Benchmarking

mmW Instrument timing measurements

30 April 2020, Vuk Obradovic, Peter Vago @ NOFFZ Technlogies

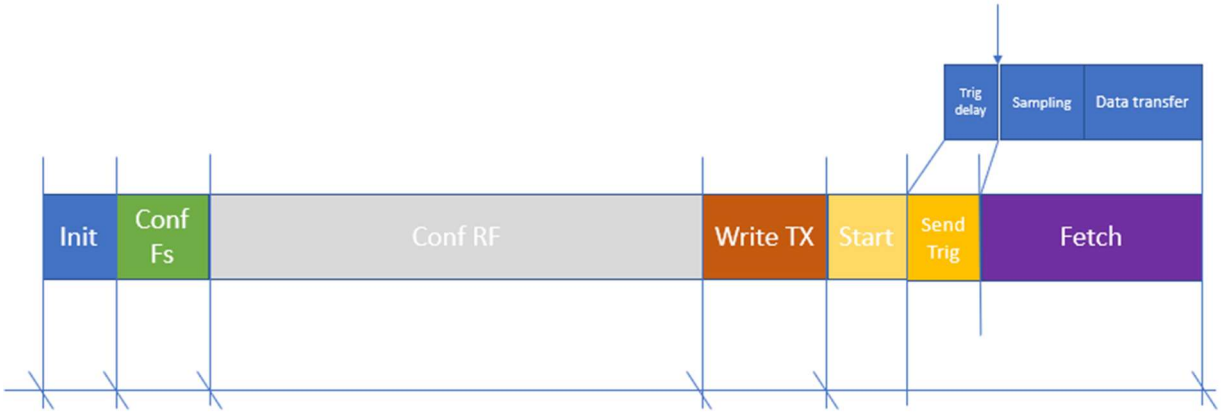
Contents

Timing benchmarks

Benchmarking a full session (singe session)

Following diagram shows the command durations that were measured.

Performance – measuring command durations

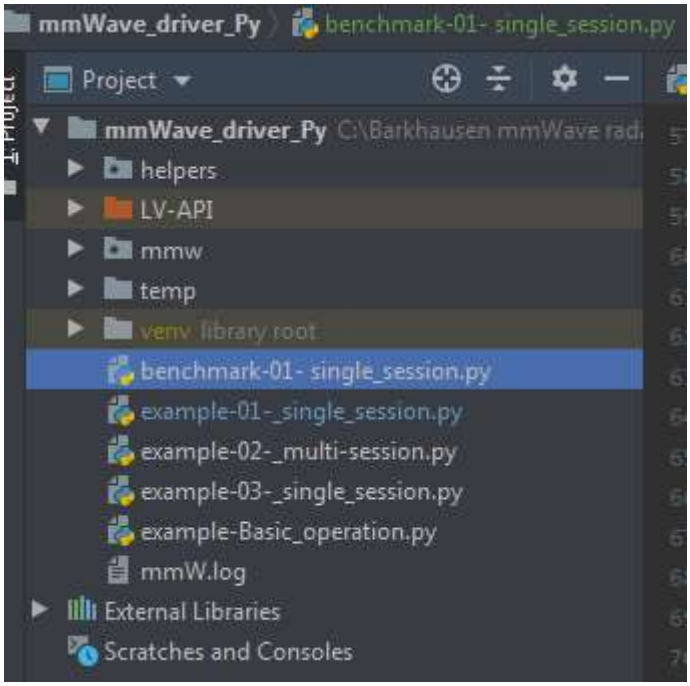


Python code was running on Sys-B , instrument was the Sys-A. Data was sent through 1Gbit LAN connection.

Data generation on “user PC” and plotting or any kind of data processing are excluded form the measurement.

All measurements were repeated several times. Reported durations are average values.

Python code for testing (main function only):




```
def main():
    hostA="192.168.40.126"
    portA=5555
    hostB="192.168.40.171"
    hostB="127.0.0.1"
```

```
    portB=5555

    global instr
    i = mmw.ni_mmw(hostA, portA)
    instr = i

    for k in range(0,1):
        # Generate Waveforms
        wfm = helpers.waveform_helpers()
        waveform, attribs = wfm.get_waveform(31e5)  # 1ms@3GSps -> 66e5 pts ->
~13MB

        perf.reset("main")
        i.initialize_hw(mmw.const.instr_hwclass.NI_mmW01, mmw.const.opmodes.RF)
        perf.record_diff("main","01_init_hw")
        fs = 3.072e9
        i.configure_fs(fs, mmw.const.ports.rx) # Set Fs, can be set to all
        i.configure_fs(fs, mmw.const.ports.tx) # Set Fs, can be set to all
        perf.record_diff("main","02_conf_fs")

        fc = 75e9  # 76.2 GHz
        i.configure_rf(fc, -10, mmw.const.ports.tx) # gain = -10dB
        i.configure_rf(fc, 10, mmw.const.ports.rx) #gain = 10dB
        perf.record_diff("main","03_conf_rf")

        #ins.s04_conf_sync(instr)       # Synhronization

        #ins.s05_equalization(instr)    # Equalization

        i.write_tx("iqsine2", waveform)  # Download waveform to instrument
        perf.record_diff("main","04_write_tx-1ms")

        i.start(["iqsine2"], 800e-6, 15000)  # burst_name, length in s (10e-6 :
10us)
        perf.record_diff("main","05_start")

        # Trigger burst(s)
        r,data = i._depr_trigger_burst("iqsine", mmw.const.burst_mode.burst)  # 1x
iqsine + mode
        perf.record_diff("main","06_trigger_burst")
        print("Data received: %d"%len(data))

        #print(instr.send_trigger())
        #response , data = instr.fetch() # data is a tuple (data, type<str>)

        #Stop
        i.stop()
        perf.record_abs("main","07_fulltime")



        # ====================
        print("--- END ---")
        print("====== performance data: %s "%(str(perf.read("main"))))
```

Result of full session (incl RF config):

[0]====== performance data:	01_init_hw: 9.16 ms	02_conf_fs: 16.04 ms	03_conf_rf: 1162.89 ms	04_write_tx-1ms: 159.46 ms	05_start: 148.48 ms	06_trigger_burst: 308.74 ms	07_fulltime: 1811.06 ms
[1]====== performance data:	01_init_hw: 9.07 ms	02_conf_fs: 17.63 ms	03_conf_rf: 1159.51 ms	04_write_tx-1ms: 159.15 ms	05_start: 147.65 ms	06_trigger_burst: 302.71 ms	07_fulltime: 1801.90 ms
[2]====== performance data:	01_init_hw: 8.85 ms	02_conf_fs: 16.67 ms	03_conf_rf: 1160.00 ms	04_write_tx-1ms: 158.28 ms	05_start: 147.04 ms	06_trigger_burst: 301.58 ms	07_fulltime: 1798.64 ms
[3]====== performance data:	01_init_hw: 8.83 ms	02_conf_fs: 16.27 ms	03_conf_rf: 1161.23 ms	04_write_tx-1ms: 159.49 ms	05_start: 146.44 ms	06_trigger_burst: 306.72 ms	07_fulltime: 1804.94 ms
[4]====== performance data:	01_init_hw: 9.30 ms	02_conf_fs: 16.80 ms	03_conf_rf: 1162.06 ms	04_write_tx-1ms: 158.04 ms	05_start: 144.78 ms	06_trigger_burst: 303.69 ms	07_fulltime: 1800.91 ms
[5]====== performance data:	01_init_hw: 8.92 ms	02_conf_fs: 16.49 ms	03_conf_rf: 1158.72 ms	04_write_tx-1ms: 158.71 ms	05_start: 145.74 ms	06_trigger_burst: 305.89 ms	07_fulltime: 1800.70 ms
[6]====== performance data:	01_init_hw: 9.19 ms	02_conf_fs: 16.62 ms	03_conf_rf: 1168.07 ms	04_write_tx-1ms: 159.03 ms	05_start: 146.99 ms	06_trigger_burst: 303.31 ms	07_fulltime: 1809.82 ms
[7]====== performance data:	01_init_hw: 8.78 ms	02_conf_fs: 16.70 ms	03_conf_rf: 1167.20 ms	04_write_tx-1ms: 158.13 ms	05_start: 148.84 ms	06_trigger_burst: 303.55 ms	07_fulltime: 1809.73 ms
[8]====== performance data:	01_init_hw: 8.68 ms	02_conf_fs: 16.37 ms	03_conf_rf: 1166.61 ms	04_write_tx-1ms: 160.03 ms	05_start: 148.83 ms	06_trigger_burst: 300.73 ms	07_fulltime: 1807.17 ms
[9]====== performance data:	01_init_hw: 8.92 ms	02_conf_fs: 16.31 ms	03_conf_rf: 1159.88 ms	04_write_tx-1ms: 159.34 ms	05_start: 147.67 ms	06_trigger_burst: 303.08 ms	07_fulltime: 1801.97 ms

Excluding RF config:

Only 1 out of 2 RF config command excluded

```
fc = 75e9  # 76.2 GHz
#i.configure_rf(fc, -10, mmw.const.ports.tx) # gain = -10dB
i.configure_rf(fc, 10, mmw.const.ports.rx) #gain = 10dB
perf.record_diff("main","03_conf_rf")
```

[0]====== performance data:	01_init_hw: 9.00 ms	02_conf_fs: 16.21 ms	03_conf_rf: 579.57 ms	04_write_tx-1ms: 161.05 ms	05_start: 148.25 ms	06_trigger_burst: 308.26 ms	07_fulltime: 1228.97 ms
[1]====== performance data:	01_init_hw: 8.87 ms	02_conf_fs: 16.80 ms	03_conf_rf: 584.60 ms	04_write_tx-1ms: 159.52 ms	05_start: 149.10 ms	06_trigger_burst: 303.85 ms	07_fulltime: 1228.72 ms
[2]====== performance data:	01_init_hw: 8.70 ms	02_conf_fs: 16.40 ms	03_conf_rf: 579.36 ms	04_write_tx-1ms: 158.95 ms	05_start: 146.84 ms	06_trigger_burst: 302.87 ms	07_fulltime: 1218.81 ms
[3]====== performance data:	01_init_hw: 8.91 ms	02_conf_fs: 16.85 ms	03_conf_rf: 581.52 ms	04_write_tx-1ms: 158.67 ms	05_start: 147.66 ms	06_trigger_burst: 303.39 ms	07_fulltime: 1223.76 ms
[4]====== performance data:	01_init_hw: 8.95 ms	02_conf_fs: 16.69 ms	03_conf_rf: 584.22 ms	04_write_tx-1ms: 159.31 ms	05_start: 147.32 ms	06_trigger_burst: 302.52 ms	07_fulltime: 1225.10 ms
[5]====== performance data:	01_init_hw: 9.05 ms	02_conf_fs: 16.50 ms	03_conf_rf: 580.58 ms	04_write_tx-1ms: 158.24 ms	05_start: 148.45 ms	06_trigger_burst: 301.67 ms	07_fulltime: 1221.02 ms
[6]====== performance data:	01_init_hw: 9.54 ms	02_conf_fs: 16.32 ms	03_conf_rf: 577.81 ms	04_write_tx-1ms: 158.41 ms	05_start: 144.61 ms	06_trigger_burst: 301.77 ms	07_fulltime: 1214.30 ms
[7]====== performance data:	01_init_hw: 9.22 ms	02_conf_fs: 17.01 ms	03_conf_rf: 582.41 ms	04_write_tx-1ms: 157.50 ms	05_start: 146.68 ms	06_trigger_burst: 303.52 ms	07_fulltime: 1223.20 ms
[8]====== performance data:	01_init_hw: 8.19 ms	02_conf_fs: 15.14 ms	03_conf_rf: 577.48 ms	04_write_tx-1ms: 158.34 ms	05_start: 147.13 ms	06_trigger_burst: 306.66 ms	07_fulltime: 1219.47 ms
[9]====== performance data:	01_init_hw: 9.27 ms	02_conf_fs: 16.17 ms	03_conf_rf: 585.22 ms	04_write_tx-1ms: 157.99 ms	05_start: 145.50 ms	06_trigger_burst: 301.57 ms	07_fulltime: 1222.15 ms

Both RF config command excluded

```
fc = 75e9  # 76.2 GHz
#i.configure_rf(fc, -10, mmw.const.ports.tx) # gain = -10dB
#i.configure_rf(fc, 10, mmw.const.ports.rx) #gain = 10dB
perf.record_diff("main","03_conf_rf")
```

[0]====== performance data:	01_init_hw: 9.36 ms	02_conf_fs: 17.97 ms	03_conf_rf: 0.00 ms	04_write_tx-1ms: 157.39 ms	05_start: 150.02 ms	06_trigger_burst: 314.53 ms	07_fulltime: 655.86 ms
[1]====== performance data:	01_init_hw: 9.04 ms	02_conf_fs: 16.58 ms	03_conf_rf: 0.00 ms	04_write_tx-1ms: 152.15 ms	05_start: 146.74 ms	06_trigger_burst: 302.50 ms	07_fulltime: 633.69 ms
[2]====== performance data:	01_init_hw: 8.99 ms	02_conf_fs: 15.81 ms	03_conf_rf: 0.00 ms	04_write_tx-1ms: 153.47 ms	05_start: 146.05 ms	06_trigger_burst: 299.15 ms	07_fulltime: 629.31 ms
[3]====== performance data:	01_init_hw: 8.34 ms	02_conf_fs: 16.08 ms	03_conf_rf: 0.00 ms	04_write_tx-1ms: 153.52 ms	05_start: 150.03 ms	06_trigger_burst: 303.34 ms	07_fulltime: 637.03 ms
[4]====== performance data:	01_init_hw: 9.06 ms	02_conf_fs: 16.74 ms	03_conf_rf: 0.00 ms	04_write_tx-1ms: 153.69 ms	05_start: 147.61 ms	06_trigger_burst: 305.10 ms	07_fulltime: 638.17 ms
[5]====== performance data:	01_init_hw: 9.28 ms	02_conf_fs: 16.25 ms	03_conf_rf: 0.00 ms	04_write_tx-1ms: 154.71 ms	05_start: 148.53 ms	06_trigger_burst: 301.35 ms	07_fulltime: 636.53 ms
[6]====== performance data:	01_init_hw: 9.08 ms	02_conf_fs: 16.37 ms	03_conf_rf: 0.00 ms	04_write_tx-1ms: 156.87 ms	05_start: 150.24 ms	06_trigger_burst: 303.11 ms	07_fulltime: 641.49 ms
[7]====== performance data:	01_init_hw: 8.83 ms	02_conf_fs: 16.30 ms	03_conf_rf: 0.00 ms	04_write_tx-1ms: 157.63 ms	05_start: 150.82 ms	06_trigger_burst: 304.23 ms	07_fulltime: 643.75 ms
[8]====== performance data:	01_init_hw: 9.03 ms	02_conf_fs: 16.98 ms	03_conf_rf: 0.00 ms	04_write_tx-1ms: 153.07 ms	05_start: 146.85 ms	06_trigger_burst: 304.30 ms	07_fulltime: 636.64 ms
[9]====== performance data:	01_init_hw: 8.89 ms	02_conf_fs: 16.23 ms	03_conf_rf: 0.00 ms	04_write_tx-1ms: 157.70 ms	05_start: 152.62 ms	06_trigger_burst: 300.74 ms	07_fulltime: 642.31 ms

Multi-session - Testing with 1ms TX and 1ms RX data

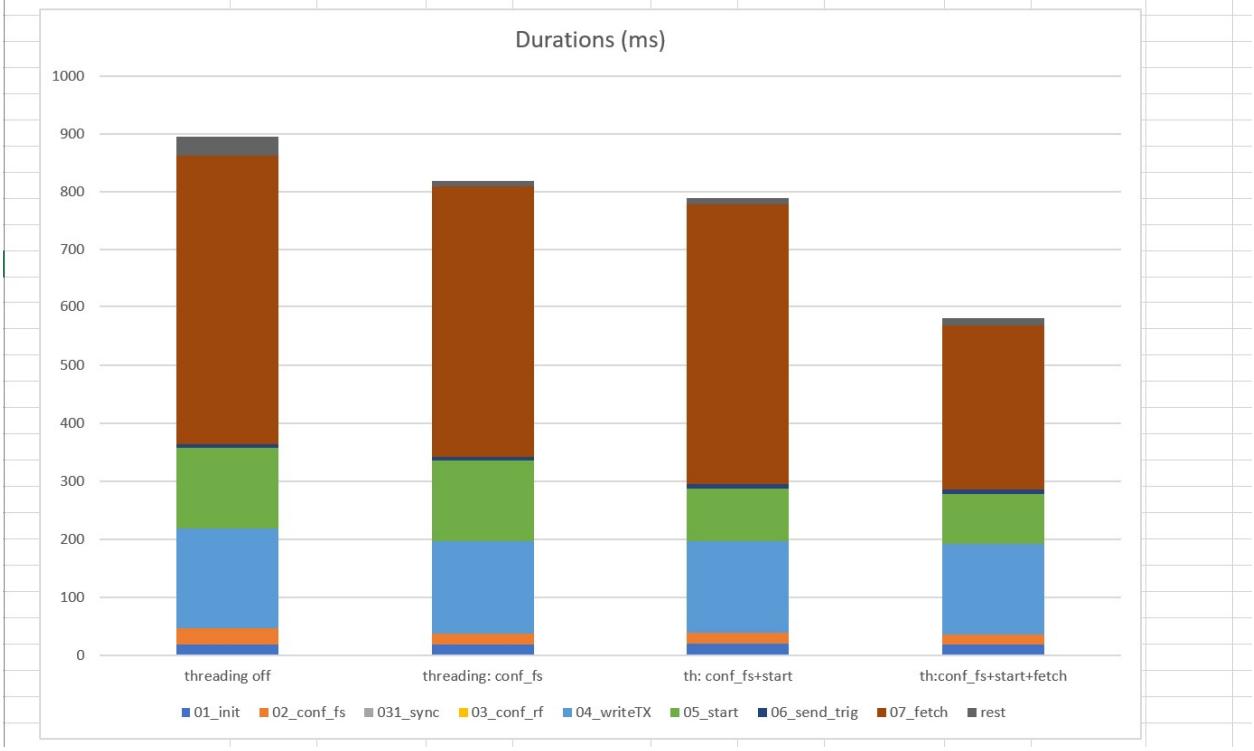
First we did not use any multithreading and we got ~900ms for the full session duration (without RF-reconfiguration, streamed 1ms waveforms to TX and from both RX channels.)

Then we started to introduce multi-threaded steps for configure_fs(), start() and fetch() functions.

Note: previously used trigger_burst() is now deprecated and we use send_trigger() and fetch() command-pair for that to make it more clear.

Finally, using threading the previously measured 900ms dropped down to **580-600ms**.

(all in ms)	01_init	02_conf_fs	031_sync	03_conf_rf	04_writeT	05_start	06_send_trig	07_fetch	rest	**08_fulltime**
threading off	19	28	0	0	171	139	7	499	33	**896**
threading: conf_fs	19	18	0	0	159	140	6	468	10	**820**
th: conf_fs+start	20	19	0	0	158	90	7	484	11	**789**
th:conf_fs+start+fetch	19	17	0	0	156	86	7	283	13	**581**



Multi-threaded commands:

```
fs = 3.072e9
fs = fs/1
# Single-threaded configure_fs
#iA.configure_fs(fs, mmw.const.ports.rx) # Set Fs, can be set to all
#iA.configure_fs(fs, mmw.const.ports.tx) # Set Fs, can be set to all
#iB.configure_fs(fs, mmw.const.ports.rx) # Set Fs, can be set to all
# Multi-threaded configure_fs
conf_fs_list = [(iA, fs, mmw.const.ports.rx),(iA, fs, mmw.const.ports.tx),(iB,
fs/2, mmw.const.ports.rx)]
responses = mmw.configure_fs_multi(conf_fs_list)
perf.record_diff("main","02_conf_fs")
```

```
length = 1000e-6 # 1ms
# Signle-threaded start()
#iA.start([wfmname], length, 5000)  # burst_name, length in s (10e-6 : 10us)
#iB.start([wfmname], length, 5000)  # burst_name, length in s (10e-6 : 10us)
# Multi-threaded start()
start_arg_list=[(iA,[wfmname], length, 5000), (iB, [wfmname], length, 5000)]
mmw.start_multi(start_arg_list)
perf.record_diff("main","05_start")
```

```
# Single-threaded fetch()
#r , data = iA.fetch() # data is a tuple (data, type<str>)
#r , data = iB.fetch() # data is a tuple (data, type<str>)
```

```
# Multi-threaded fetch()
r, data = mmw.fetch_multi([(iA,5000),(iB,5000)])
for resp, d in r, data:
    print(resp) # {'Errcode': 'OK', 'Command': 'fetch', 'Parameters':
{'generic_info': '', 'datatype': <datatypes.interleaved_I16: 2>, 'source': ''}}
    print("Data length: %d"% len(d)) # Data length: 3072000
perf.record_diff("main","07_fetch")
```

Conclusion

Timing fits to the requirements (1 second).

Notes:

- fits without RF front-end re-configuration (fs, fs, gain, synchronization properties).
- write_tx() function should accept Double data too (now it accepts I16 array).
  In this case DBL -> I16 conversion should be taken into account (first benchmarks showed 100ms processing time for 1ms waveform length, /tested on i7-3650 CPU/)

IF and BB testing

// This section is not part of the timing report. We only tested the functionality.

mmW instrument has three opertation modes: RF, IF, BB (RF, Intermediate frequency, Baseband)

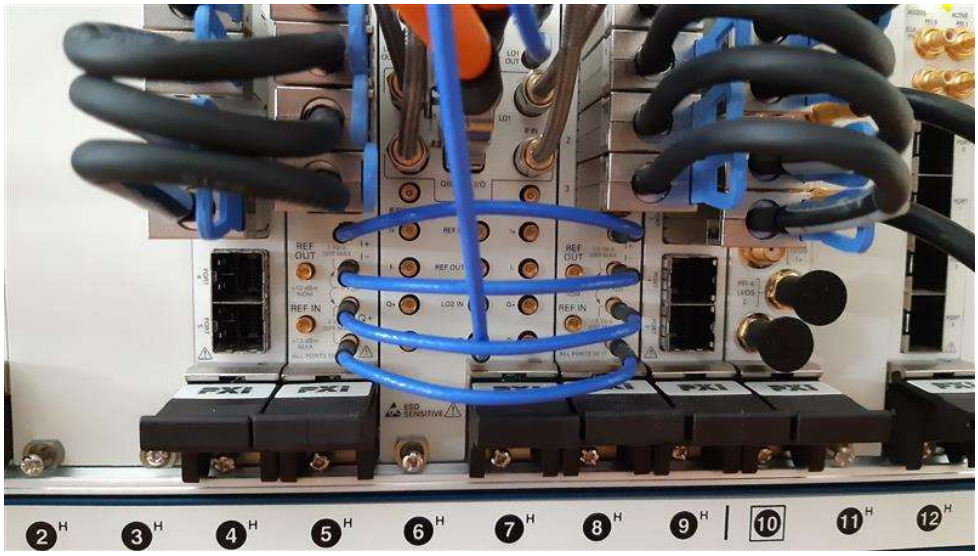
Op.mode can be selected at initialization (initialize_hw command) and running this command takes appr. 36-38 seconds.

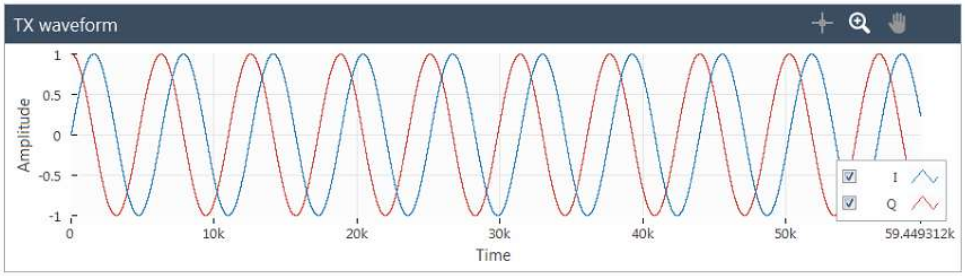
IF

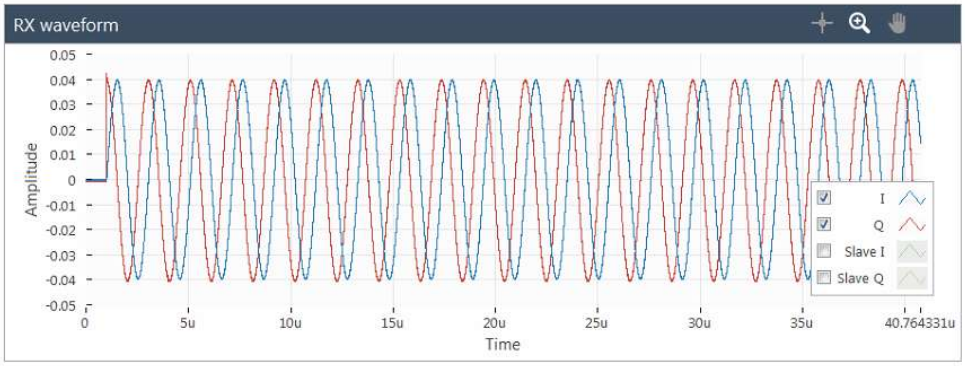
```
i.initialize_hw(mmw.const.instr_hwclass.NI_mmW01, mmw.const.opmodes.IF , 60000) # This is done automatically . Need to be called when reinit needed.
```

BB

```
iA.initialize_hw(mmw.const.instr_hwclass.NI_mmW01, mmw.const.opmodes.BB , 45000)
```







HW response

{
"system info":{
"mmw instr":
"niMmWave_Transciever",

Slave response

disconnected

Plot

Save