Network communication protocol for SNOWLeoSDR platform

Version	Date	Comments
V1.0	2013/12	Initial version
V1.1	2014/5	Add SNOWLeoSDR support
V1.2	2014/7	Change freq configure mode
V1.3	2015/1	Support matlab& gnuradio

Brief

Data link
TX Port: 5005
RX Port: 5004
Data format(e.g. 2 samples,8Bytes)

I1	**Q1**	**I2**	**Q2**

Control link
Port:5006

packet head	Control Word	Param	Param	Param	Param	Param	Param

1、Control Word Definitions

```
#define   PACKAGE_HEAD             0xF0
#define   SDR_HANDSHAKE            0x16

#define   SDR_RF_CTRL_TX_FREQ      0x17
#define   SDR_RF_CTRL_RX_FREQ      0x18
#define   SDR_RF_CTRL_TX_VGA       0x19
```

```
#define    SDR_RF_CTRL_RX_VGA        0x20
#define    SDR_RF_CTRL_TX_DC         0x21
#define    SDR_RF_CTRL_RX_DC         0x23
#define    SDR_RF_CTRL_SAMPRATE      0x24
```

2、Network Command Format

Network command length is 8 bytes,command format as follows

MSB LSB

packet head	Control Word	Param	Param	Param	Param	Param	Param

3、Network Command Details

SDR_HANDSHAKE ----PC send to Snowleosdr

MSB LSB

F0	16	DIR	ID	Param

DIR: 0, TX; 1, RX

ID: 1, gnuradio mode；2, matlab mode

Param: When DIR was 0(TX)，Param is 0

When DIR was 1(RX)，Param is size of data PC request(in byte)

SDR_RF_CTRL_TX_FREQ ----Snowleosdr’s TX RF configuration

MSB LSB

F0	17	0	0	Frequency (Hz)

Frequency: 300MHz~3800MHz step:100KHz

e.g. 0xF0170000 7F8DCF00 (2140MHz)

SDR_RF_CTRL_RX_FREQ ---- Snowleosdr’s RX RF configuration

MSB LSB

F0	18	0	0	Frequency (Hz)

SDR_RF_CTRL_TX_VGA ---- Snowleosdr’s TX Gain configuration

MSB LSB

F0	19	VGA2	VGA1	PA	GPIOSEL	0	0

VGA1:0~0x1f range:-35dB~-4dB

VGA2: 0~0x1f range:0dB~25dB

PA GPIOSEL reference example configuration

e.g. 0xF0190016 0B000000(VGA1=0x16; VGA2=0; PA=0x0b, GPIOSEL=0)

SDR_RF_CTRL_RX_VGA ---- Snowleosdr’s RX Gain configuration

MSB LSB

F0	**20**	**LNA**	**VGA**	**GPIOSEL**	**0**	**0**	**0**

注：

VGA2：0~20 range:0~60dB

LNA GPIOSEL reference example configuration

e.g. 0xF020D000 03000000(LNA=0xd0; VGA=0; GPIOSEL=0x03)

SDR_RF_CTRL_TX_DC ---- Snowleosdr’s TX DC configuration

MSB LSB

F0	**21**	**DCI**	**DCQ**	**0**	**0**	**0**	**0**

SDR_RF_CTRL_RX_DC ---- Snowleosdr’s RX DC configuration

MSB LSB

F0	**23**	**DCI**	**DCQ**	**0**	**0**	**0**	**0**

SDR_RF_CTRL_SAMPLE ---- Configure RF sample rate clock

MSB LSB

F0	**24**	**0**	**0**	**Sample Rate(Hz)**			

Sample Code

Configure TX RF

C code

```
int set_freq(unsigned int d_tx_freq)
{
        unsigned int cmd_buf[2]={0,0};
        cmd_buf[0] = 0xF0170000; /*set tx freq*/
        cmd_buf[1] = d_tx_freq;
        if(send(sockfd, cmd_buf, sizeof(cmd_buf), 0) > 0)
                return 1;
        else
                return -1;
}
```

Matlab code

```
%% open control port
ctrl_link = udp('192.168.1.10', 5006);
fopen(ctrl_link);

%% send tx freq set cmd
freq_hex=dec2hex(1090e6,8);
tx_freq=[0 0 hex2dec('17') hex2dec('f0') hex2dec(freq_hex(7:8)) hex2dec(freq_hex(5:6))
hex2dec(freq_hex(3:4)) hex2dec(freq_hex(1:2))];
fwrite(ctrl_link,tx_freq,'uint8');
```

Recv data from network

C code

```
#define RX_SAMPLES_NUM    8*1024
void *recv_sample(void *)
{
        unsigned int cmd_buf[2]={0,0};
        int len = 0, i = 0;
        cmd_buf[0] = 0xF0160102;
        cmd_buf[1] = 0x00002000; /*set rx size，recv 8KBytes */
        send(sockfd, cmd_buf, sizeof(cmd_buf), 0);

        do {
                len += recv(sockfd, data+len, RX_SAMPLES_NUM-len, 0);
        }while(len != RX_SAMPLES_NUM);

        return NULL;
}
```

Matlab code

```
clc;
clear;
close all;
warning off;
cyc=1;
%% open control port
ctrl_link = udp('192.168.1.10', 5006);
fopen(ctrl_link);
%% open data port
data_link = tcpip('192.168.1.10', 5004);
set(data_link,'InputBufferSize',256*1024);
set(data_link,'OutputBufferSize',16*1024);
fopen(data_link);

%% send rx freq
freq_hex=dec2hex(1090e6,8);
rx_freq=[0 0 hex2dec('18') hex2dec('f0') hex2dec(freq_hex(7:8)) hex2dec(freq_hex(5:6))
hex2dec(freq_hex(3:4)) hex2dec(freq_hex(1:2))];
fwrite(ctrl_link,rx_freq,'uint8');

%% send rx vga
rx_vga=[0 hex2dec('d0') hex2dec('20') hex2dec('f0') 0 0 0 hex2dec('03')];
fwrite(ctrl_link,rx_vga,'uint8');

%% send handshake cmd
size_hex=dec2hex(8*1024,8);
handshake=[2 1 hex2dec('16') hex2dec('f0') hex2dec(size_hex(7:8)) hex2dec(size_hex(5:6))
hex2dec(size_hex(3:4)) hex2dec(size_hex(1:2))];
fwrite(ctrl_link,handshake,'uint8');
pause(0.5);

while (1)
    %% read 256*1024 bytes data from zing
    data = fread(data_link,8*1024,'uint8');

    %% receive
    figure(1);clf;
    datah=data(2:2:end);
```

```
        datal=data(1:2:end);
        datah_hex=dec2hex(datah,2);
        datal_hex=dec2hex(datal,2);
        data_hex(:,1:2)=datah_hex;
        data_hex(:,3:4)=datal_hex;
        dataun=hex2dec(data_hex);
        datain=dataun-(dataun>32767)*65536;
        a1=datain(2:2:end);
        a2=datain(1:2:end);
        [uV sV] = memory;
        mem=round(uV.MemUsedMATLAB/2^20);
        subplot(221);
        plot(a1,'b');
        hold on
        plot(a2,'r');
        subplot(222);
        plot(a1,a2);
        title(['plot cyc=',num2str(cyc),';mem=',num2str(mem),'MB']);
%           pause;
        cyc=cyc+1;
end

%% close all link
fclose(data_link);
delete(data_link);
clear data_link;
```

Send data to network

C code

```
int send_sample()
{
          unsigned int cmd_buf[2]={0,0};
          int len = 0, i = 0, nbyte = 0;

          cmd_buf[0] = 0xF0160001;
          cmd_buf[1] = 0;
          send(sockfd, cmd_buf, sizeof(cmd_buf), 0);
```

```
        usleep(10000);
        send(sockfd, read_buffer, TX_SAMPLES_NUM, 0);

        return NULL;
}
```

Matlab code

```
clc;
clear all;
close all;
%% open control port
ctrl_link = udp('192.168.1.10', 5006);
fopen(ctrl_link);
%% open data port
data_link = tcpip('192.168.1.10', 5005);
set(data_link,'InputBufferSize',16*1024);
set(data_link,'OutputBufferSize',64*1024);
fopen(data_link);

%% send tx sample rate set cmd
samp_rate=dec2hex(10e6,8);
tx_samp=[0 0 hex2dec('24') hex2dec('f0') hex2dec(samp_rate(7:8)) hex2dec(samp_rate(5:6))
hex2dec(samp_rate(3:4)) hex2dec(samp_rate(1:2))];
fwrite(ctrl_link,tx_samp,'uint8');
%% send tx freq set cmd
freq_hex=dec2hex(1090e6,8);
tx_freq=[0 0 hex2dec('17') hex2dec('f0') hex2dec(freq_hex(7:8)) hex2dec(freq_hex(5:6))
hex2dec(freq_hex(3:4)) hex2dec(freq_hex(1:2))];
fwrite(ctrl_link,tx_freq,'uint8');
%% send tx vga set cmd
tx_vga=[hex2dec('0') hex2dec('00') hex2dec('19') hex2dec('f0') 0 0 hex2dec('00') hex2dec('0b')];
fwrite(ctrl_link,tx_vga,'uint8');

%% send handshake cmd
handshake=[2 0 hex2dec('16') hex2dec('f0') 0 0 0 0];
fwrite(ctrl_link ,handshake, 'uint8');
pause(0.5);

%% main
fid1=fopen('F:\matlab_work\tone_16bit.dat','r');
```

```
txdata=fread(fid1,'int16');
txd1=(txdata<0)*65536+txdata;
txd2=dec2hex(txd1,4);
txd3=txd2(:,1:2);
txd4=txd2(:,3:4);
txd5=hex2dec(txd3);
txd6=hex2dec(txd4);
txd7=zeros(length(txd6)*2,1);
txd7(1:2:end)=txd6;
txd7(2:2:end)=txd5;
fclose('all');

%% Write data to the zing and read from the host.
fwrite(data_link,txd7,'uint8');

%% close all link
fclose(data_link);
delete(data_link);
clear data_link;
fclose(ctrl_link);
delete(ctrl_link);
clear ctrl_link;
```

Note: The above codes are the reference code.