GNU RADIO BLOCK CREATION

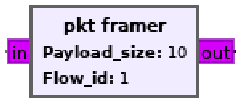
Igor Povarich

Comp Eng 5430 Lab C-1

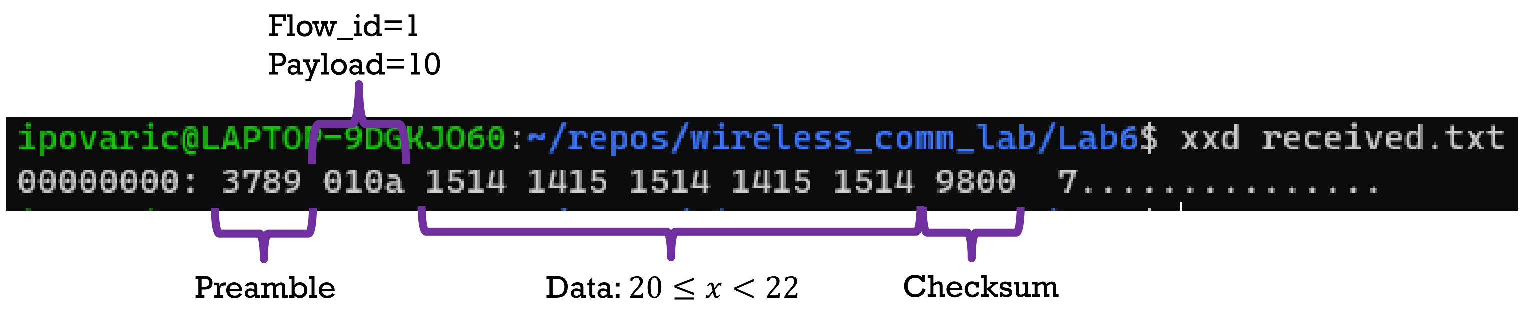
GOALS

- Create 2 GNURadio blocks
  - Transmitter block
    - Preamble (fixed)
    - Flow ID (user configurable)
    - User Data (user configurable)
    - Checksum (calculation)
  - Receiver block
    - Catch the preamble to identify the start of the data stream
    - Filter based on the flow ID
    - Check for errors using the checksum
    - Pass user data, if validated

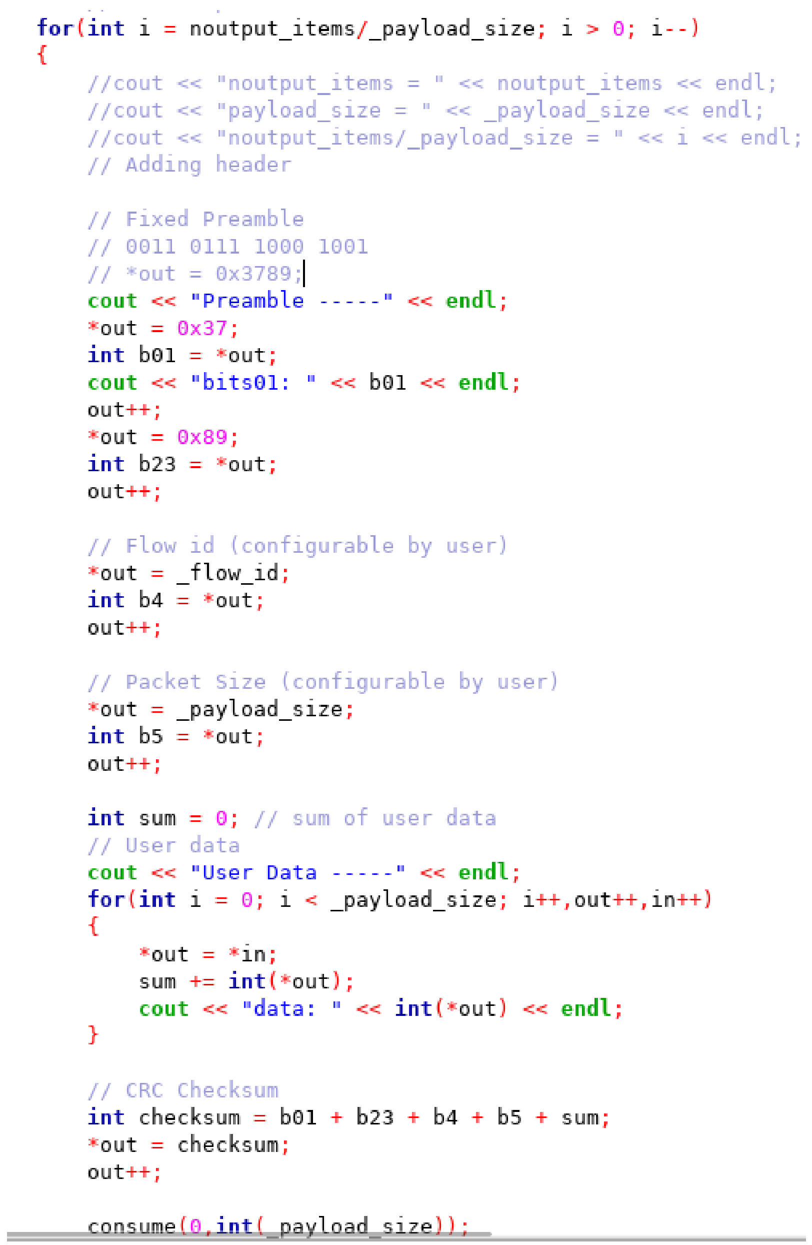
TRANSMITTER BLOCK



- Functioning transmitter block that writes the correct byte stream

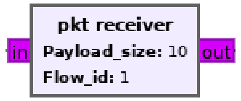


```
ipovaric@LAPTOP-9DGKJO60:~/repos/wireless_comm_lab/Lab6$ xxd received.txt
00000000: 3789 010a 1514 1415 1514 1415 1514 9800  7...............
```

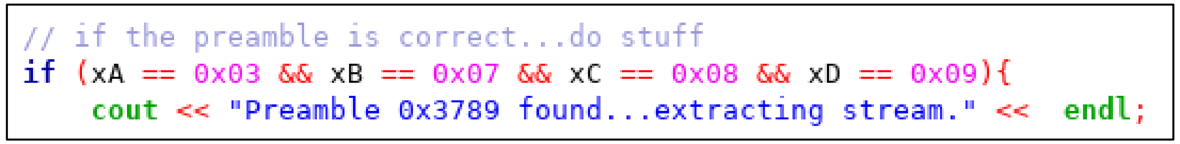


```
for(int i = noutput_items/_payload_size; i > 0; i--)
{
    //cout << "noutput_items = " << noutput_items << endl;
    //cout << "payload_size = " << _payload_size << endl;
    //cout << "noutput_items/_payload_size = " << i << endl;
    // Adding header

    // Fixed Preamble
    // 0011 0111 1000 1001
    // *out = 0x3789;
    cout << "Preamble -----" << endl;
    *out = 0x37;
    int b01 = *out;
    cout << "bits01: " << b01 << endl;
    out++;
    *out = 0x89;
    int b23 = *out;
    out++;

    // Flow id (configurable by user)
    *out = _flow_id;
    int b4 = *out;
    out++;

    // Packet Size (configurable by user)
    *out = _payload_size;
    int b5 = *out;
    out++;

    int sum = 0; // sum of user data
    // User data
    cout << "User Data -----" << endl;
    for(int i = 0; i < _payload_size; i++,out++,in++)
    {
        *out = *in;
        sum += int(*out);
        cout << "data: " << int(*out) << endl;
    }

    // CRC Checksum
    int checksum = b01 + b23 + b4 + b5 + sum;
    *out = checksum;
    out++;

    consume(0,int(_payload_size));
```

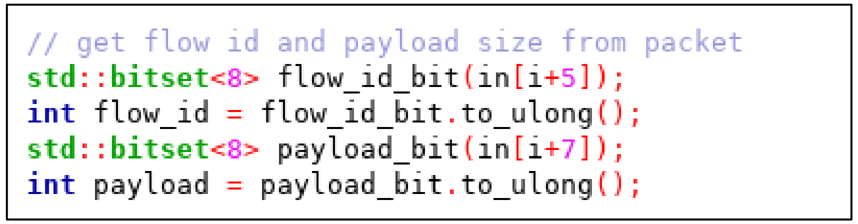
RECEIVER BLOCK



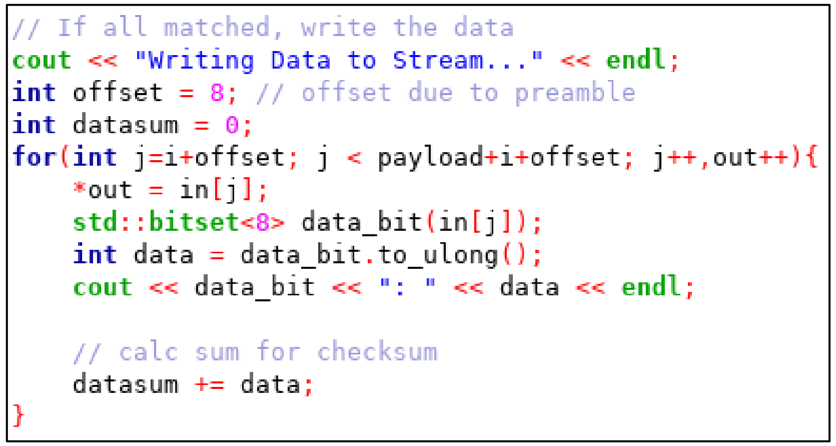
- Continuously grabs four sequential 4-bit bytes and compares them
  - As soon as they match the preamble, execute the main code
- Extract flow id and payload
  - Filter based on Flow ID
- If all matches, write the data to the output and sum it for the checksum
- Currently a bug in the checksum due to the byte packing/unpacking used



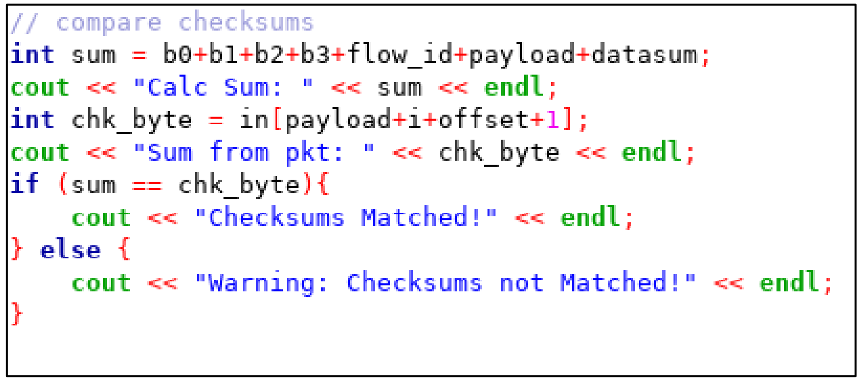
```
// if the preamble is correct...do stuff
if (xA == 0x03 && xB == 0x07 && xC == 0x08 && xD == 0x09){
    cout << "Preamble 0x3789 found...extracting stream." <<  endl;
```



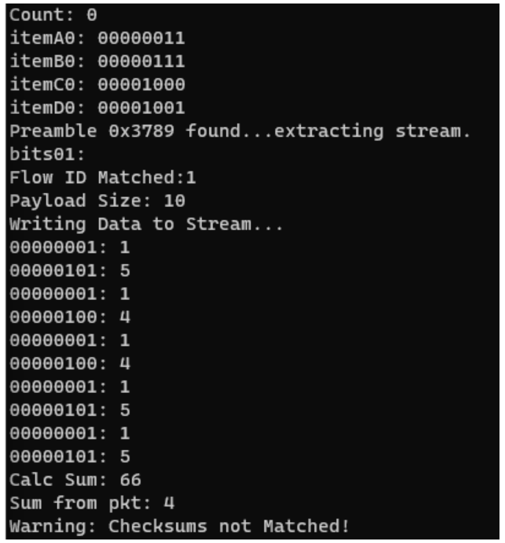
```
// get flow id and payload size from packet
std::bitset<8> flow_id_bit(in[i+5]);
int flow_id = flow_id_bit.to_ulong();
std::bitset<8> payload_bit(in[i+7]);
int payload = payload_bit.to_ulong();
```



```
// If all matched, write the data
cout << "Writing Data to Stream..." << endl;
int offset = 8; // offset due to preamble
int datasum = 0;
for(int j=i+offset; j < payload+i+offset; j++,out++){
    *out = in[j];
    std::bitset<8> data_bit(in[j]);
    int data = data_bit.to_ulong();
    cout << data_bit << ": " << data << endl;

    // calc sum for checksum
    datasum += data;
}
```



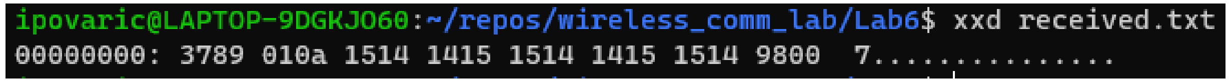
```
// compare checksums
int sum = b0+b1+b2+b3+flow_id+payload+datasum;
cout << "Calc Sum: " << sum << endl;
int chk_byte = in[payload+i+offset+1];
cout << "Sum from pkt: " << chk_byte << endl;
if (sum == chk_byte){
    cout << "Checksums Matched!" << endl;
} else {
    cout << "Warning: Checksums not Matched!" << endl;
}
```



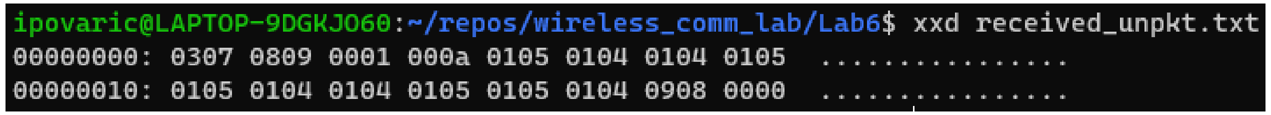
```
Count: 0
itemA0: 00000011
itemB0: 00000111
itemC0: 00001000
itemD0: 00001001
Preamble 0x3789 found...extracting stream.
bits01:
Flow ID Matched:1
Payload Size: 10
Writing Data to Stream...
00000001: 1
00000101: 5
00000001: 1
00000100: 4
00000001: 1
00000100: 4
00000001: 1
00000101: 5
00000001: 1
00000101: 5
Calc Sum: 66
Sum from pkt: 4
Warning: Checksums not Matched!
```

CURRENT TEST CODE

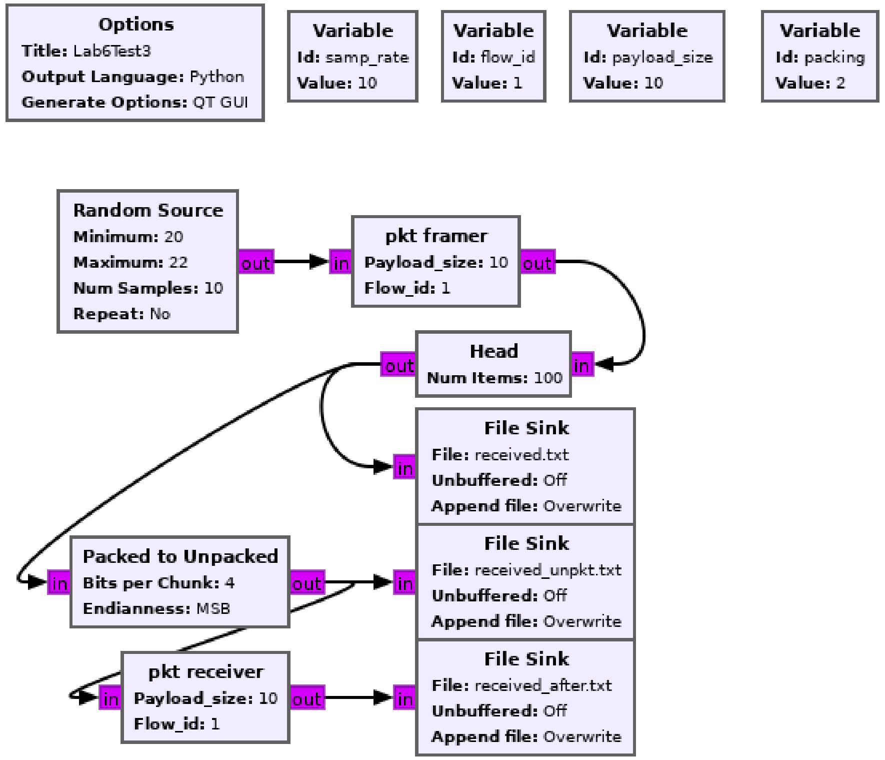
- Simply writing packed and unpacked data to file sinks to compare the block outputs
  - Use Packed to Unpacked block in to simulate packing of a modulator block



```
ipovaric@LAPTOP-9DGKJO60:~/repos/wireless_comm_lab/Lab6$ xxd received.txt
00000000: 3789 010a 1514 1415 1514 1415 1514 9800  7...............
```



```
ipovaric@LAPTOP-9DGKJO60:~/repos/wireless_comm_lab/Lab6$ xxd received_unpkt.txt
00000000: 0307 0809 0001 000a 0105 0104 0104 0105  ................
00000010: 0105 0104 0104 0105 0105 0104 0908 0000  ................
```



Options
Title: Lab6Test3
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_rate
Value: 10

Variable
Id: flow_id
Value: 1

Variable
Id: payload_size
Value: 10

Variable
Id: packing
Value: 2

Random Source
Minimum: 20
Maximum: 22
Num Samples: 10
Repeat: No

pkt framer
Payload_size: 10
Flow_id: 1

Head
Num Items: 100

File Sink
File: received.txt
Unbuffered: Off
Append file: Overwrite

Packed to Unpacked
Bits per Chunk: 4
Endianness: MSB

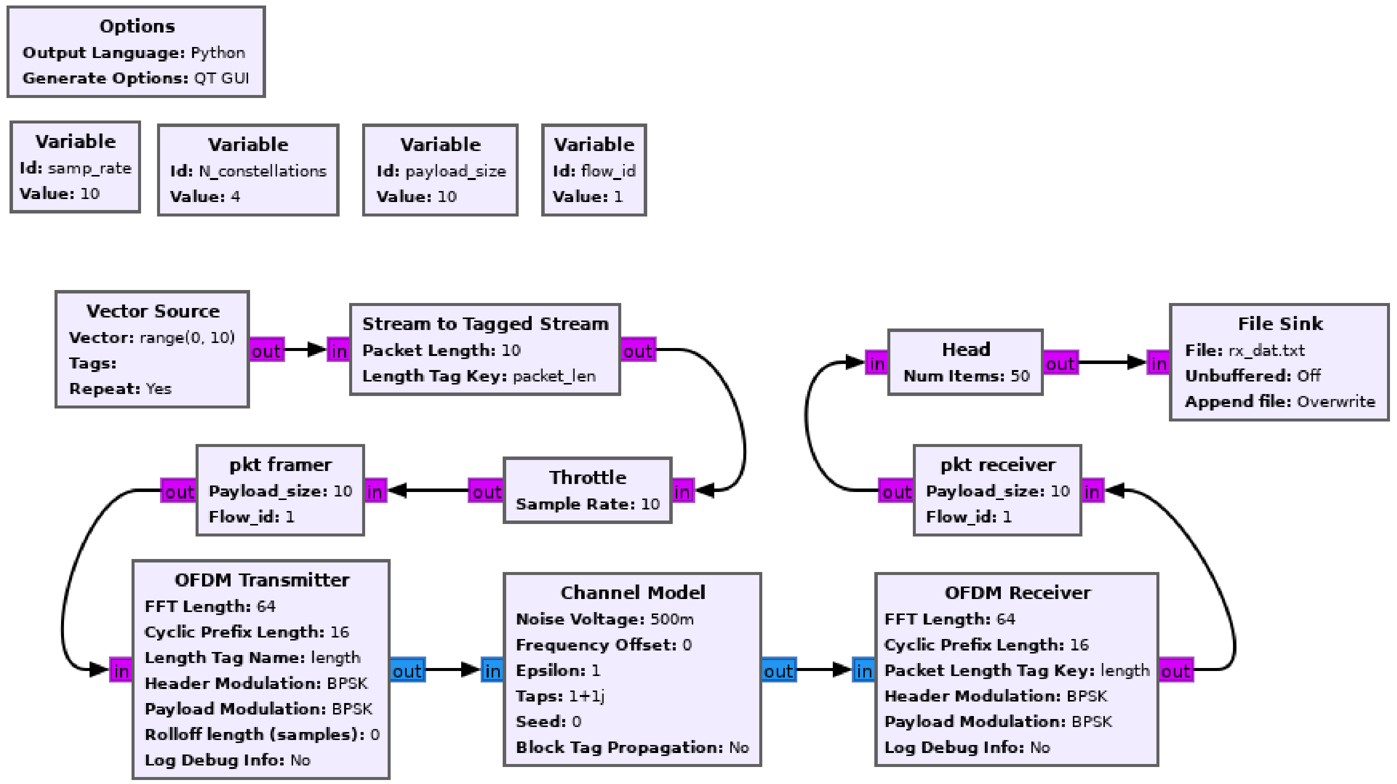
File Sink
File: received_unpkt.txt
Unbuffered: Off
Append file: Overwrite

pkt receiver
Payload_size: 10
Flow_id: 1

File Sink
File: received_after.txt
Unbuffered: Off
Append file: Overwrite

PLANNED EXAMPLE/TEST CODE

- Use OFDM transmitter/receiver model
  - Vary the noise voltage in the channel model to determine the resilience of the receiver block to errors and noise



Options
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_rate
Value: 10

Variable
Id: N_constellations
Value: 4

Variable
Id: payload_size
Value: 10

Variable
Id: flow_id
Value: 1

Vector Source
Vector: range(0, 10)
Tags:
Repeat: Yes

Stream to Tagged Stream
Packet Length: 10
Length Tag Key: packet_len

Throttle
Sample Rate: 10

pkt framer
Payload_size: 10
Flow_id: 1

OFDM Transmitter
FFT Length: 64
Cyclic Prefix Length: 16
Length Tag Name: length
Header Modulation: BPSK
Payload Modulation: BPSK
Rolloff length (samples): 0
Log Debug Info: No

Channel Model
Noise Voltage: 500m
Frequency Offset: 0
Epsilon: 1
Taps: 1+1j
Seed: 0
Block Tag Propagation: No

OFDM Receiver
FFT Length: 64
Cyclic Prefix Length: 16
Packet Length Tag Key: length
Header Modulation: BPSK
Payload Modulation: BPSK
Log Debug Info: No

pkt receiver
Payload_size: 10
Flow_id: 1

Head
Num Items: 50

File Sink
File: rx_dat.txt
Unbuffered: Off
Append file: Overwrite