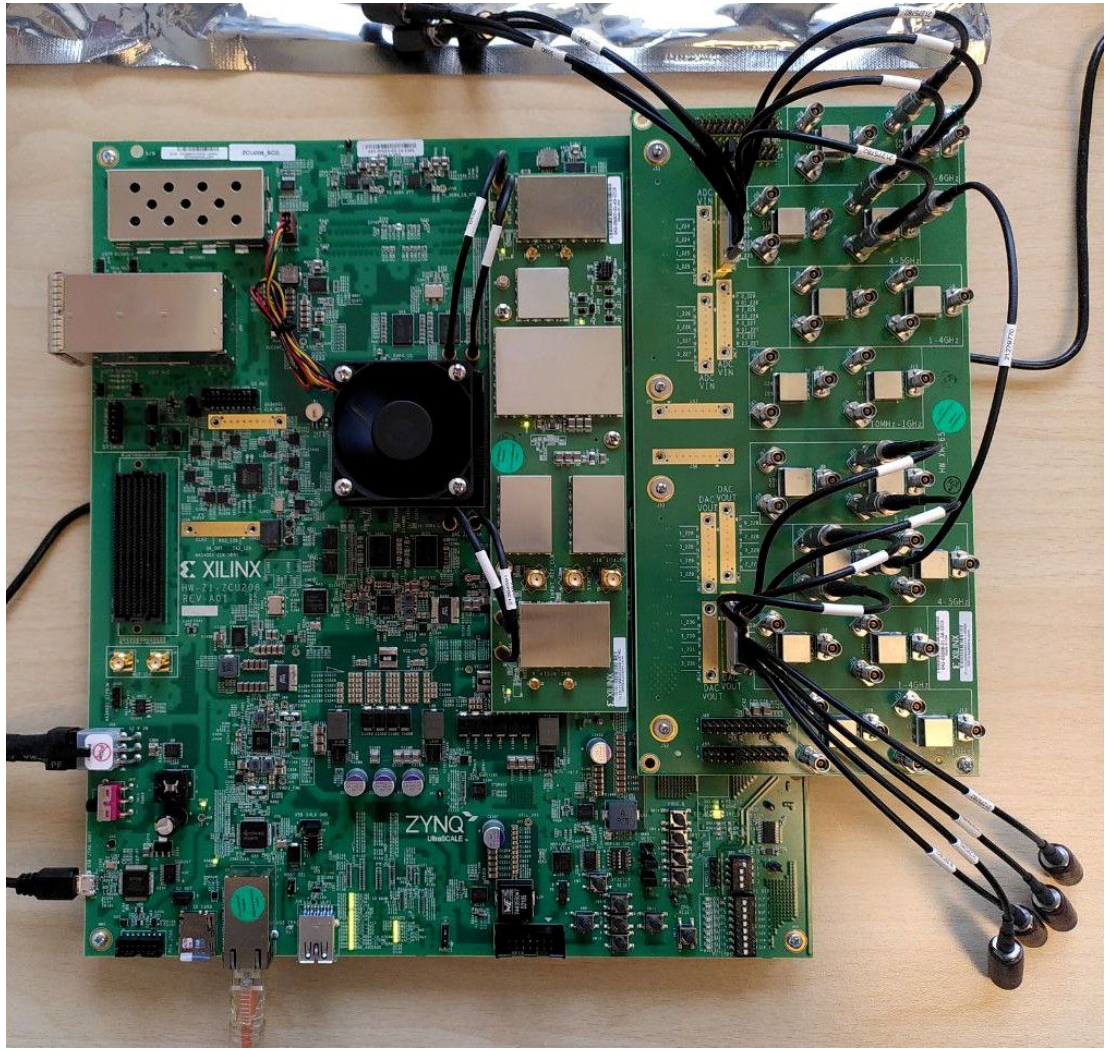
Qubit control using QICK on ZCU208 RFSoC



Xilinx ZCU208 RFSoC Installed at Ferrara

Done:

- 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

To do:

- **Validation of the QICK installation and code cleanup**
- Integration of QiboSoq + QICK
- (Support for QICK 0.2.135, but 0.2.165 installed)

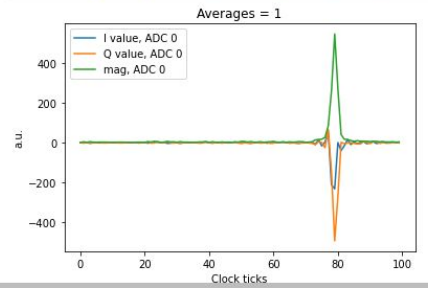
Qubit control using QICK on ZCU208 RFSoC

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

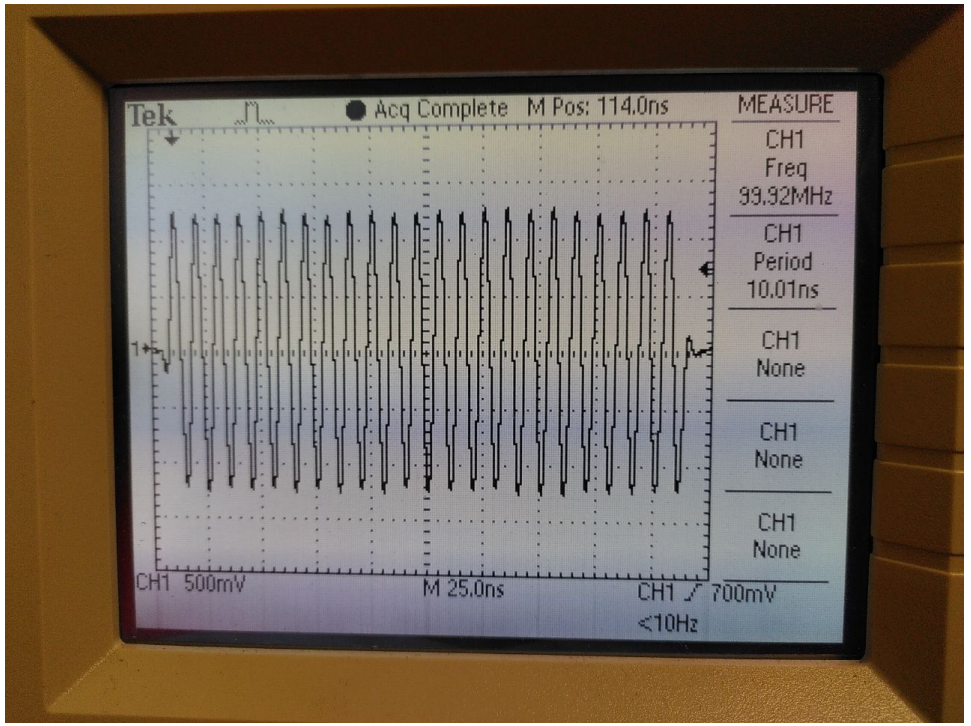
[49]:
```
config={"res_ch":6, # --Fixed
        "ro_chs":[0], # --Fixed
        "reps":1, # --Fixed
        "relax_delay":1.0, # --us
        "res_phase":0, # --degrees
        "pulse_style": "const", # --Fixed

        "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 seems 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 seems to be much shorter, and also the “adc_trig_offset” setting seems not to affect the signal as expected.