BLACK PERL

『RF, new hacking tools』

무선통신?

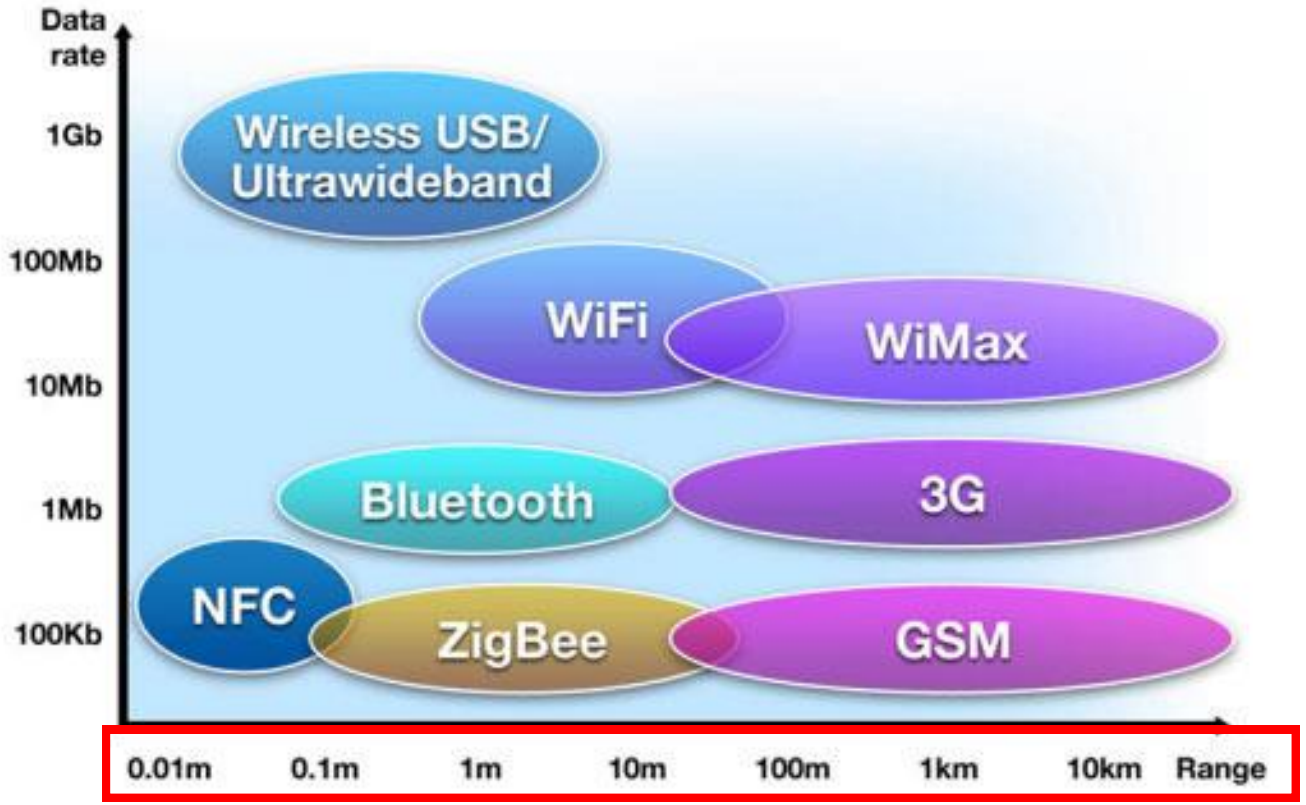
On-Line

무선통신?

Propagation Wave

무선통신?

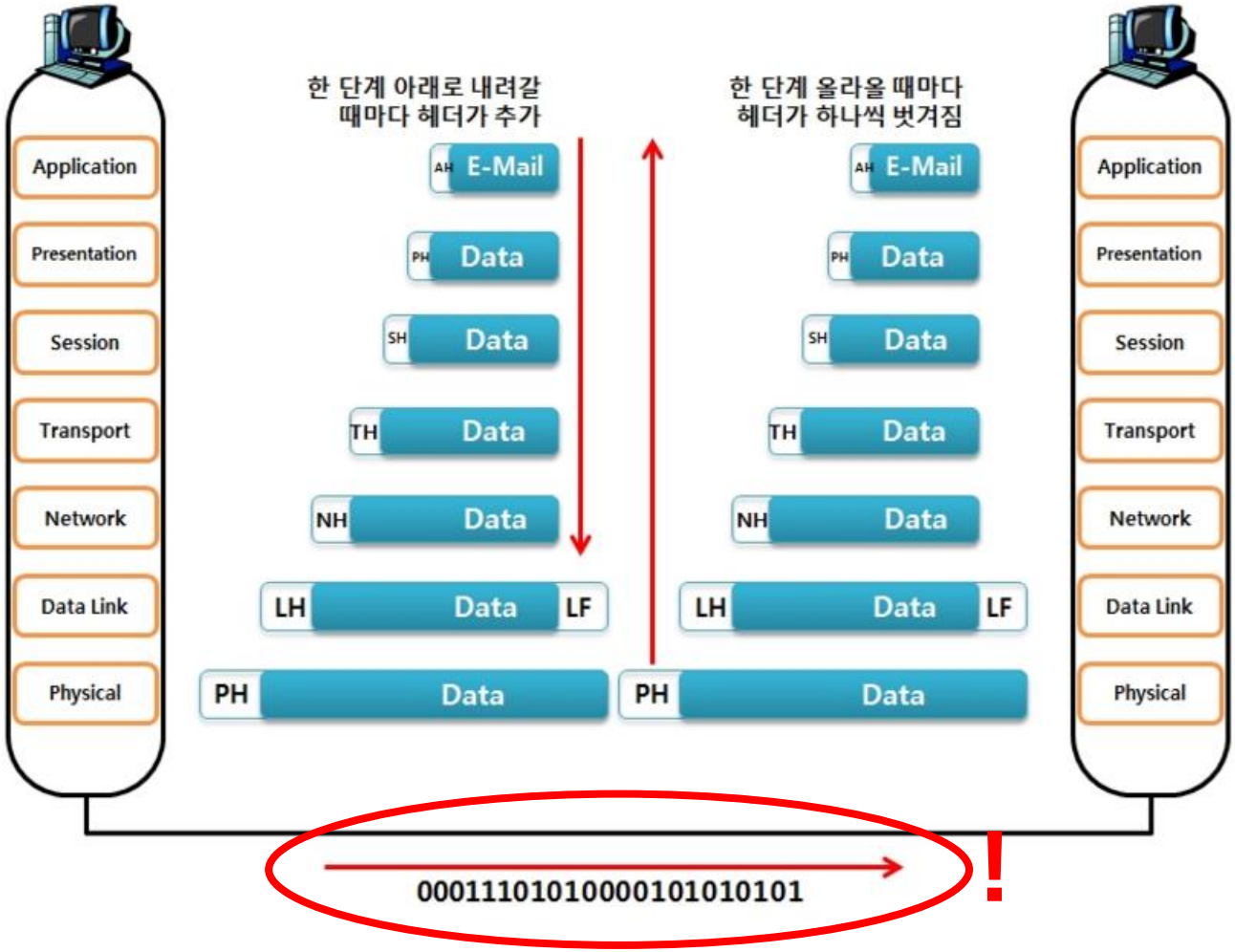
Not Distanse, Range



무선 통신별 연결범위 및 전송속도 비교

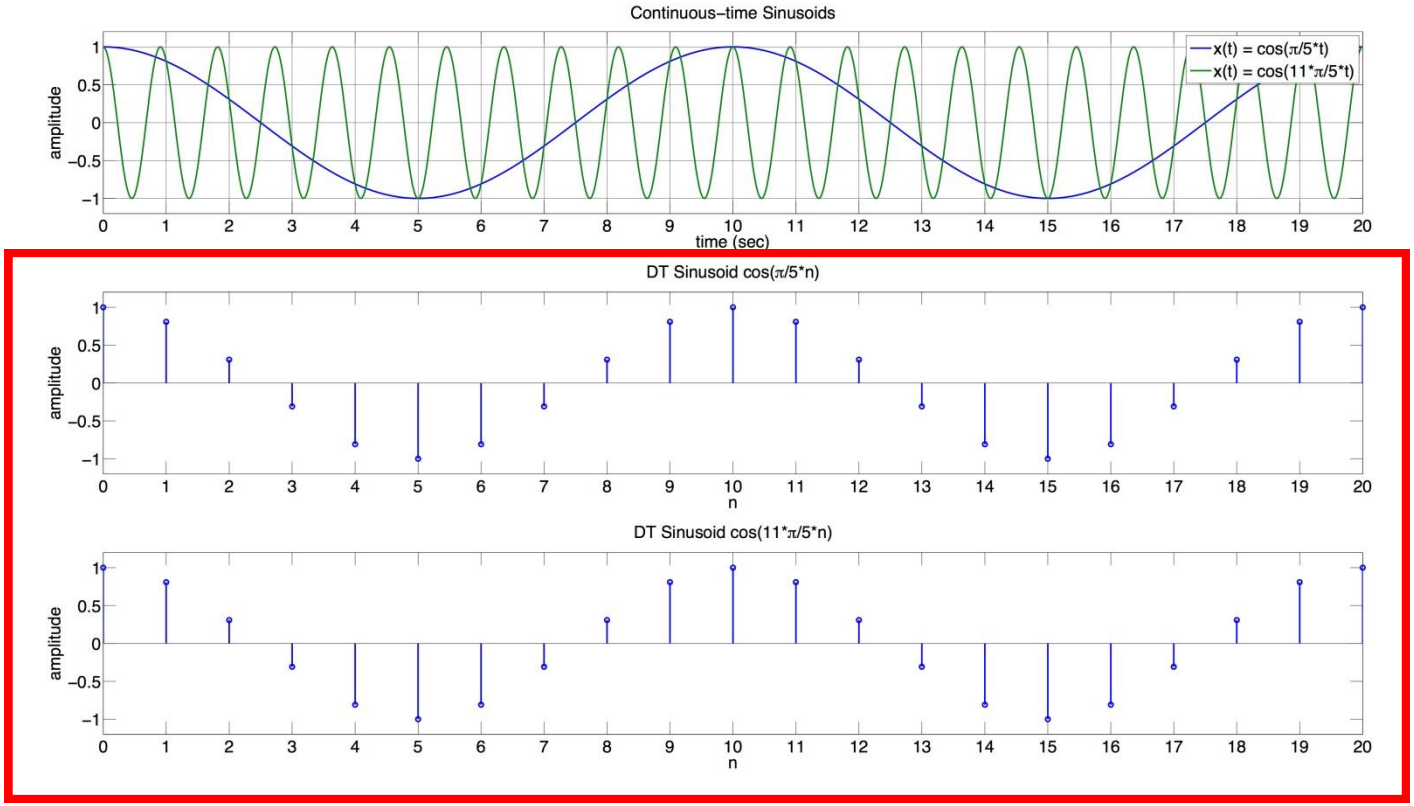
무선통신?

신호의 유(1), 무(0)



무선통신?

Continuous time : Discrete time = Analog : Digital



01000101011· · ·

무선통신?

장점?

SDR

SDR(Software Defined Radio)

SDR

Low Cose:

RTL2832U – Realtek SDR dongle	24-1766MHz

Medium Cose:

FunCube DonglePro +	150kHz-2.05GHz
HackRF One	10MHz-6GHz
BlacdRF	300MHz-3.8GHz

High Cost:

USRP1	
USRP B200	70MHz-6GHz
USRP B210	70MHz-6GHz
UmTRX	300MHz-3.8GHz
Matchstiq	300MHz-3.8GHz

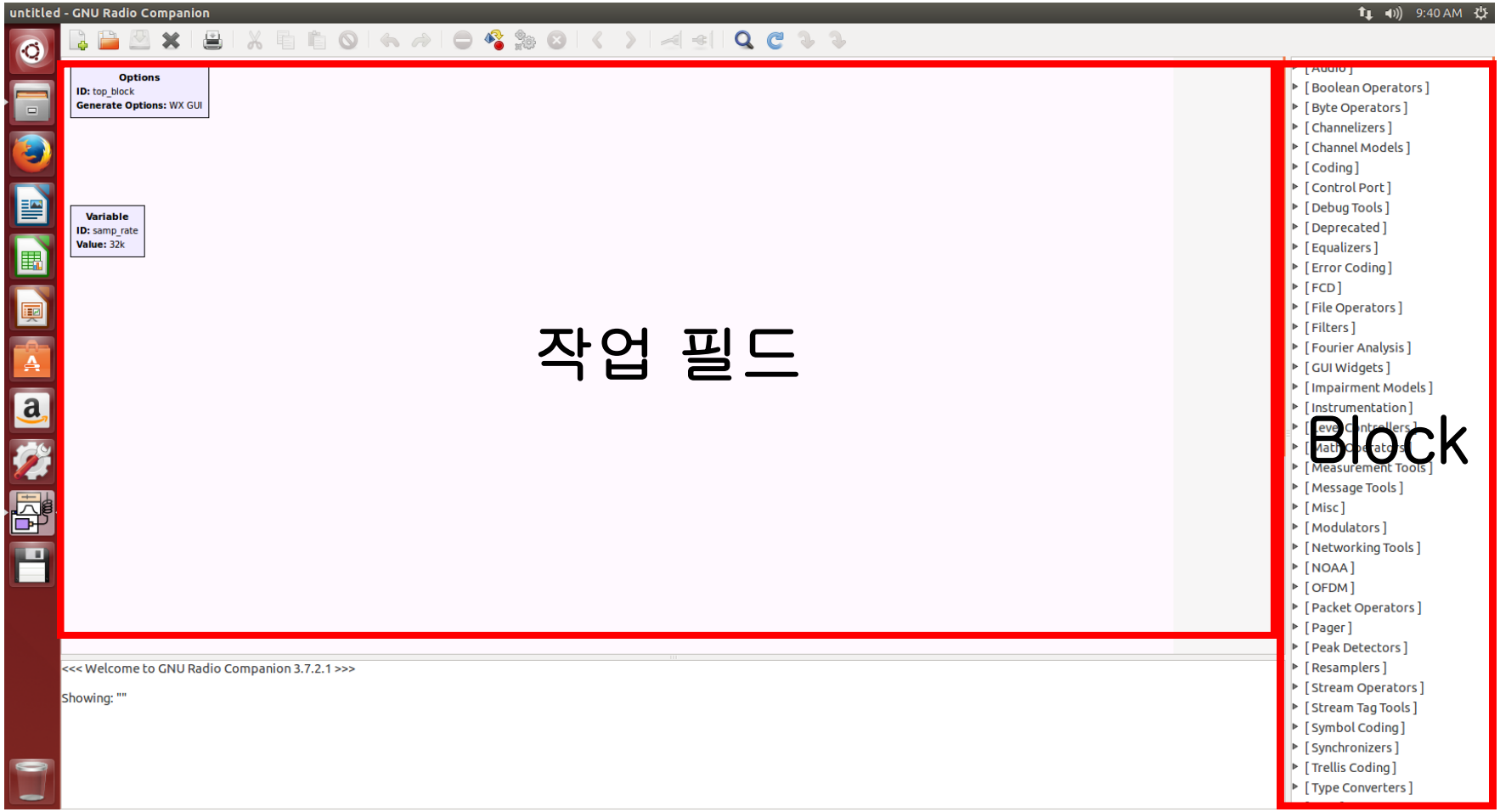
GNURadio

시각적 블록
다이어그램 설계

설계를 바탕으로
Python 코드 생성

SDR 장비에
코드 삽입

FM Radio receiver

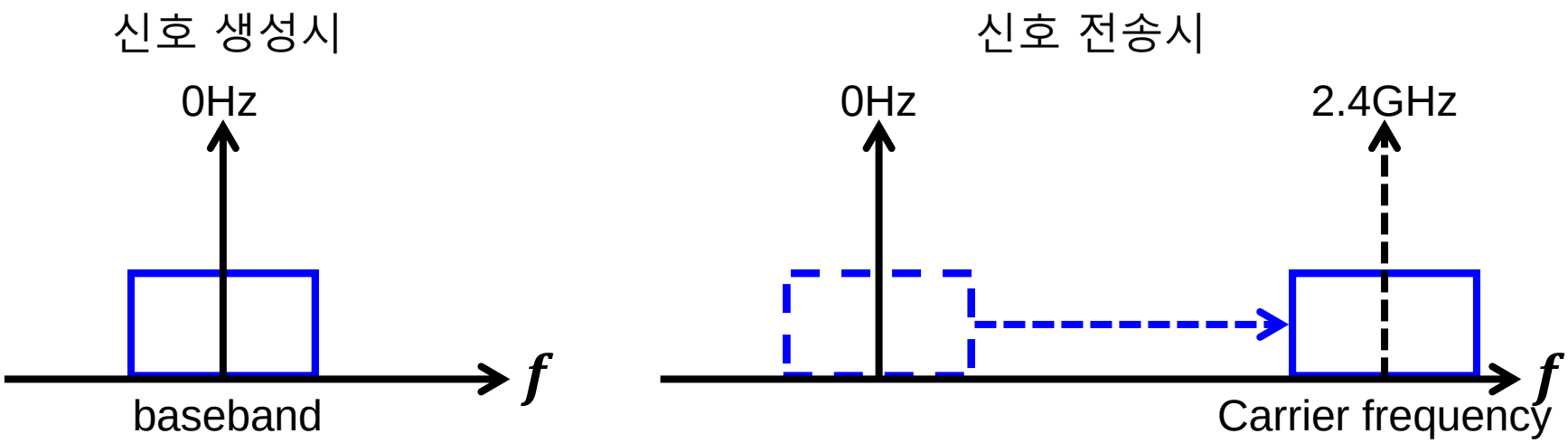


FM Radio receiver

Carrier Frequency

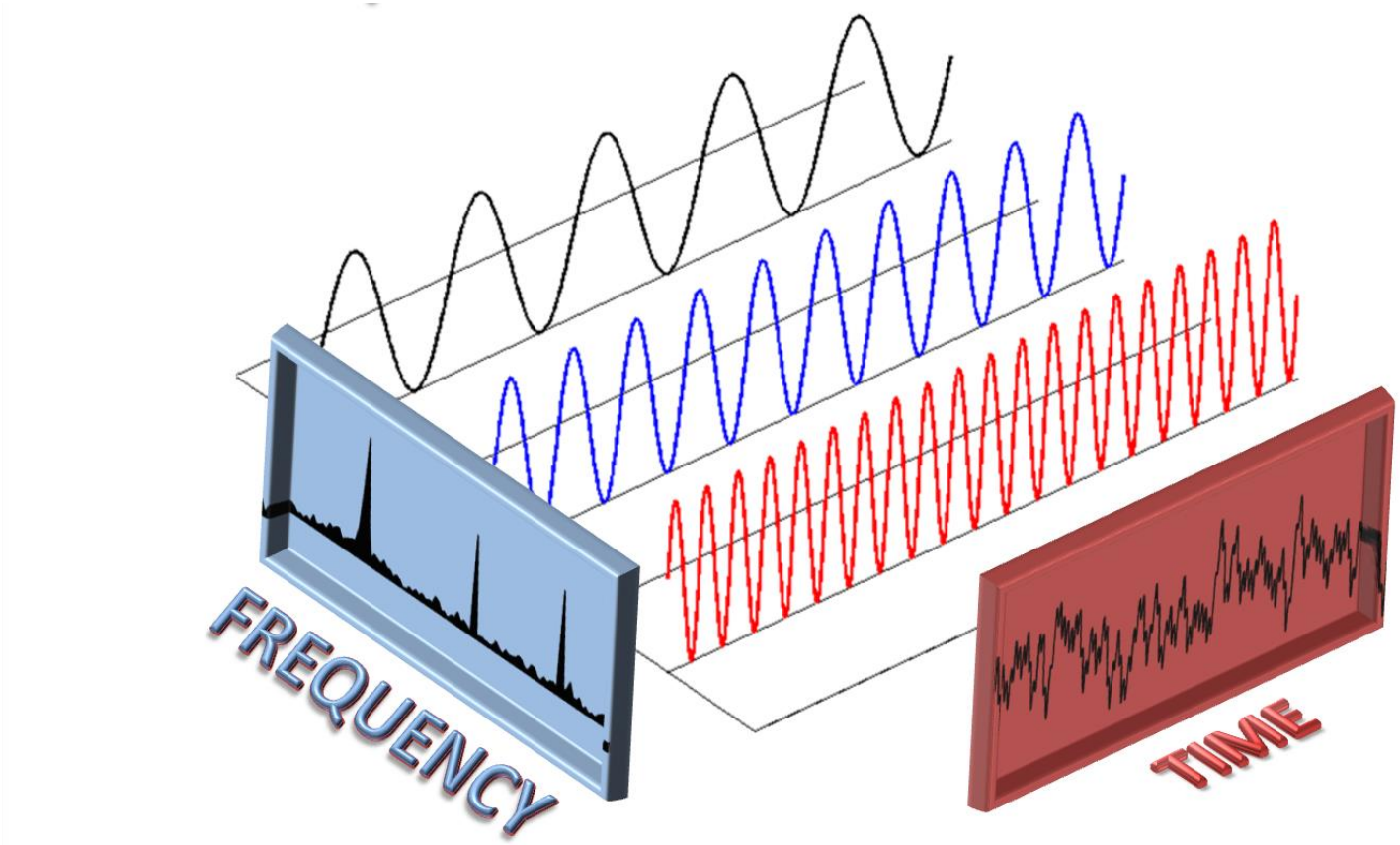
Time domain product = Frequency domain shift

433MHz? 2.4GHz?



FM Radio receiver

FFT(Fast Fourier Transform)



FM Radio receiver

Channel Detection

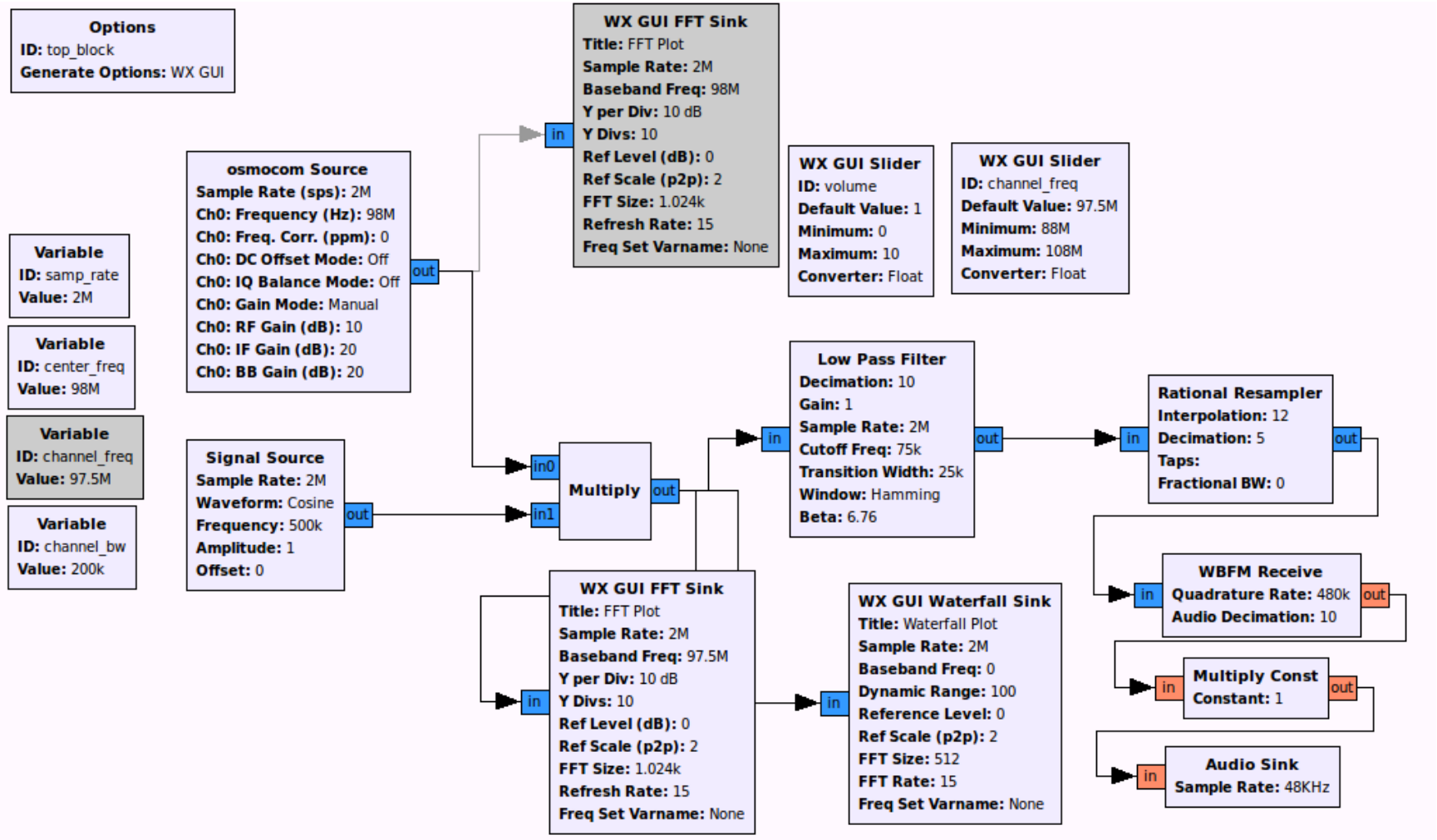
$$\int \{A(cos(w_1 t + \theta_1) + \cos(w_2 t + \theta_2)\} \times cos(w_1 t + \theta_3) dt$$

$$= \int Acos(w_1 t + \theta_1) \times cos(w_1 t + \theta_3) dt$$

$$+ \int \mathrm{A}\cos(w_2 t + \theta_2) \times cos(w_1 t + \theta_3) dt$$

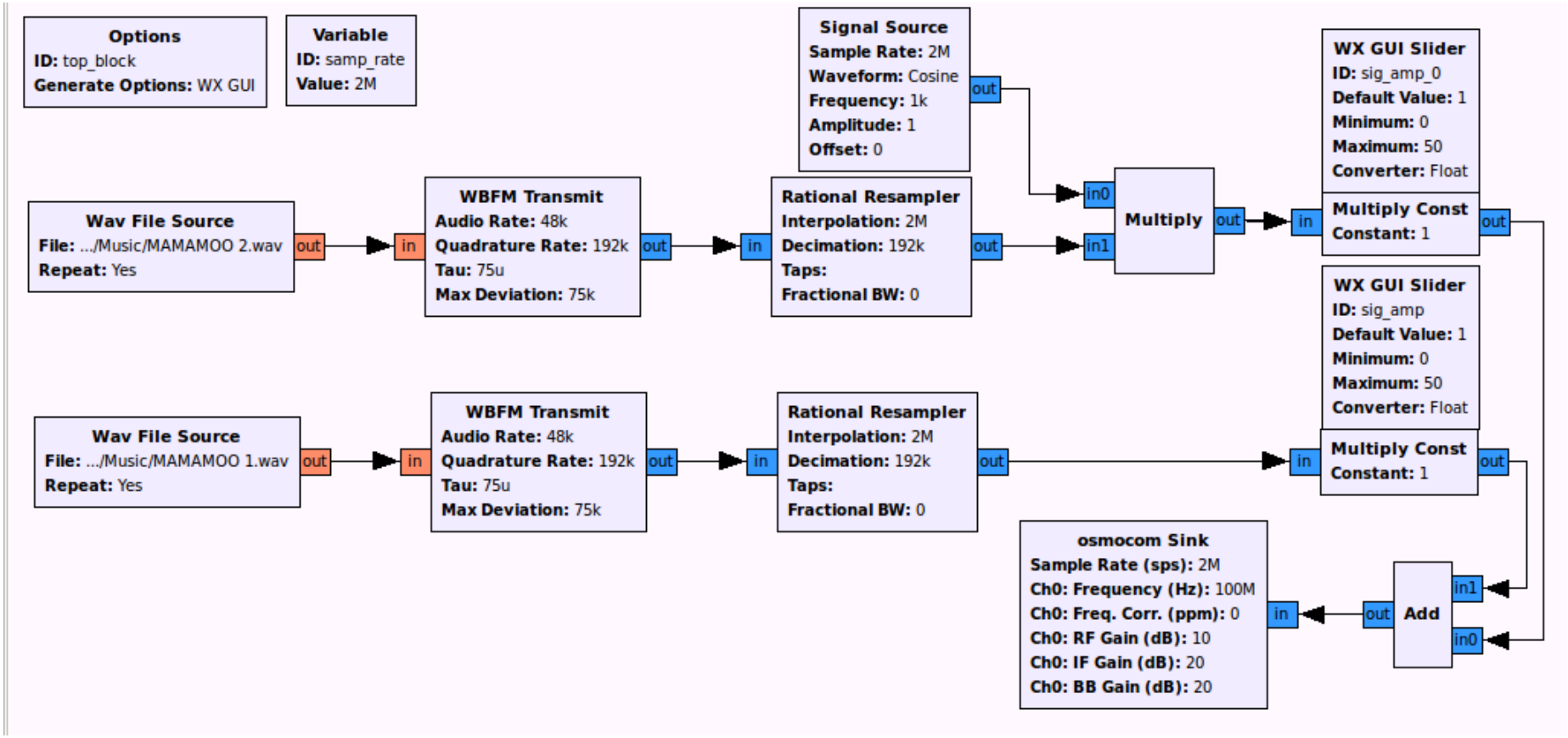
$$= A'$$

FM Radio receiver



FM Radio Transmitter

2 Channel Multicasting

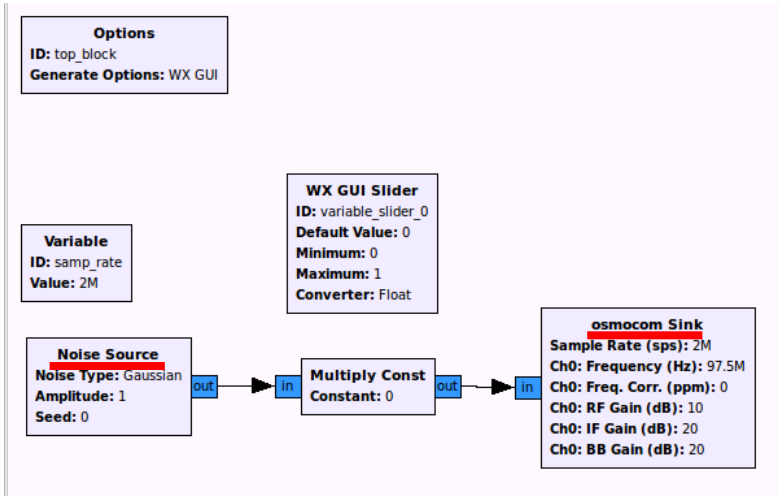


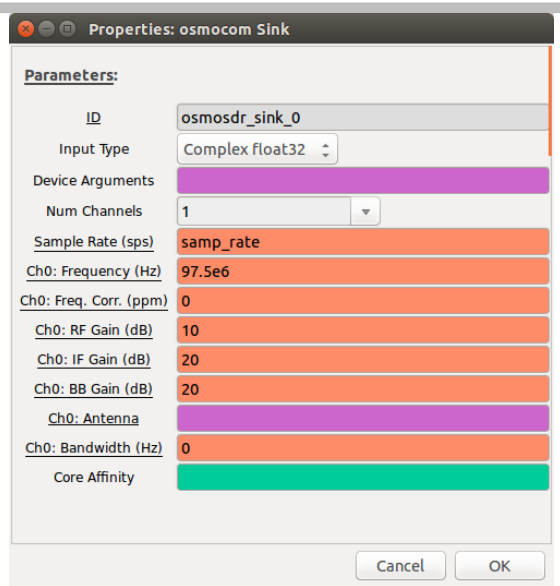
FM Radio Jammer

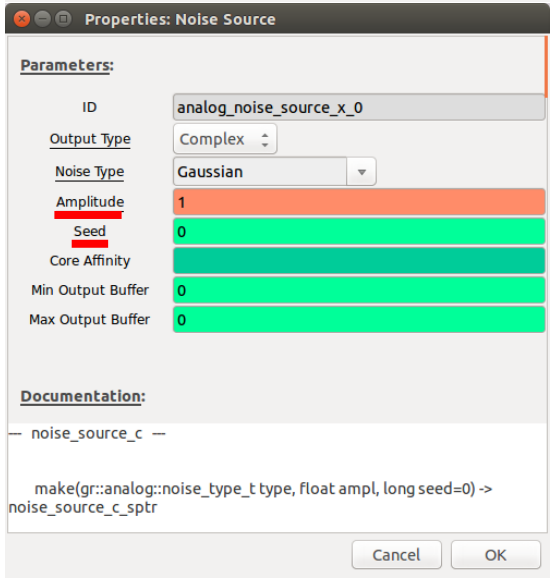
HackRF One

10 MHz to 6 GHz operating frequency
half-duplex transceiver
up to 20 million samples per second
8-bit quadrature samples (8-bit I and 8-bit Q)
compatible with GNU Radio, SDR#, and more
software-configurable RX and TX gain and baseband filter
software-controlled antenna port power (50 mA at 3.3 V)
SMA female antenna connector
SMA female clock input and output for synchronization
convenient buttons for programming
internal pin headers for expansion
Hi-Speed USB 2.0
USB-powered
open source hardware

FM Radio Jammer





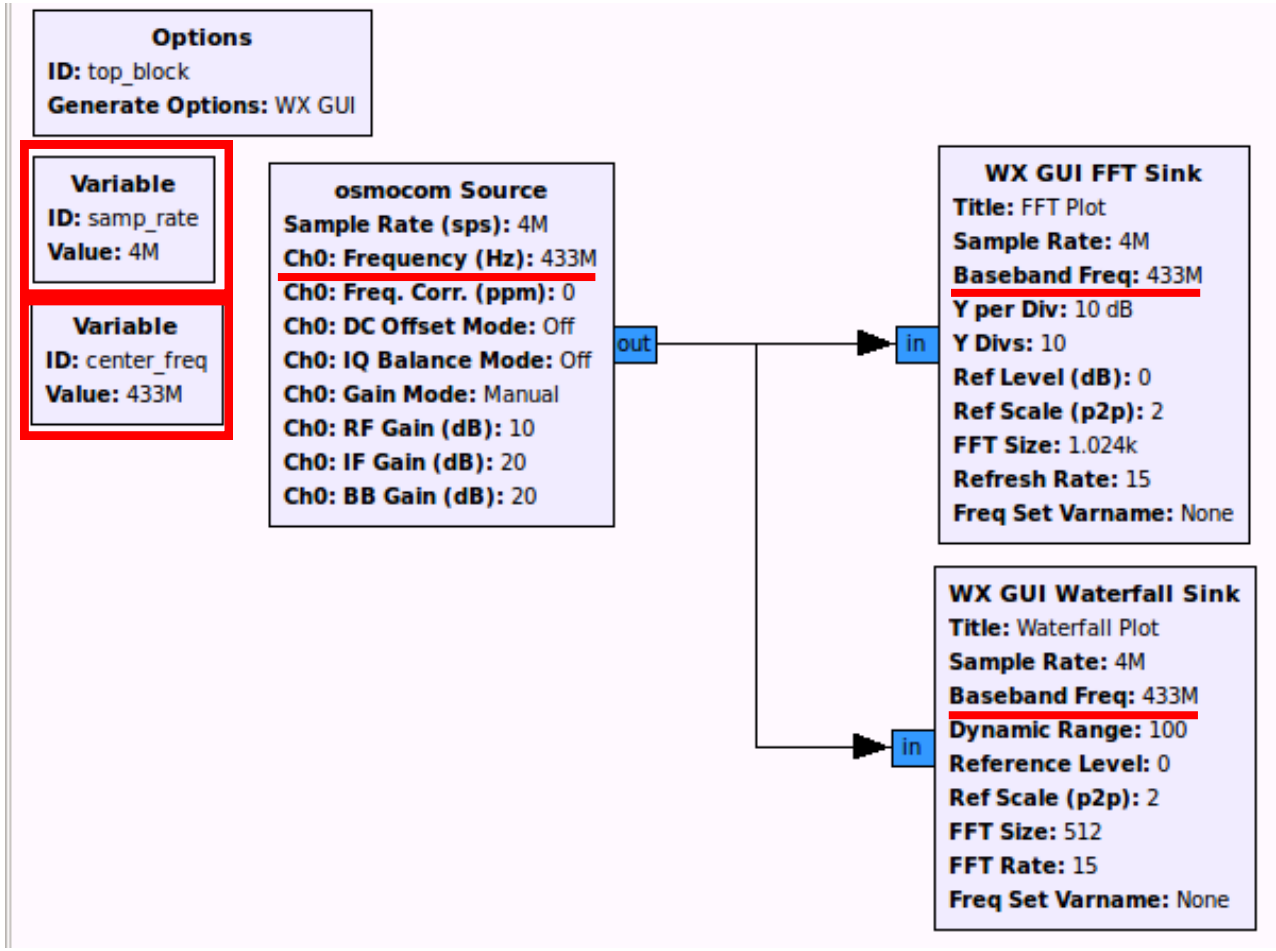


RKE replay

RKE(Remote Keyless Entry)

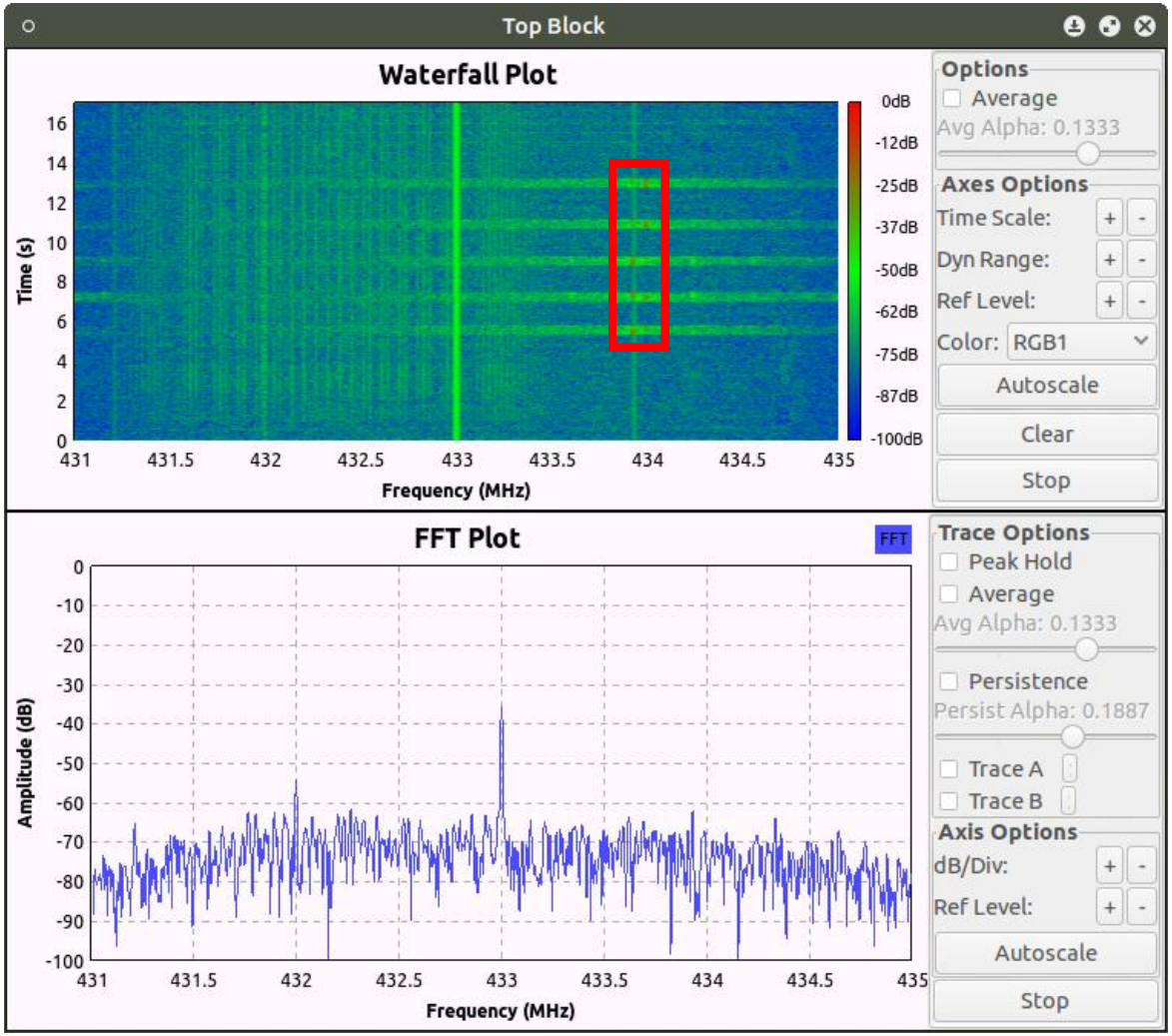
RKE replay

Signal Detection



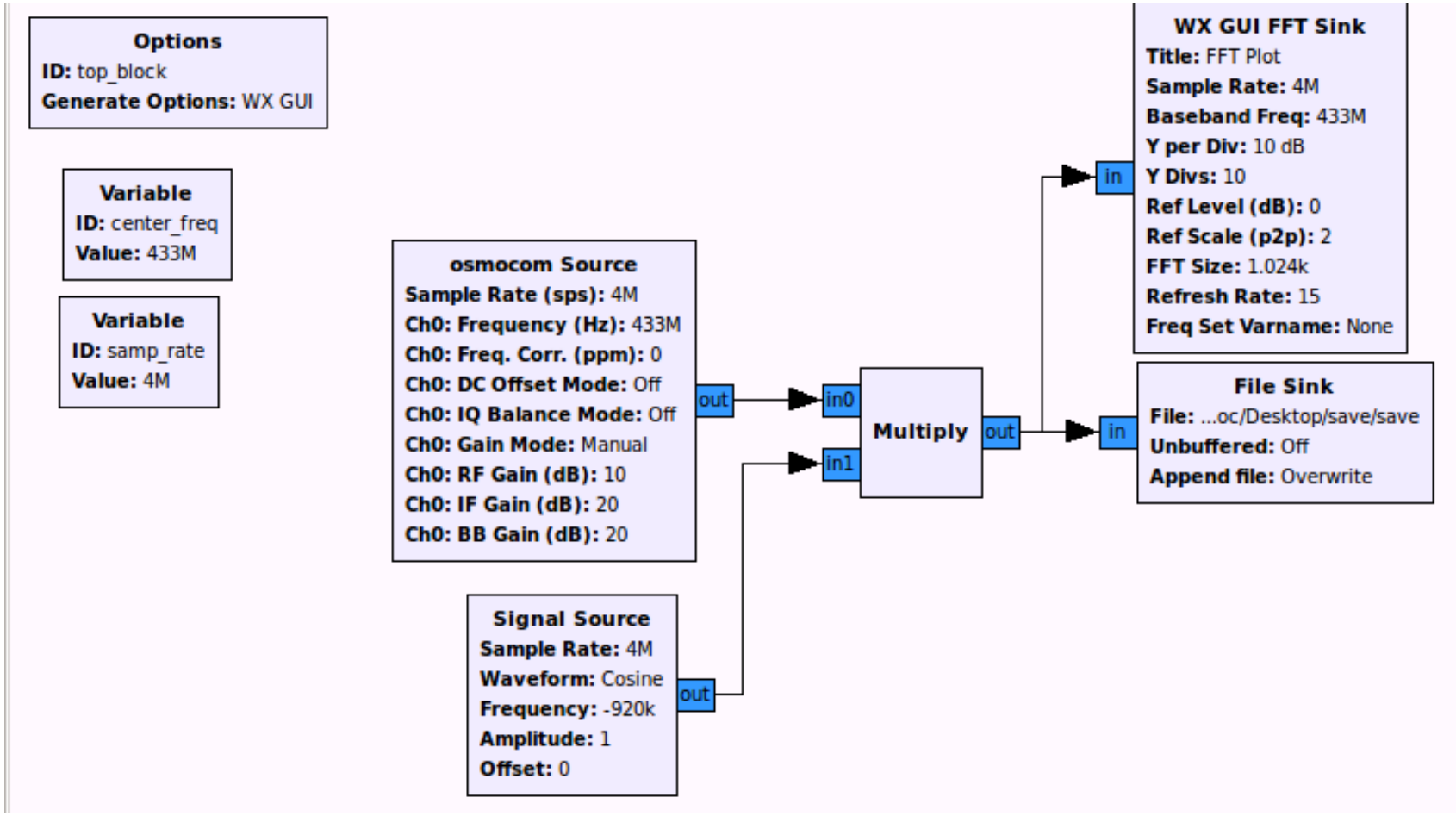
RKE replay

Signal Detection



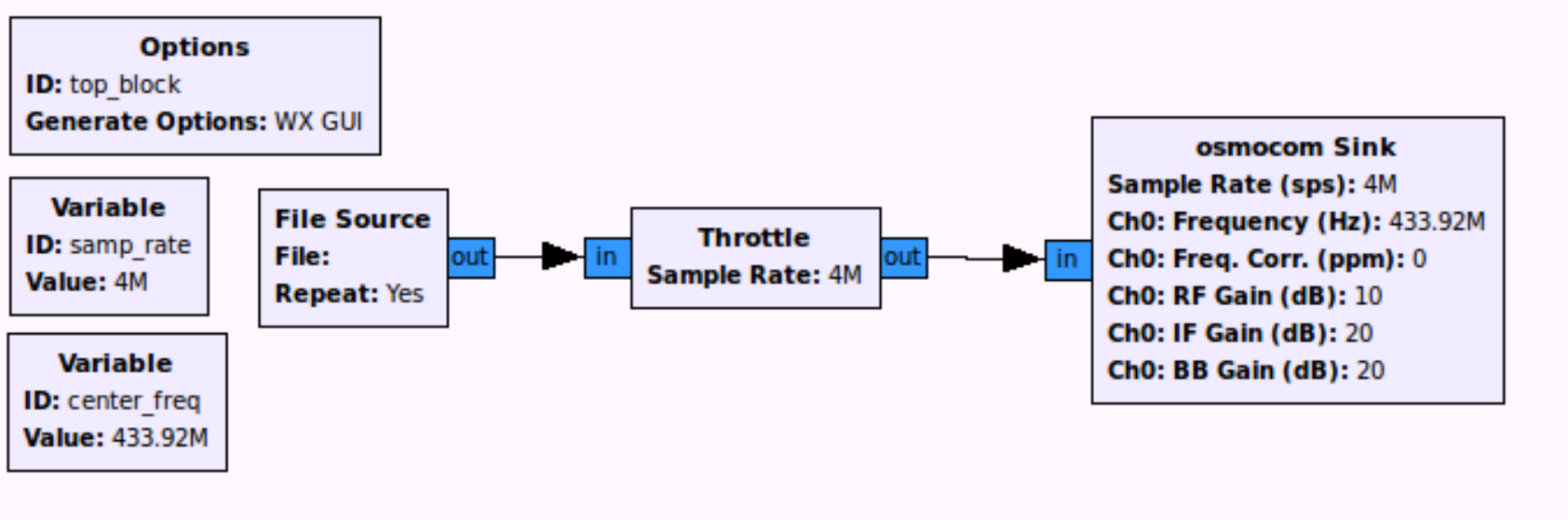
RKE replay

Signal save

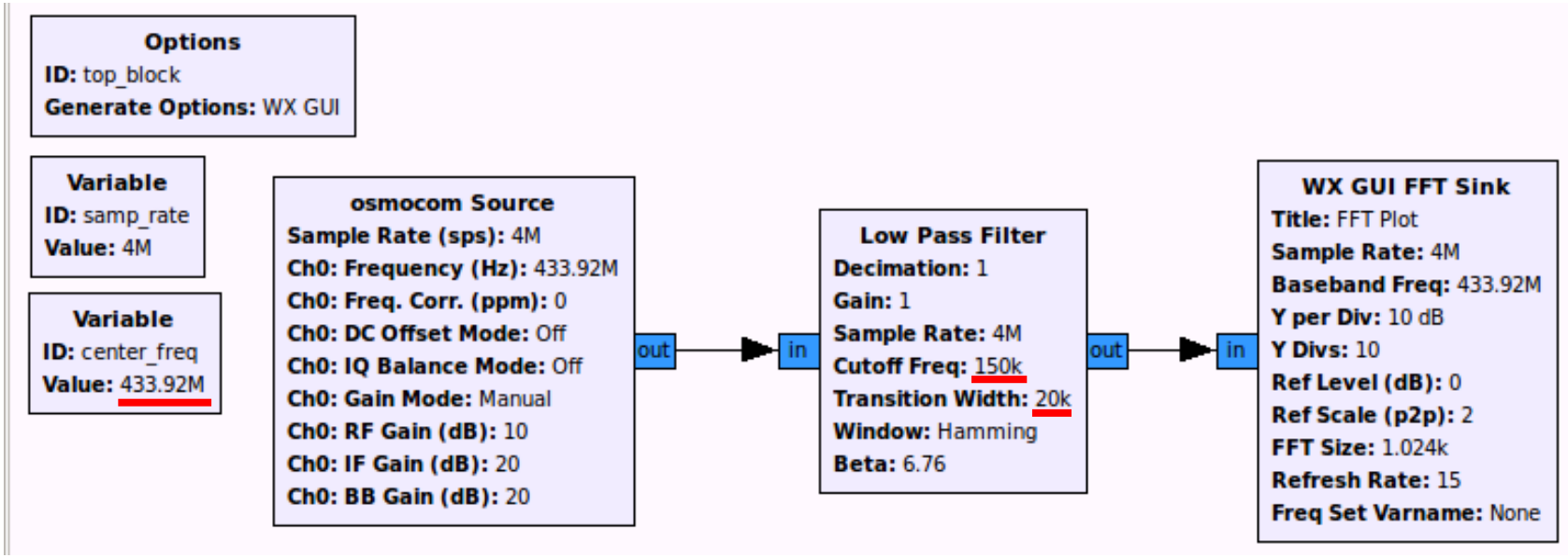


RKE replay

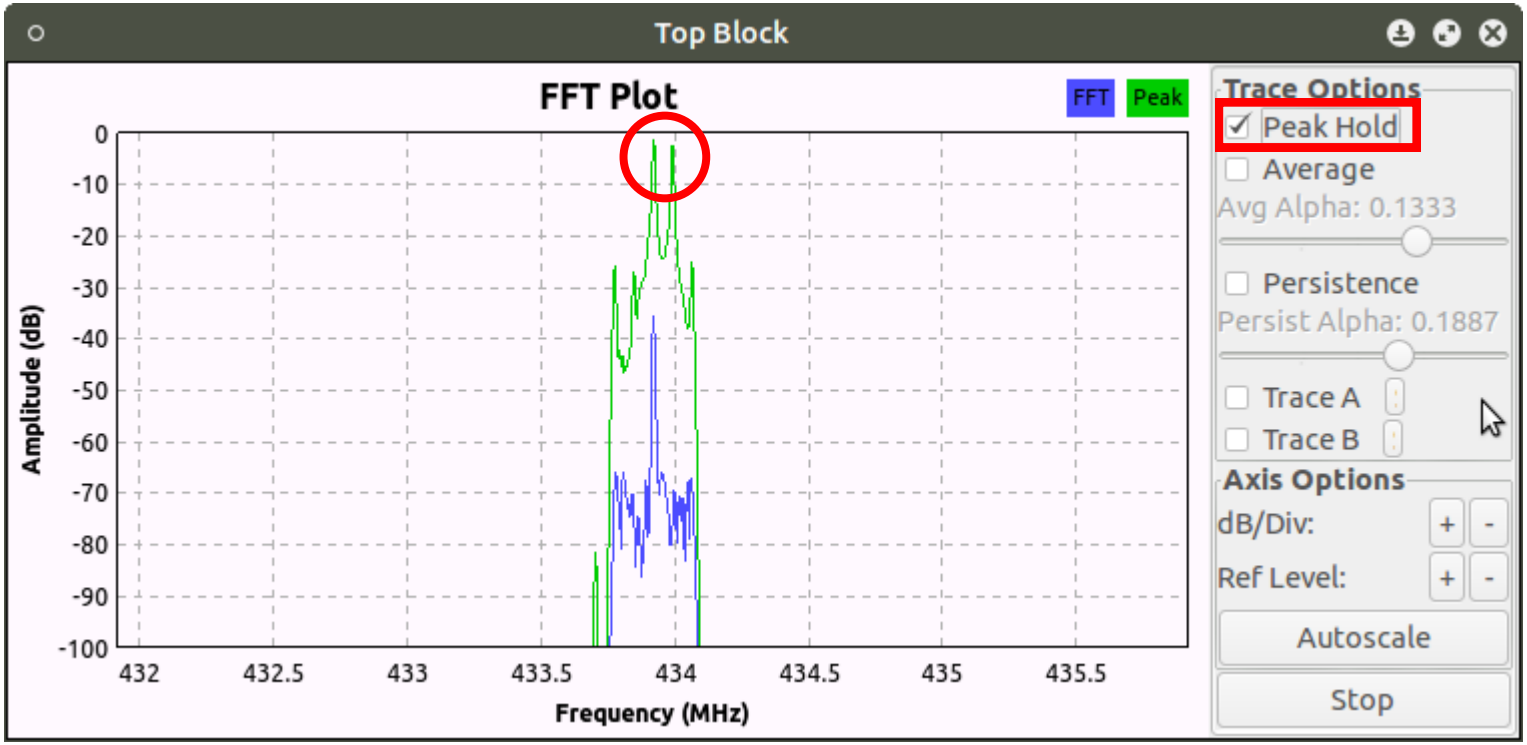
Play



RKE signal analysis



RKE signal analysis



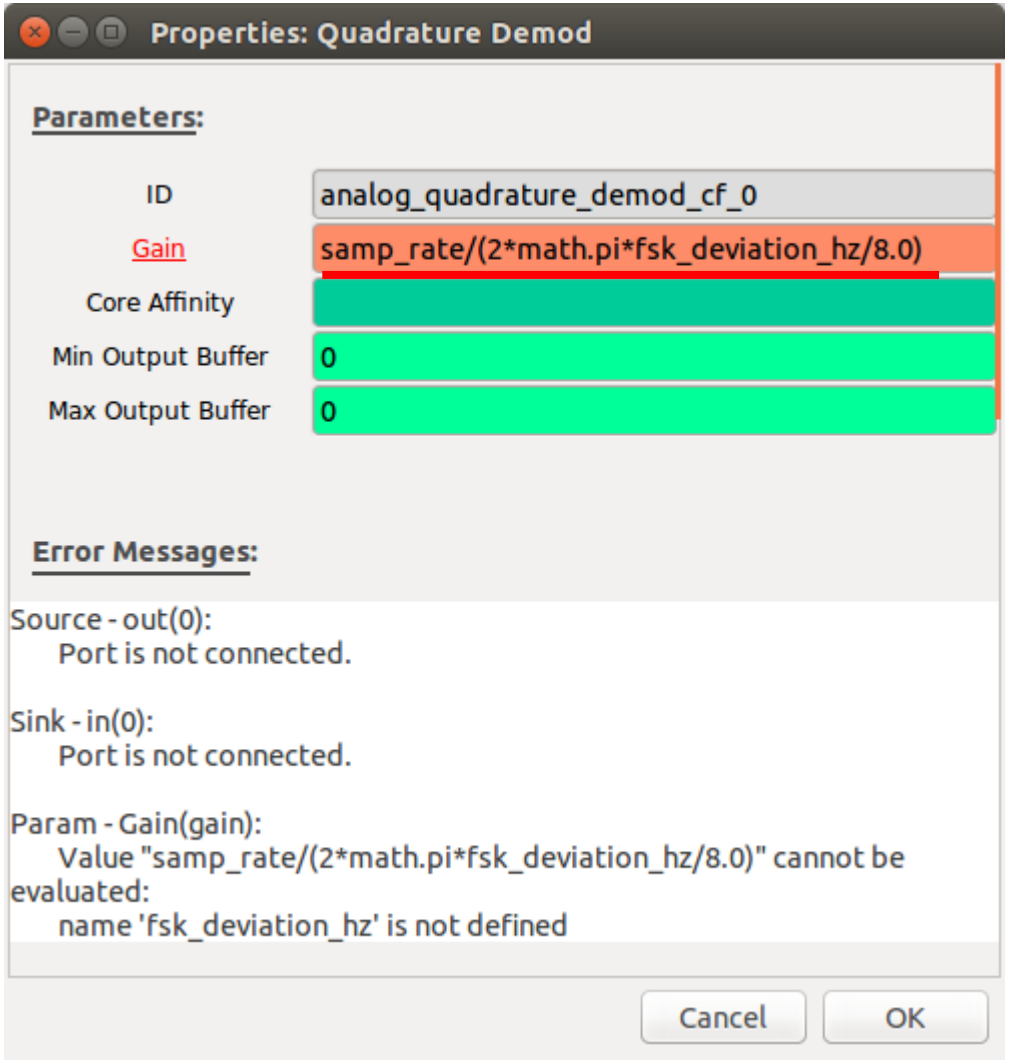
RKE signal analysis

math.pi = π

fsk_deviation_hz
: fsk modulation에 사용된
주파수간 차이

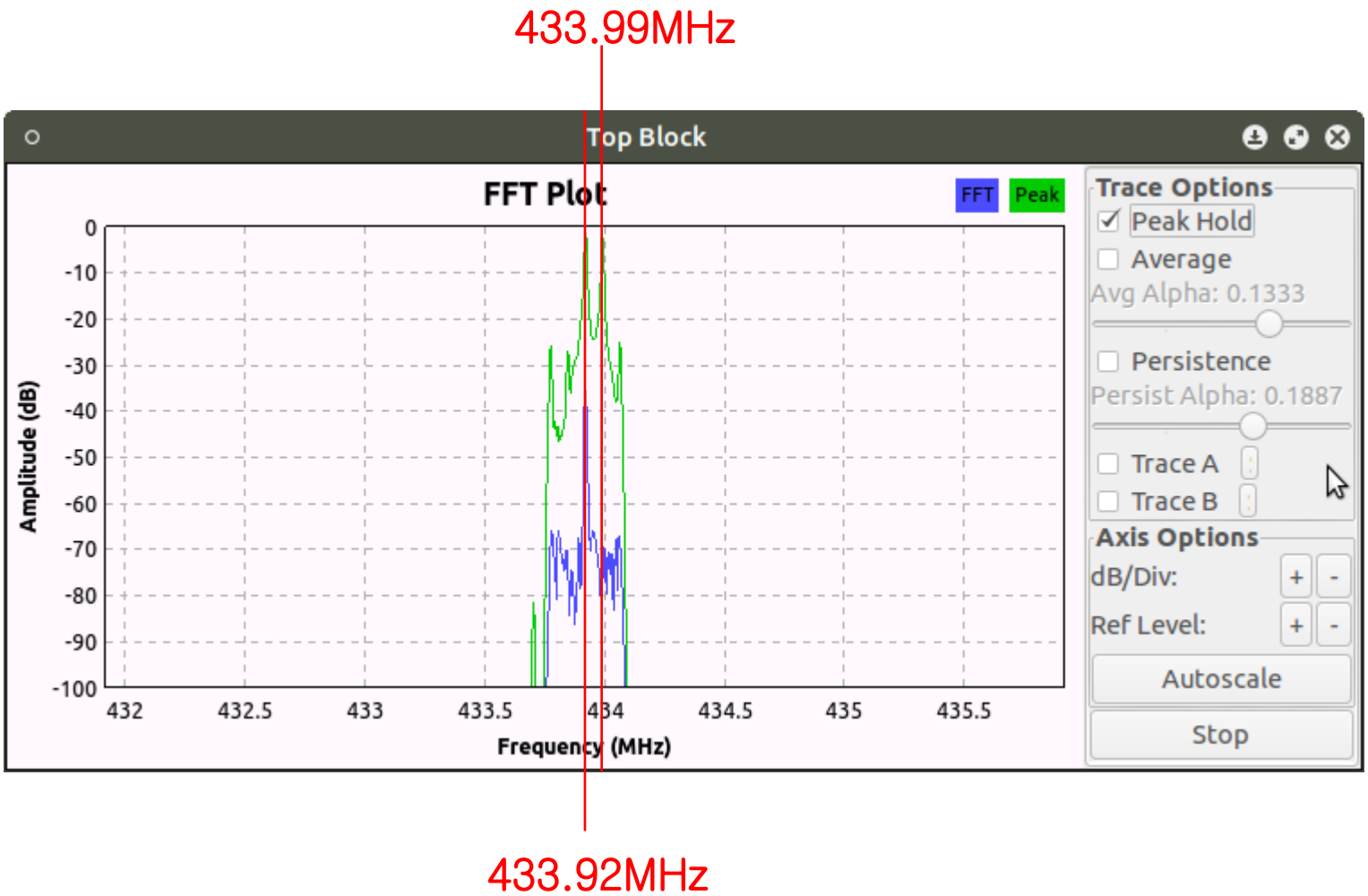
$$s_1(t) = A_c \cos(2\pi(f_c - \Delta f)t)$$
$$s_2(t) = A_c \cos(2\pi(f_c + \Delta f)t)$$

Here Dδf is called the **frequency deviation**.

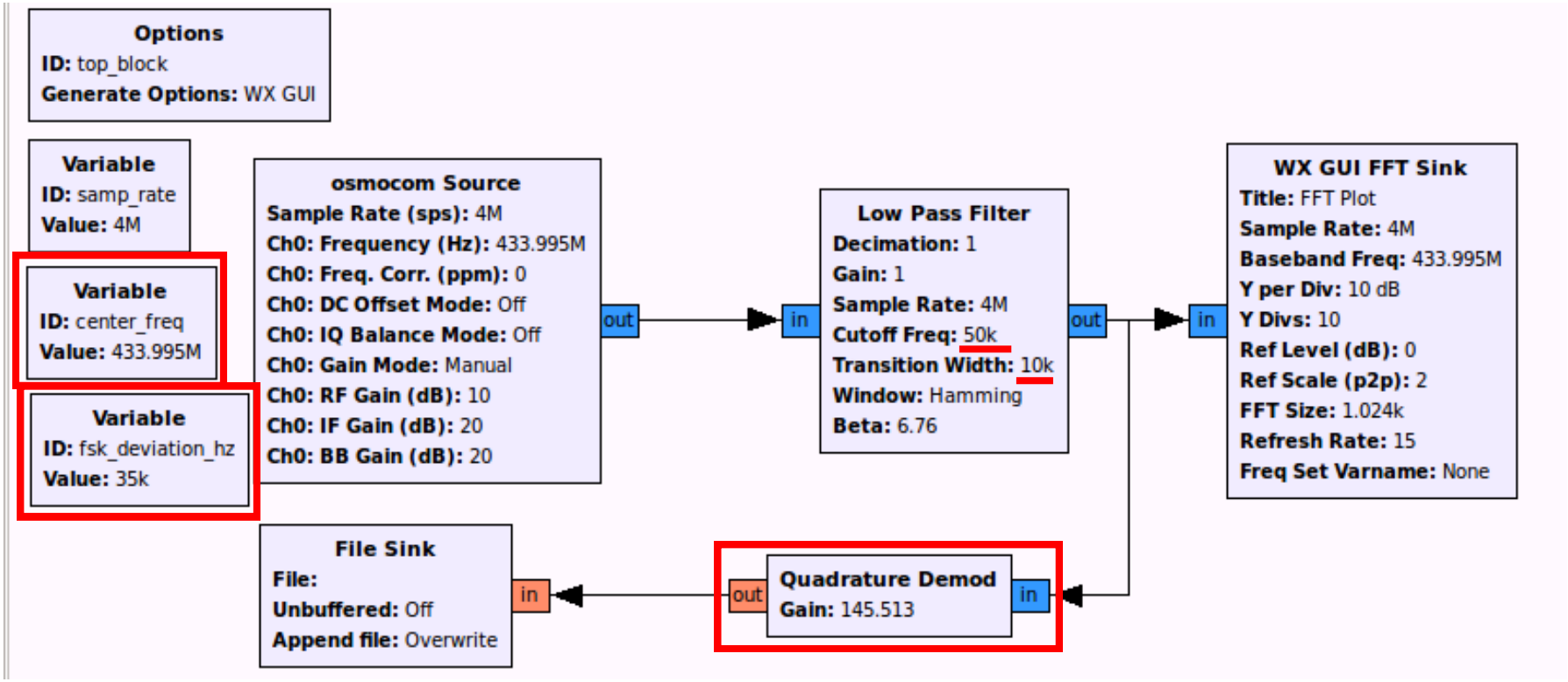


RKE signal analysis

$f_c = 433.995MHz$
$\Delta f = 35kHz$

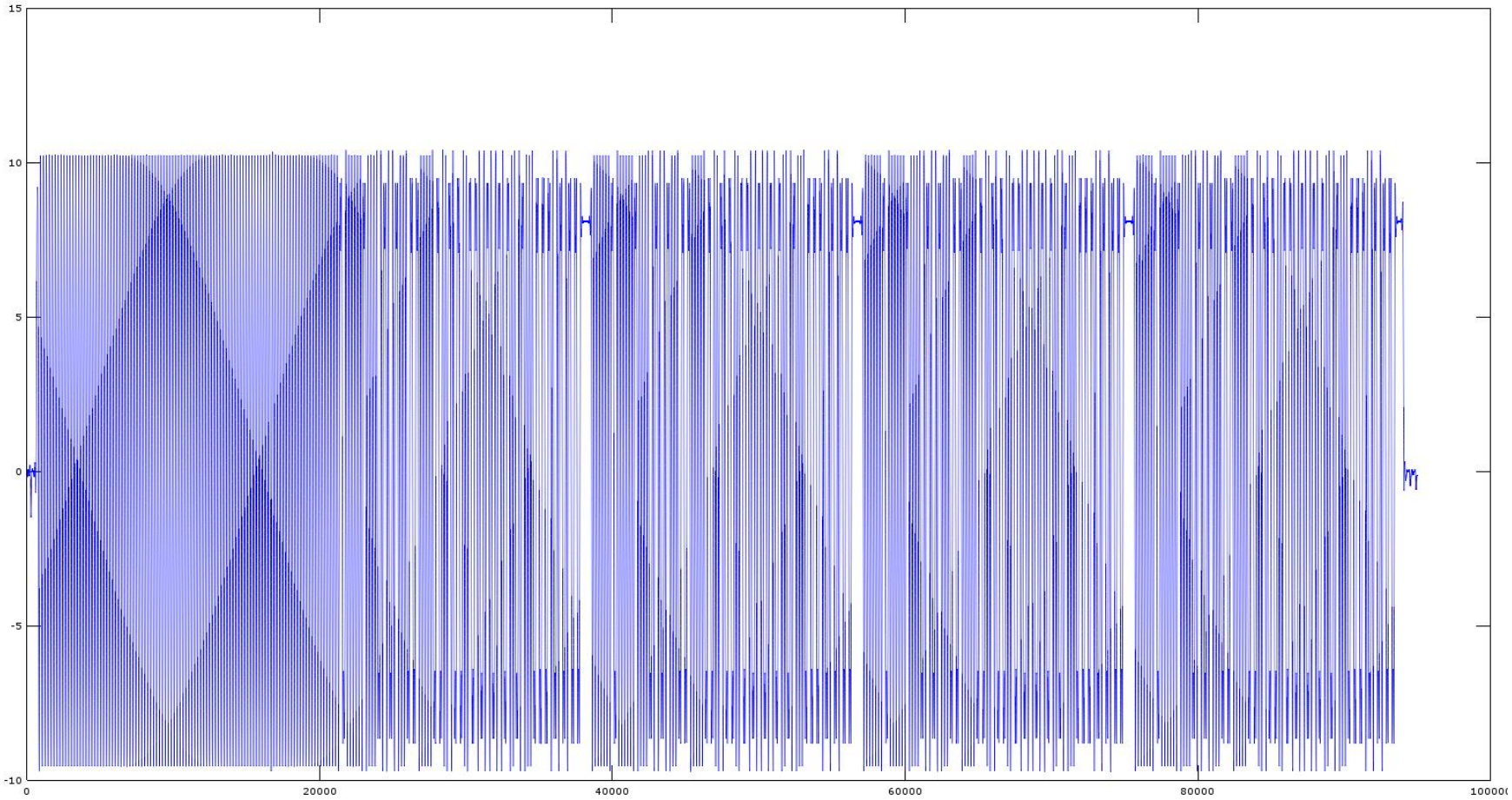


RKE signal analysis



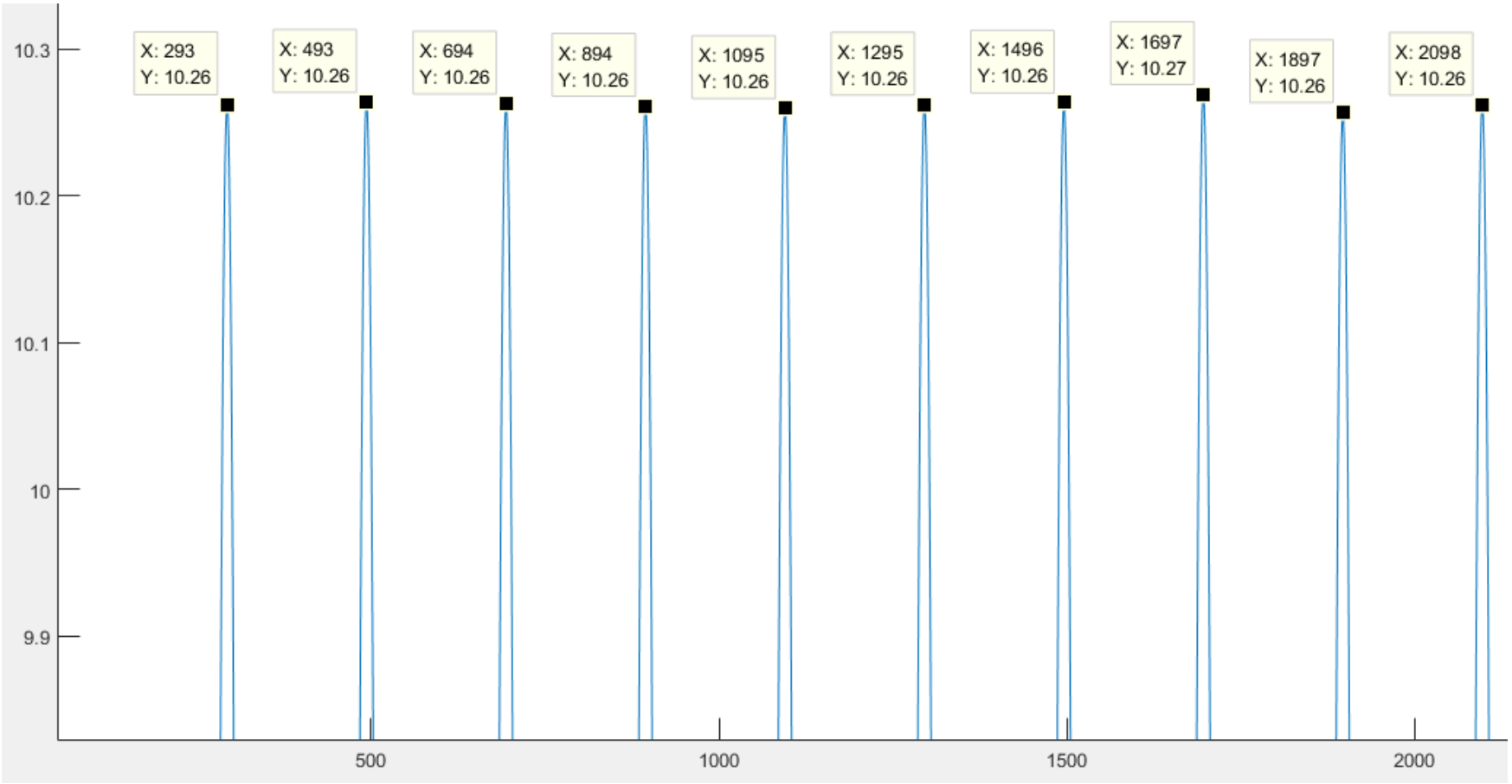
Bits Extraction

About 200 samples



Bits Extraction

About 200 samples



Bits Extraction

1. 200 samples = 1 bit

2. 0 crossing = bit change

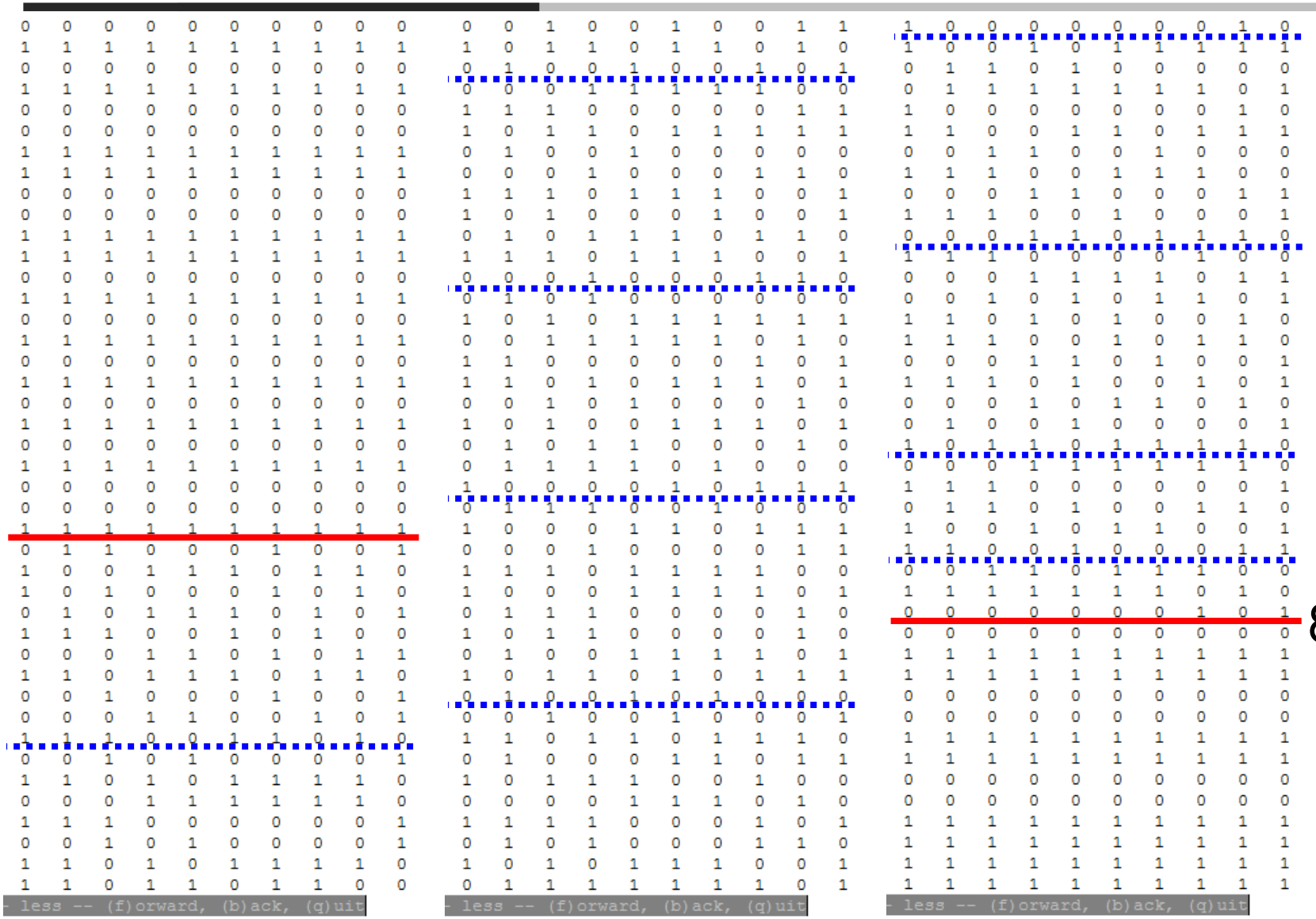
3. Total about 1000 bit

Bits Extraction

```
clc; clear all; close all;

%% open file
f = fopen('D:\close_signal\50\morning_test');
close_sig= fread(f,'float');
fclose(f);

%% detect length define
detect_len = length(close_sig);

%% detect vector define (for FOR loop)
detect_range = 1:detect_len;

%% starting point finding vector
inv_detect_range = 1:20;

%% starting point find
for i = detect_range
  if (close_sig(i) > 4) % find over +4 point,
    for j = inv_detect_range % back to negati
      if (close_sig(i-j) <0)
        break;
      endif
    endfor
    break;
  endif
endfor
```

```
  % negative condition
  elseif(sign_flag ==0)
    % maintain negative condition
    if (close_sig(i) <= 0)
      % coundting symbol
      symbol_cnt++;
                %printf("symbol_cnt = %d\n", symbol_cnt);

    % positive value detected
    elseif (close_sig(i) >0)
      % bits number carculating
      bits_cnt = round(symbol_cnt/100);
                %printf("bits_cnt = %d, ",bits_cnt);

      % record bits '0'
      bits = [bits ; zeros(bits_cnt,1)];
                %printf("%d\n", bits);

      % reset the symple/symbol
      symbol_cnt = 0;
      % change sign condition to positive
      sign_flag = 1;
      % increase bits length
      bits_len = bits_len + bits_cnt;
    endif
  endif
endfor

save('demod_bits','bits');

plot(close_sig)
figure();
plot(bits);
```

Bits Extraction



88

BLACK PERL

Bits Extraction

1. Preamble = 97 bits
2. ID = 14 bits
3. Action bits = 2 bits
4. Rolling code = 44 bits
5. End bits = 17 bits

총 3번 반복

Bits Generation

FSK Modulation

$$S_0(t) = S_0, i(t) + jS_0, q(t)$$

$$S(t) = \mathrm{Re}\left\{S_0(t)e^{j2\pi f_c t}\right\}$$

$$= \mathrm{Re}\left\{\left[S_0, i(t) + jS_0, q(t)\right]\left[\cos 2\pi f_c t + j \sin 2\pi f_c t\right]\right\}$$

$$= S_0, i(t) \cos 2\pi f_c t - S_0, q(t) \sin 2\pi f_c t$$

...

$$S_0(t) = \cos\left[2\pi f_d \int m(\tau)d\tau\right] + j \sin\left[2\pi f_d \int m(\tau)d\tau\right]$$

Bits Generation

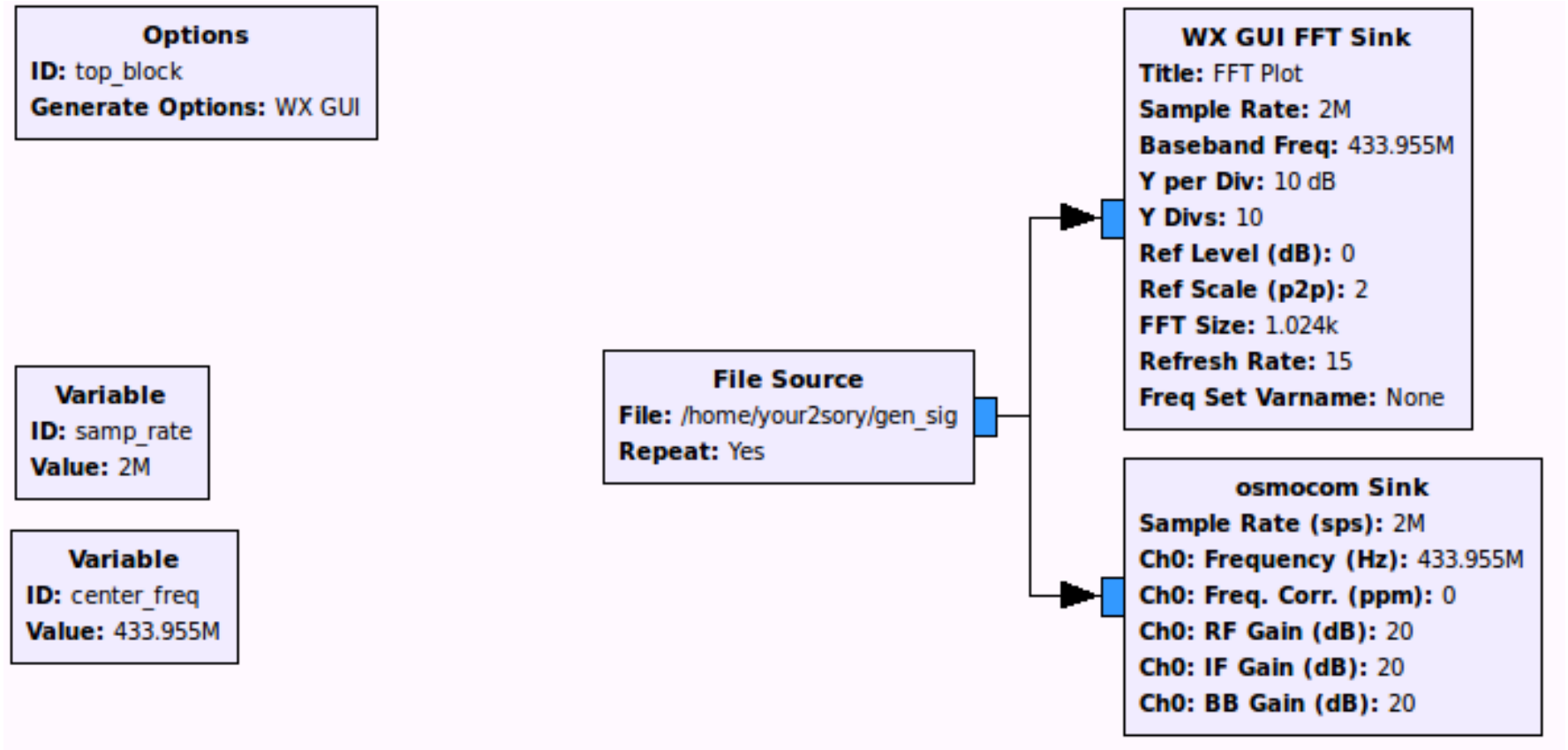
```
8  center_freq = 433.92e6;      % 433.92MHz
9  samp_rate = 2e6;             % 4MHz
10 fsk_deviation_hz = 70e3;     % 35kHz
11
12 %% Preamble Gen.
13
14 preamble_len = 97;
15 preamble = ones(preamble_len,1);
16 man_preamble = kron(preamble, [0 ; 1]);
17
18 %% ID & action Gen.
19
20 open_action = [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1].';
21 close_action = [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0].';
22
23 ID = [ 0 0 0 0 1 1 0 0 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1].';
24
25 man_open_ID_0 = kron([xor(open_action,1) ; xor(ID,1)], [1 ; 0]);
26 man_close_ID_0 = kron([xor(close_action,1) ; xor(ID,1)], [1 ; 0]);
27
28 man_open_ID_1 = kron([open_action ; ID], [0 ; 1]);
29 man_close_ID_1 = kron([close_action ; ID], [0 ; 1]);
30
31 man_open_ID = man_open_ID_0 + man_open_ID_1 ;
32 man_close_ID = man_close_ID_0 + man_close_ID_1 ;
```

```
49 t = 0:(1/(2e6)):0.0010025;
50 fc_0 = 0;          %30kHz
51 fc_1 = 70e3;        %100kHz
52 fsk_signal_0 = cos(2*pi*fc_0*t);
53 fsk_signal_1 = cos(2*pi*fc_1*t);
54
55 base_tx_0_sig = kron(xor(tx_bit,1),fsk_signal_0.');
56 base_tx_1_sig = kron(tx_bit,fsk_signal_1.');
57
58 base_tx_sig = base_tx_0_sig + base_tx_1_sig;
59
60 t = 0:1/2e6:1;
61 len_sig = length(base_tx_sig);
62 carrier = cos(2*pi*center_freq*t(1:len_sig)).';
63
64 tx_sig = base_tx_sig .* carrier;
65
66 fft_tx_sig = fft(tx_sig);
67
68 subplot(2,1,1);
69 plot(real(fftshift(fft_tx_sig)));
70 subplot(2,1,2);
71 plot(imag(fftshift(fft_tx_sig)));
72
73 fid = fopen('gen_sig','wb');
74 fwrite(fid,tx_sig,'float32');
75 fclose(fid);
```

30MB 크기의 파일 생성

BLACK PERL

Bits Generation



Bits Transmission

시연

Conclusion

1. 신호 수신 방법을 알면 생성도 가능

2. 생성 방법은 수식의 코드전환 수준

3. 대부분의 소형 장비의 신호는 매우 단순

4. 대해적의 시대가 왔습니다.

Q & A

감사합니다.