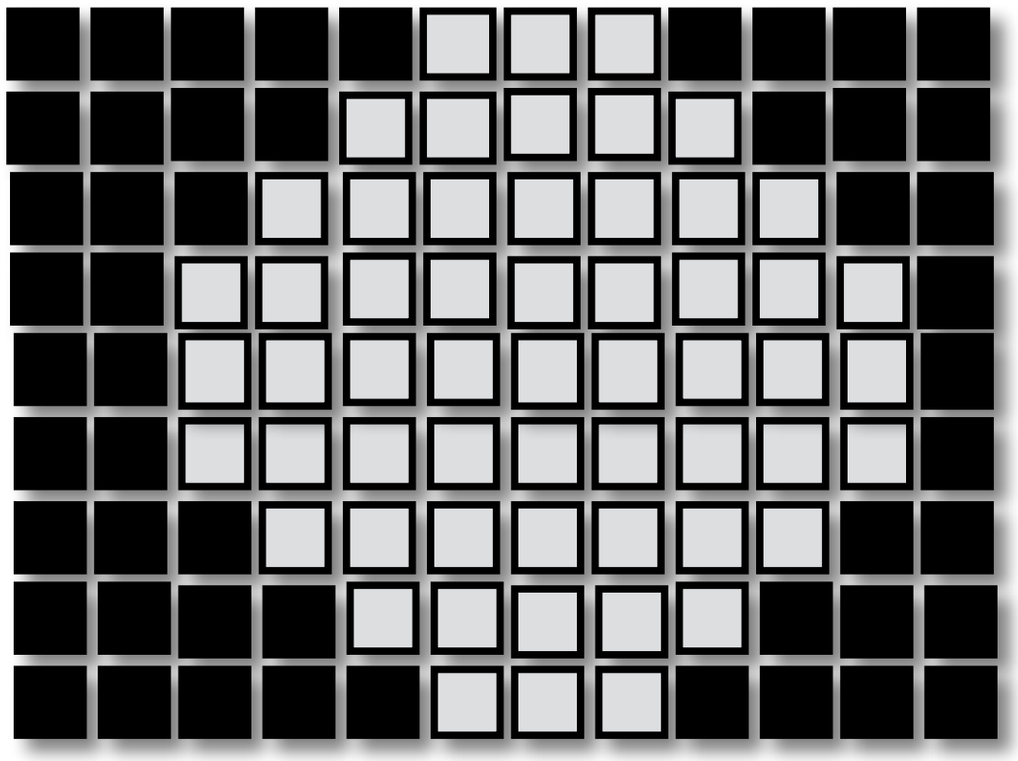
沈雯萱&吳浩平&黃禹程

What you will learn

- Implement a simple Camcom system
- Get familiar with equipments related to VLC
  - USRP & GNURadio
  - VLC front-end board
- Basic modulation schemes: **frequency shift keying**

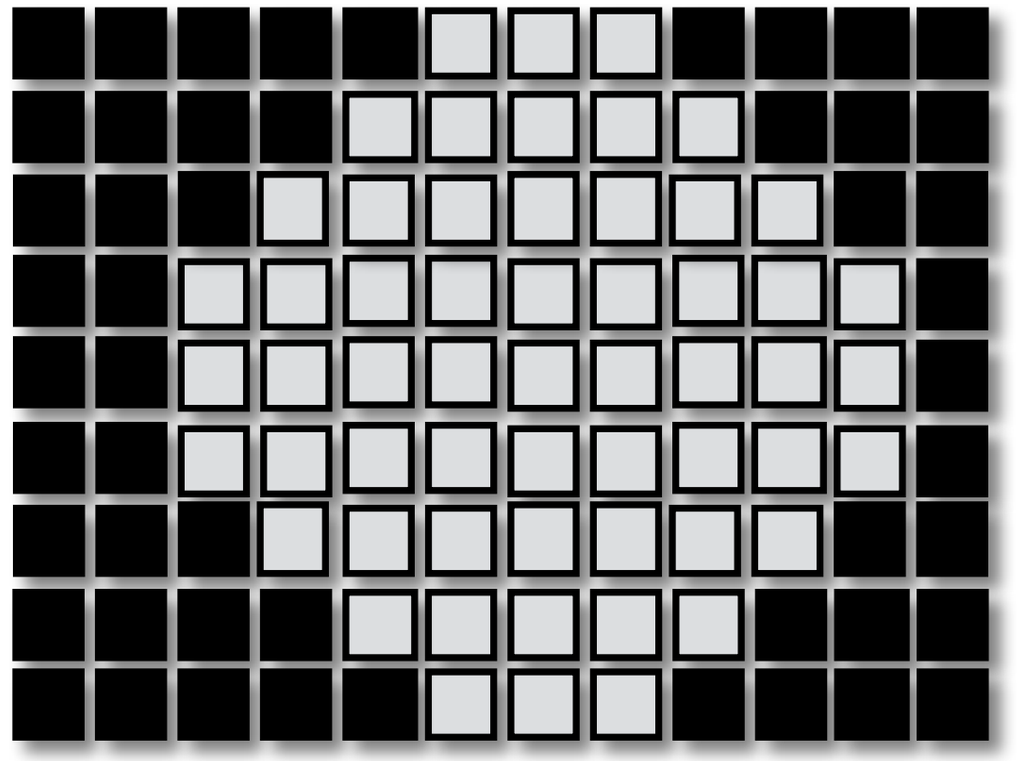
Camera -
Two kinds of shutter

Global shutter



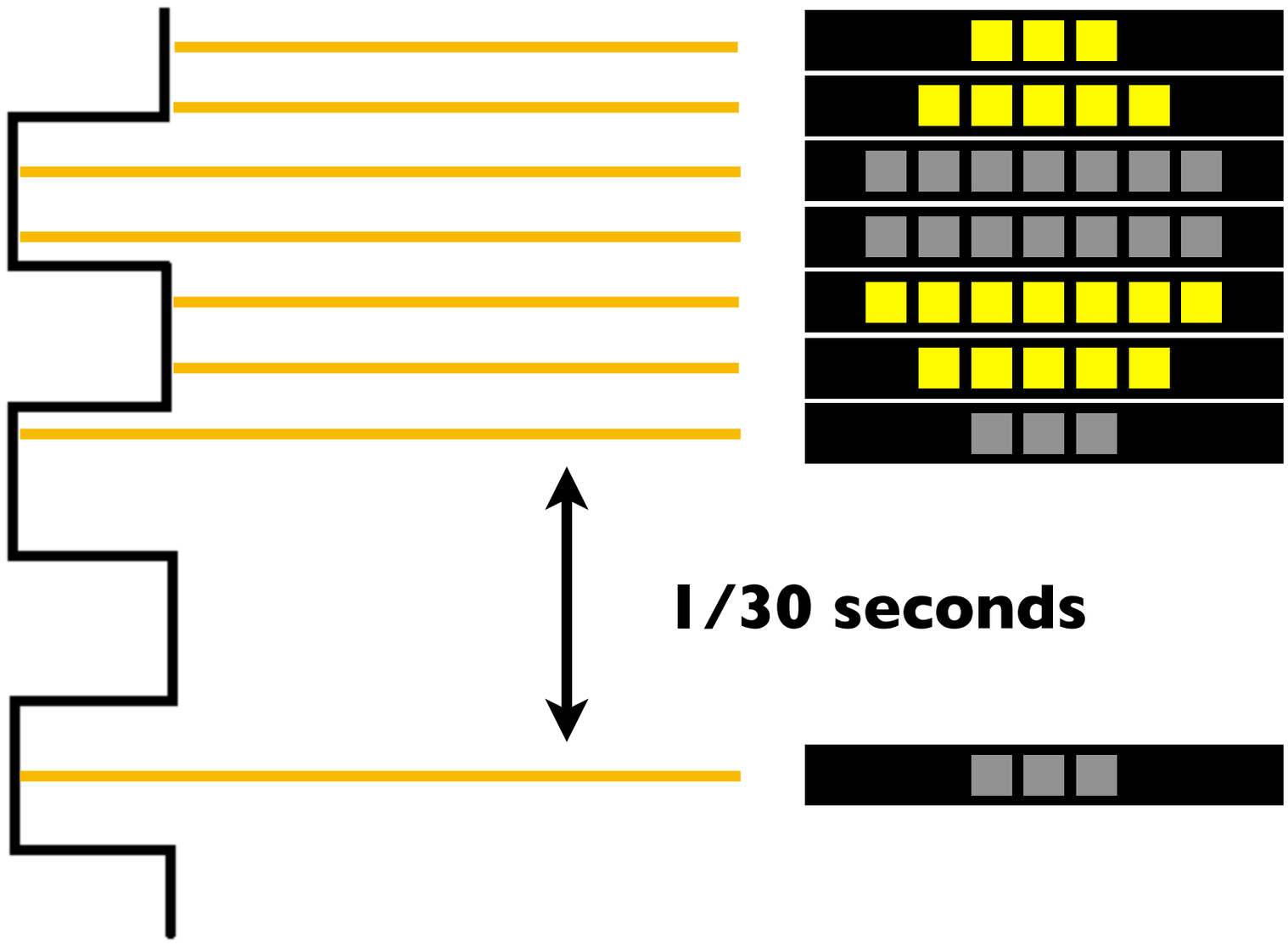
Pixels exposed SIMULTANEOUSLY

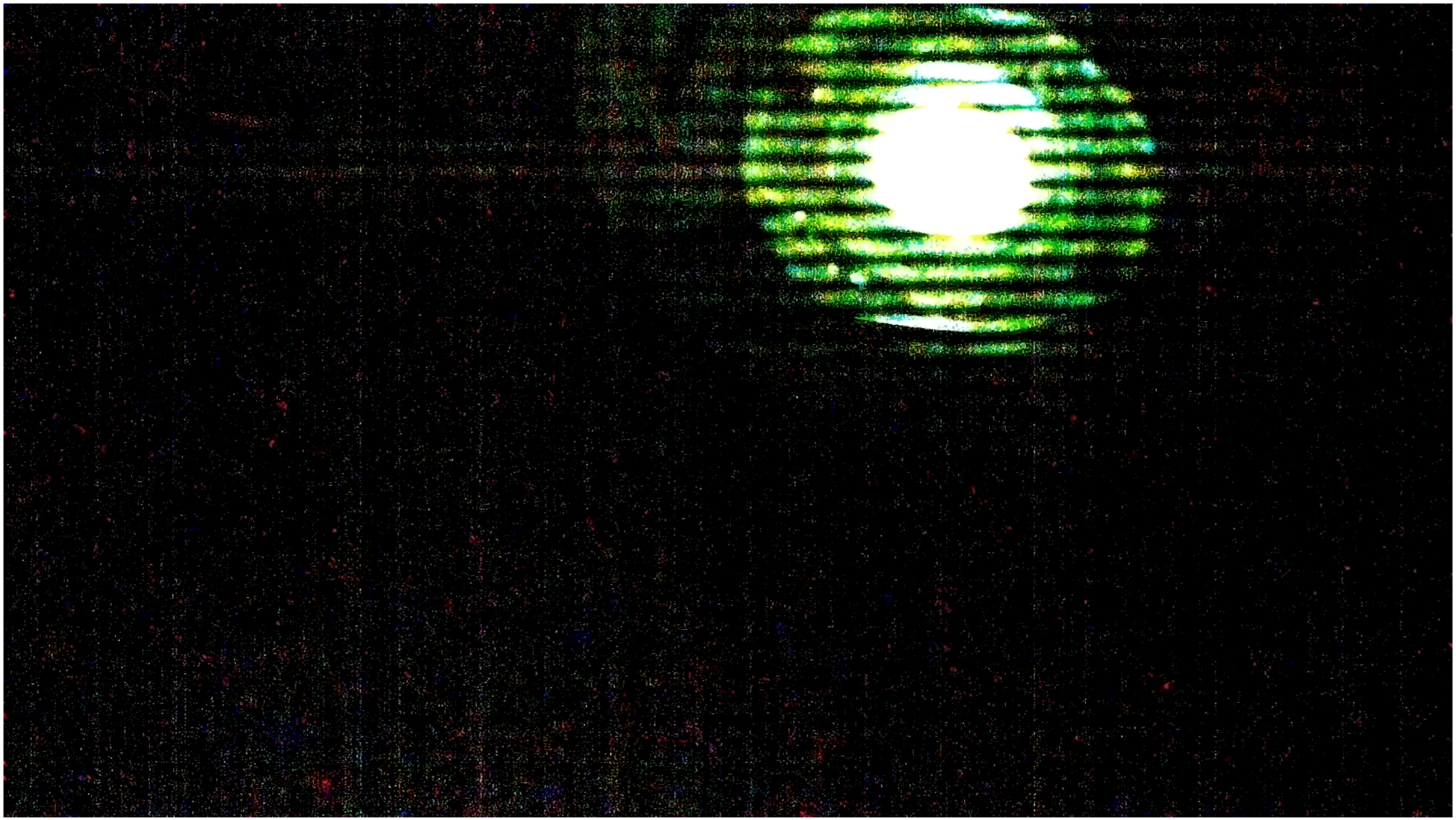
Rolling shutter



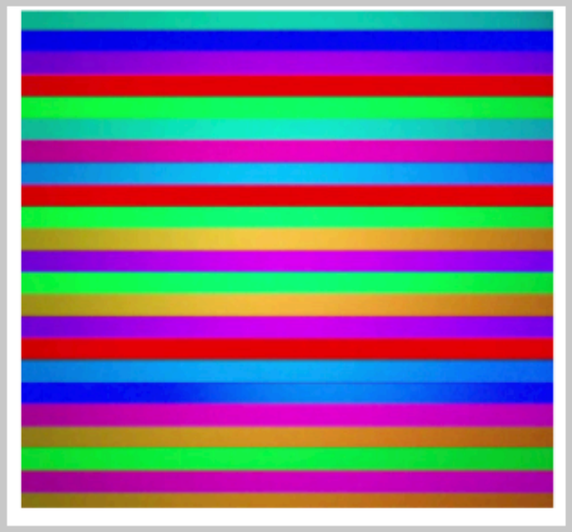
Pixels exposed Row by Row

How rolling shutter works?





Modulations

On Off Keying	Frequency Shift Keying	Color Shift Keying
		
Only **On** or **Off**	A set of **frequencies**	A set of **colors**

Goal - TX

Matlab

- Generate signal for LED **Create a .bin file**

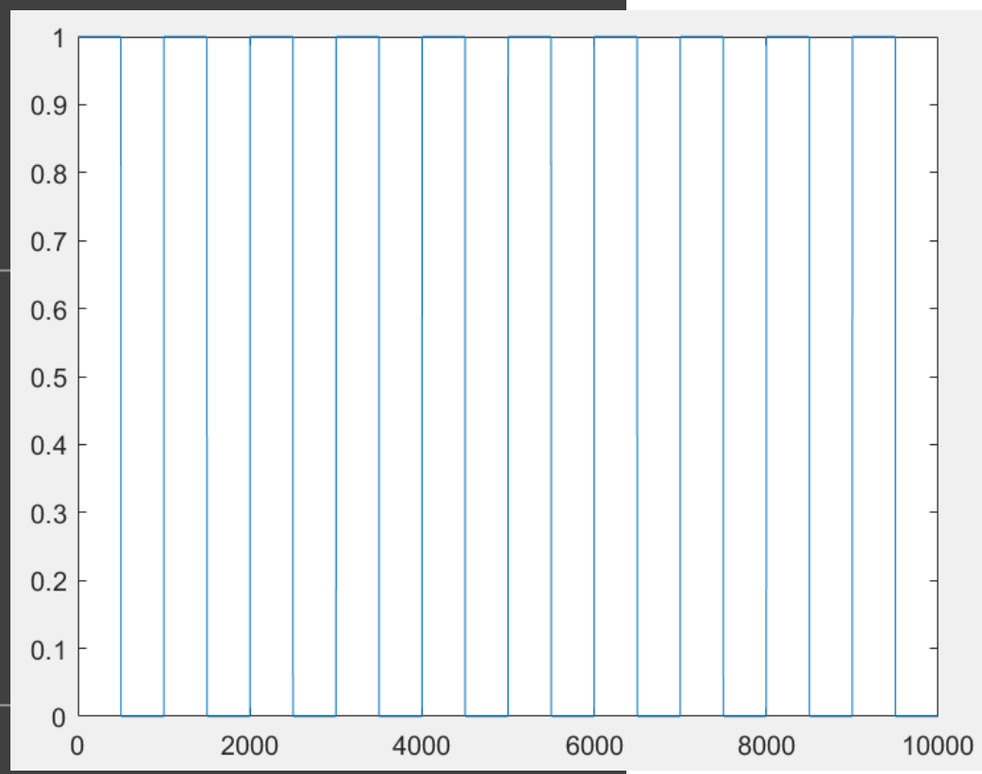
USRP

- Use GNU Radio to control USRP

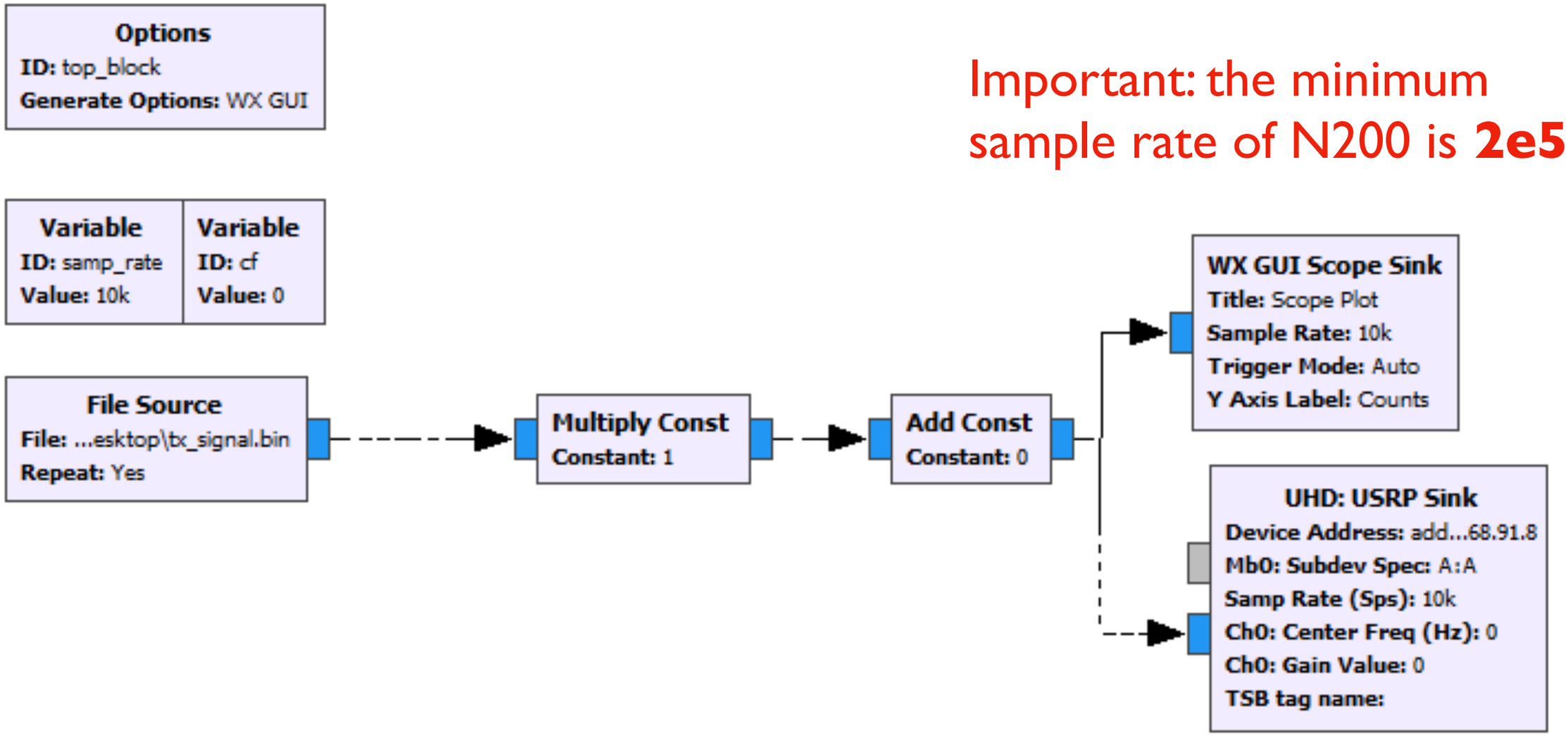
LED

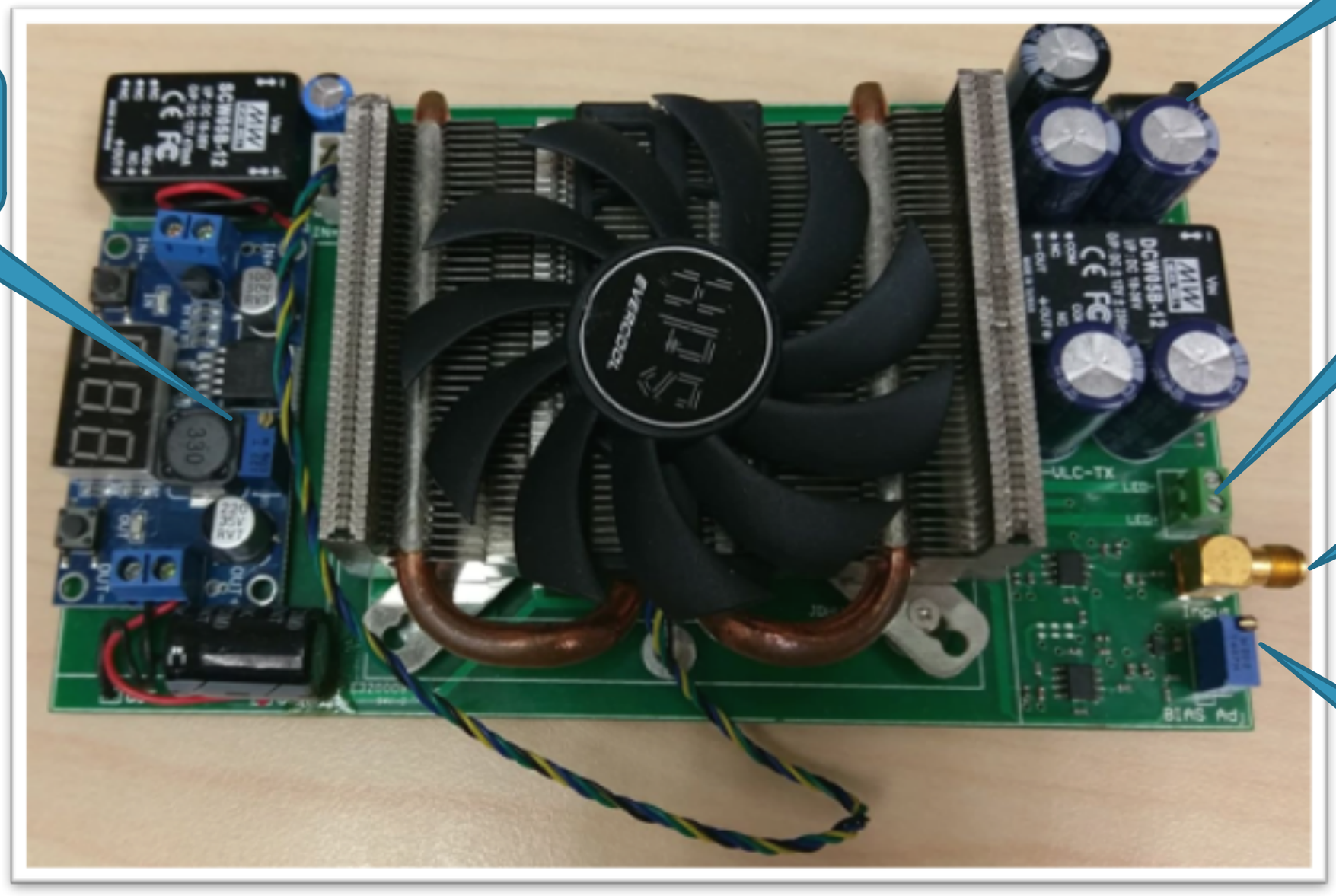
- Use front-end board to convert voltage signal to current signal to LED

```
%% Parameters
N_sam = 1e4; % number of samples per second
N_sym = 1e1; % number of symbols per second == symbol frequency
T_sam = 1/N_sam; % sample duration(unit: microsecond)
T_sym = 1/N_sym; % symbol duration(unit: microsecond)
samPerSym = N_sam/N_sym; % samples per symbol
%% Create a symbol
ON = ones(1,samPerSym/2);
OFF = zeros(1,samPerSym/2);
sym = [ON OFF];
%% Create signal
data = [];
for i=1:N_sym
    data = [data sym];
end
plot(data);
%% Write signal as binary file
fid = fopen(strcat('tx_signal.bin'), 'w');
fwrite(fid, [real(data); imag(data)], 'float');
fclose(fid);
```

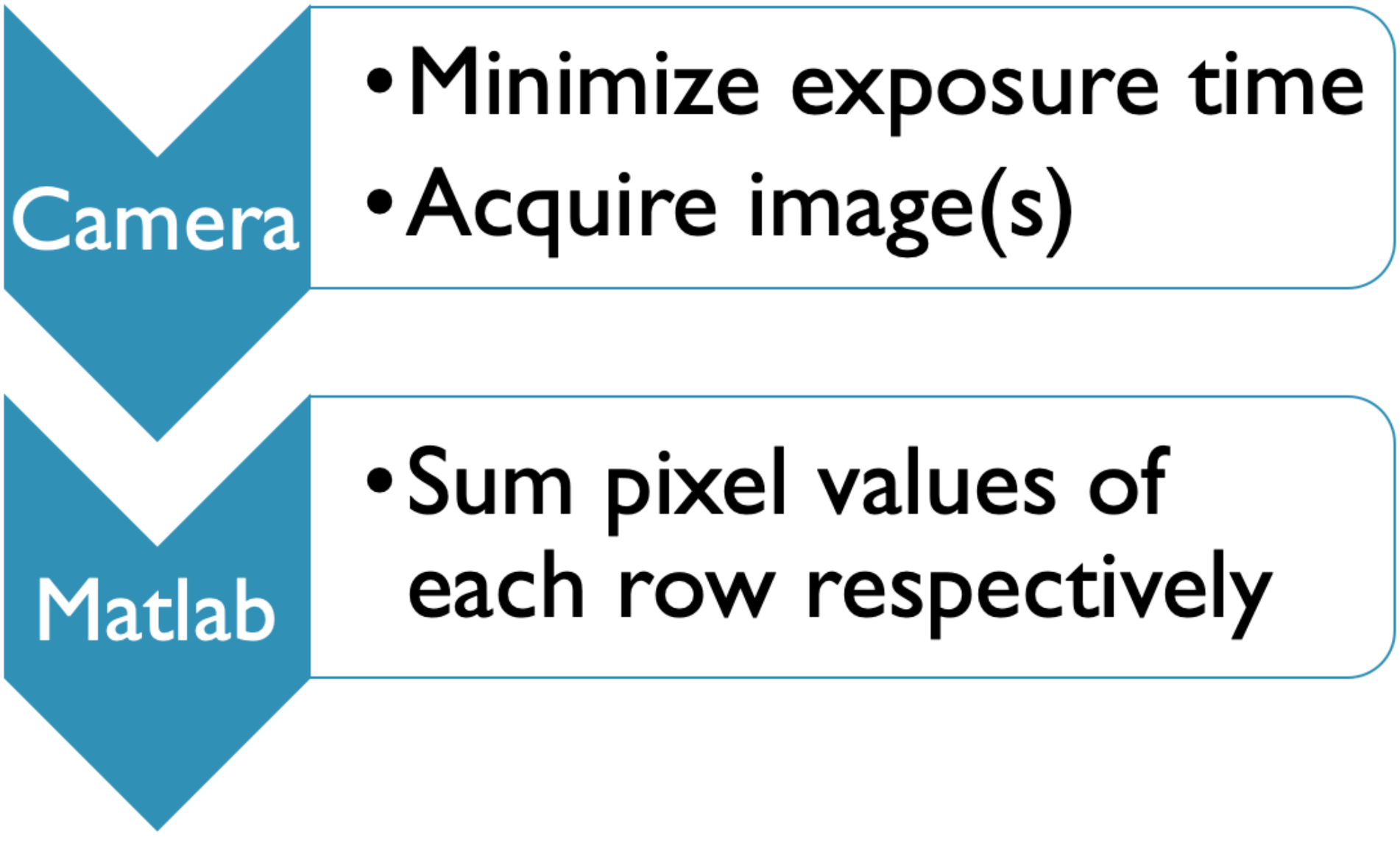


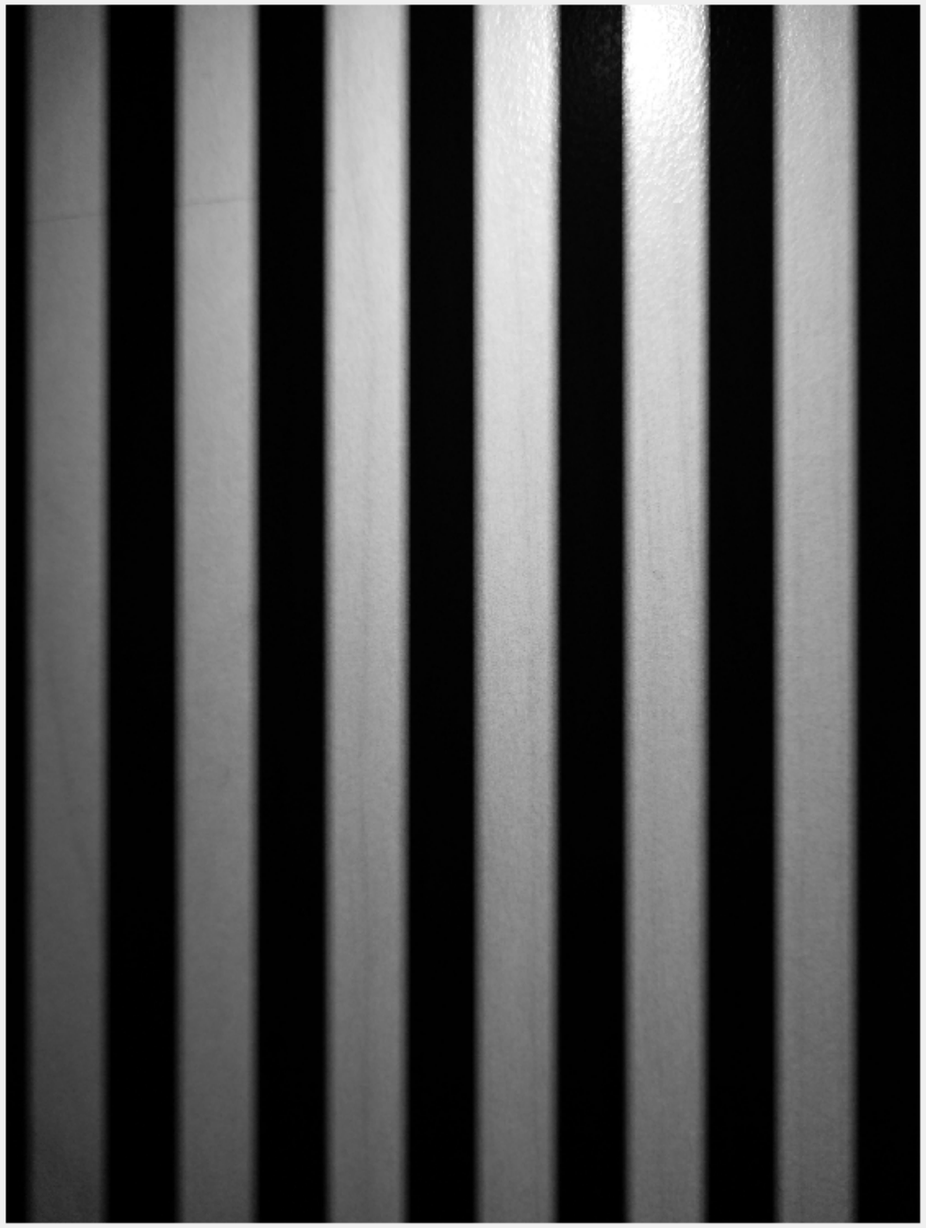
Important: the minimum sample rate of N200 is **2e5**

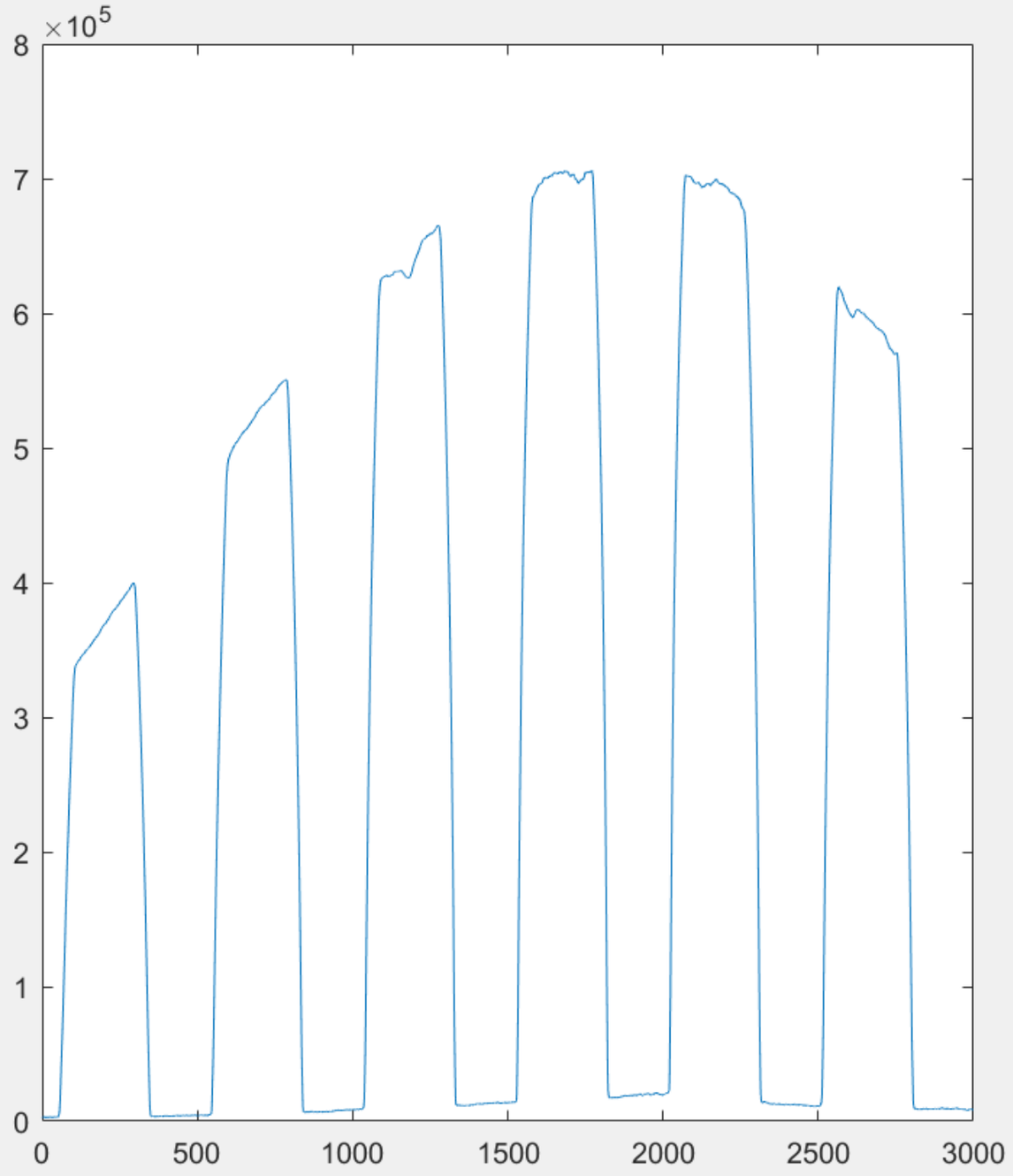



Step1: Plug 30V/2A adaptor
Step2: Set $V_{working}$ for LED
6~8V
Step4: Connect LED
Step5: Connect to USRP
Step3: Set current bias for linearity

Goal - RX


Camera
• Minimize exposure time
• Acquire image(s)
Matlab
• Sum pixel values of each row respectively




×10^5
8
7
6
5
4
3
2
1
0
0
500
1000
1500
2000
2500
3000

Todo

- Goal
  - Observe frequency = 10, 1000, 2000 square waves
  - Understand the relationship between strip width and frequency
- Instruction
  - Tx.m and Rx.m will be provided
  - Generate frequency = 10, 1000, 2000 square waves respectively with Tx.m
  - Setup USRP(Tx.grc) & Front-end board to control LED
  - Acquire images with your phone with low exposure time
  - Sum pixel values of each rows with Rx.m