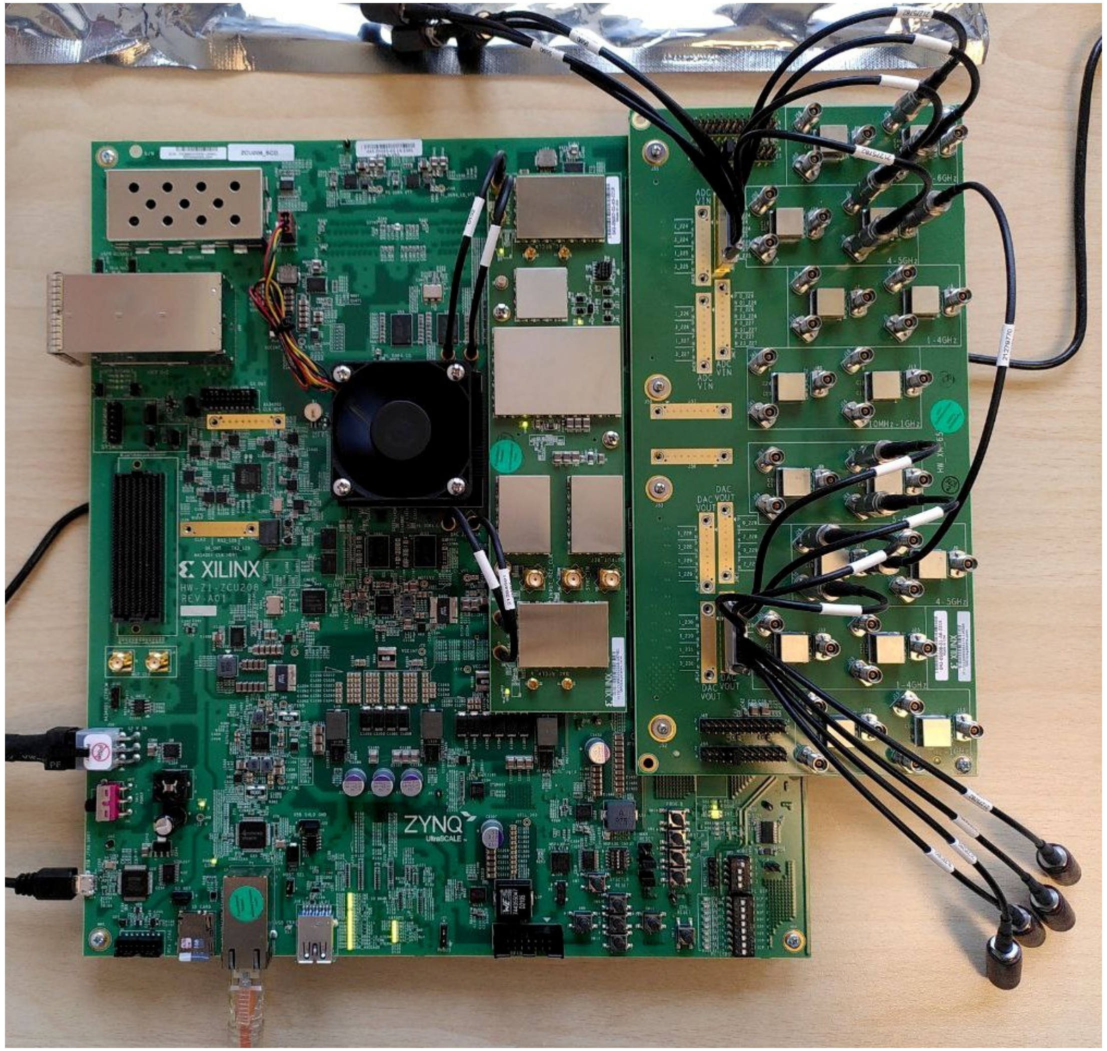
Qubit control using QICK on ZCU208 RFSoC

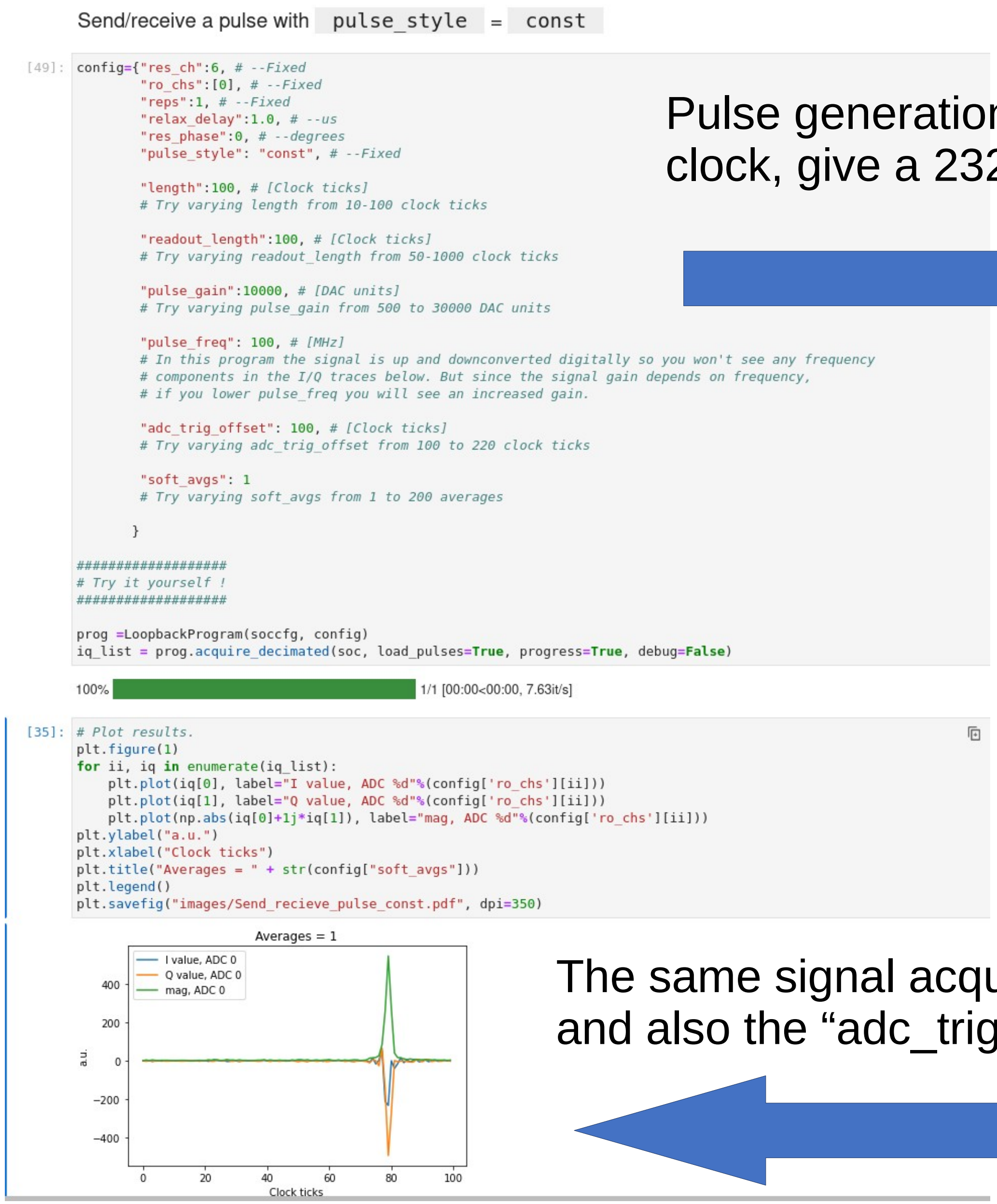


Xilinx ZCU208 RFSoC

In the context of the QUB-IT Project:

- Procurement of the ZCU208 board
- Installation of PYNQ 3.0.1
- Signals generation and reading tests with PYNQ
- QICK (0.2.165) firmware and code porting to ZCU208
  - Essential interaction with QICK developers
  - Most modifications required on the firmware side

Qubit control using QICK on ZCU208 RFSoC



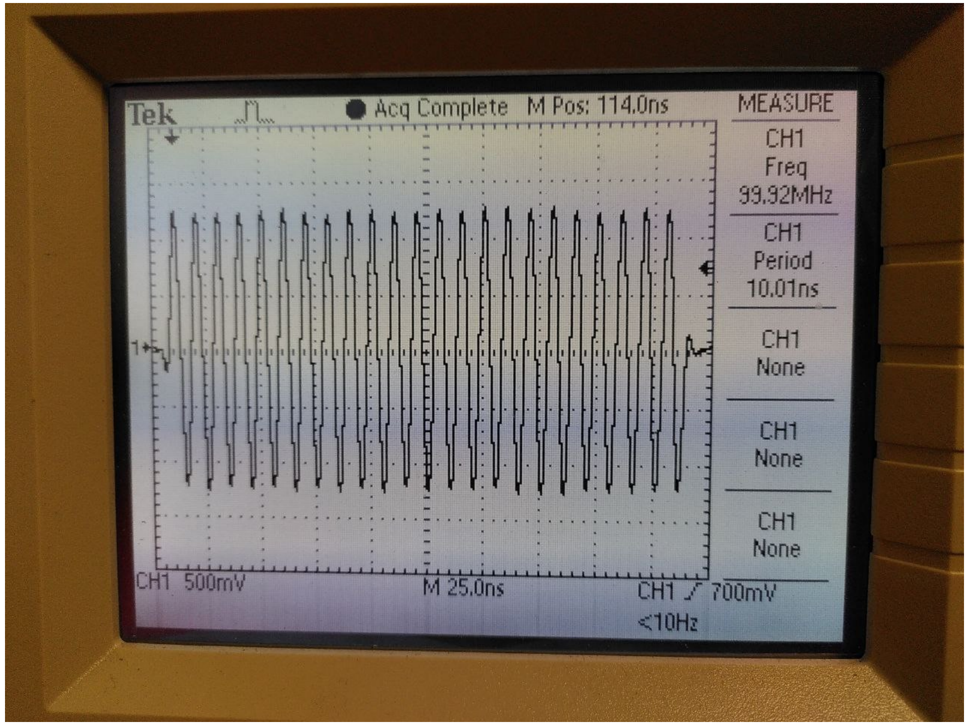
Send/receive a pulse with `pulse_style` = `const`

```
[49]: config={"res_ch":6, # --Fixed
              "ro_chs":[0], # --Fixed
              "reps":1, # --Fixed
              "relax_delay":1.0, # --us
              "res_phase":0, # --degrees
              "pulse_style": "const", # --Fixed

              "length":100, # [Clock ticks]
              # Try varying length from 10-100 clock ticks

              "readout_length":100, # [Clock ticks]
              # Try varying readout_length from 50-1000 clock ticks

              "pulse_gain":10000, # [DAC units]
              # Try varying pulse_gain from 500 to 30000 DAC units

              "pulse_freq": 100, # [MHz]
              # In this program the signal is up and downconverted digitally so you won't see any frequency
              # components in the I/Q traces below. But since the signal gain depends on frequency,
              # if you lower pulse_freq you will see an increased gain.

              "adc_trig_offset": 100, # [Clock ticks]
              # Try varying adc_trig_offset from 100 to 220 clock ticks

              "soft_avgs": 1
              # Try varying soft_avgs from 1 to 200 averages

             }

      ###################
      # Try it yourself !
      ###################

      prog =LoopbackProgram(soccfg, config)
      iq_list = prog.acquire_decimated(soc, load_pulses=True, progress=True, debug=False)
```

```
100% 1/1 [00:00<00:00, 7.63it/s]
```

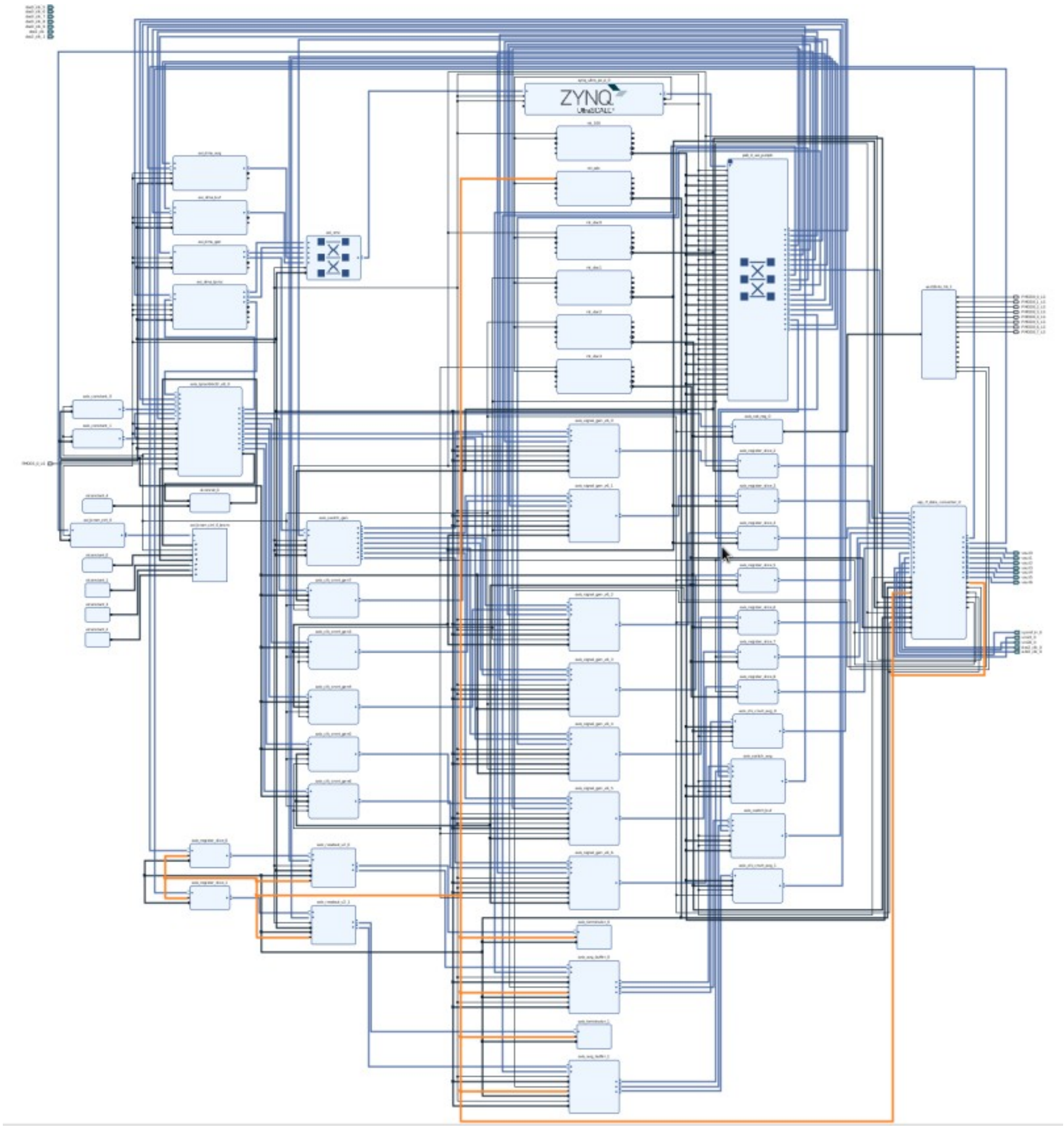
```
[35]: # Plot results.
      plt.figure(1)
      for ii, iq in enumerate(iq_list):
          plt.plot(iq[0], label="I value, ADC %d"%(config['ro_chs'][ii]))
          plt.plot(iq[1], label="Q value, ADC %d"%(config['ro_chs'][ii]))
          plt.plot(np.abs(iq[0]+1j*iq[1]), label="mag, ADC %d"%(config['ro_chs'][ii]))
      plt.ylabel("a.u.")
      plt.xlabel("Clock ticks")
      plt.title("Averages = " + str(config["soft_avgs"]))
      plt.legend()
      plt.savefig("images/Send_recieve_pulse_const.pdf", dpi=350)
```

Pulse generation seemed to work as expected: 100 clock ticks with a 430MHz clock, give a 232ns pulse length with the expected frequency (i.e. 100MHz).



The same signal acquired from ADC on the other hand seemed to be much shorter, and also the “adc_trig_offset” setting seemed not to affect the signal as expected.

Qubit control using QICK on ZCU208 RFSoC



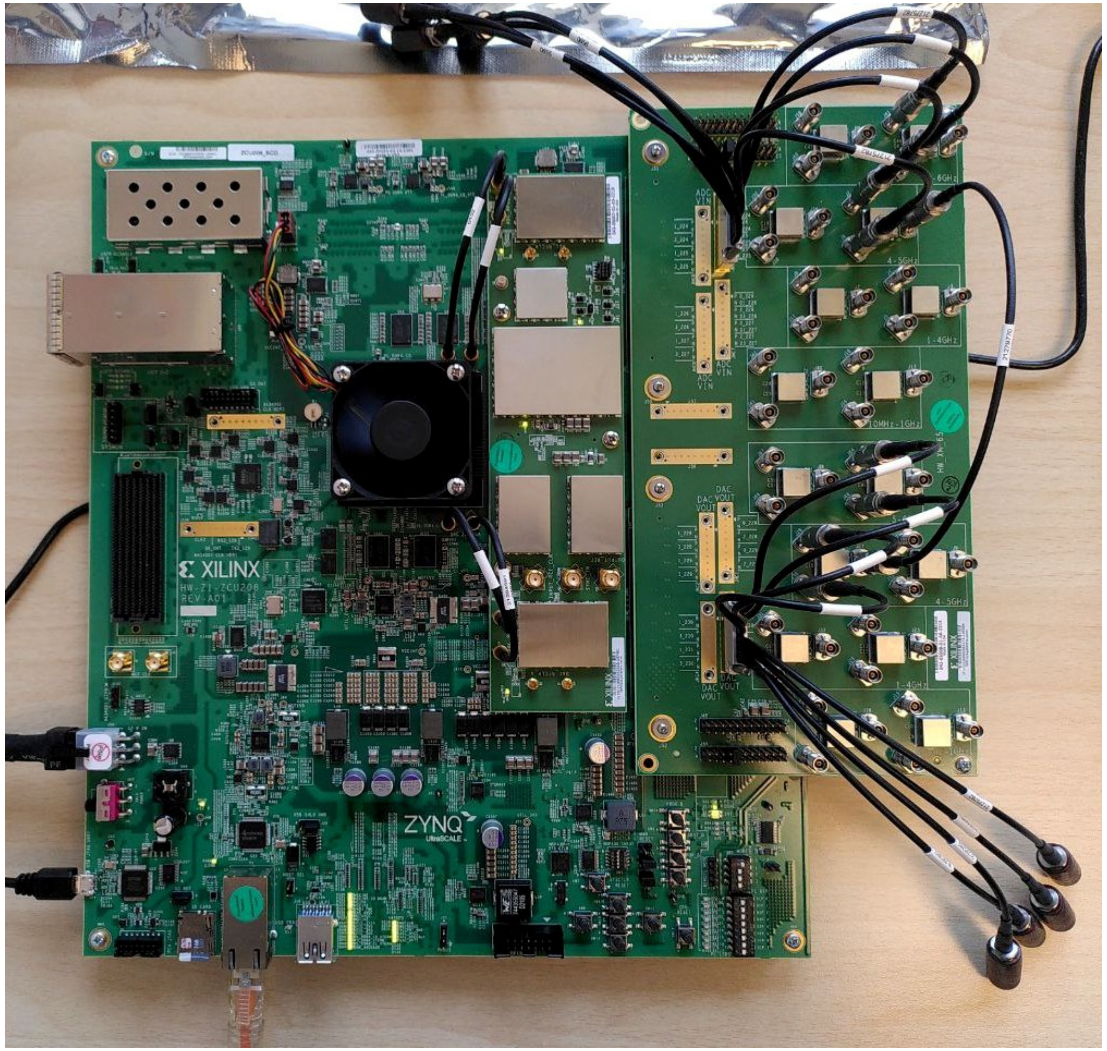
GitHub Issue describing some of the important steps for the hardware design porting:

https://github.com/openquantumhardware/qick/issues/57

Further support on the Unitary Fund QICK Discord Channel:

https://discord.com/channels/764231928676089909/1036774653479362630

Qubit control using QICK on ZCU208 RFSoC

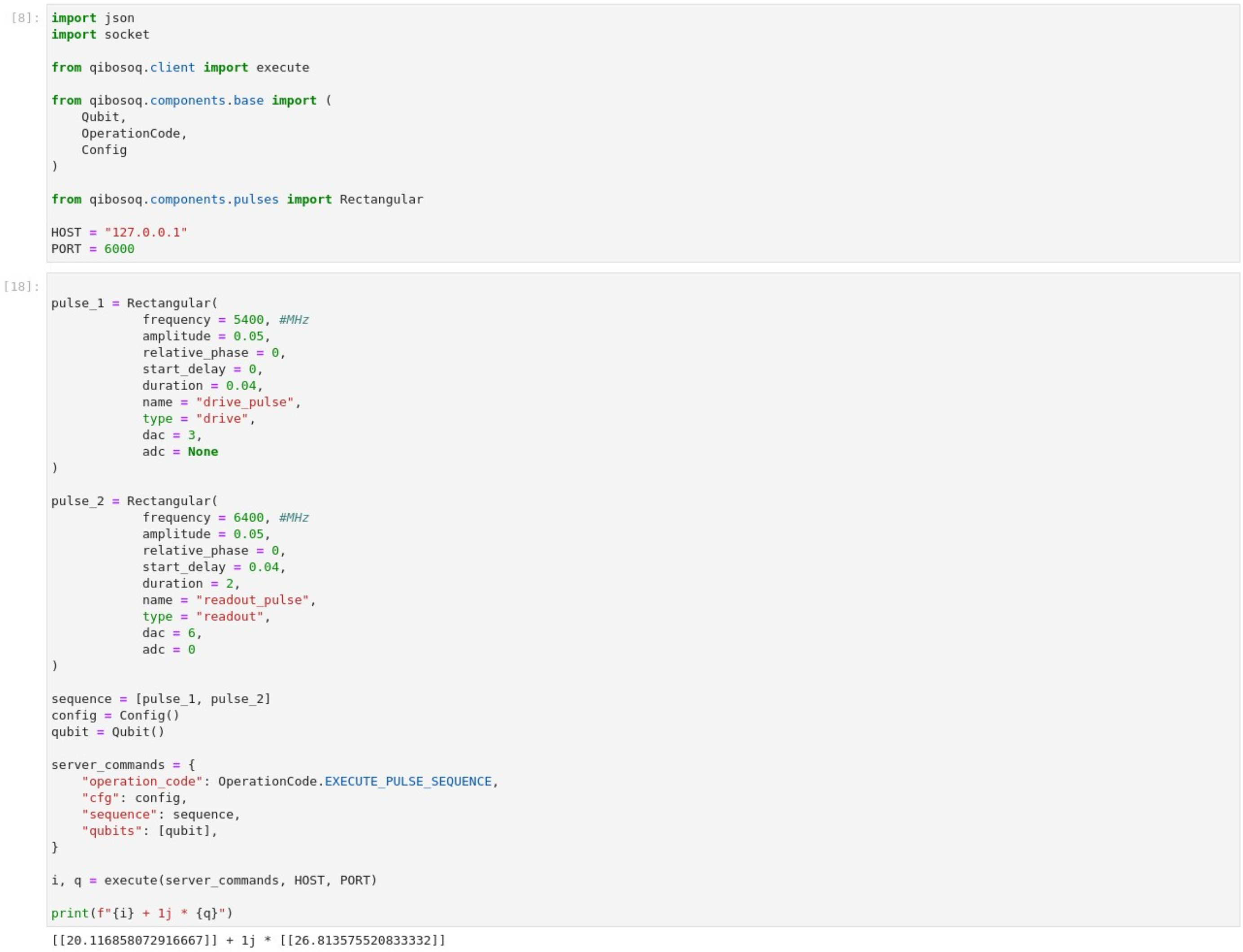


Xilinx ZCU208 RFSoC

In the context of the QUB-IT Project:

- Procurement of the ZCU208 board
- Installation of PYNQ 3.0.1
- Signals generation and reading tests with PYNQ
- QICK (0.2.165) firmware and code porting to ZCU208
    - Essential interaction with QICK developers
    - Most modifications required on the firmware side
- Integration of QiboSoq + QICK
    - Initially supporting QICK 0.2.135, but 0.2.165 installed... now solved.
    - Validation of the QiboSoq + QICK installation

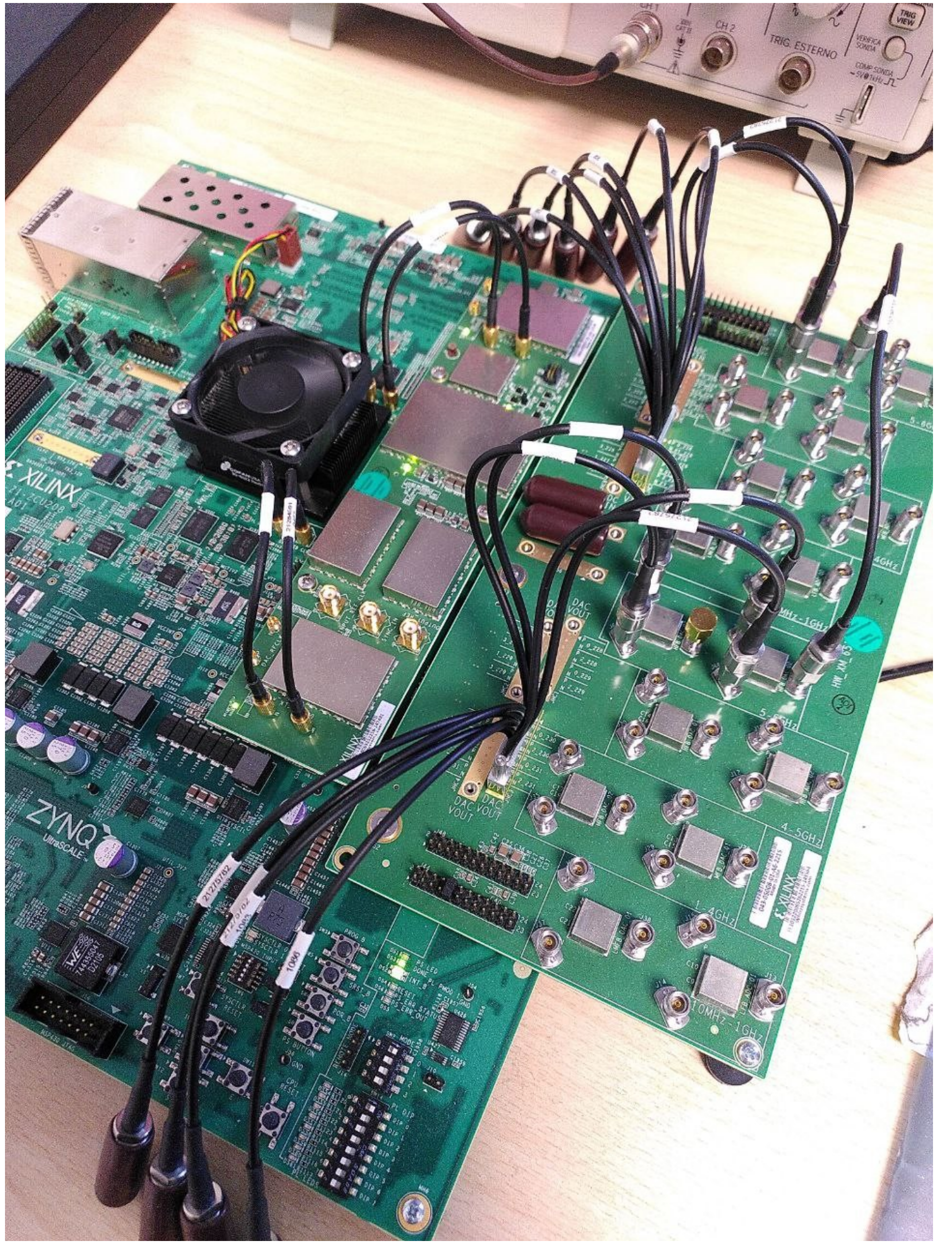
Qubit control using QICK on ZCU208 RFSoC



```
[8]:
import json
import socket

from qibosoq.client import execute

from qibosoq.components.base import (
    Qubit,
    OperationCode,
    Config
)

from qibosoq.components.pulses import Rectangular

HOST = "127.0.0.1"
PORT = 6000
```

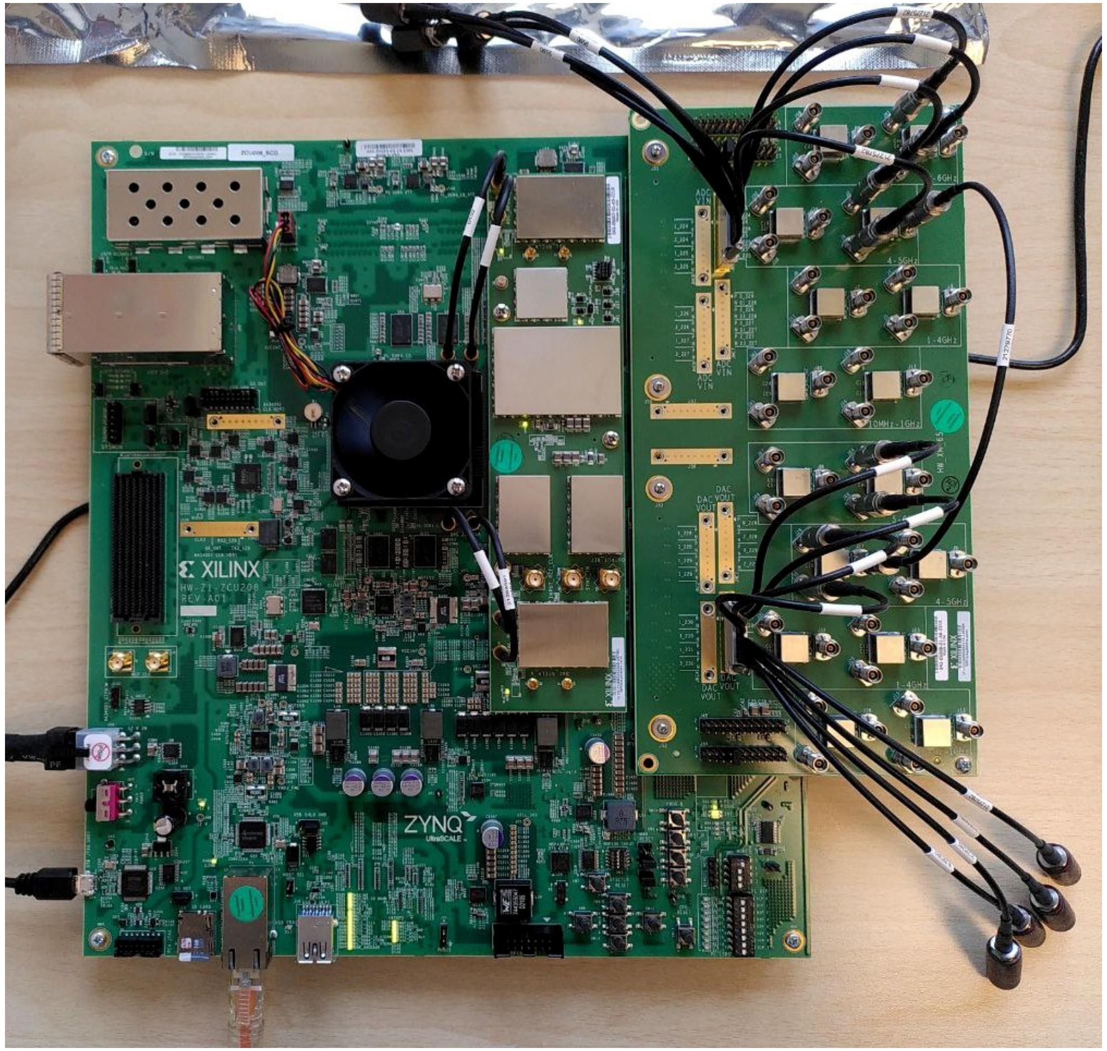
```
[18]:

pulse_1 = Rectangular(
            frequency = 5400, #MHz
            amplitude = 0.05,
            relative_phase = 0,
            start_delay = 0,
            duration = 0.04,
            name = "drive_pulse",
            type = "drive",
            dac = 3,
            adc = None
)

pulse_2 = Rectangular(
            frequency = 6400, #MHz
            amplitude = 0.05,
            relative_phase = 0,
            start_delay = 0.04,
            duration = 2,
            name = "readout_pulse",
            type = "readout",
            dac = 6,
            adc = 0
)

sequence = [pulse_1, pulse_2]
config = Config()
qubit = Qubit()

server_commands = {
    "operation_code": OperationCode.EXECUTE_PULSE_SEQUENCE,
    "cfg": config,
    "sequence": sequence,
    "qubits": [qubit],
}

i, q = execute(server_commands, HOST, PORT)

print(f"{i} + 1j * {q}")
```

```
[[20.116858072916667]] + 1j * [[26.813575520833332]]
```



Tests performed both at LNF and Ferrara

Qubit control using QICK on ZCU208 RFSoC



Xilinx ZCU208 RFSoC

In the context of the QUB-IT Project:

- Procurement of the ZCU208 board
- Installation of PYNQ 3.0.1
- Signals generation and reading tests with PYNQ
- QICK (0.2.165) firmware and code porting to ZCU208
    - Essential interaction with QICK developers
    - Most modifications required on the firmware side
- Integration of QiboSoq + QICK
    - Initially supporting QICK 0.2.135, but 0.2.165 installed... now solved.
    - Validation of the QiboSoq + QICK installation

TODO: Code cleanup and push to a git repository