Ettus UHD Research Project

Extended from Ettus Research example code txrx_loopback_to_file

Cheng-en, Sung

Department of Computer Science Engineering University of Yuan ze

26th June, 2017

Ettus UHD Research Project

Cheng-en, Sung

Flow chart

Implementation

variable declare setup po

parse command line and map with option create usrp device detect/parsing which channel to use

lock mdboard and setup TX/RX subdev check TX/RX rate, TX freq

config rx channel create tx streamer read file in binary

etup spb, md heck Ref and

neck Ref and LC ock eset USRP time

start transmit worke receiver

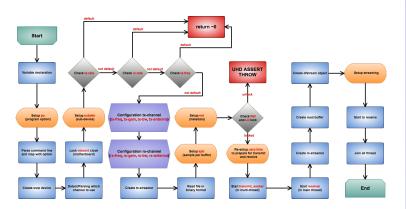
create RX streamer, read buffer, ofstream object

setup streaming start to receive

Result

tx data result

Flow chart



Ettus UHD Research Project

Cheng-en, Sung

Flow chart

create RX streamer.





Variable declare

```
235 //transmit variables to be set by po
236 std::string tx_args, tx_ant, tx_subdev, ref, otw, tx_channels, tx_data;
237 double tx_rate, tx_freq, tx_gain, tx_bw;
238 float ampl:
240 //receive variables to be set by po
241 std::string rx_args, file, type, rx_ant, rx_subdev, rx_channels;
242 size t total num samps, spb:
   double rx_rate, rx_freq, rx_gain, rx_bw;
244 float settling;
```

Ettus UHD Research Project

Cheng-en, Sung

variable declare

create RX streamer. setup streaming



239

243

Setup po (program options)

```
246 //setup the program options
247 po::options description desc("Allowed options"):
248 desc.add options()
249
        ("help", "help message")
250
        ("tx-args", po::value<std::string>(&tx_args)->default_value(""), "
             uhd transmit device address args")
251
        ("rx-args", po::value<std::string>(&rx_args)->default_value(""), "
             uhd receive device address args")
        ("file", po::value<std::string>(&file)->default_value("usrp_samples.
             dat"), "name of the file to write binary samples to")
        ("type", po::value < std::string > (&type) -> default_value ("short"), "
             sample type in file: double, float, or short")
254
255
        . . .
257
        ("otw", po::value<std::string>(&otw)->default value("sc16"), "
             specify the over-the-wire sample mode")
258
        ("tx-channels", po::value < std::string > (&tx_channels) -> default_value (
             "0"). "which TX channel(s) to use (specify \"0\". \"1\".
             \"0.1\", etc)")
        ("rx-channels", po::value < std::string > (&rx_channels) -> default_value (
             "0"). "which RX channel(s) to use (specify \"0\". \"1\".
             \"0,1\", etc)")
260
        ("tx-int-n", "tune USRP TX with integer-N tuning")
261
        ("rx-int-n", "tune USRP RX with integer-N tuning")
262
        ("tx-data", po::value < std::string > (&tx data), "transmit data")
```

Ettus UHD Research Project

Cheng-en, Sung

Flow chart

Incolous content con

riable declare

setup po

arse command line may with option reate usrp device etect/parsing which hannel to use bock mdboard and etup TX/RX subdev heck TX/RX rate, X freq onlig tx channel onlig tx channel reate tx streamer ead file in binary

etup spb, md neck Ref and LO

reset USRP time start transmit worker receiver create RX streamer.

object
setup streaming

5 1.

lata result



263 ;

Parse command line and map with option

279 //parse command line and map with option

po::variables_map vm;

po::notifv(vm):

```
po::store(po::parse_command_line(argc, argv, desc), vm);
```

Ettus UHD Research Project

Cheng-en, Sung

parse command line and map with option

create RX streamer. setup streaming





Create usrp device

Ettus UHD Research Project

Cheng-en, Sung

Flow char

Implementation

setup po parse command line

and map with option

create usrp device

detect/parsing which channel to use lock mdboard and setup TX/RX subdev check TX/RX rate.

onfig rx channel reate tx streamer and file in binary ormat

heck Ref and LO ock eset USRP time

reset USRP time start transmit worker, receiver create RX streamer.

object setup streaming

. Is

Result



Detect/Parsing which channel to use

```
298 //detect/parsing which channels to use
299 //default chnnel: 0. input channel foramt(0.1.2 or 0\1\2).
300
   std::vector<std::string> tx_channel_strings;
301
   std::vector < size_t > tx_channel_nums;
302
   boost::split(tx_channel_strings, tx_channels, boost::is_any_of("\"',"));
303
   for(size_t ch = 0; ch < tx_channel_strings.size(); ch++){
304
        size_t chan = boost::lexical_cast < int > (tx_channel_strings[ch]);
305
       if(chan >= tx usrp->get tx num channels()){
306
            throw std::runtime error("Invalid TX channel(s) specified."):
307
308
       else tx channel nums.push back(boost::lexical cast<int>(
             tx channel strings[ch]):
309 }
   std::vector<std::string> rx_channel_strings;
   std::vector<size t> rx channel nums:
   boost::split(rx_channel_strings, rx_channels, boost::is_any_of("\"',"));
   for(size_t ch = 0; ch < rx_channel_strings.size(); ch++){</pre>
314
        size t chan = boost::lexical cast<int>(rx channel strings[ch]);
       if(chan >= rx_usrp->get_rx_num_channels()){
           throw std::runtime_error("Invalid RX channel(s) specified.");
317
       }
       else rx channel nums.push back(boost::lexical cast<int>(
             rx_channel_strings[ch]));
319 }
```

Ettus UHD Research Project

Cheng-en, Sung

Flow chart

mplementation

parse command line and map with option

detect/parsing which channel to use lock mdboard and

lock maboard and setup TX/RX subdev check TX/RX rate, TX freq config tx channel config rx channel

read file in binary format setup spb, md check Ref and LO lock

reset USRP time start transmit worker receiver create RX streamer.

object
setup streaming

5 1.

data result data result



Lock mboard and Setup TX/RX subdev

```
//Lock mboard(mother-board) clocks
tx_usrp->set_clock_source(ref);
rx_usrp->set_clock_source(ref);

// setup tx/rx subdev if any of these parameter is given.

if (vm.count("tx-subdev")) tx_usrp->set_tx_subdev_spec(tx_subdev);

if (vm.count("rx-subdev")) rx_usrp->set_rx_subdev_spec(rx_subdev);
```

Ettus UHD Research Project

Cheng-en, Sung

Flow chart

variable declare setup po parse command lin and map with opti create usrp device

channel to use lock mdboard and setup TX/RX subdev check TX/RX rate,

check TX/RX rate,
TX freq
config tx channel
config rx channel
create tx streamer
read file in binary

up spb, md
ck Ref and LO
ct USRP time

create RX streamer, read buffer, ofstream object setup streaming

S Is

data result data result



Check TX/RX rate, TX freq

```
332 //check the transmit sample rate
333 if (not vm.count("tx-rate")){
334
       std::cerr << "Please specify the transmit sample rate with --tx-rate
             " << std::endl;
335
       return ~0;
336 }
337 std::cout << boost::format("Setting TX Rate: %f Msps...") % (tx_rate/1e6
         ) << std::endl;
338 tx_usrp->set_tx_rate(tx_rate);
339 std::cout << boost::format("Actual TX Rate: %f Msps...") % (tx usrp->
         get_tx_rate()/1e6) << std::endl << std::endl;</pre>
340
341 //check the receive sample rate
342 if (not vm.count("rx-rate")){
343
       std::cerr << "Please specify the sample rate with --rx-rate" << std
             ::endl:
344
       return ~0:
345 }
346 std::cout << boost::format("Setting RX Rate: %f Msps...") % (rx_rate/1e6
         ) << std::endl:
347 rx_usrp->set_rx_rate(rx_rate);
348 std::cout << boost::format("Actual RX Rate: %f Msps...") % (rx usrp->
         get rx rate()/1e6) << std::endl << std::endl:</pre>
349
350 //check the transmit center frequency
351 if (not vm.count("tx-freq")){
352
       std::cerr << "Please specify the transmit center frequency with --tx</pre>
             -freq" << std::endl;
353
       return ~0:
354 }
```

Ettus UHD Research Project

Cheng-en, Sung

Flow char

nplementation
variable declare
vetup po
varse command line
and map with option
vereate usrp device
letect/parsing which
channel to use
ock mdboard and

check TX/RX rate, TX freq config tx channel config rx channel create tx streamer

etup spb, md heck Ref and LO ock eset USRP time

start transmit worker receiver create RX streamer.

object setup streaming start to receive

. 1 .

Config TX channel

```
356 //config each tx channel
357 for(size t ch = 0: ch < tx channel nums.size(): ch++) {
358
       size_t channel = tx_channel_nums[ch];
359
       if (tx_channel_nums.size() > 1) {
360
            std::cout << "Configuring TX Channel " << channel << std::endl;</pre>
361
       }
362
       std::cout << boost::format("Setting TX Freq: %f MHz...") % (tx_freq
             /1e6) << std::endl:
363
       uhd::tune request t tx tune request(tx freq):
364
       if (vm.count("tx-int-n")) tx_tune_request.args = uhd::device_addr_t("
             mode n=integer"):
365
       tx_usrp->set_tx_freq(tx_tune_request, channel);
366
       std::cout << boost::format("Actual TX Freq: %f MHz...") % (tx_usrp->
             get_tx_freq(channel)/1e6) << std::endl << std::endl;</pre>
367
       if (vm.count("tx-gain")){
368
            std::cout << boost::format("Setting TX Gain: %f dB...") %
                 tx_gain << std::endl;</pre>
369
            tx usrp->set tx gain(tx gain, channel):
            std::cout << boost::format("Actual TX Gain: %f dB...") % tx usrp
                 ->get_tx_gain(channel) << std::endl << std::endl;
       if (vm.count("tx-bw")){
            std::cout << boost::format("Setting TX Bandwidth: %f MHz...") %
                 tx_bw << std::endl;
374
            tx_usrp->set_tx_bandwidth(tx_bw, channel);
            std::cout << boost::format("Actual TX Bandwidth: %f MHz...") %
                 tx_usrp->get_tx_bandwidth(channel) << std::endl << std::</pre>
                 endl:
376
       if (vm.count("tx-ant")) tx_usrp->set_tx_antenna(tx_ant, channel);
378 }
```

Ettus UHD Research Project

Cheng-en, Sung

Flow char

nplementation

setup po parse command line and map with option create usrp device detect/parsing which channel to use lock mdboard and setup TX/PX subdev

config tx channel

reate tx streamer ead file in binary ormat etup spb, md

ck set USRP time

t transmit wor

create RX streamer, read buffer, ofstream object setup streaming

join all th

Recul



```
Ettus UHD
Research Project
```

Cheng-en, Sung

```
config rx channel
```

start transmit worke receiver create RX streamer,

object setup streaming

join all threa

Result

```
for(size t ch = 0: ch < rx channel nums.size(): ch++) {
387
       size_t channel = rx_channel_nums[ch];
       if (rx channel nums.size() > 1) {
388
389
            std::cout << "Configuring RX Channel " << channel << std::endl:
390
       }
391
392
       //set the receive center frequency
393
       if (not vm.count("rx-freq")){
394
            std::cerr << "Please specify the center frequency with --rx-freq
                 " << std::endl:
395
            return ~0:
397
       std::cout << boost::format("Setting RX Freq: %f MHz...") % (rx_freq</pre>
             /1e6) << std::endl:
398
       uhd::tune_request_t rx_tune_request(rx_freq);
399
       if(vm.count("rx-int-n")) rx_tune_request.args = uhd::device_addr_t("
             mode_n=integer");
       rx_usrp->set_rx_freq(rx_tune_request, channel);
400
       std::cout << boost::format("Actual RX Freq: %f MHz...") % (rx_usrp->
401
             get rx freq(channel)/1e6) << std::endl << std::endl:</pre>
402
403
       //set the receive rf gain
       if (vm.count("rx-gain")){
404
405
            std::cout << boost::format("Setting RX Gain: %f dB...") %
                 rx_gain << std::endl;
406
            rx_usrp->set_rx_gain(rx_gain, channel);
407
            std::cout << boost::format("Actual RX Gain: %f dB...") % rx usrp
                 ->get_rx_gain(channel) << std::endl << std::endl;
408
       }
```

Config RX channel (cont.)

```
//set the receive analog frontend filter bandwidth
       if (vm.count("rx-bw")){
411
412
            std::cout << boost::format("Setting RX Bandwidth: %f MHz...") %</pre>
                 (rx_bw/1e6) << std::endl;
413
            rx_usrp->set_rx_bandwidth(rx_bw, channel);
            std::cout << boost::format("Actual RX Bandwidth: %f MHz...") % (
414
                 rx_usrp->get_rx_bandwidth(channel)/1e6) << std::endl << std
                 ::endl;
415
416 }
417 //set the receive antenna
418 if (vm.count("ant")) rx usrp->set rx antenna(rx ant):
```

Ettus UHD Research Project

Cheng-en, Sung

Flow char

Implementation variable declare

parse command line and map with option create usrp device detect/parsing which channel to use lock mdboard setup TX/RX subdev

> eate tx streamer and file in binary armat stup spb, md neck Ref and LO ck

config rx channel

reset USRP time start transmit worker receiver

object setup streaming start to receive

Result

data result data result



Create TX streamer

Ettus UHD Research Project

Cheng-en, Sung

```
create ty streamer
```

create RX streamer.

setup streaming

```
421 //create a transmit streamer
422 /*
423
       otw stands for "over-the-wire sample mode"
424
       Setting the OTW ("over-the-wire") format is, in theory, transparent
             to the application, but changing this can have some side
             effects.
       Using less bits for example (e.g. when going from otw_format sc16 to
              sc8) will reduce the dynamic range, and increases quantization
              noise.
       On the other hand, it reduces the load on the data link and thus
             allows more bandwidth (a USRP N210 can work with 25 MHz
             bandwidth for 16-Bit complex samples, and 50 MHz for 8-Bit
             complex samples).
       The following otw format have been implemented:
        * sc16 (default using)
        * sc12
430
        * sc8
432 */
433 uhd::stream_args_t stream_args("fc32", otw);
434
   stream_args.channels = tx_channel_nums;
435
   // Get the max number of samples per buffer per packet.
436
   uhd::tx_streamer::sptr tx_stream = tx_usrp->get_tx_stream(stream_args);
```



425

426

427

431



* read file function

60 61

63

64

65 66

67

68

69

70

71 72

74

76

80

81

83

84

85

86 87 88

receiver

create RX streamer.

```
vector < std::complex <double > read file(string tx data)
  {
       std::vector < std::complex <double > > buff;
       std::ifstream is:
       is.open (tx_data, std::ios::binary | std::ios::in);
       is.seekg (0, std::ios::end);
       int rlen = is.tellg():
       is.seekg (0, std::ios::beg);
       char *rbuff = new char [rlen]:
       is.read (rbuff, rlen);
       is.close();
       double* pbuff = (double*)rbuff:
       std::vector<double> tmp_buff(pbuff, pbuff + (rlen / sizeof(double)))
       double real = 0.0, imag = 0.0:
       for ( int i = 0; i < tmp_buff.size(); i++) {</pre>
           if (i \% 2 == 0) {
               real = tmp buff[i]:
           } else {
               imag = tmp_buff[i];
           std::complex <double > cp(real, imag);
           buff.push_back(cp);
       retrun buff:
89 }
```

Setup spb (sample per buffer), md (metadata)

```
Ettus UHD
Research Project
```

Cheng-en, Sung

setup spb, md

create RX streamer.

```
449 //setup sample per buffer
450 if (spb == 0) spb = tx_stream -> get_max_num_samps()*10; // ?
451
   int num channels = tx channel nums.size();
452
453 //setup the metadata flags
454 uhd::tx_metadata_t md;
455 md.start of burst = true:
456 md.end_of_burst
                      = false;
457
   md.has_time_spec
                    = true;
458
   md.time_spec = uhd::time_spec_t(0.1); //give us 0.1 seconds to fill the
         ty buffers
```





Check Ref and LO lock

```
460 //Check Ref and LO Lock detect
461 std::vector<std::string> tx sensor names. rx sensor names:
462 tx sensor names = tx usrp->get tx sensor names(0):
463 if (std::find(tx_sensor_names.begin(), tx_sensor_names.end(), "lo_locked
         ") != tx_sensor_names.end()) {
464
       uhd::sensor value t lo locked = tx usrp->get tx sensor("lo locked"
465
       std::cout << boost::format("Checking TX: %s ...") % lo_locked.
             to pp string() << std::endl:
466
       UHD ASSERT THROW(lo locked.to bool()):
467 }
468 rx_sensor_names = rx_usrp->get_rx_sensor_names(0);
469 if (std::find(rx sensor names.begin(), rx sensor names.end(), "lo locked
        ") != rx_sensor_names.end()) {
470
       uhd::sensor_value_t lo_locked = rx_usrp->get_rx_sensor("lo_locked"
             .0):
471
       std::cout << boost::format("Checking RX: %s ...") % lo_locked.
             to_pp_string() << std::endl;
       UHD ASSERT THROW(lo locked.to bool()):
472
473 }
474 tx_sensor_names = tx_usrp->get_mboard_sensor_names(0);
475
476 if ((ref == "mimo") and (std::find(tx sensor names.begin().
         tx_sensor_names.end(), "mimo_locked") != tx_sensor_names.end())) {
477
       uhd::sensor_value_t mimo_locked = tx_usrp->get_mboard_sensor("
             mimo locked".0):
       std::cout << boost::format("Checking TX: %s ...") % mimo_locked.</pre>
478
             to_pp_string() << std::endl;
479
       UHD ASSERT THROW(mimo locked.to bool()):
480 }
```

Ettus UHD Research Project

Cheng-en, Sung

check Ref and I O

lock

create RX streamer.



Check Ref and LO lock (cont.)

```
482 if ((ref == "external") and (std::find(tx_sensor_names.begin(),
        tx_sensor_names.end(), "ref_locked") != tx_sensor_names.end())) {
483
       uhd::sensor value t ref locked = tx usrp->get mboard sensor("
             ref_locked",0);
484
       std::cout << boost::format("Checking TX: %s ...") % ref_locked.</pre>
             to pp string() << std::endl:
485
       UHD ASSERT THROW(ref locked.to bool()):
486 }
487
488 rx_sensor_names = rx_usrp->get_mboard_sensor_names(0);
489 if ((ref == "mimo") and (std::find(rx_sensor_names.begin(),
         rx_sensor_names.end(), "mimo_locked") != rx_sensor_names.end())) {
       uhd::sensor value t mimo locked = rx usrp->get mboard sensor("
490
             mimo_locked",0);
491
       std::cout << boost::format("Checking RX: %s ...") % mimo_locked.
             to pp string() << std::endl:
       UHD ASSERT THROW(mimo locked.to bool()):
492
493 }
494 if
      ((ref == "external") and (std::find(rx sensor names.begin().
         rx sensor names.end(), "ref locked") != rx sensor names.end())) {
495
       uhd::sensor_value_t ref_locked = rx_usrp->get_mboard_sensor("
             ref_locked",0);
       std::cout << boost::format("Checking RX: %s ...") % ref_locked.</pre>
496
             to_pp_string() << std::endl;
497
       UHD_ASSERT_THROW(ref_locked.to_bool());
498 }
```

Ettus UHD Research Project

Cheng-en, Sung

Flow char

mplementation

arse command line and map with option eate usrp device etect/parsing which hannel to use ck mdboard and

eate tx streamer ad file in binary rmat

check Ref and LO lock

reset USRP time start transmit worker, receiver create RX streamer.

object setup streaming start to receive

Docul+



Reset USRP time

Ettus UHD Research Project

Cheng-en, Sung

reset USRP time

create RX streamer.

setup streaming

510 //reset usrp time to prepare for transmit/receive 511 std::cout << boost::format("Setting device timestamp to 0...") << std:: endl: 512 tx_usrp->set_time_now(uhd::time_spec_t(0.0));

Start transmit worker, recevier

```
514 //start transmit worker thread
515 boost::thread_group transmit_thread;
transmit_thread.create_thread(boost::bind(&transmit_worker, buff,
         tx stream. md. num channels)):
517
518 //recv to file
519 if (type == "double") recy to file < std::complex < double > > (rx usrp, "fc64
         ", otw, file, spb, total_num_samps, settling, rx_channel_nums);
520 else if (type == "float") recv_to_file <std::complex <float >> (rx_usrp,
         fc32", otw, file, spb, total_num_samps, settling, rx_channel_nums);
   else if (type == "short") recv_to_file <std::complex <short> >(rx_usrp,
         sc16", otw, file, spb, total_num_samps, settling, rx_channel_nums);
522 else {
523
       //clean up transmit worker
524
       stop_signal_called = true;
525
       transmit_thread.join_all();
526
       throw std::runtime error("Unknown type " + type):
527 }
```

Ettus UHD Research Project

Cheng-en, Sung

start transmit worker. receiver

create RX streamer.



Create RX streamer, read buffer, ofstream object

Ettus UHD Research Project

Cheng-en, Sung

create RX streamer.

read buffer, ofstream object setup streaming

```
145 //create a receive streamer
146 uhd::stream_args_t stream_args(cpu_format,wire_format);
147
   stream args.channels = rx channel nums:
148
   uhd::rx streamer::sptr rx stream = usrp->get rx stream(stream args):
149
150
   // Prepare buffers for received samples and metadata
   uhd::rx metadata t md:
   std::vector <std::vector < samp_type > > buffs(
       rx_channel_nums.size(), std::vector< samp_type >(samps_per_buff)
154
   ):
155
   //create a vector of pointers to point to each of the channel buffers
156
   std::vector<samp_type *> buff_ptrs;
   for (size t i = 0: i < buffs.size(): i++) {
       buff ptrs.push back(&buffs[i].front());
159 }
```





Setup streaming

```
//setup streaming

uhd::stream_cmd_t stream_cmd((num_requested_samples == 0)?

uhd::stream_cmd_t::STREAM_MODE_START_CONTINUOUS:

uhd::stream_cmd_t::STREAM_MODE_NUM_SAMPS_AND_DONE

177 );

stream_cmd.num_samps = num_requested_samples;

stream_cmd.stream_now = false;

stream_cmd.time_spec = uhd::time_spec_t(settling_time);

178 tream_sisue_stream_cmd(stream_cmd):
```

Ettus UHD Research Project

Cheng-en, Sung

Flow char

Implementation

setup po parse command line and map with option create usrp device detect/parsing which channel to use

ock mdboard and etup TX/RX subdev heck TX/RX rate, X freq onfig tx channel onfig rx channel reate tx streamer

etup spb, md neck Ref and LO

reset USRP time start transmit worker receiver create RX streamer.

object setup streaming

n all threa

Result



Start to receive

```
while (not stop signal called and (num requested samples >
     num total samps or num requested samples == 0)) {
    size_t num_rx_samps = rx_stream->recv(buff_ptrs, samps_per_buff, md,
          timeout):
    timeout = 0.1f; //small timeout for subsequent recv
      (md.error code == uhd::rx metadata t::ERROR CODE TIMEOUT) {
    }
       (md.error code == uhd::rx metadata t::ERROR CODE OVERFLOW){
    }
       (md.error_code != uhd::rx_metadata_t::ERROR_CODE NONE){
    }
    num_total_samps += num_rx_samps;
    for (size t i = 0: i < outfiles.size(): i++) {
        outfiles[i]->write((const char*) buff_ptrs[i], num_rx_samps*
             sizeof(samp type)):
    }
```

Ettus UHD Research Project

Cheng-en, Sung

Flow chart

nplementation ariable declare

area command line and map with option reate usrp device etect/parsing which nannel to use cck mdboard and

create tx streamer read file in binary format setup spb, md

reset USRP time start transmit worke

create RX streamer, read buffer, ofstream object setup streaming

start to receive join all thread

Result

resuit

data result



184

186

190

194 195 196

197

200

201

202 203 }

Join all thread

```
145 if (type == "double") recv_to_file<std::complex<double> >(rx_usrp, "fc64
         ", otw, file, spb, total_num_samps, settling, rx_channel_nums);
146 else if (type == "float") recv_to_file < std::complex < float > > (rx_usrp, "
        fc32", otw, file, spb, total_num_samps, settling, rx_channel_nums);
   else if (type == "short") recv_to_file <std::complex <short> >(rx_usrp,
        sc16", otw. file. spb. total num samps, settling, rx channel nums);
148
   else {
149
       //clean up transmit worker
       stop_signal_called = true;
       transmit_thread.join_all();
       throw std::runtime_error("Unknown type " + type);
153 }
```

Ettus UHD Research Project

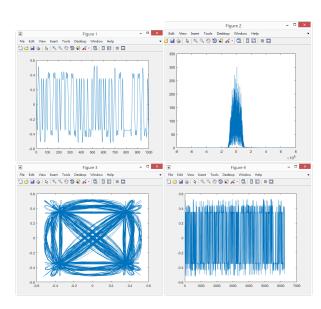
Cheng-en, Sung

create RX streamer. setup streaming

ioin all thread



TX data result





Cheng-en, Sung

Flow char

Implementation

variable declare

parse command line and map with option create usrp device detect/parsing which channel to use lock mdboard and

check TX/RX ra TX freq config tx channel

reate tx strear ead file in bina ormat

check Ref and L lock

reset USRP time start transmit worker receiver

create RX streamer, read buffer, ofstream

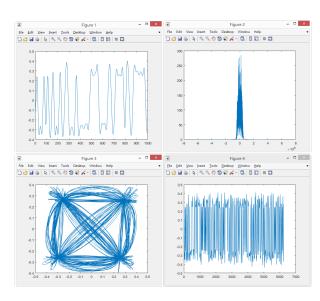
setup streaming start to receive

Result

tx data result rx data result



RX data result





Cheng-en, Sung

Flow char

Implementation

variable declare

parse command line and map with option create usrp device detect/parsing which channel to use lock mdboard and setup TX/RX subdev

onfig tx channe onfig rx channe eate tx stream

ead file in bina ormat etup spb. md

heck Ref and L ock

reset USRP time start transmit worke receiver

create RX streamer, read buffer, ofstream

setup streaming start to receive

Result

tx data result rx data result



Reference

- ► Getting Started with UHD and C++ https://kb.ettus.com/Getting_Started_with_UHD_and_C++
- EttusResearch project https://github.com/EttusResearch/uhd/
- ► USRP Hardware Driver and USRP Manual https://files.ettus.com/manual/
- ► EttusResearch example code https://githubċom/EttusResearch/uhd/txrx_loopback_to_file.cpp
- ► Source code link http://codepad.org/iQoh4MCZ

Ettus UHD Research Project

Cheng-en, Sung

Flow chart

Implementation

pariable declare setup po parse command line and map with option create usrp device detect/parsing which channel to use ock mdboard and setup TX/RX subdev

reate tx streame ead file in binary ormat etup spb, md

tup spb, ma leck Ref and Li ck

tart transmit worke eceiver

setup streaming start to receive

Result

rx data resul

