Calibration Unit CalU and Controller CuC

Calibration of a HF/VHF/UHF solar radio telescope

System and units below describe a system to calibrate raw-data from CALLISTO (FIT-files) into antenna temperature Tant and into solar radio flux S, in case antenna gain is known all over the observation time. Calibration is performed with respect to an internal noise source, based on an Avalanche diode which frequency range is limited from ~10 MHz to ~900 MHz. Excess noise ratio ENR is in the order of +15 … +20 dB, depending on component availability. A calibration protocol will be provided with order. Calibration is performed by injecting two known noise temperatures Tcold and Thot into the receiving system at pre-defined times, usually modulo 15 minutes synchronous with data collection by Callisto.

System to perform such a calibration is composed out of:

1. Hardware:
   - Calibration unit CalU between antenna and LNA, containing rf-switch and noise source
   - Calibration controller CuC next to the computer to control and feed CalU
   - Antenna with access to the feed such as CLP-5130 or BICONE or Yagi (not LWA!)
2. Software:
   - Python script to control CalU
   - Python script to perform calibration, once FIT-file is written to disc
   - Required: Python 3.7 or higher, libraries installed astropy, scipy, os, configparser, tkinter, time and serial (Pyserial)
   - System scheduler ssfree.exe (Windows) or crontab (LINUX)

Important aspects:

- Antenna gain must be known as precise as possible all-over observed frequencies and observation times.
- In case antenna is fixed at sky-position and not tracking the Sun a model of gain variation versus date&time must be included. Usually, calibration only works when antenna gain is known for all frequencies of observation over all dates and times.
- In case any nearby transmitters appear at Callisto antenna input with power > -60 dBm, Callisto suffers from saturation and calibration goes wrong. At many places worldwide, a FM-notch filter can help to cope with strong radio transmitters. In addition, at high rfi-level, it might be useful to reduce software-controlled gain of CALLISTO (pwm-value in Callisto.cfg).

Given currently available rf-tight enclosures, maximum 10 CalU can be produced. For more units, PCB requires a re-design and a re-evaluation of the enclosure. Currently is one CalU on stock and one is used for endurance testing and software improvements.

System Layout:

A system is composed out of a directional antenna tracking the Sun, the calibration unit CalU with its controller CuC, low noise amplifier LNA (e.g. LNA2000), optional heterodyne converter and Callisto. Controller as well as Callisto should be controlled by one and the same computer to ensure time synchronization of calibration sequence and observation. Calibration controller CuC also allows to supply LNA and Callisto where control-software offers the option to switch power off/on for LNA and CALLISTO. This to save power in case of photo-voltaic power source and/or to reset CALLISTO in case of blockage. Controller also provides +28V in case of a commercial noise source together with a L-band calibration unit, based on low loss microwave relays. The calibration unit described here is designed to cover 10 MHz … 1 GHz and low cost given by low cost SPDT UltraCMOS RF-Switch PE4250 from Peregrine Semiconductors.

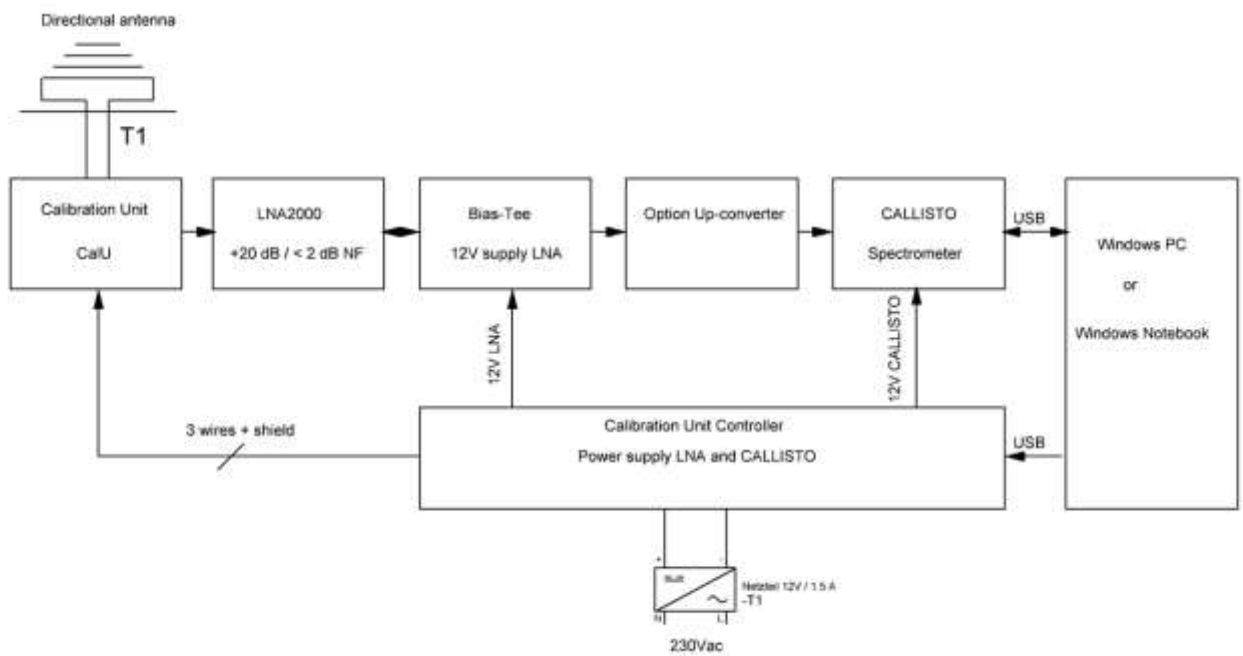

Fig. 1: System layout solar radio telescope with flux calibration unit and optional frequency up-converter, Bias-T is used to supply LNA2000 with 12Vdc.

Parameter	Calibration Unit CalU	Calibration unit controller CuC
Frequency range	10 MHz … 900 MHz	N/A
Attenuation (loss)	< 2 dB	N/A
Wave impedance	50 Ω	N/A
Power consumption	15 V * 30 mA max ~0.45 W max	< 12 V x 1.5 A including Callisto + LNA
Dimensions	155 x 95 x 25 mm^3	180 x 110 x 55 mm^3
Weight	305 grams	800 grams incl. power adapter {tbc}
Prize tag Sept. 2024	350 CHF	610 CHF / 310 CHF

Table 1: Specifications calibration unit CalU and calibration unit controller CuC

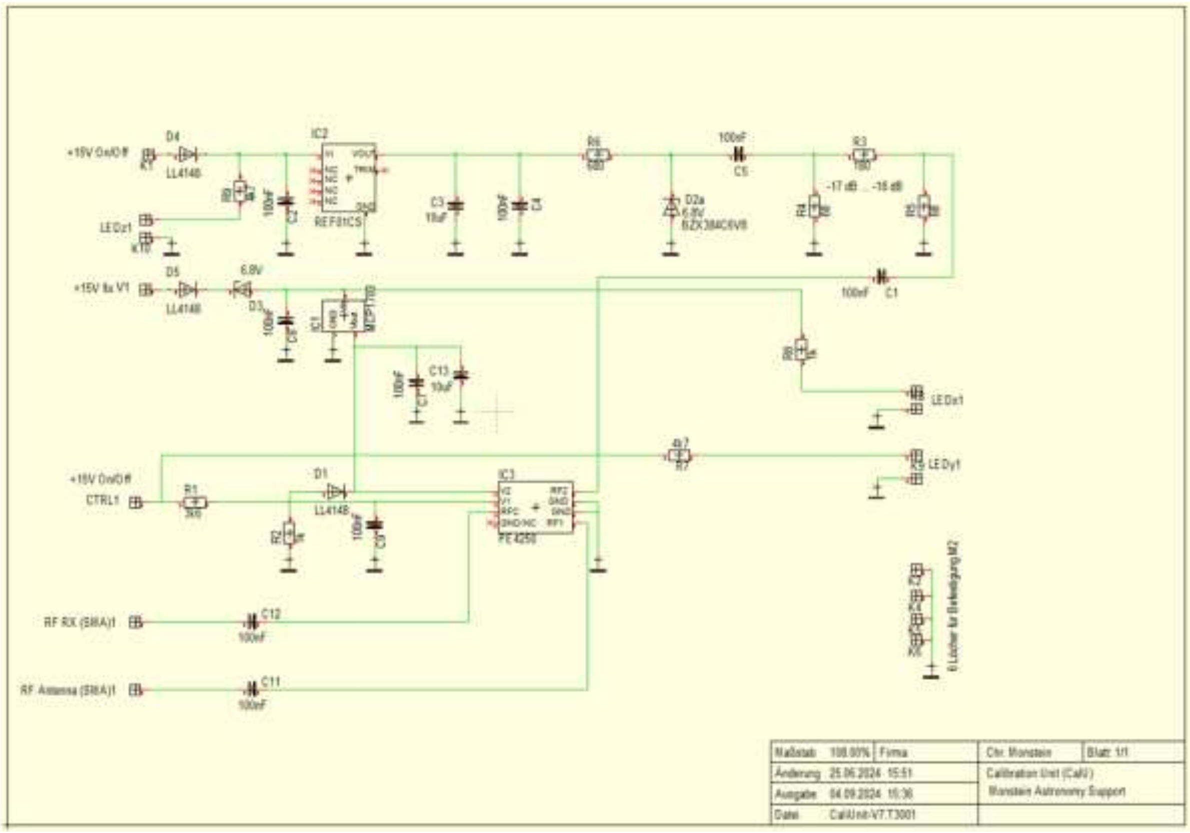

Fig. 2: Schematics calibration unit CalU with SPDT, Avalange noise diode and voltage regulators 10V and 3.3V.

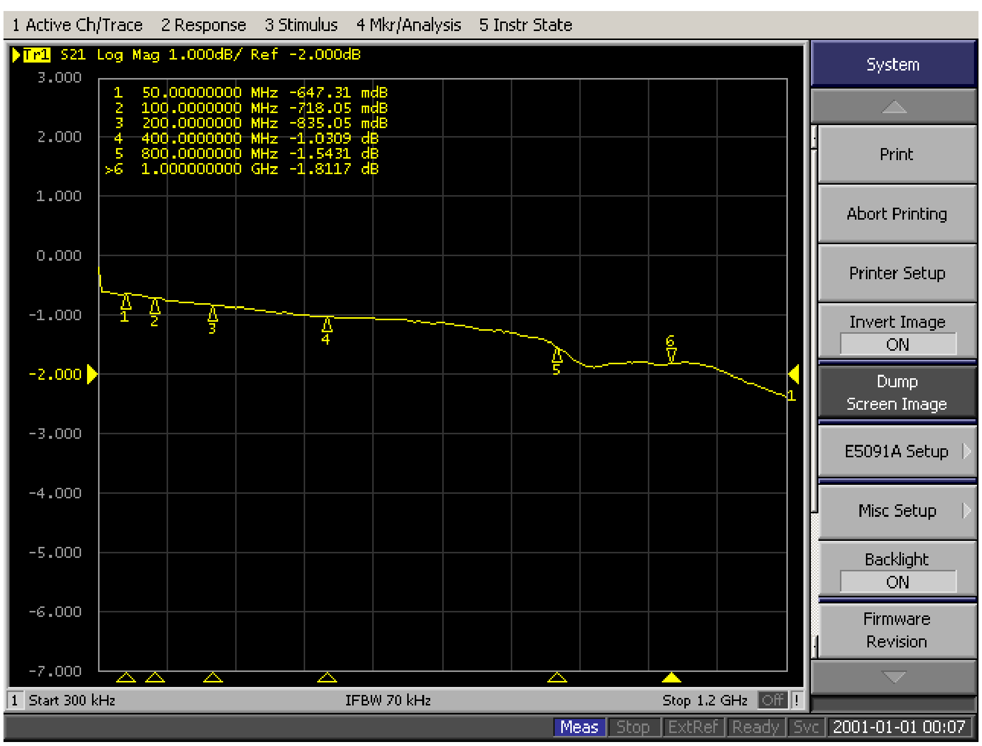

Fig. 3: Measurement of s21 between antenna port and LNA-port with < 2dB up to 1 GHz. For low loss application such as in L-band rf-microwave relays are required. This is not part of this project as cost will be much higher wit rf-microwave relays.

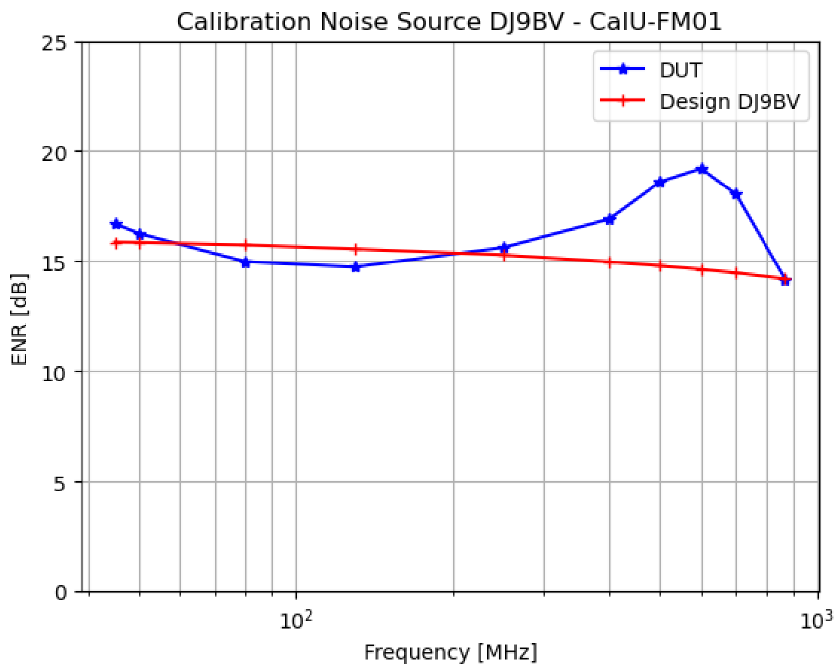

Fig. 4: Calibration plot of internal noise diode. Red plot show reference noise source after DJ9BV, while blue plot shows ENR of internal noise source, based on Avalange diode BZX384C6V8. Blue dots are transferred to a table in calibration script (Python) and later on used together with interpolation function.

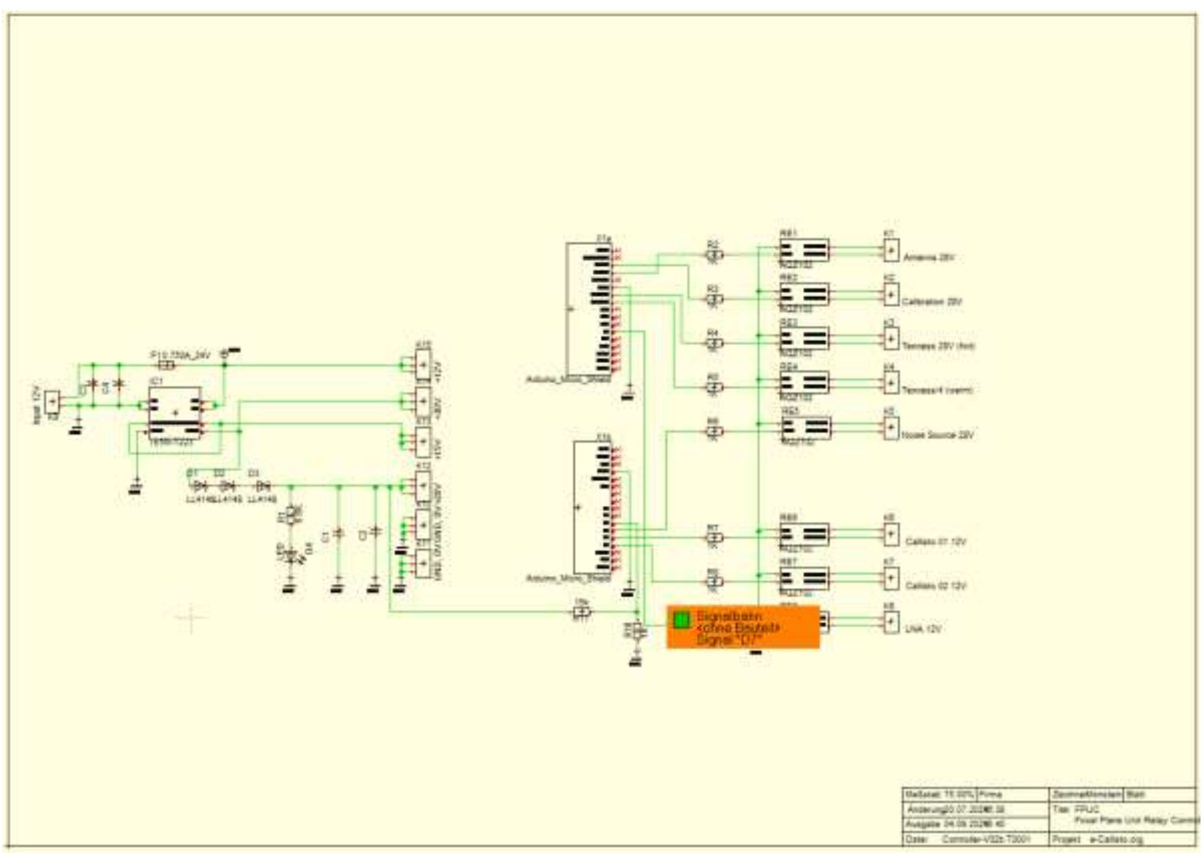

Fig. 5: Schematics of control unit CuC, based on Arduino Micro with USB-port, dc/dc/converter 12V -> +/- 15V.

Device connectors at backside of CuC:

DIN-connector ccw	Signal	PCB Terminal	External cable 3 wire + shield
1 (blue wire)	GND	K10-1	Shield=GND=0V
2 (white wire)	~Calibration/Antenna	K4-innercontact	White
3 (green wire)	Noise source	K3-innercontact	Green
4 (red wire)	+15 Volt < 20 mA	K13-1	Red/Purple

Table 2: Circular female DIN chassis-connector at calibration controller back-side for CalU

DC-connector 12V in	Signal	PCB terminal
DC centre lug	GND	K9minus
Left lug (seen from solder side)	+12 volts > 400 mA	K9plus

Table 3: DC-male connector 5.5/2.1mm 12 volts at calibration controller back-side

4 mm Banana connector female	Signal	PCB terminal
CALLISTO black (blue wire)	GND	K10-3
CALLISTO yellow (yellow wire)	+12 volts 250 mA	K1-innercontact
LNA2000 blue (blue wire)	GND	K10-2
LNA2000 green (green wire)	+12 volts 120 mA	K2-innercontact

Table 4: Banana connectors at calibration controller back-side to feed LNA and/or CALLISTO (Option)

Printed circuits boards Calu and CuC:

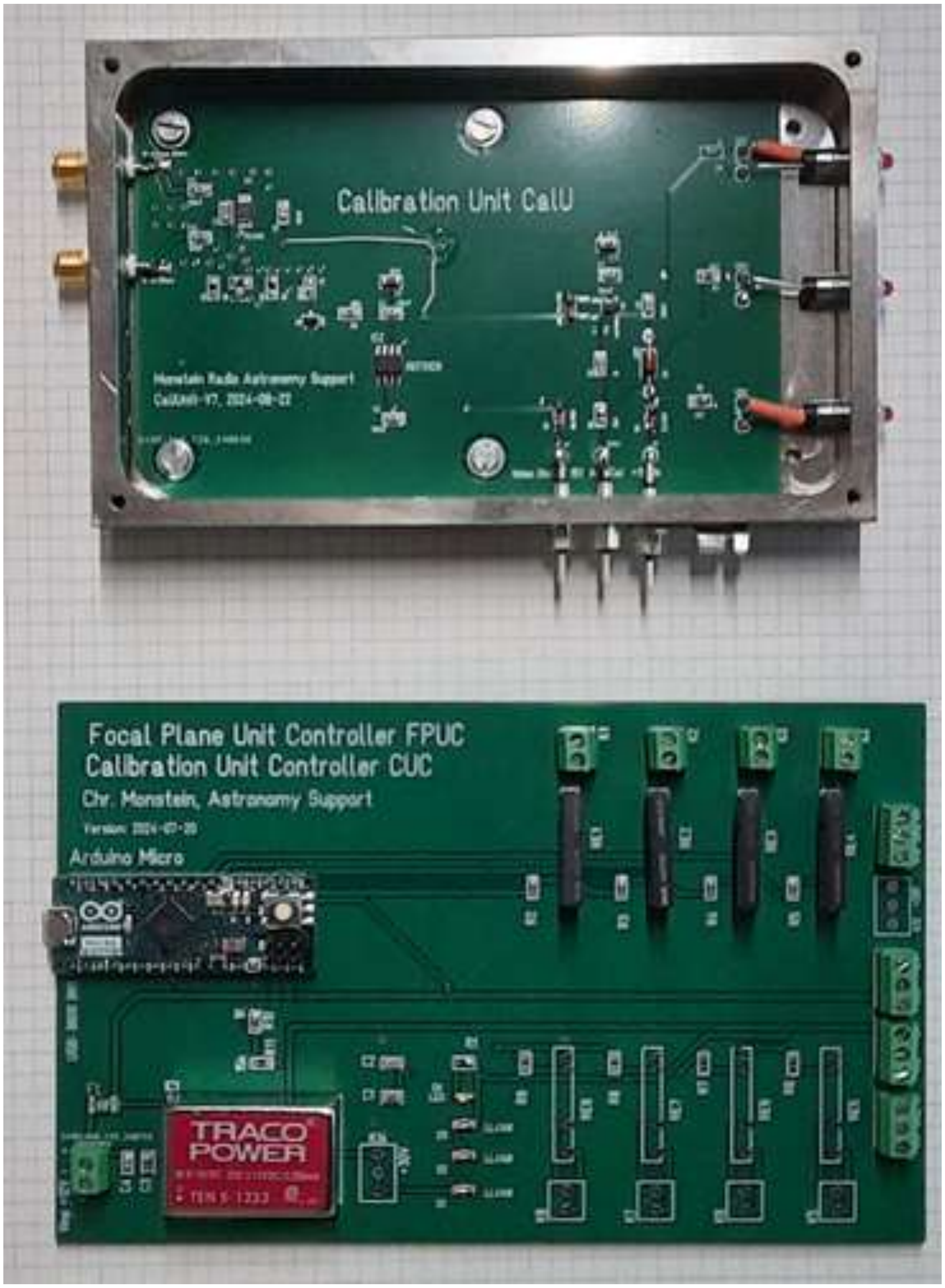

Fig. 6a: Image of outdoor calibration unit with open cover. Top left: antenna input, bottom left output to LNA and finally to receiver. Right side three LED showing status of operation (antenna/calibration, noise excess and 15Vdc). Underneath feed through capacitor for antenna, noise and power.

Fig. 6b: Image of indoor calibration unit controller board. On top right 4 MOS-relay (power opto-coupler) to switch on/off CALLISTO with 12 volts at K1, LNA2000 with 12 volts at K2, noise source with 15 volts at K3 and switching between calibration and observation with 15 volts at K4. On the left side ARDUINO MICRO as controller with USB-connector and TRACO dc up-converter to produce +/-15 volts out of 12 volts DC input. Also available +28 volts for commercial noise sources.
Arduino firmware is written in C to control relays R1-R4 and to test +28 volts for extra noise source.

Space left for additional 4 relays which are foreseen for L-band calibration unit based on expensive electro-mechanical rf-relays (not part of this low-cost project).

Calibration control Python3:

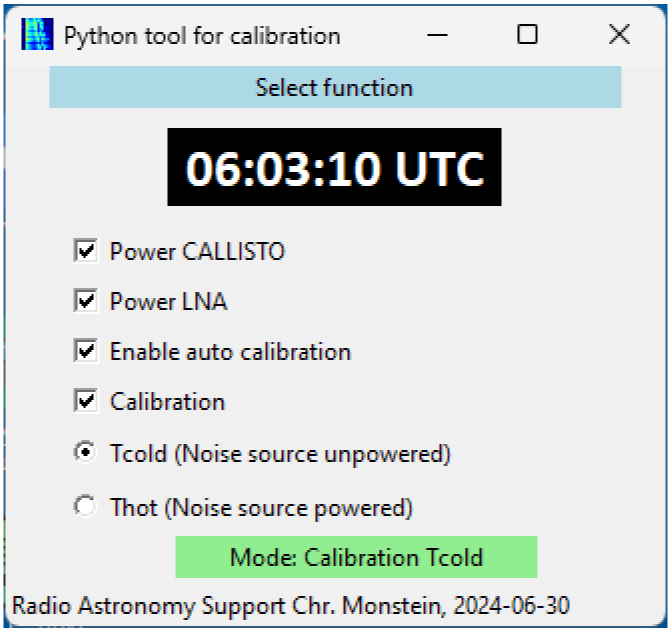

Fig. 7: Python application to control calibration unit, Callisto and LNA2000. There is also a C++ version available but, it doesn't work properly on Windows 11, only Win XP, 7, 8 and 10.

Every time modulo 15 minutes it automatically activates calibration mode and injects Tcold for 10 seconds and after 10 seconds Thot. Then back to observation mode by connecting antenna to the receiver system.
This application is based on TkInter().

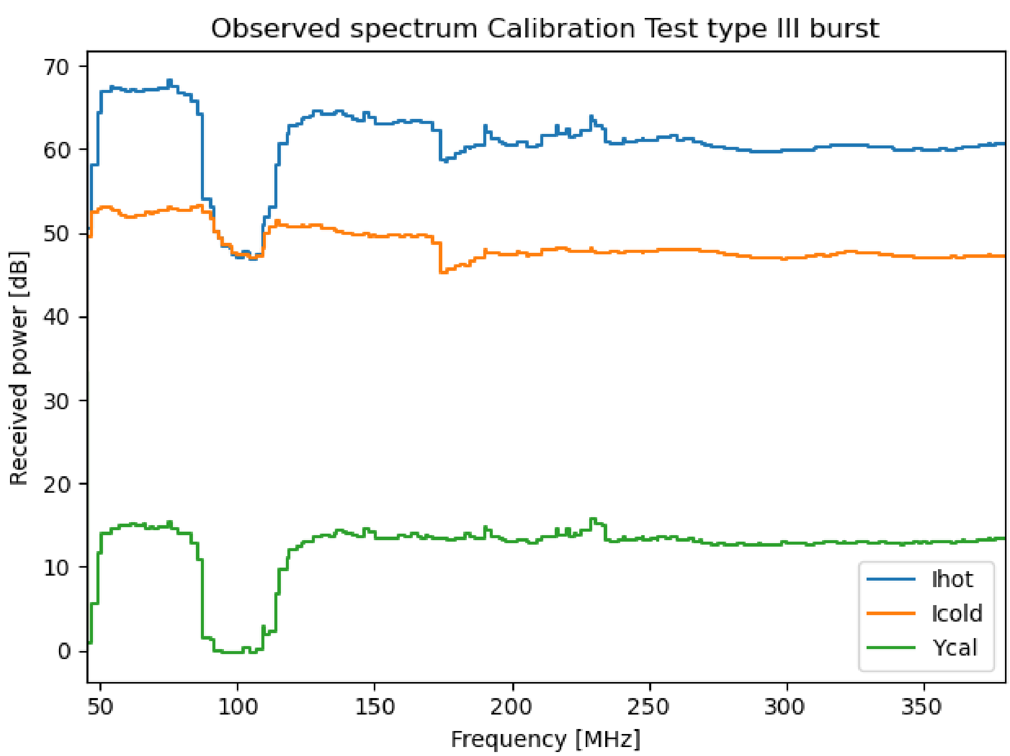

Fig. 8: Calibration signals with notch-filter at FM 88 – 108 MHz and low pass filter at 450 MHz to avoid saturation of LNA and/or CALLISTO by commercial transmission services. Ycal denotes to received Y-factor Thot/Tcold = Ihot/Icold

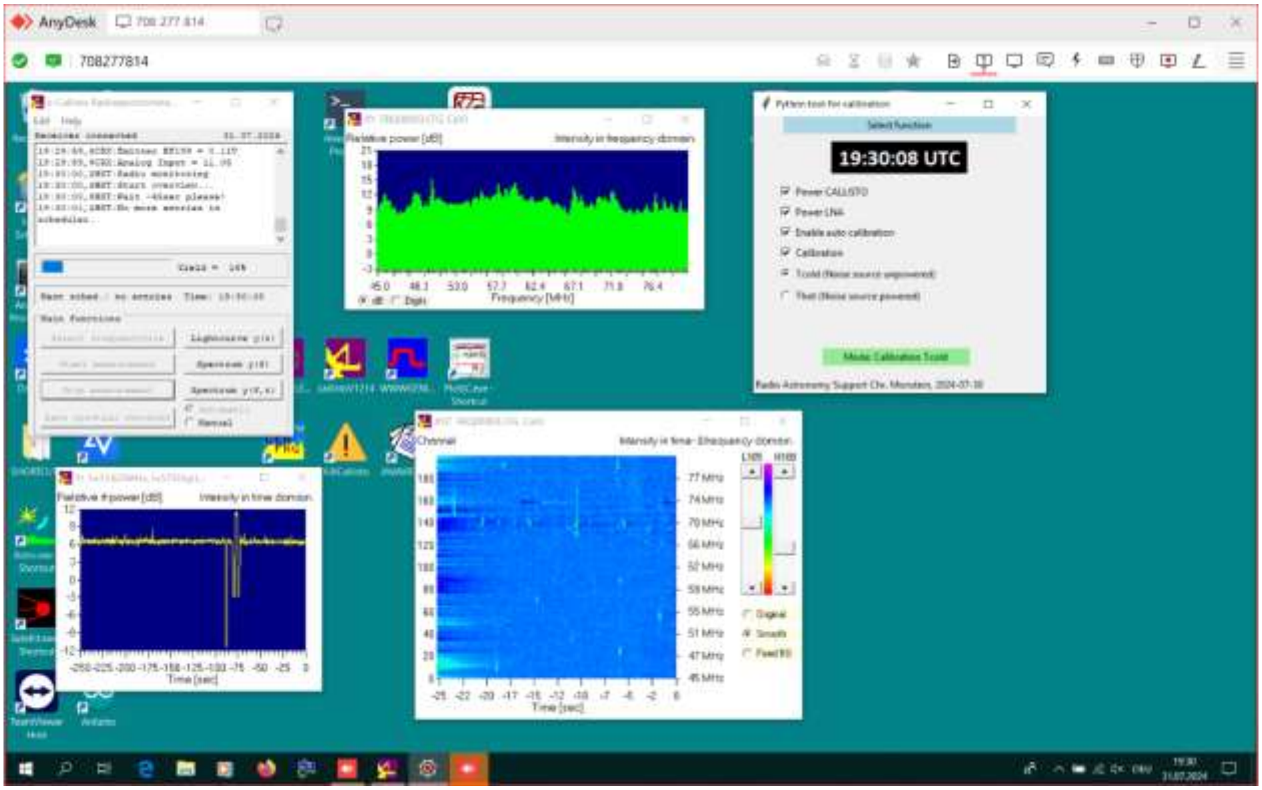

Fig. 9: Running CALLISTO application (C+) and control application (top right in Python3). Image = remote screen-shot via AnyDesk. Light curve bottom left shows a calibration sequence Tcold, Thot.

Example from raw-data to SFU:

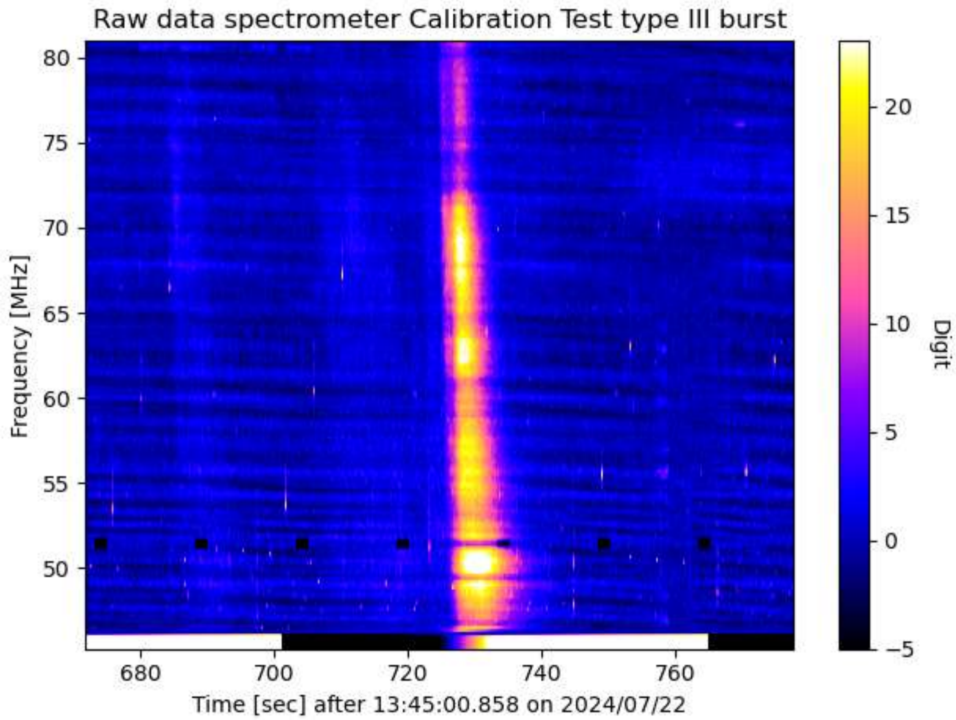

Fig. 10: Zoom into raw data (FIT-file from CALLISTO) before calibration. Intensity in digits.

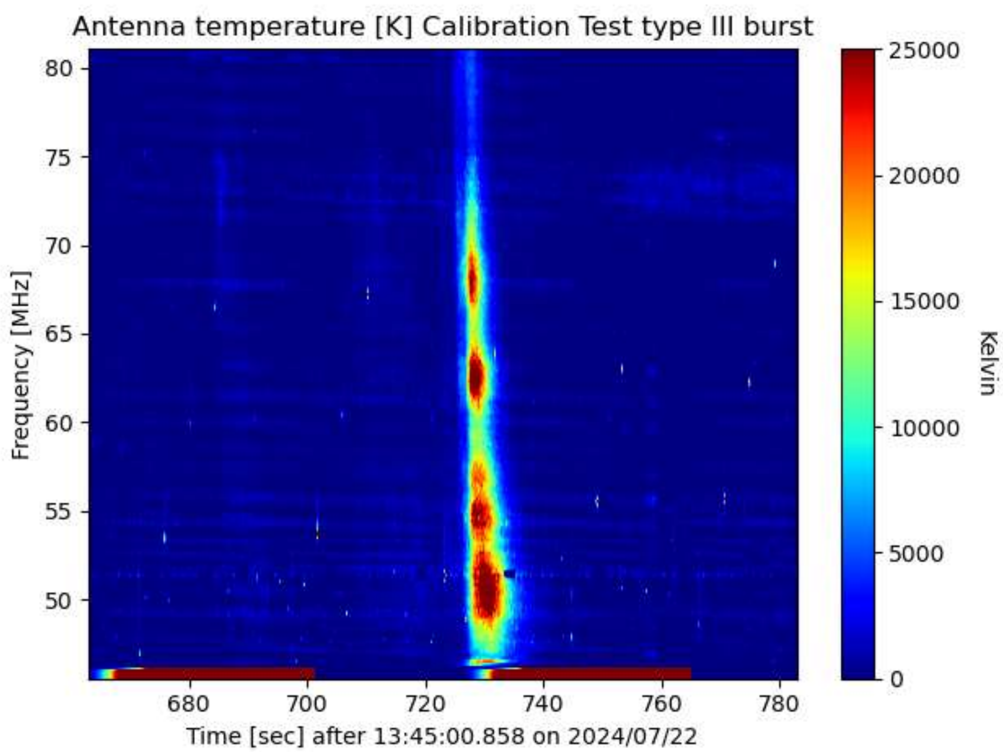

Fig. 11: Calibration in antenna temperature.

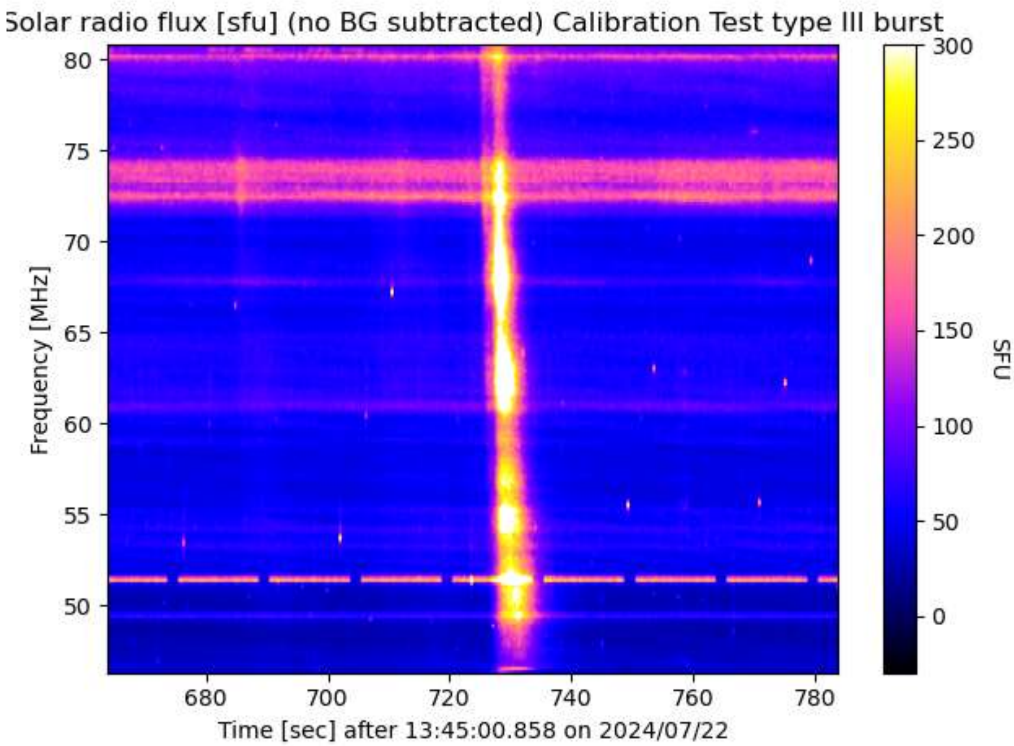

Fig. 12: Calibration in SFU, plot: no background subtracted. Thus, rfi is also calibrated!

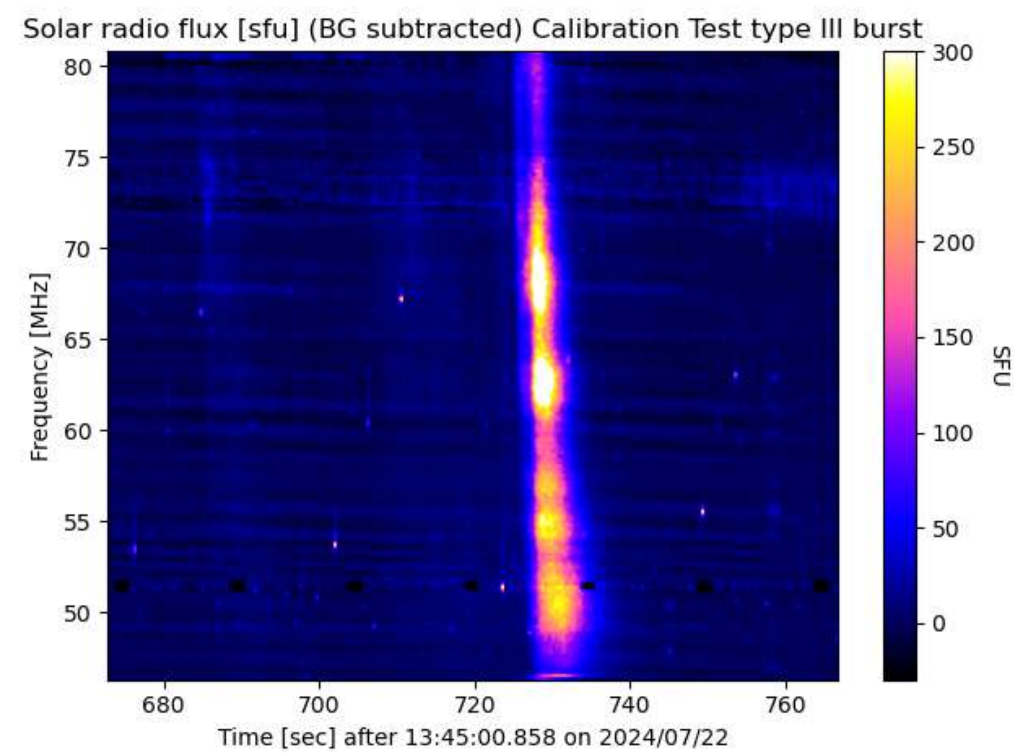

Fig. 13: Calibration in SFU, plot: with background subtracted

Other examples of calibrated solar radio bursts and lightning.

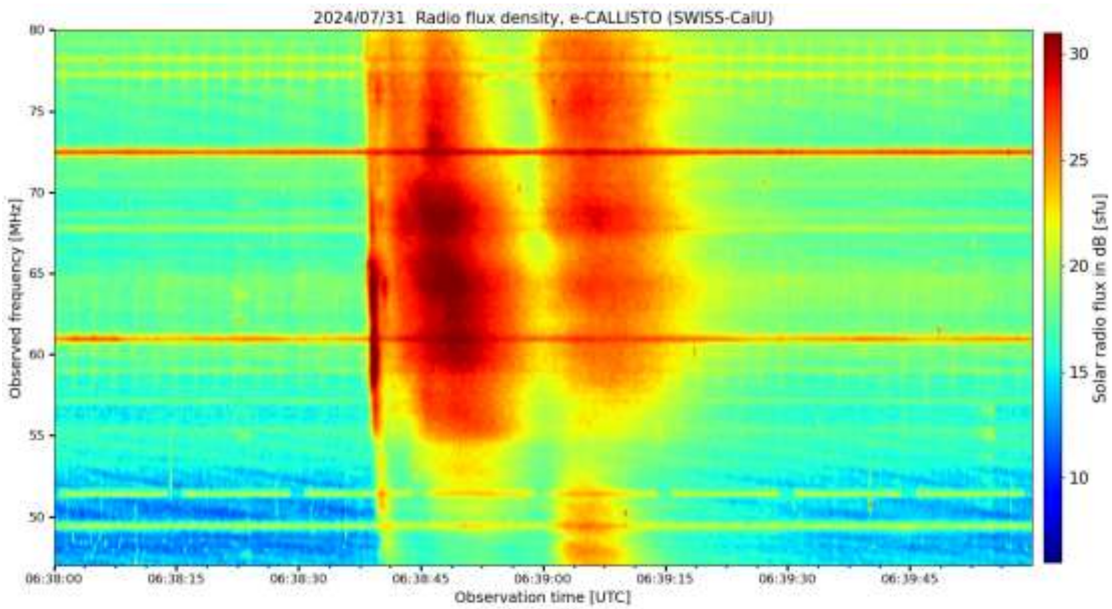

Fig. 14: Type III burst observed with CLP-5130-1N, data calibrated in SFU. Plot in Python3.

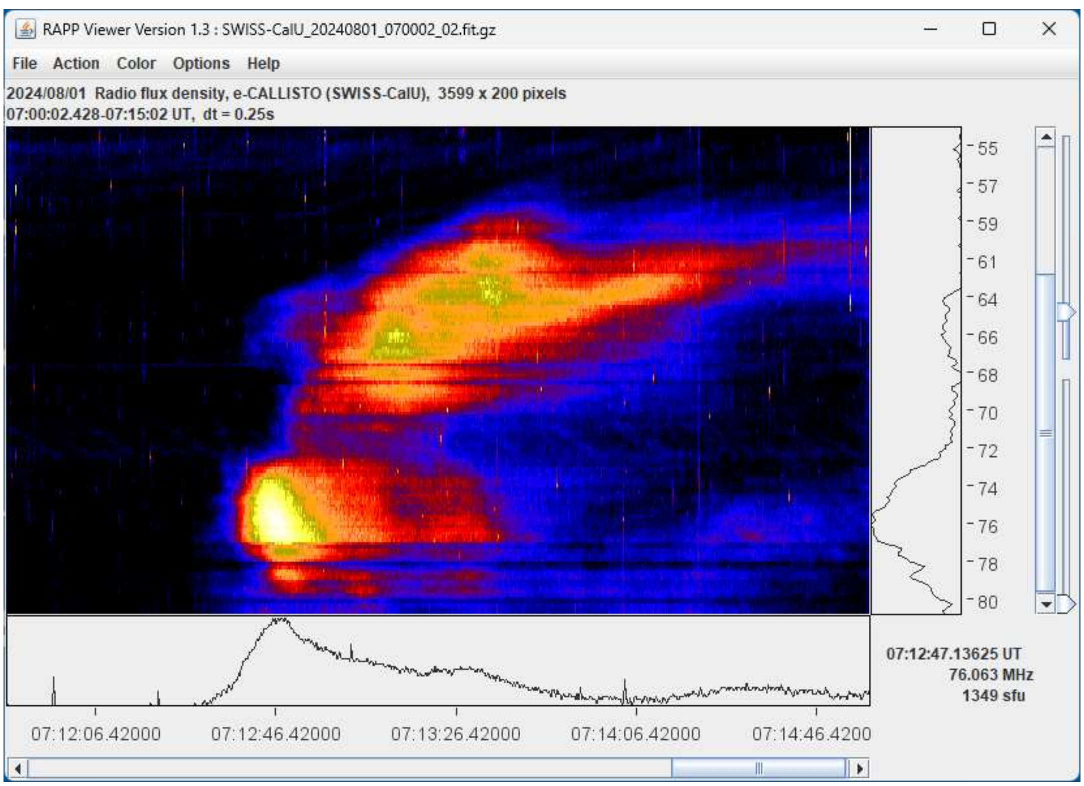

Fig. 15: Calibrated type II radio burst during Swiss National Holiday. Peak flux 1349 sfu at 76 MHz at position of mouse cursor at 07:12:47 UT. Plot presented as screen-hot from JavaViewer https://e-callisto.org/Software/jv_20070109.jar

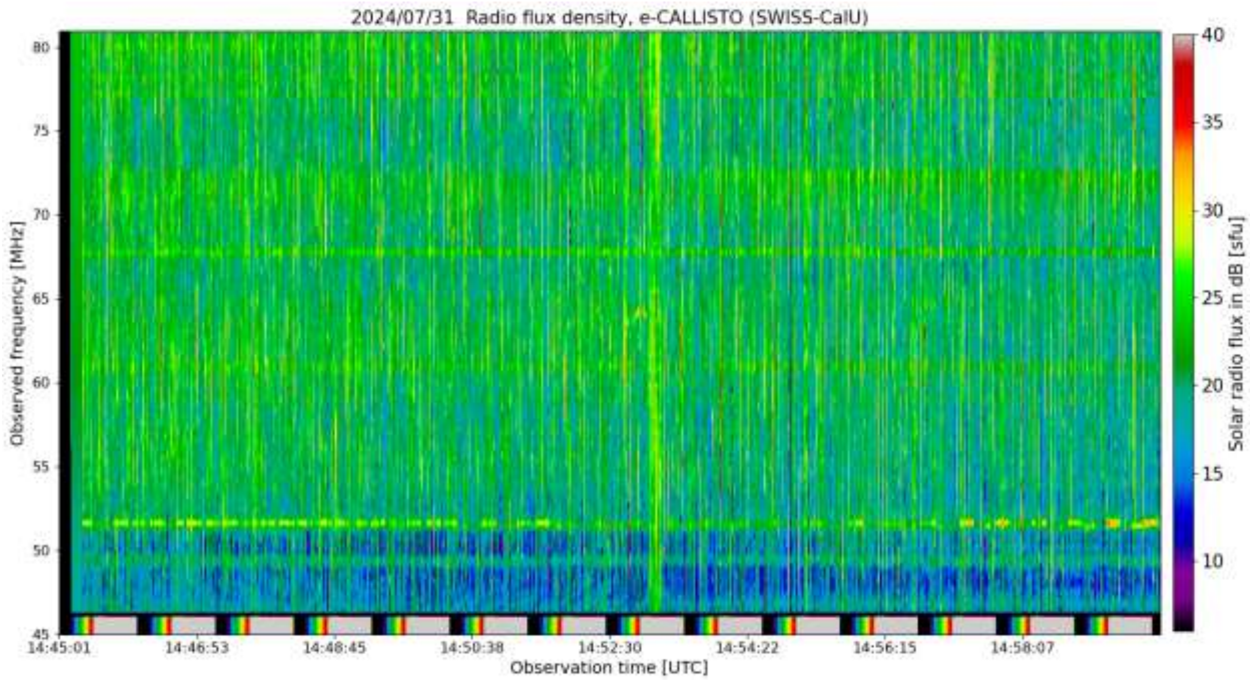

Fig. 16: Heavy lightning in the afternoon at a distance of 1…2 km. On the far-left side we can see 10 seconds Tcold with ~300 kelvin (black), followed by Thot ~12’000 kelvin for another 10 seconds (green). Antenna temperature of lightning strikes exceeds 12’000 kelvins.

Real-time control process MOS-ctrl.pyw:

A Python script running permanently takes care about control of the calibration unit CalU. Every modulo 15 minutes a calibration sequence is send to the calibration control unit CuC via USB-interface. Fist 10 seconds switch to 50Ω termination resistance of cold noise source at ambient temperature -> Tcold, usually in the order of 300 kelvins. In the standard calibration script this reference temperature is set to an average constant value. For more precise calibration ambient temperature should be measured and integrated in the calibration script (not implemented yet). Second 10 seconds the internal noise source is powered with 15 volts. At seconds 20 noise source power is switched off and rf-switch is changed to antenna for remaining 14 minutes and 40 seconds.
Parameter in script:
- Communication port, in my case ‘COM13’
- Print-/Logging-mode, in my case False. Set to True for testing and experimenting
- Calibration period, in my case 15 minutes
- Time for Tcold and Thot fix coded in state-machine of Python-script to 10 seconds

Real-time calibration process calibrate.py:

A Python script calibrate.py which is triggered modulo 15 minutes at minutes 01, 16, 31 and 46 takes care about calibration of raw FIT-file from CALLISTO. Input parameter are stored in file configCalU.ini
This configuration file contains following calibration parameter:
- Path where FIT-files are primarily saved by CALLISTO
- Ambient temperature = reference temperature = Tcold, in my case 25°C
- Minimum Y-factor taken from calibration steps, in my case 9 dB
- Logging flag, whether Y-factor shall be logged, in my case True
- Antenna gain as a table with one array for frequency and a second table with gain in dB. Tables MUST cover frequency range of observation for later interpolation during calibration.
- Calibration tables of internal noise source, one for frequency and a 2nd one for ENR in dB. Tables MUST cover observation frequency range.

Calibration steps are:

1. Read all calibration parameter from configuration file
2. Get a list of all FIT-files in designated folder
3. For each FIT-file check if unit != 'sfu', then perform calibration
4. Define start- and stop-time for Tcold and Thot observation
5. Read data-array and frequency array from FIT-file
6. Initiate interpolation of excess noise ratio and antenna gain for frequency axis of FIT-file
7. Convert raw data into dB
8. Read calibration data Icold and Ihot, representing Tcold and Thot from FIT-file and average
9. Calculate Y-factor of calibration sequence and perform logging in case of debug = True
10. Convert dB into linear scale for Iref=Icold, Ihot and Iantenna
11. Calculate Tref=Tcold and Thot
12. Check if Thot>Tcold. In case of interference it might be wrong
13. Calculate antenna temperature Tant = (Thot - Tcold) / (Ihot - Icold) * (Iant - Iref) + Tref
14. Calculate wavelength lambda from frequency array out of FIT-file
15. Calculate flux as S = 8 k pi / (G * lambda**2) * Tant * 1e22 with k = Boltzmann constant
16. Clip flux data between 10 sfu and 450'000 sfu (~56 dB sfu)
17. Perform logarithmic compression Sc = 45 * log10(S) and squeeze into 8-bit resolution
18. Update data-array with flux instead of digits and update unit with 'sfu'

After calibration at times modulo 15 minutes at minutes 02, 17, 32 and 46 calibrated FIT-files are uploaded to the central server in Switzerland by FTP. Then the file is moved to a local backup-folder. This process is activated by the application ssfree.exe from https://www.bytesin.com/software/Download-System-Scheduler/ in case of Windows computer. In case of LINUX crontab can be used to trigger calibrate.py via Python3.

Reading and plotting calibrated FIT-files in Python:

After reading a calibrated FIT-file the compressed data need to be de-compressed by the following steps:

1. Read the FIT-file as usual
2. Decompress S_linear = 10** (data/45.0) – 10
3. Plot if you prefer linear scale [sfu]
4. In case of logarithmic scale perform S_dB = 10*log10(S_linear + 0.1) # 0.1 to avoid log10(0)
5. Plot flux in dB [sfu]
6. In all plot cases you need to play with colour-table and colour-table clipping parameter vmin and vmax to get best visual quality of the calibrated observation. Also relevant regarding visual quality is whether you subtract background or not, see figures 12 and 13.

In case of quick-view plotting with JavaViewer, de-compression is performed automatically. For JavaViewer, get it from here: https://e-callisto.org/Software/jv_20070109.jar

Prize indication, status September 2024:

Qty	Unit	Fully tested in enclosure	PCB populated
1	Calibration unit CalU	350 CHF	Not supported
1	Calibration control unit CuC	610 CHF	310 CHF
1	Control SW: MOS-ctrl.pyw	Free of charge	N/A
1	Calibration SW: calibrate.py	Free of charge	N/A
1	Shipping cost per one set CalU+CuC	< 150 CHF	< 150 CHF

Table 5: Preliminary cost estimation

CalU can only be delivered with enclosure, with feed-through, with LEDs and with SMA-connectors. Otherwise neither any tests nor cross-calibration of internal source can be performed. But control unit CuC can be ordered without any connectors, without enclosure and without power adapter for those cases where the customer wants to integrate PCB in an existing unit with its own power supply and connectors.
Ask for shipping cost, it depends on country of delivery and type of delivery (economy or urgent). In case of several units you may add another 90 CHF for appropriate shipping container.

Appendix: Configuration of System Scheduler ssfree.exe for Windows users:

System scheduler is used to trigger calibration.py at hh:01, hh:16, hh:31 and hh:46 as well as to trigger FTP-upload to central server at hh:02, hh:17, hh:32 and hh:47 whereas 00 < hh < 24.

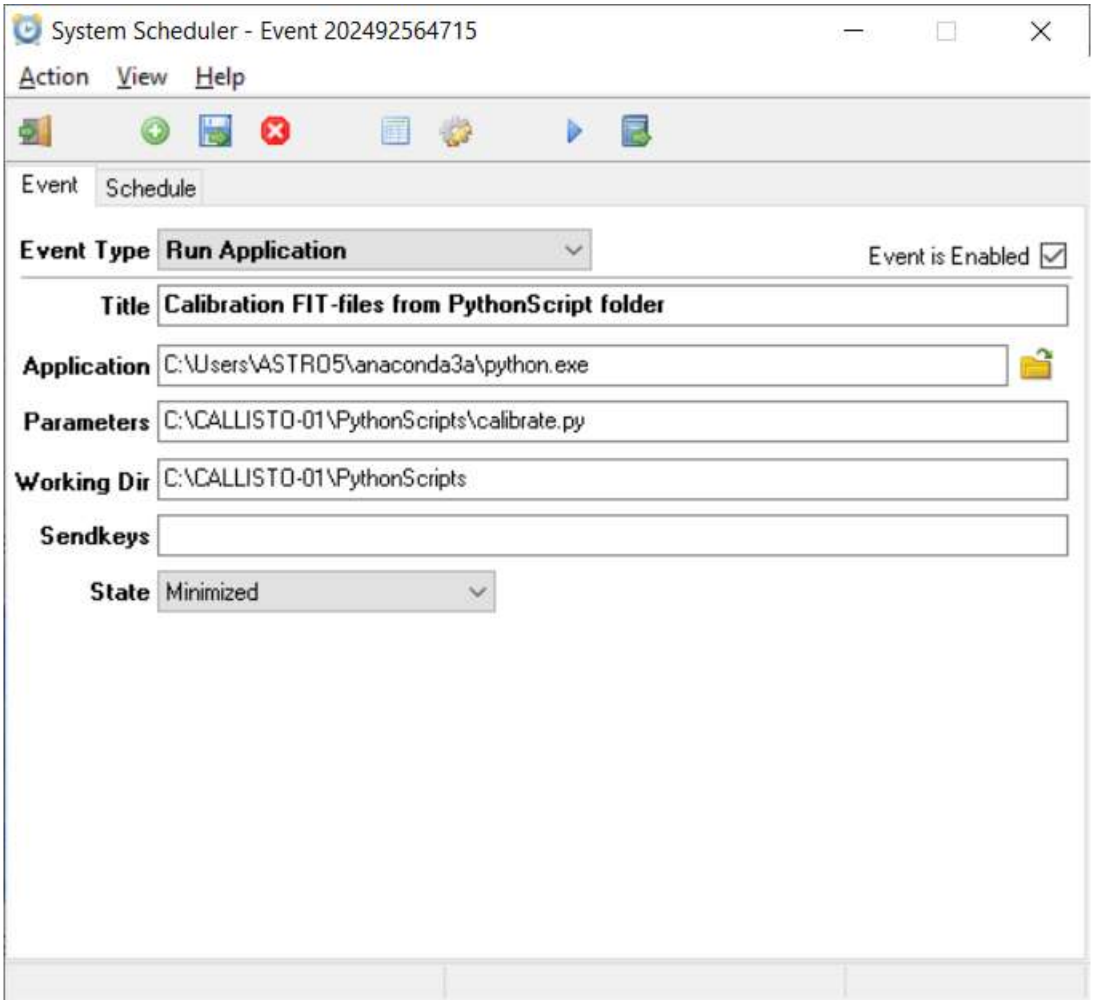

Fig. 17: Application configuration for FIT-file calibration. Path to Python.exe might be different for every user. You need to search for python.exe using file-explorer. Parameter-folder as well as working folder might also look different, depending on chosen drive and instrument-code = focus-code (00…63).

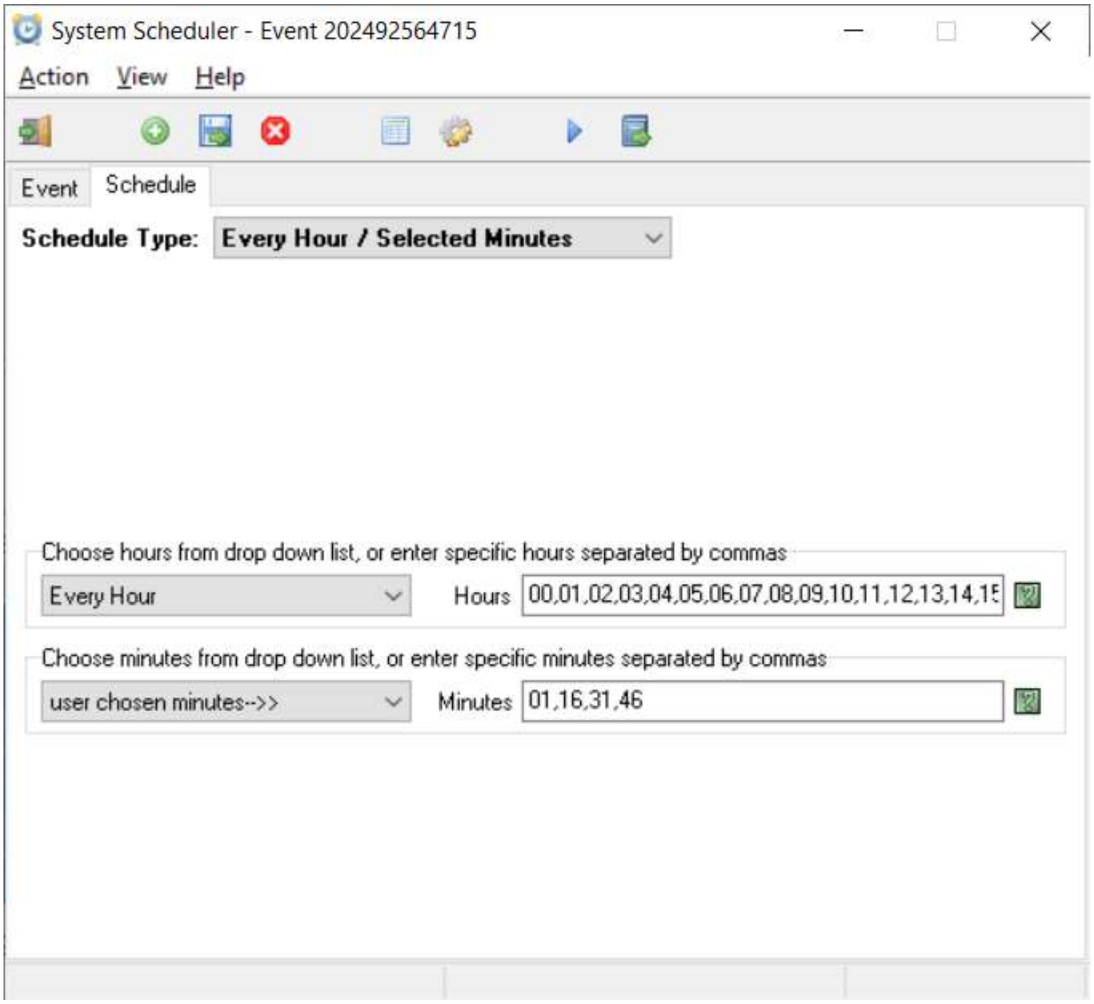

Fig. 18: Scheduling for calibration process

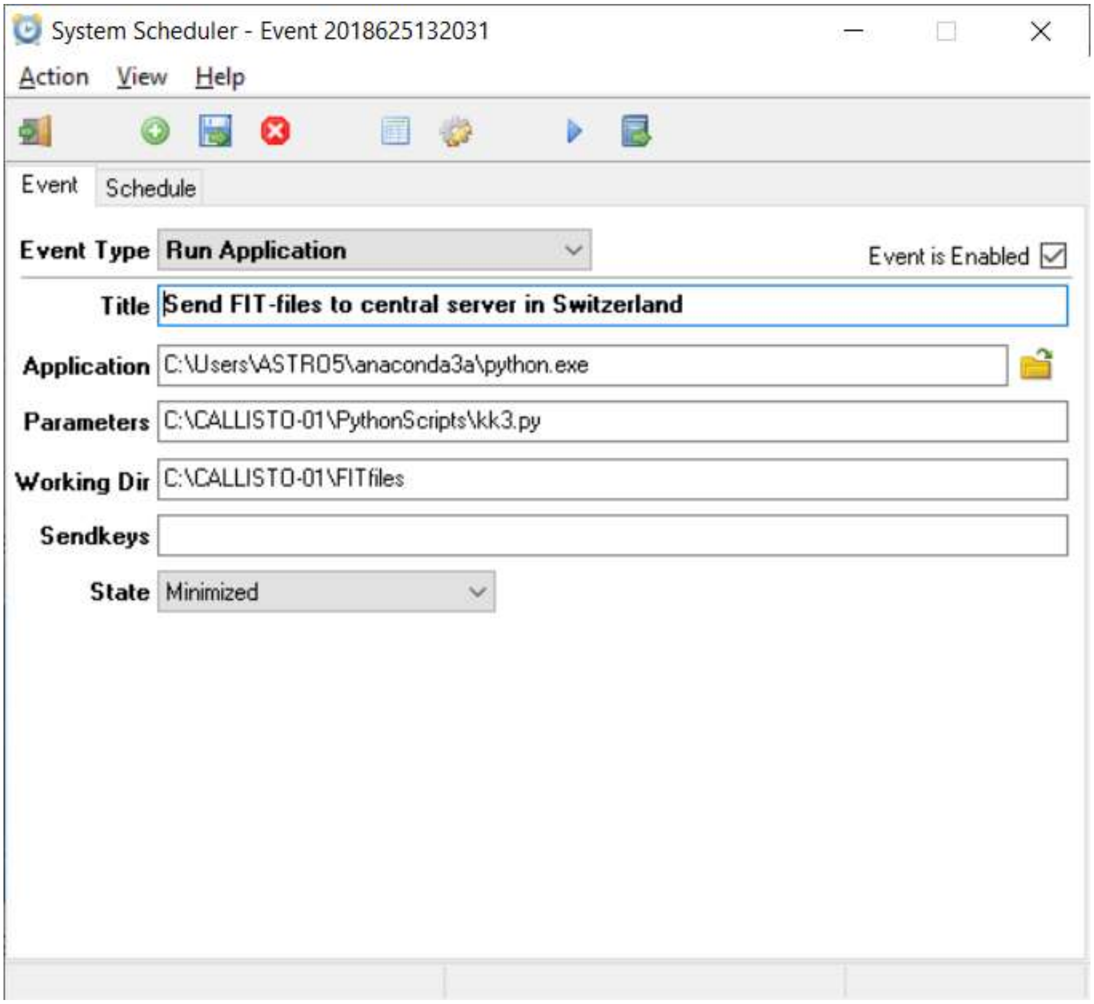

Fig. 19: Application configuration for FIT-file upload to central server. Path to Python.exe might be different for every user. You need to search for python.exe using file-explorer. Parameter-folder as well as working folder might also look different, depending on chosen drive and instrument-code = focus-code (00…63).

After final commissioning Python script kk3.py will probably be renamed to UploadFTP.py [tbd]

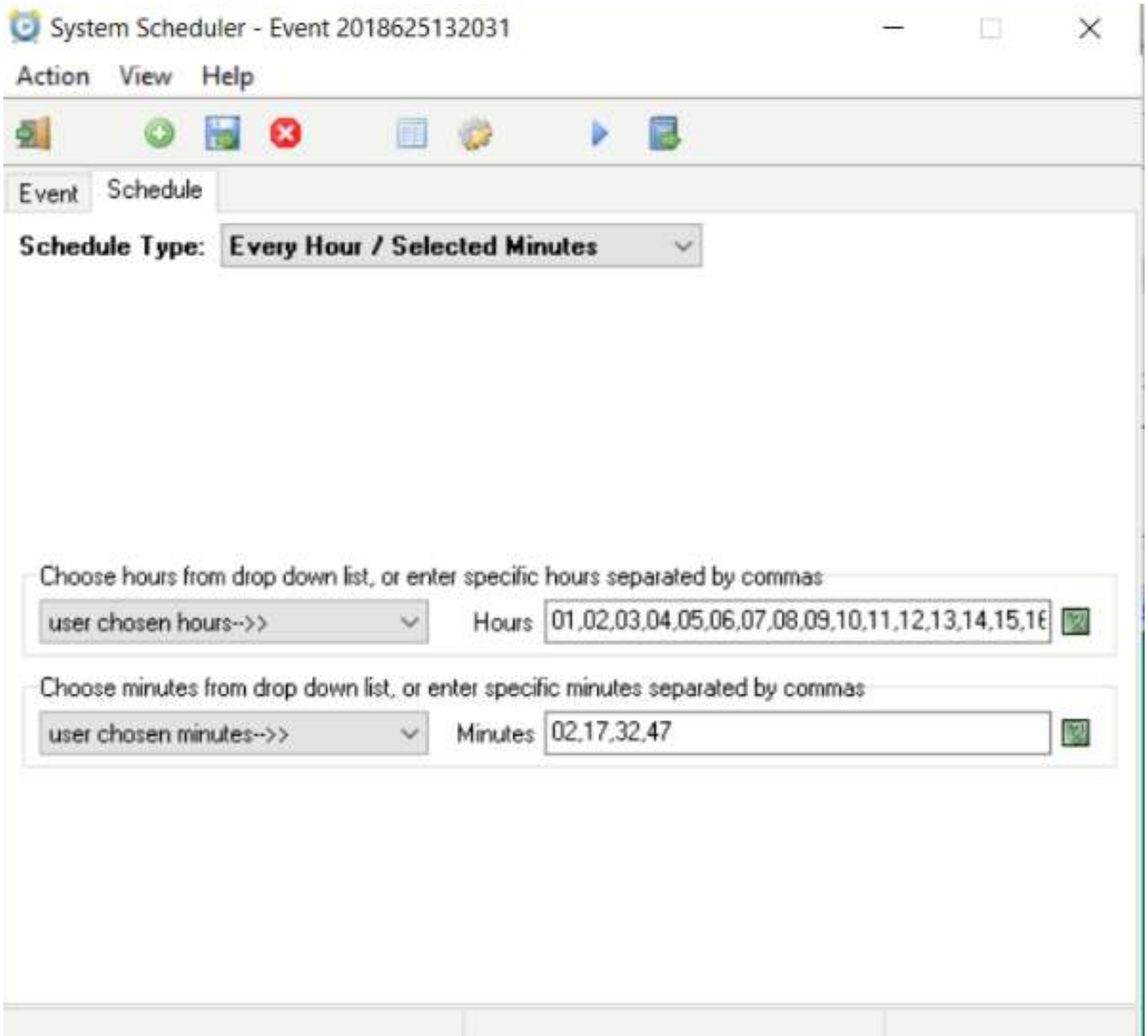

Fig. 20: Scheduling for FTP-upload

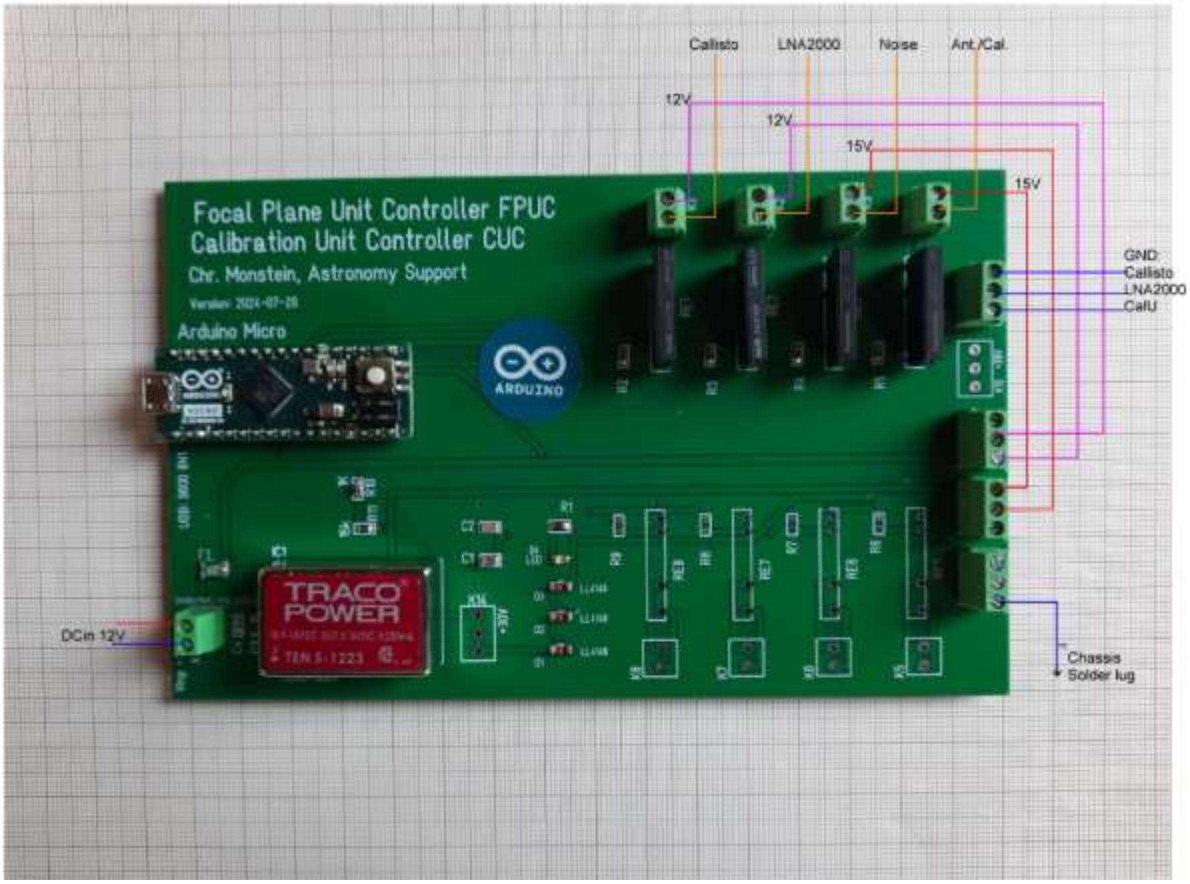

Fig. 21: Harnessing of CuC inside aluminium enclosure. Be aware that all supply voltages (purple and red wires) have to be connected the outer contacts of K1…. K4 due to internal diode of the MOS-relay. Inner contacts (orange wires) need to be connected with the connectors at the back-plate of the enclosure! Blue wires denote to ground connections.

Version control:

Version	Date
Preliminary draft CalUandControl.docx	09.09.2024
CalUandControl-V01.docx	27.09.2024
CalUandControl-V02.docx	