Short Topic: Bolometer DAQ Updates

P. Hacker[1,2], F. Reimold[1]

[1] Max Planck Institute for Plasma Physics, Wendelsteinstr. 1, D-17491 Greifswald, Germany

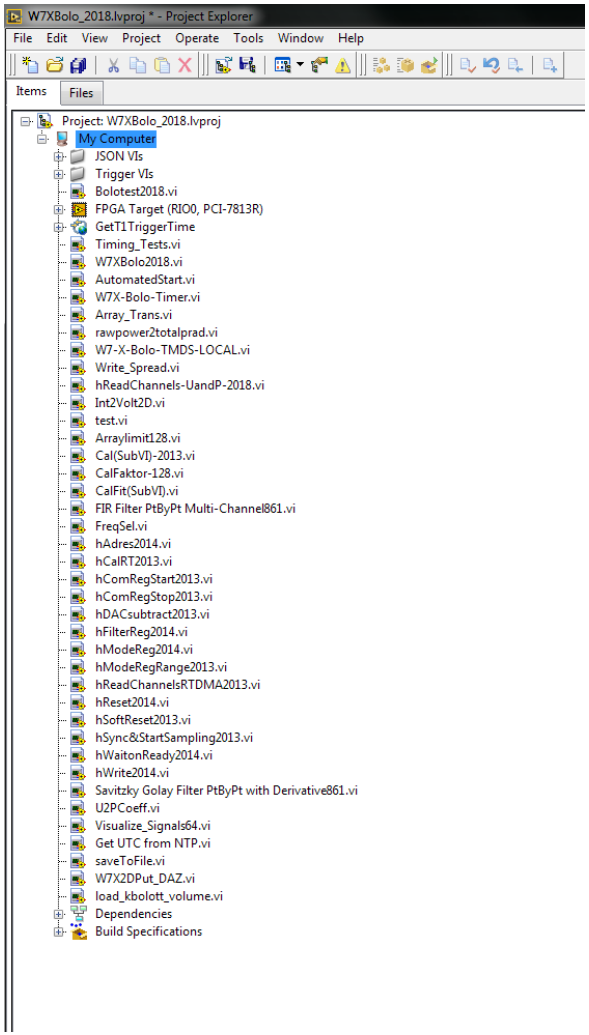
[2] Ernst-Moritz-Arndt University Greifswald, Domstr. 11, D-17489 Greifswald, Germany

July 6, 2018

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No. 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

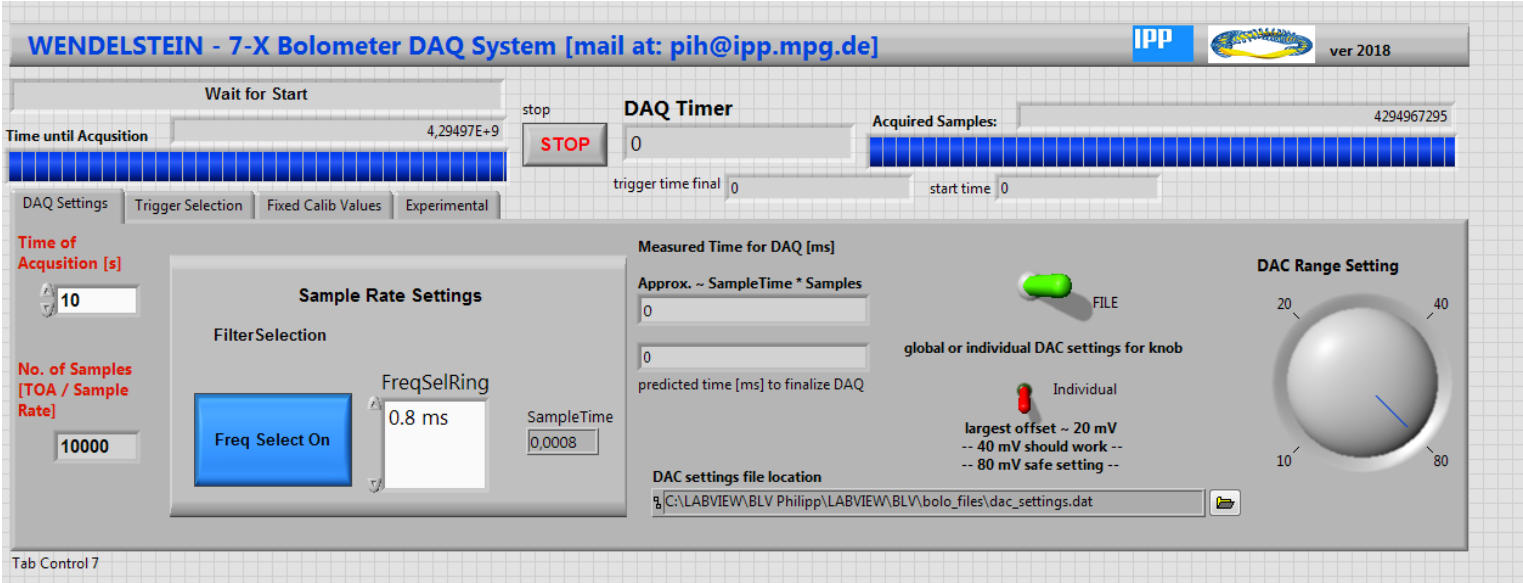
software versioning

- current and previous states of software development in git-controlled repository at `https://gitlab.mpcdf.mpg.de/pih/bolometer_labview`
- including documentation, software and hardware setup
- device manuals and supportfiles located in dedicated structures



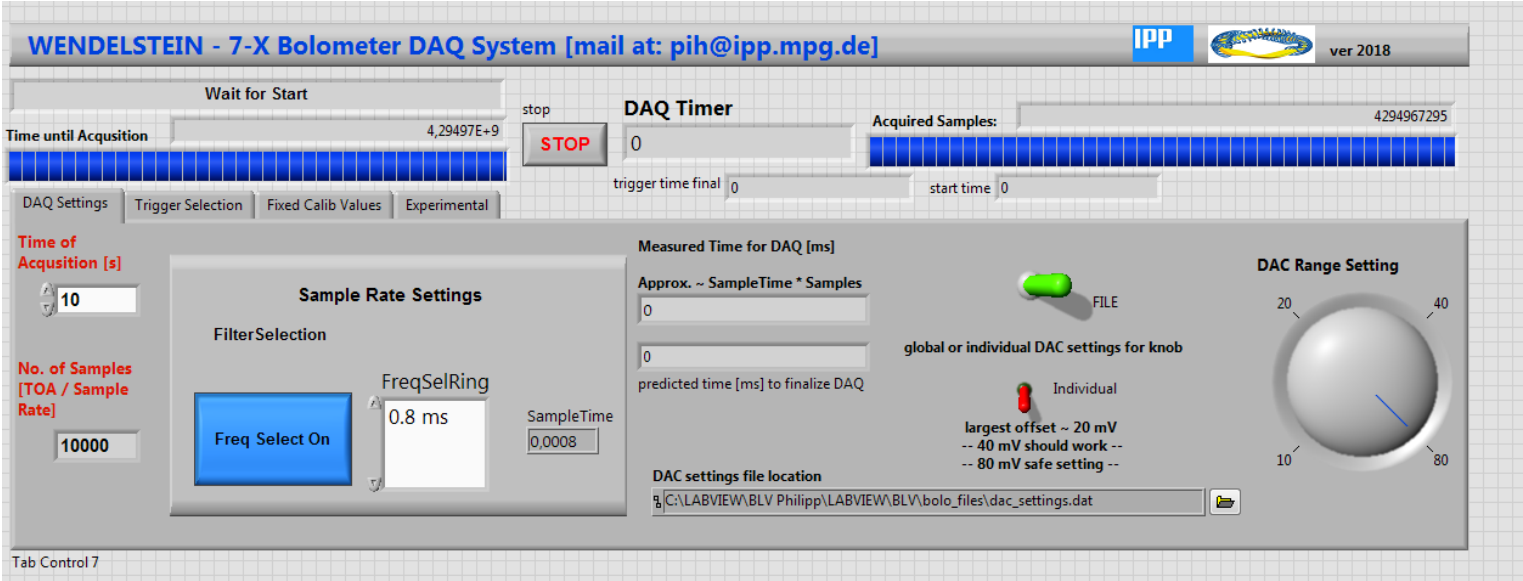
full software project with all dependencies; FPGA target with current build of executable and bit file, untouched (leave untouched!)

- important files are ‘AutomatedStart.vi’ & ‘W7XBolo2018.vi’
- latter is main VI in where calibration, acquisition and output are queued
- ’hReadChannels-UandP-2018.vi’ does measurement/estimation, ‘test.vi’ is a timing and reference test (disabled in the build)
- ‘rawpower2totalprad.vi’ calculates the final P_{rad} after acquisition



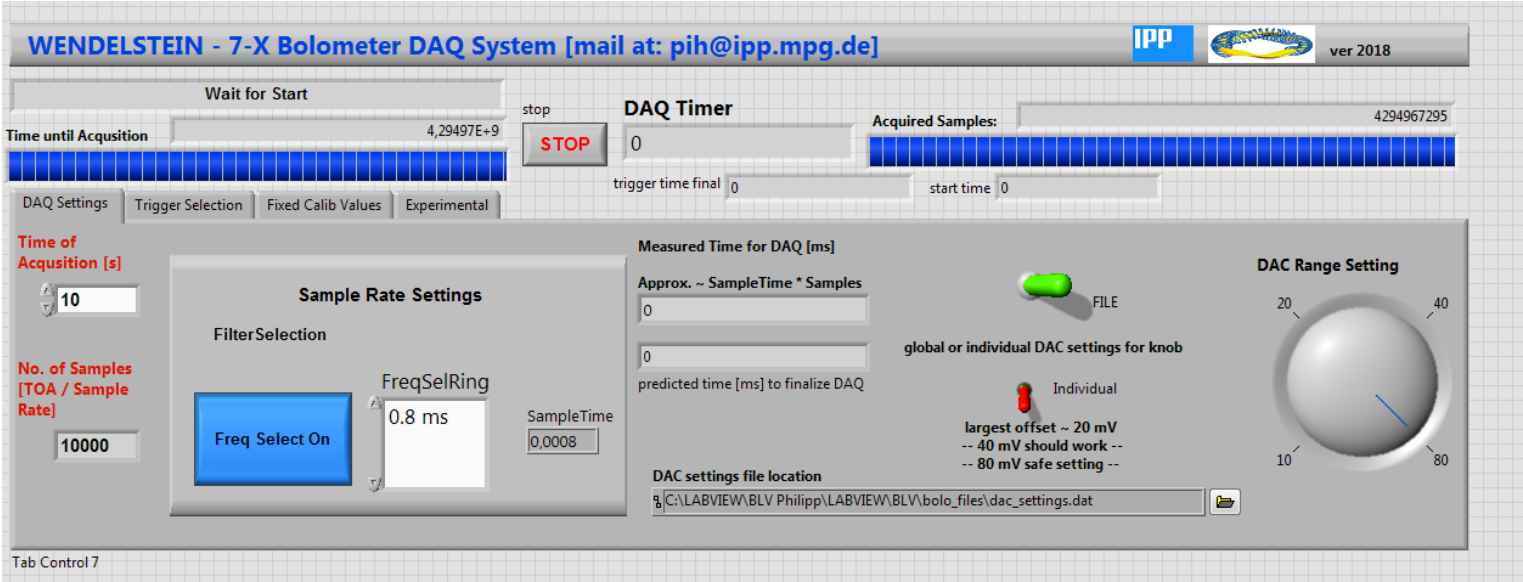
front panel: data acquisition and range settings for the voltage boundaries of the controller, e.g. turn/dial or file import of individual/global bits

- time of acq. defined in seconds (divide by sampling rate in ms yields number of samples)
- comparison of predicted DAQ duration and actual time passed acquiring samples (performance)



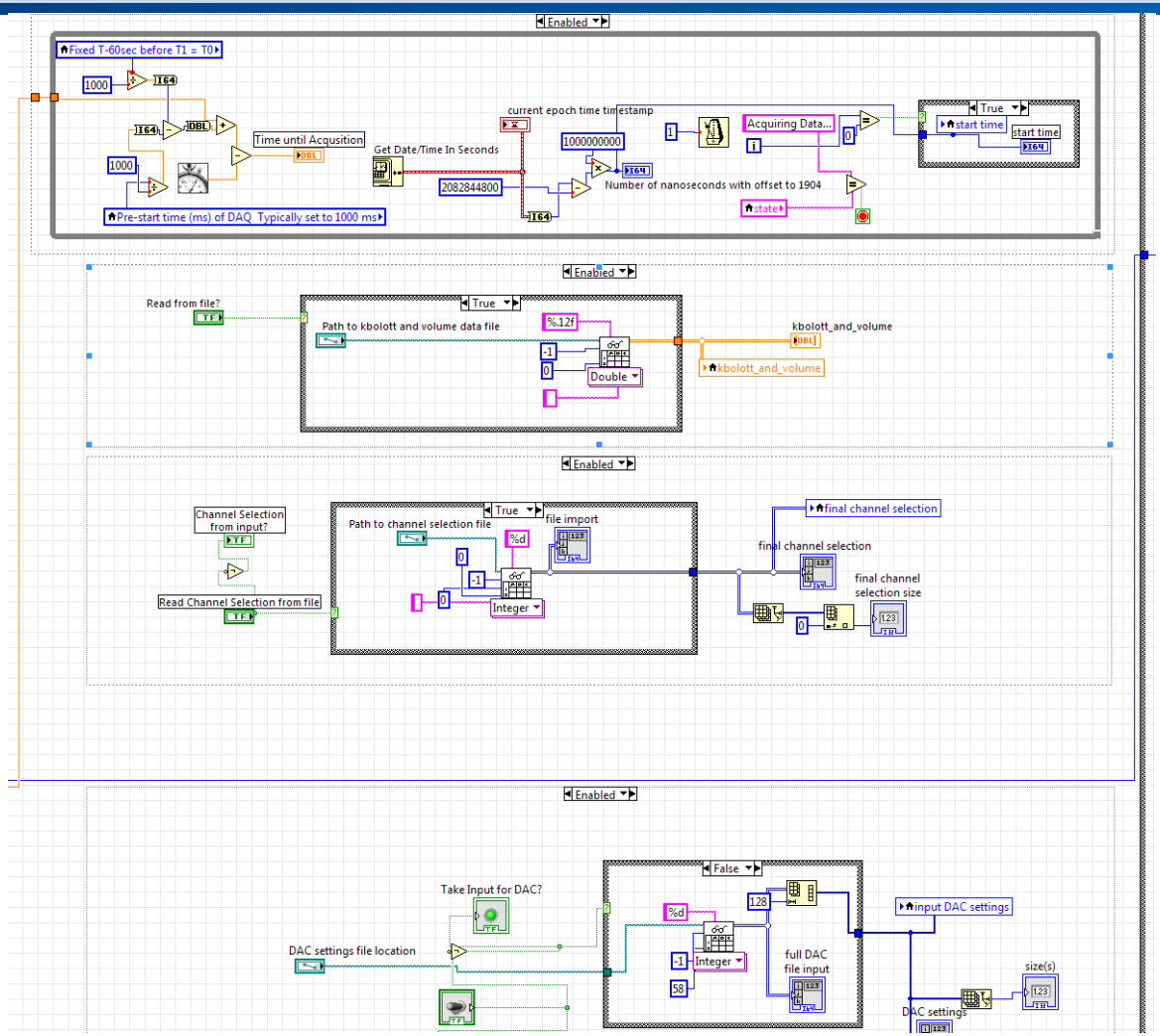
front panel: data acquisition and range settings for the voltage boundaries of the controller, e.g. turn/dial or file import of individual/global bits

- switch between global (all channels the same), individual bit settings for DAC ranges
- switch for input or file loaded settings of DAC ranges; individual input settings located in bottom ‘Test’ tab

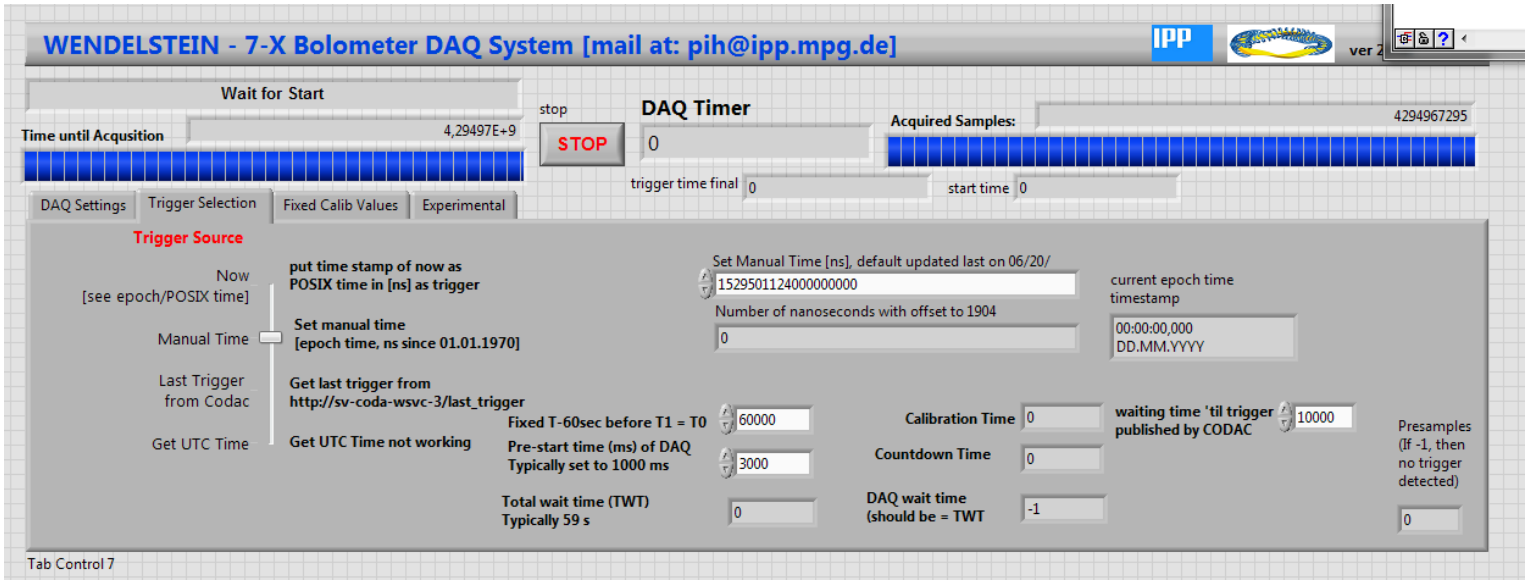


front panel: data acquisition and range settings for the voltage boundaries of the controller, e.g. turn/dial or file import of individual/global bits

- trigger and current epoch time displayed at top of VI
- countdown to start of data acquisition

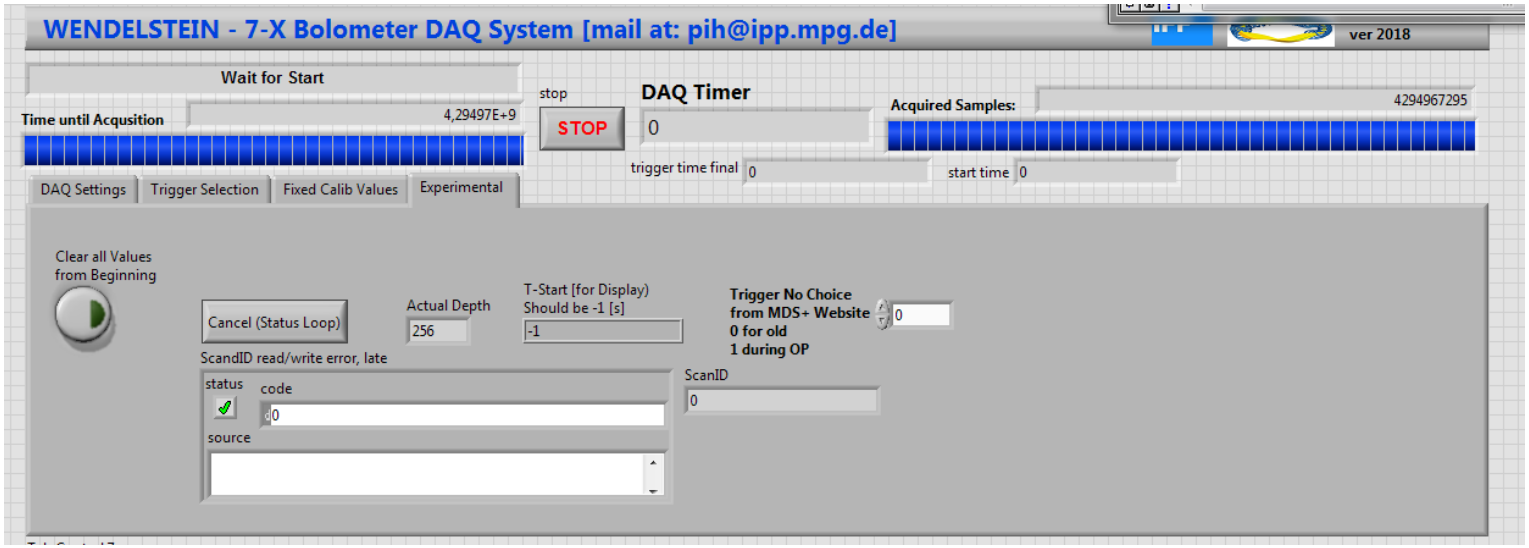


block diagram 'W7XBolo2018.vi': of previous front panel parts (top - timing, middle top - geometric factors import/input, middle bottom - channel selection, bottom - DAC range settings)

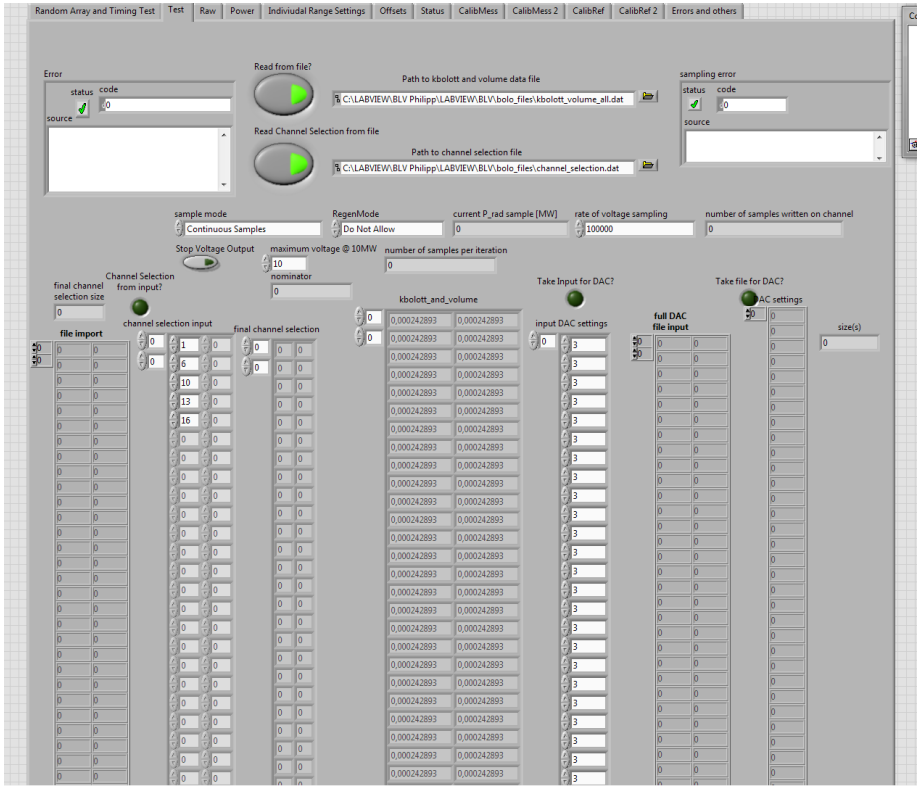


front panel: trigger source settings and timings; pre-start time before T0 (60s fixed after hardware trigger)

- time of acq. defined in seconds (divide by sampling rate in ms yields number of samples)
- comparison of predicted DAQ duration and actual time passed acquiring samples (performance)

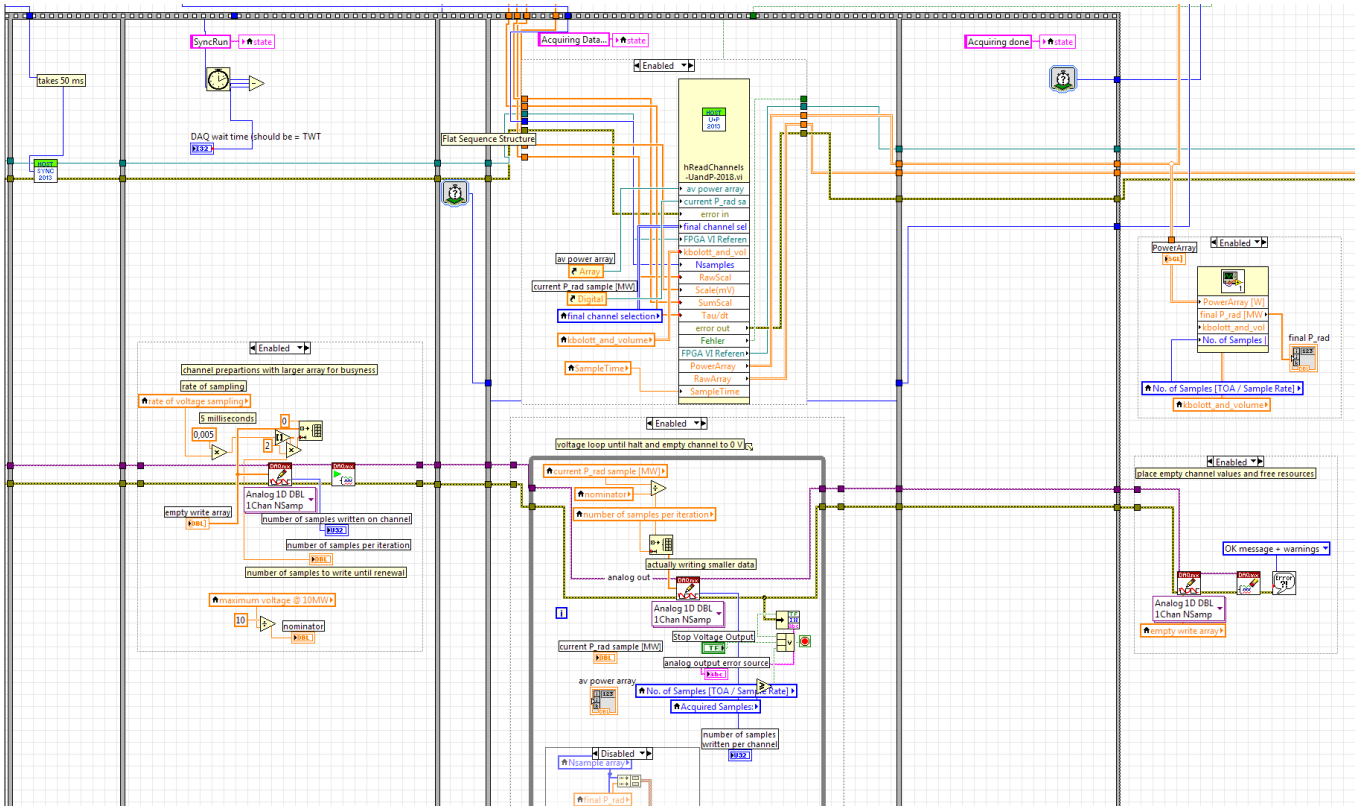


front panel: error top panel (first place to look at if something goes wrong/seems off) 'ScanID read/write error' refers to the stashed sample number on the FPGA target

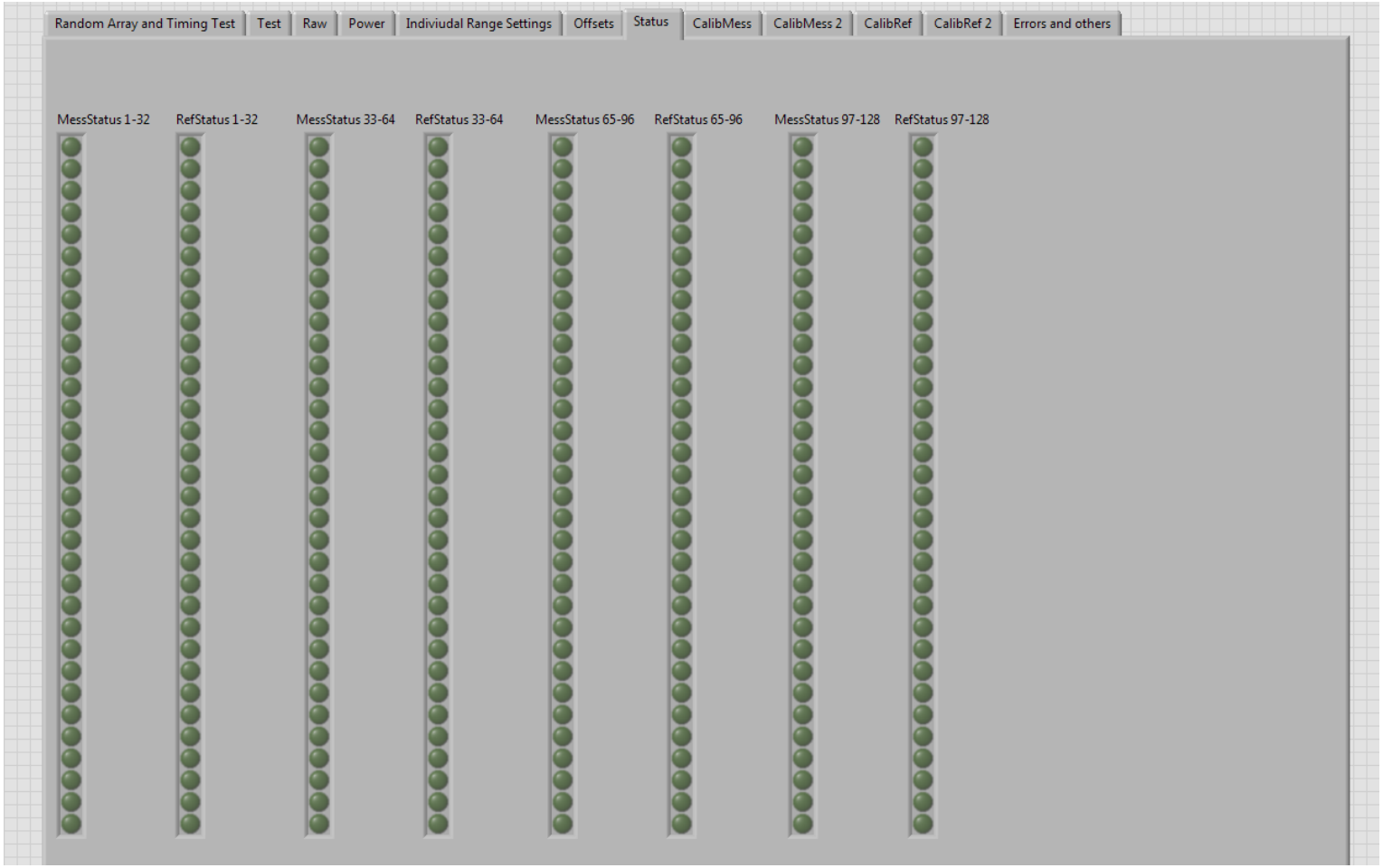


front panel: 'Test' tab; various DAC range import and and load settings; channel selection settings for P_{rad} estimation; array debugging and error codes

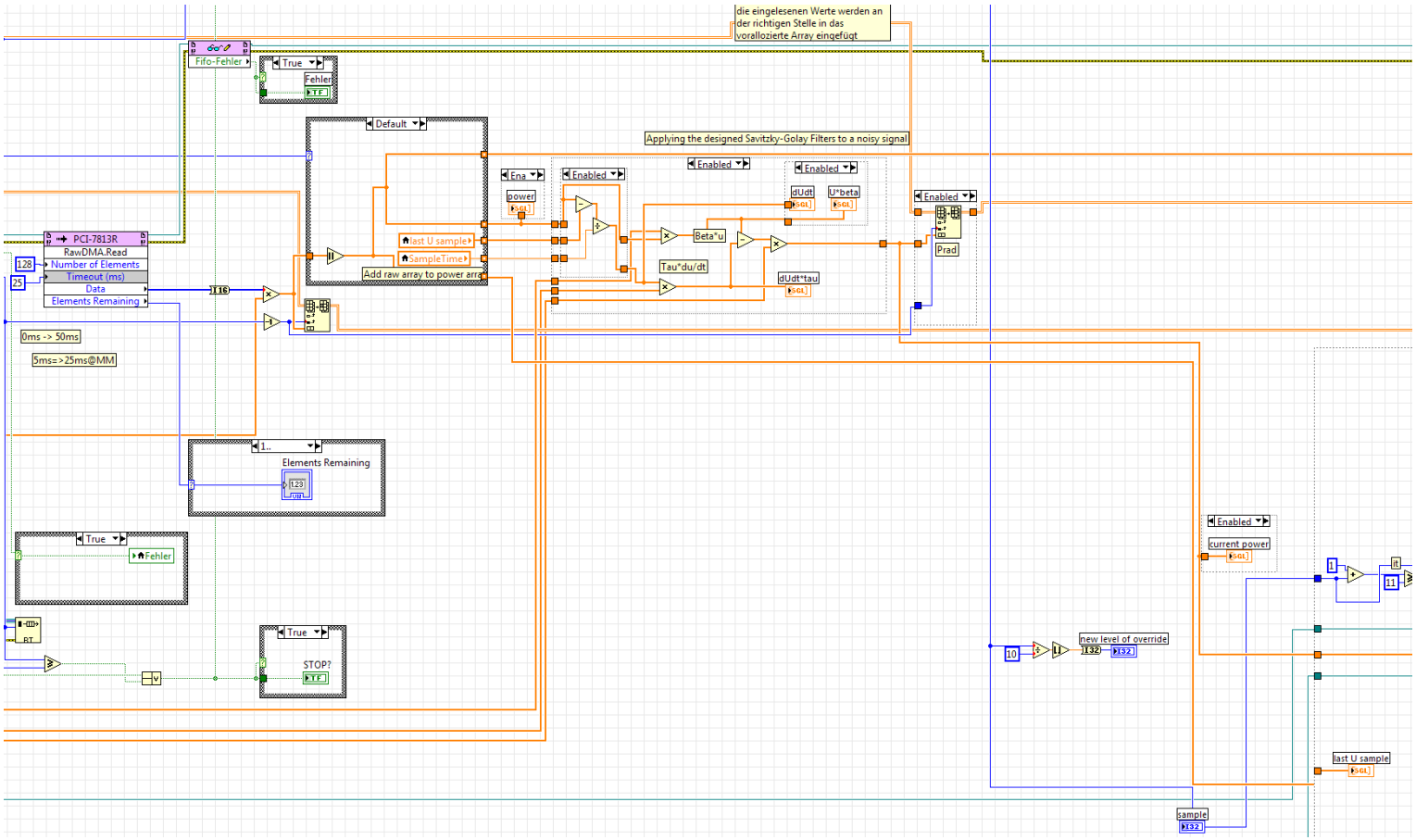
- top: switches on what to import for DAC range/channel selection
- below: sampling and frequency settings for voltage output of P_{rad} estimate, plus scaling voltage @ 10 MW
- bottom: debugging for arrays of imports/input



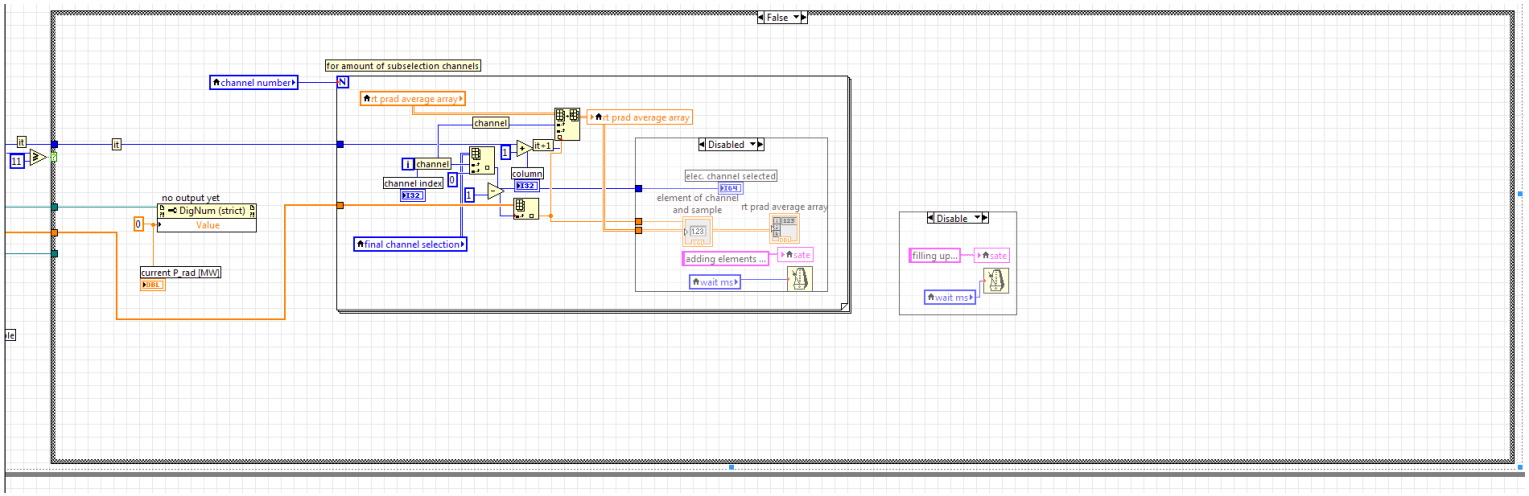
block diagram 'W7XBolo2018.vi': flat sequence of DAQ 'hReadChannels-UandP-2018.vi' with local variables inputs; bottom: channel set up for voltage and sampling on analog; right, outside: end of analog sampling and 'rawpower2totalprad.vi', where the channel power is used to calculate a 'full' P_{rad}



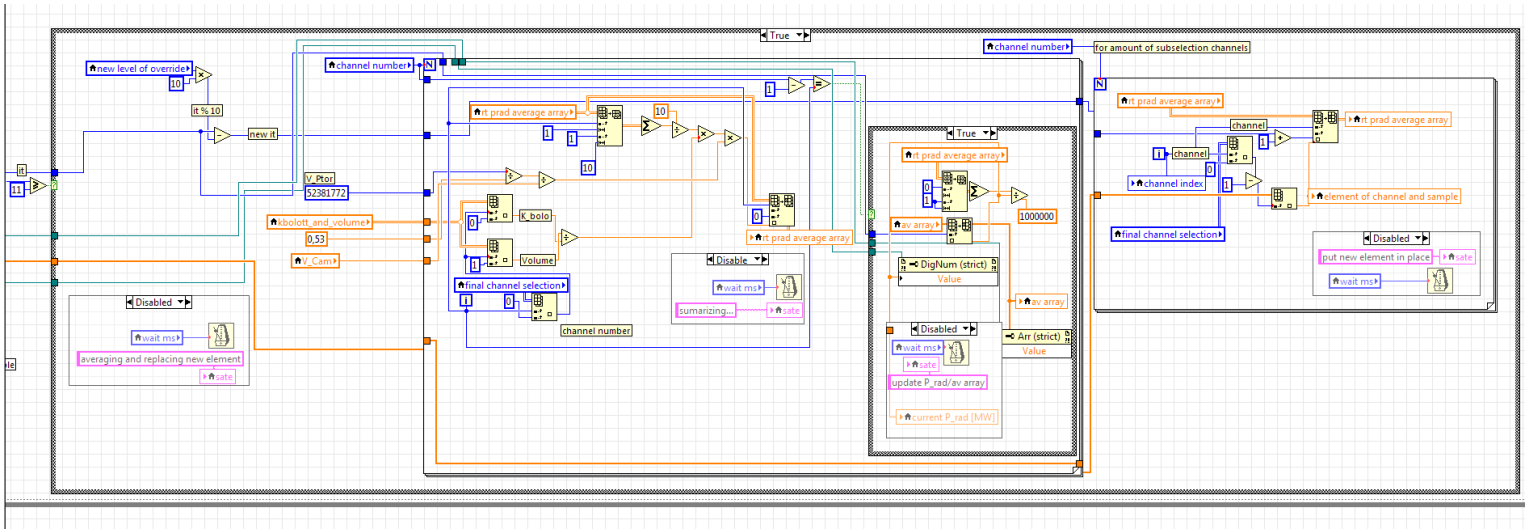
front panel: 'Status' panel tab; if calibration or reference values are off, this shows the channel status accordingly



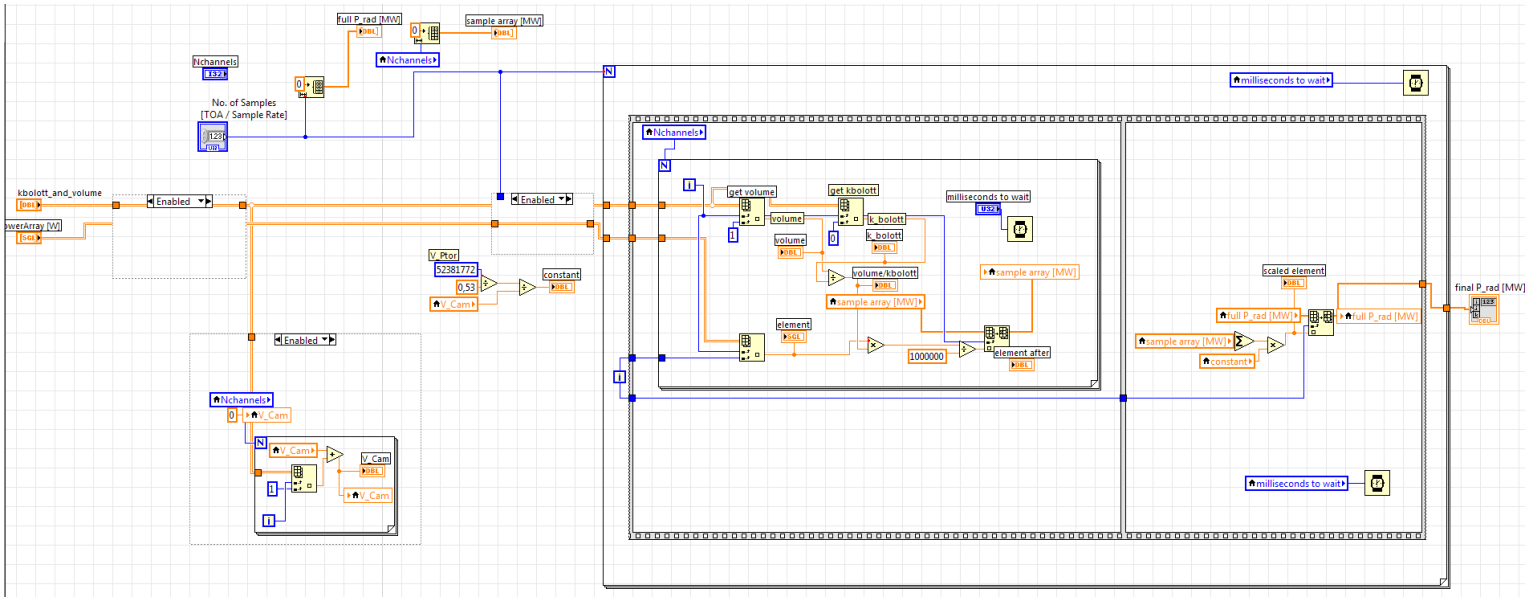
block diagram 'hReadChannels-UandP-2018.vi': conversion from raw voltage signal of channel and sample to power; bottom right: input for estimation routine and FIFO shifting



block diagram 'hReadChannels-UandP-2018.vi': first part of estimation - before 10th sample, filling up averaging FIFO array



block diagram 'hReadChannels-UandP-2018.vi': after 10th sample, averaging with geometry factors and scaling; then FIFO push the first sample out for new one (far right) and writing to the referenced properties for the top level 'W7XBolo2018.vi'



block diagram 'rawpower2totalprad.vi': processing routine calculating the 'full' P_{rad} after acquisition has finished, hence for all channels and samples the same calculation as in the previous subVI

Pros:

- performance good for channel selection of up to $\sim$10
- acquisition times for one sample actually improved due to some touch ups compared to old version
- technicality of imports, inputs and loading has been proven to not fail in previous (trigger-) tests
- stash of local files to load configuration, reproducibility
- calculation of estimate P_{rad} seems fine, tests to be made with comparison signal (manual/virtual)

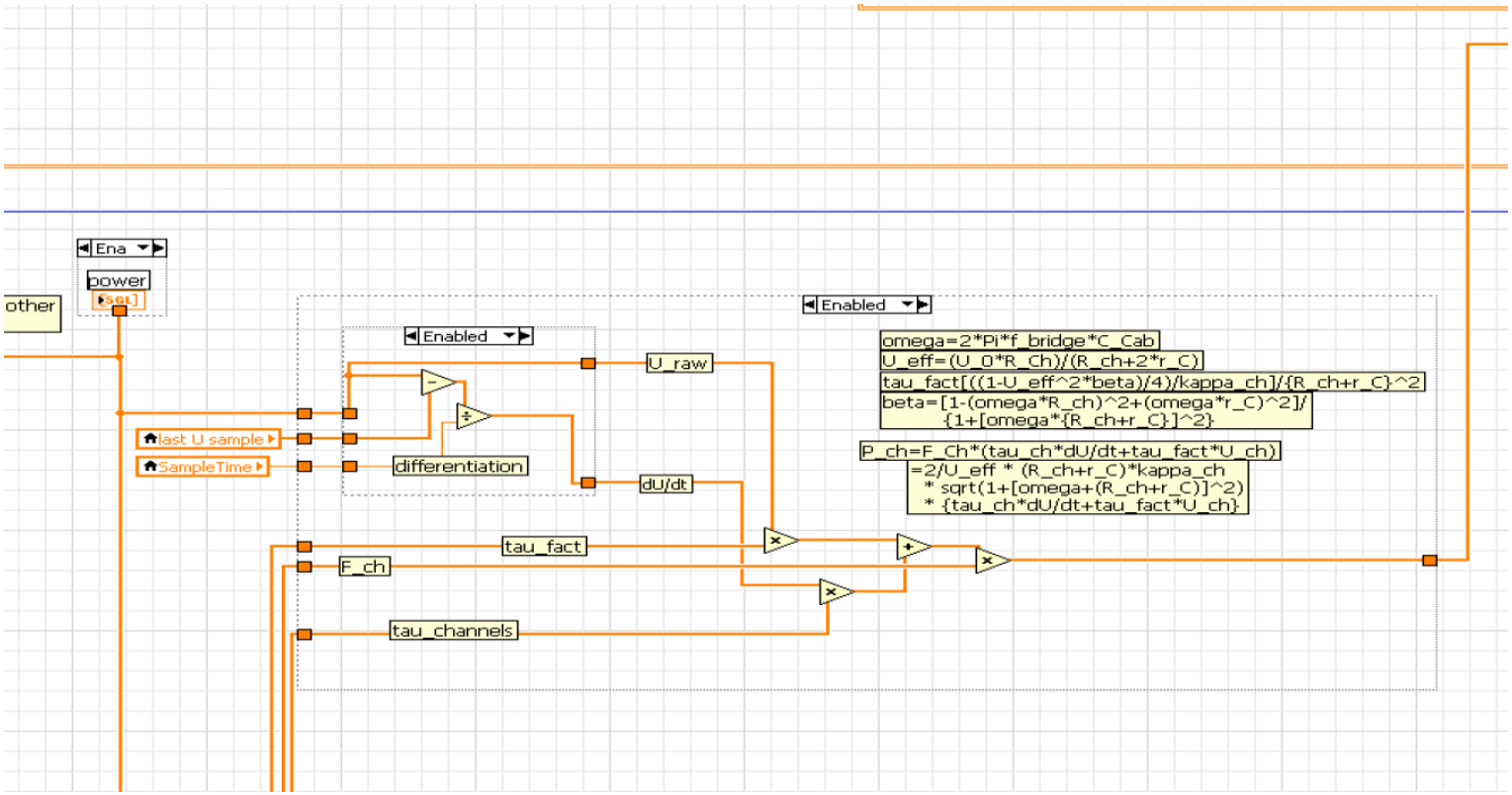
Cons:

- critical bug/feature in LabView's abort/stop function that does not clear FPGA targets local memory, hence e.g. samples acquired is stuck on a specific value
- hence, no proper measurement is possible directly after a failed software deployment/execution
- calculation inside 'rawpower2totalprad.vi' seem to be off by some orders of magnitude 10^x to no apparent reason (exactly the same calculation process as in 'hReadChannels-UandP-2018.vi')
- ?! to be proven: upload to archive broken ?!

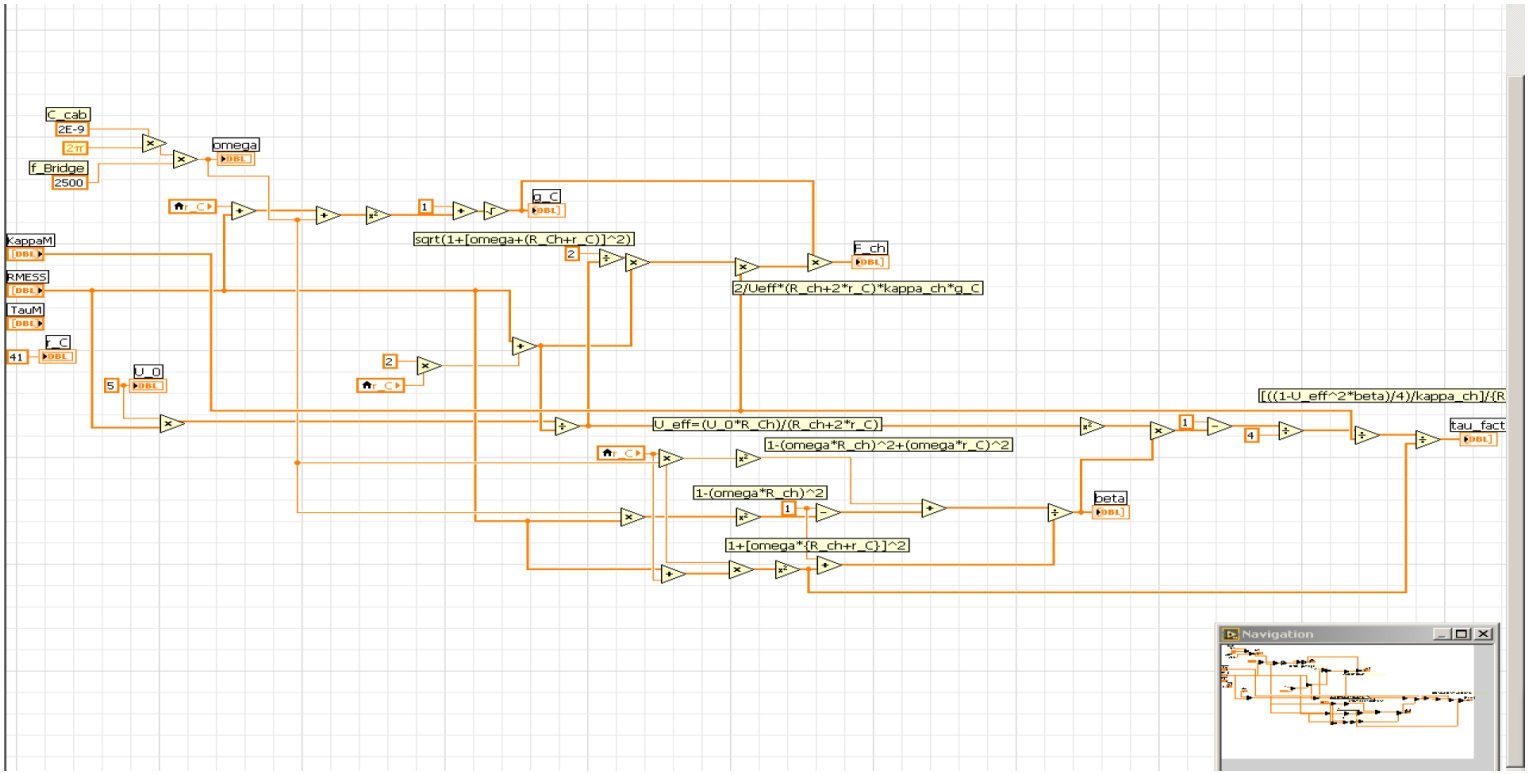
TODO:

- fix upload cycle and add the current configurations to it in the form of a JSON, add estimation time line to archive
- proof of calculations and estimation
- find possible fix for memory cache, local flashing routine of FPGA target in the start up phase of VI maybe
- fix full P_{rad} routine

Appendix


Ena
power
other
Enabled
Enabled
U_raw
dU/dt
differentiation
last U sample
SampleTime
tau_fact
F_ch
tau_channels
omega=2*Pi*f_bridge*C_Cab
U_eff=(U_0*R_Ch)/(R_ch+2*r_C)
tau_fact[((1-U_eff^2*beta)/4)/kappa_ch]/{R_ch+r_C}^2
beta=[1-(omega*R_ch)^2+(omega*r_C)^2]/
{1+[omega*{R_ch+r_C}]^2}
P_ch=F_Ch*(tau_ch*dU/dt+tau_fact*U_ch)
=2/U_eff * (R_ch+r_C)*kappa_ch
* sqrt(1+[omega+(R_ch+r_C)]^2)
* {tau_ch*dU/dt+tau_fact*U_ch}

Appendix


C_cab
2E-9
2π
omega
f_Bridge
2500
r_C
g_C
sqrt(1+[omega+(R_Ch+r_C)]^2)
F_ch
2/Ueff*(R_ch+2*r_C)*kappa_ch*g_C
KappaM
RMESS
TauM
r_C
41
U_0
5
2
r_C
U_eff=(U_0*R_Ch)/(R_ch+2*r_C)
1-(omega*R_ch)^2+(omega*r_C)^2
1-(omega*R_ch)^2
1+[omega*(R_ch+r_C)]^2
beta
[((1-U_eff^2*beta)/4)/kappa_ch]/(R
tau_fact
Navigation