Philip's Lab book

Week 1

14.09.2020 - 18.09.2020

Onboarding

Registration of a new D-PHYS account

1. Go to this address to activate a new physics account: https://account.phys.ethz.ch/new/from_nethz
2. You will receive the login data in a file in poly box (no need to go to the Help-Desk)
3. Then you can log in here: https://account.phys.ethz.ch/login
4. If you are added to the qo-folder you can access it via: smb:*group-data.phys.ethz.ch/qo (MacOS: To connect to this remote-folder go to: Finder→Go to→Connect to Server) - You can access your personal folder in the same way via: smb:*home.phys.ethz.ch/USERNAME

Week 2

21.09.2020 - 25.09.2020

Semester Project

Presentation of the semester project: Fast Digital Notch Filter

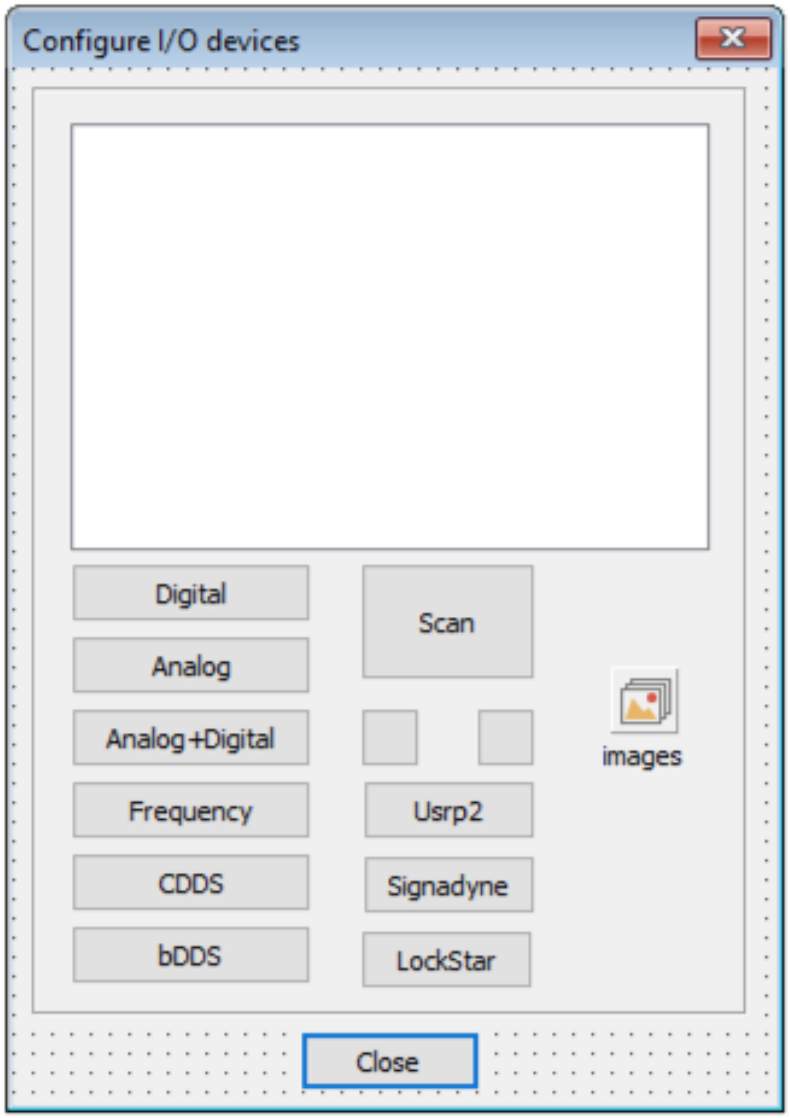
Abstract

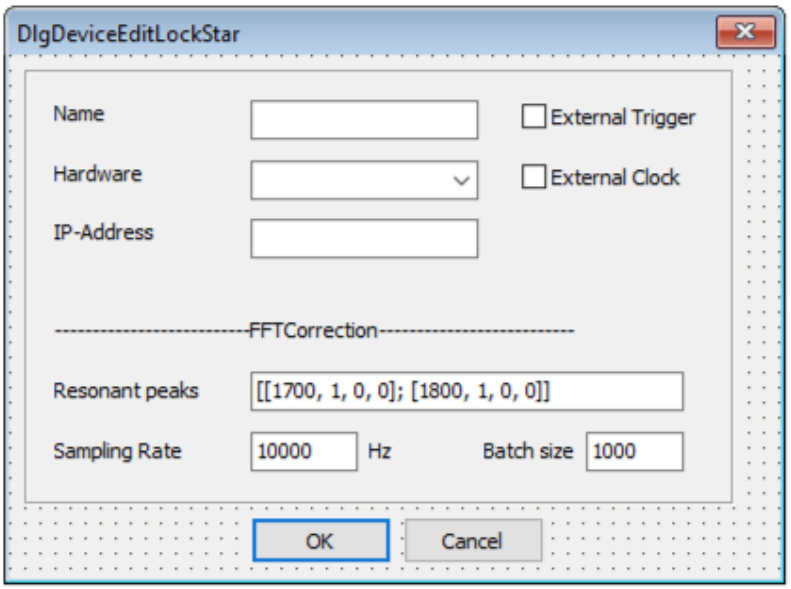
During the semester project a fast digital notch filter on a microcontroller-based feedback device 'LockStar' was implemented for improved frequency stabilization in optical cavities. With this filter algorithm the device is able to cancel or modulate multiple resonant frequencies at once and deliver real-time feedback to the system. Due to the customizable initialization parameters the algorithm can be easily applied to various different situations without any additional development while reaching suppressions up to -60 dB with a sampling rate of up to 100 kHz.

Week 3

28.09.2020 - 02.10.2020

Matrix-LockStar-Interface





New Trap Geometry Project

Relevant references for the calculation of a new trap geometry with Gaussian laser beams:

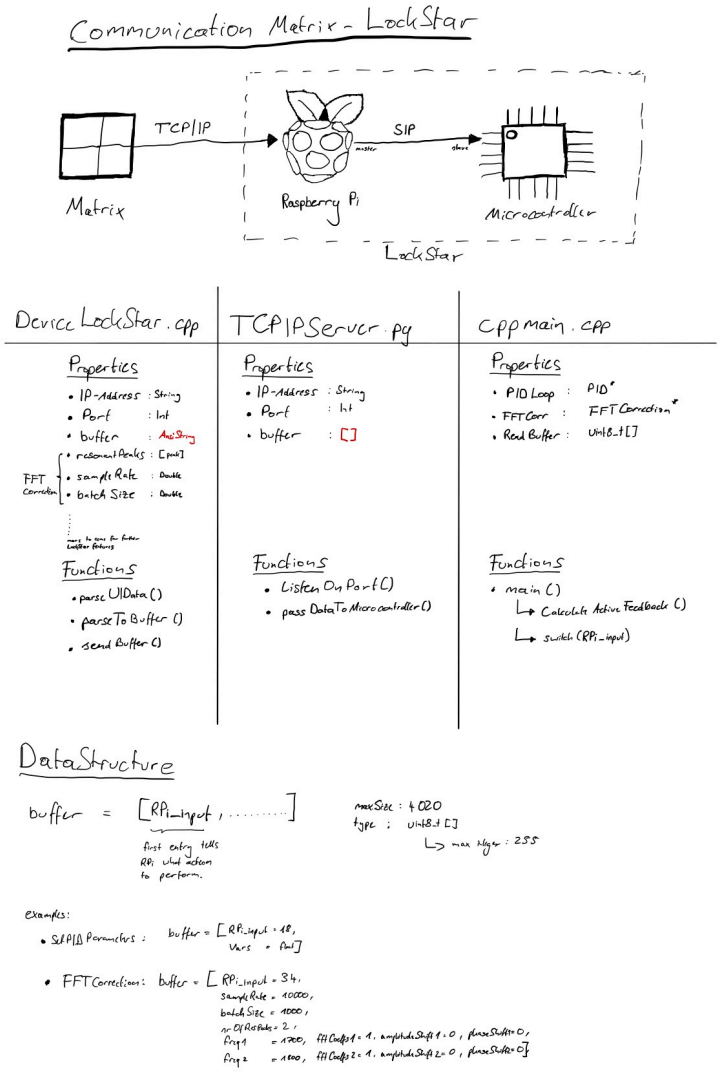
- [OPTICAL DIPOLE TRAPS FOR NEUTRAL

ATOMS](https://www.sciencedirect.com/science/article/pii/S1049250X0860186X)
- [Optical tweezers](https://en.wikipedia.org/wiki/Optical_tweezers)
- [Gaussian beam](https://en.wikipedia.org/wiki/Gaussian_beam)
- [Bose-Einstein Condensation of Atoms in a Uniform Potential](https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.110.200406)

Week 4

05.10.2020 - 09.10.2020

Matrix-LockStar Communication



Implementation

The main issues with the communication of the matrix and the TCPIPServer were the following two functions in the ExpWiz Project (the Matix Source Code):

Control/IODevices/TCPIPDevice.cpp

```
void __fastcall TCPIPDevice::instrOpen() throw (EHardwareException) {
//   socket = new TClientSocket(); // Was commented out? Fuck embarcadero.
    socket = new TClientSocket(NULL); // ---- This line is new
    try {
        runner->hostname = ipAddress;
        runner->port = tcpPort;
        runner->startSession(socket);
        stream = new TWinSocketStream( socket->Socket , TCPIP_TIMEOUT );
    } catch (Exception& e) {
        throw EHardwareException( "Cannot open TCPIP session with bDDS at "
+ ipAddress + ": " + e.Message, NULL);
    }

} // instrOpen
```

- Error code: HARDWARE ERROR: Cannot open TCPIP session with bDDS at 192.168.0.12: Access violation at address 007DC556 in module 'ExperimentRunner.exe'. Write of address 65676E75
- Reason for error: The TClientSocket was never initialised.
- Solution: Initialise the TClientSocket with parameter NULL! (Not as the commented-out line above, where someone seems to already have come across the same error earlier.)

Common/RemoteRunner.cpp

```
void __fastcall RemoteRunner::startSession( TClientSocket* socket ) throw
(ESocketError) {

    socket->Tag = (int)this;
    socket->Host = hostname;
    socket->Port = port;

    socket->ClientType = ctBlocking; // ---- This line is new

    socket->Open();

} // end startSession
```

- Reason for error: The TClientSocket needs the Socket to be in Blocking Mode.
- Solution: Set the ClientType to ctBlocking. (There is also a non-blocking mode called ctNonBlocking, which can not be used since TCPIPDevice is using TWinSocketStream.)

Now the basic TCPIP communication between the Matrix and a python server over a given ipAddress and Port works!

Example for simple python server

```
import asyncio

async def handle_echo(reader, writer):
    data = await reader.readline()
    message = data.decode()
    addr = writer.get_extra_info('peername')
```

```
    print(f"Received {message!r} from {addr!r}")

async def main():
    server = await asyncio.start_server(
        handle_echo, '127.0.0.66', 10785)
    addr = server.sockets[0].getsockname()

    print(f'Serving on {addr}')

    async with server:
        await server.serve_forever()

asyncio.run(main())
```

Week 5

12.10.2020 - 16.10.2020

Matrix-LockStar Communication

Running the Matrix-RPI-Microcontroller Communication requires both the matrix computer and the Lockstar being in the same (local) network.

To test the communication I also installed the matrix on the qo-impact-locks.dhcp.phys.ethz.ch computer. Installing the matrix requires the following steps:

1. copying the ExpWiz folder and performing step
2. Performing Step 2 in the Setting up an Experiment Runner computer on Windows 10 section on [Nuclino ExpWiz](https://app.nuclino.com/ExpWiz/ExpWiz/Experiment-Wizard-6a5ca12f-5a68-4f59-8c02-1caa5cd28303)

In order to communicate from the Raspberry Pi to the Microcontroller one has to do the following:

```
from spi import *

MCU = MCU_Handler(None, speed=20000000)

DP = DataPackage()
DP.addUInt8(Command_SetFFTCorrectionParameters)
DP.addFloat(10000) # sampleRate
DP.addFloat(2000) # batchSize
DP.addFloat(2) # nrOfResonantPeaks
DP.addFloat(1800) # freq
DP.addFloat(1) # fftCoeffs
DP.addFloat(0) # amplitudeShift
DP.addFloat(0) # phaseShift
```

```
DP.addFloat(1900) # freq
DP.addFloat(1) # fftCoeffs
DP.addFloat(0) # amplitudeShift
DP.addFloat(0) # phaseShift

MCU.append_write_queue(DP)

MCU.get_data()

time.sleep(3.0)
MCU.stop_event.set()
```

In this example the DataPackage sent to the Microcontroller contains FFTCorrection parameters, which are then updated on the microcontroller.

Next Steps

Implement the DeviceLockStar on the Matrix also as an AnalogOutputDevice (similar to the DevicePCI67xx) in order to enable ramps, etc. ...

Matrix Internal Clock Test

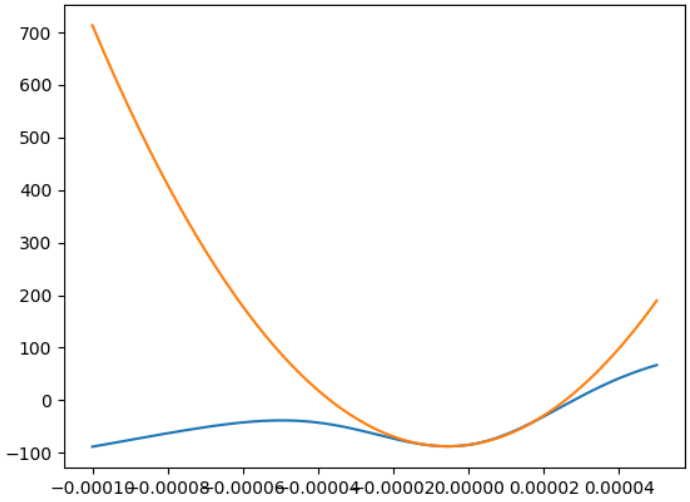
In order to decide whether the internal clock of the microcontroller is exact enough for working together with the matrix, following test was performed:

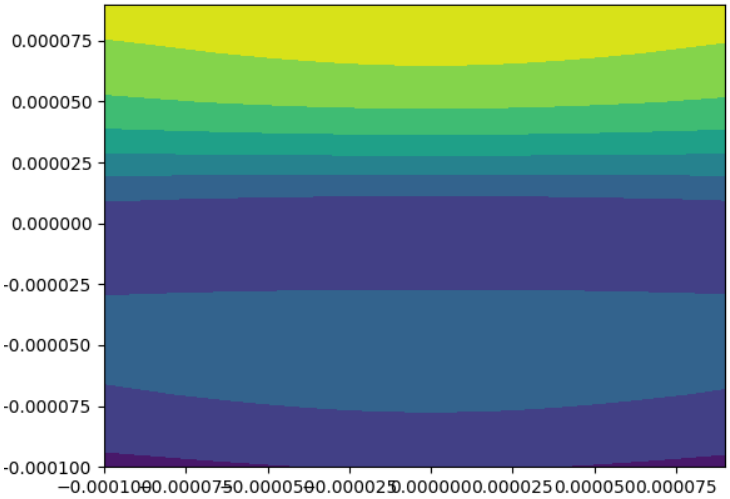
1. Use the DigitalOutHigh() and DigitalOutLow() on the microcontroller to generate a square function on the Picoscope
2. Use a function generator to generate also a square function on the Picoscope (with the same frequency as the first one)
3. Trigger on one of these functions on the Picoscope
4. Determine the drift, phase shift, absolute time difference ...

Trap Geometry Project

Some figures form the dipolepotentials.py:

(Work in progress, thus no axis names etc. ..)





Additional Resources:

- [Davide Dreon PHD Thesis](https://tel.archives-ouvertes.fr/tel-01571420v2/document)
- [Degenerate Bose Gases: Tuning Interactions & Geometry](http://www-amop.phy.cam.ac.uk/amop-zh/Publications/alg51_thesis.pdf)
- [Two-Dimensional Homogeneous Fermi Gases](https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.120.060402)

Week 6

19.10.2020 - 23.10.2020

Matrix-LockStar Communication

In order to implement the DeviceLockStar also as AnalogOutputDevice one has to implement the following functions:

- generateEvents()
- prepareRun()
- run()
- afterRun()
- stop()

For a more detailed overview see [ExpWiz Documentation Nuclino](https://app.nuclino.com/ExpWiz/ExpWiz/General-Concept-of-ExpWiz-7183886c-898e-4987-bae1-f5414e541a72) - ExperimentRunner.VSD

New Computer & Embarcadero Update

Since the current computer for the ExpWiz development is rather slow, a new computer was installed. During that process Embarcadero was updated from 10.3 to 10.4.

This switch to the new computer and the update to the new Embarcadero version caused the ExpWiz to no longer compile. Following errors arised:

1. stdlib.h, itoa
2. [ilink32 Error] Fatal: Unable to open file 'SDLCHARTPACK_RT_106XR.LIB'
3. [ilink32 Error] Fatal: Unable to open file 'VCL.IMAGING.JCONSTS.OBJ'
4. ...

Status: Still unsolved ...

Week 7

26.10.2020 - 30.10.2020

Embarcadero Error

- Ordered & installed the new SDL Version 10.7 (to replace the 10.6 version) → did not work

Status: Still unsolved ...

New Trap Geometry Project

Wrote a python script to simulate the following:

- Gaussian beam (spherically symmetric)
- Gaussian beam (ellitpic)
- Dipole trap (1 spherically symmetric Beam)
- Dipole Trap (2 crossed spherically symmetric Beams)
- Impact Lab (2 crossed elliptical beams)
- Thomas-Fermi radii (harmonic potential)

All simulations include derivations of the most important properties.

Week 8

2.11.2020 - 6.11.2020

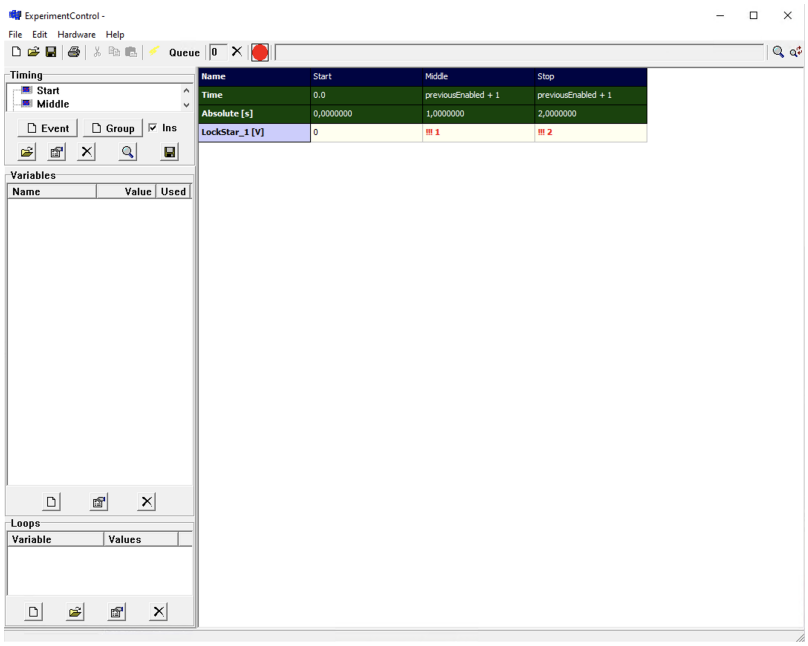
Embarcadero Error

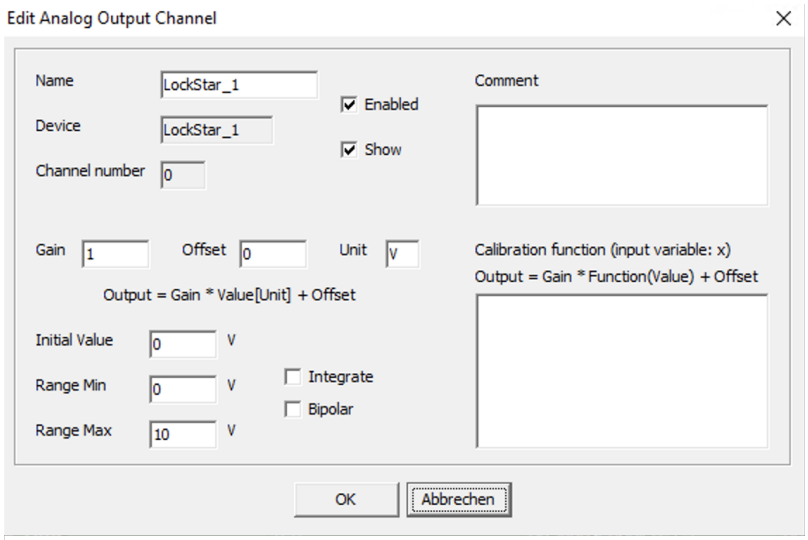
Solved via the following steps (for the installation on the new Windows 10 Computer):

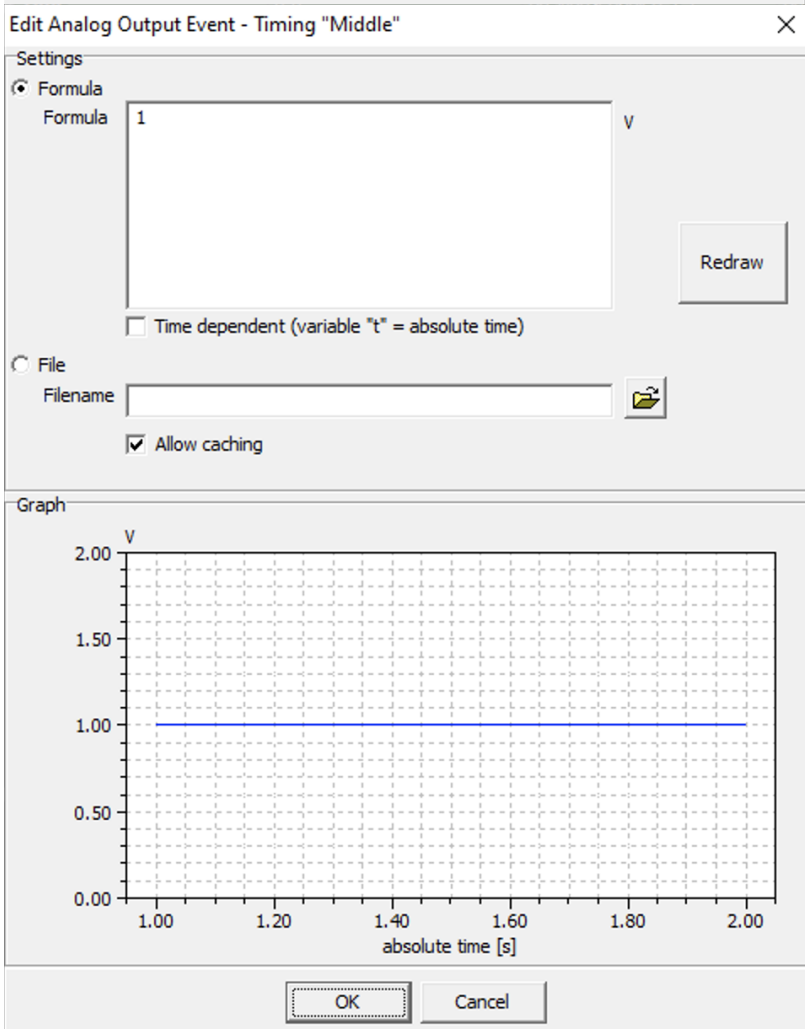
1. Install Embarcadero Version 10.3 (not the 10.4 version)
2. Install SDL Package 10.6 (not the 10.7 version)
3. Copy&Paste the original ExpWiz folder from the old to the new computer (since git ignore might cause some problems while using git)
4. Go to **Tools→Options→Language→Delphi→Library** and make sure to copy the Library path and Browsing path from the win64 version (also see: https://stackoverflow.com/questions/5656975/delphi-xe-f1027-unit-not-found-system-pas-or-binary-equivalents-dcu-upon)

Status: Solved!

Matrix-LockStar as AnalogOutputDevice







Week 9

9.11.2020 - 13.11.2020

Matrix-LockStar as AnalogOutputDevice

- Floating Point error: Solved

- Low Memory Problem on LockStar: Started implementing variable sampling rate
- …

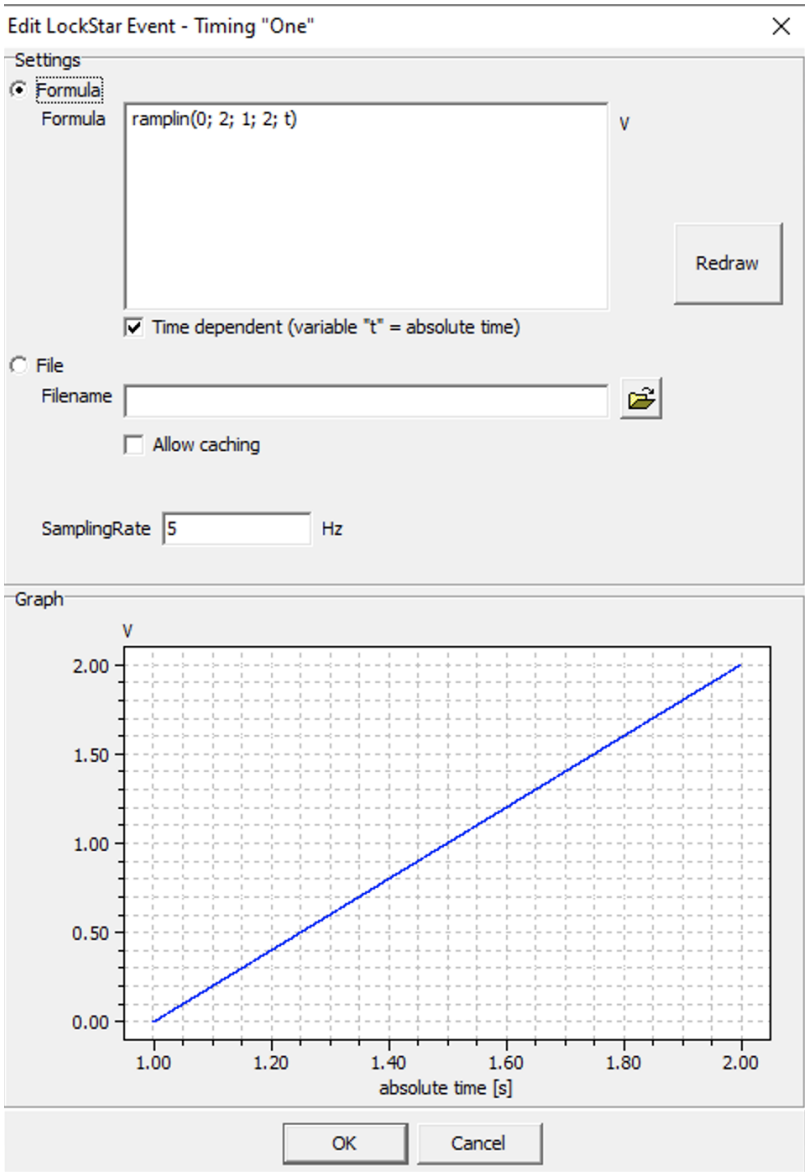
Week 10

16.11.2020 - 20.11.2020

Matrix-LockStar as AnalogOutputDevice

The current status of the Matrix-LockStar integration for using the LockStar as an AnalogOutputDevice is the following:

- In order to store and display the variable samplingRate in the User Dialog in the matrix, the following new classes (similar to the AnalogOutputDevice) were implemented:
    1. GUIChannelEditLockStar
    2. LockStarChannel
    3. LockStarEvent
    4. LockStarEventFormula
    5. LockStarEventFile



- In the backend of the Matrix (in DeviceLockStar) the variable samplingRate is used to drastically decrease the amount of data which is being sent to the LockStar, since the LockStar has only about 100kB of available RAM (flash memory would be too slow to read out for the real time operation). Since the DeviceLockStar is implemented in analogy to the AnalogOutputDevice, it has the same general features (such as output range checking or arbitrary formula parsing).
- The Voltage-Data which is being sent to the LockStar is formated in the following way: [time-step-1, voltage-1, time-step-2, voltage-2, ..., time-step-n, voltage-n]
- Then the Matrix-Server running on the Raspberry Pi passes this data to the Microcontroller. The Microcontroller saves the data in a vector (voltageOuts) and outputs the voltages via DAC_1 at the corresponding time steps.
- Additionally for each output linear interpolation is performed (https://en.wikipedia.org/wiki/Linear_interpolation)

Week 11

23.11.2020 - 27.11.2020

Build own Test-LockStar

Since the existing LockStars (installed in the IMPACT Lab) can only be debugged via a single LED on the frontpannel (Note: or also via connecting a ST-Link, which was not possible since the LockStars could not be removed from the rack), the idea was to build an additional Test-LockStar which consists only of the MC and the RPi, but can then be debugged via the STM32CubeIDE (using a simple USB Connection directly to the MC).

Setup of the RPi

1. Install the 2020-02-13-raspbian-buster-full.img file (located in: qo-impact-locks.dhcp.phys.ethz.ch → Windows (C:)→ Development → Raspberry Pi Images) on a MicroUSB Card using the Programm: Raspberry Pi Imager.
2. Insert MicroUSB into RPi and startup (also connect a screen via HDMI, Mouse and Keyboard if needed)
3. Enable SPI communication on the RPi via:
    1. sudo raspi-config
    2. Interfacting Options → SPI
    3. Click on YES to enable SPI connection and exit
4. Ask ISG#Helpdesk to enable network connection via Ethernet
5. Install the python package: spidev (pip install spidev)
6. To use the RPi via Remote Desktop do the following:
    1. sudo apt-get update
    2. sudo apt-get install xrdp
    3. hostname -I

Setup of the Microcontroller

1. Use a usb-connection to de microcontroller itself in order to flash it directly form the STM32CubeIDE (instead of flashing use the Debug-Mode in STM32CubeIDE).
2. Connect SPI cables to RPi. Make sure that all the pins on the MC are working (which was not the case for the one I tested). if necessary use a Oscilloscope to visualise the SPI signals.
3. Use breakpoints etc. to debug the Microcontroller

Week 12

30.11.2020 - 04.12.2020

Final Communication Protocol Matrix->LockStar

Todo: Write Documentation here in Wiki

Build LockStar-Rack

Parts List

The following list represents all the parts which are needed in oder to assemble a LockStar-Rack

(which can contain up to 4 LockStars). Not included in this list are the the parts for the LockStars itself (which can be mounted later), and also not included are the parts for the power supply (it is assumed to be already fully assembled, since it has to be mounted in the rack).

Pos	Amount	Device Name	Shop Number	(Specification)	Note
1	1	System case	30411.1	-	-
2	2	Threaded rail.	30411.4	M2.5	Also the shorter ones are possible.
3	1	Rearpanel	-	QO-WSO-21638	Has to be ordered extra. It is generally included in the System case, but whithout the correct holes.
4	1	Panel-mount appliance inlet	30298.1	-	-
5	1	Ethernet-cabel	-	Adafruit 4215	Has to be ordered extra.
6	1	Connector	30115.1	-	-
8	8	Flat plugs insulated (blue)	30117.2	-	-
9	8	Distance Bolts	20236.22	M3 x 6	-
10	8	Distance Bolts	20236.25	M3 x 12	-
11	8	Distance Bolts	20236.33	M3 x 40	-
12	8	Screws	20417.15	M3 x 16	-
13	8	Screws	20417.19	M3 x 35	-
14	4	Screws	20417.4	M2.5 x 8	-
15	1	Cable (blue)	30230.4	∅1.5mm², ~0.2m	-
16	1	Cable (white)	30225.32	∅1.5mm², ~0.2m	Also ∅1.0mm² is possible.
17	1	Cable (green-yellow)	-	∅1.5mm², ~0.5m	Get from Alex F. Office
18	1	Cable (black)	30225.33	∅1.0mm², ~0.2m	-
19	1	Cable (orange)	-	∅1.0mm², ~0.2m	Get from Alex F. Office
20	1	Cable (red/pink)	-	∅1.0mm², ~0.2m	Get from Alex F. Office
21	1	Cable (blue)	-	∅1.0mm², ~0.2m	Get from Alex F. Office
22	1	Power Supply LockStar	-	v3.1	Has to be built separately.
23	3	Cable ties	30309.5	-	-
24	1	Screw nut	20177.9	M4	-
25	1	Screw	20144.9	M4 x 8	-
26	3	Ground Slug	20210.4	M4 x 4.3	-
27	1	Insulator Plate (red)	-	-	Has to be ordered extra.
28	2	Screw	20417.12	M3 x 8	-
29	2	Screw nut	20177.8	M3	-
30	1	Circuit board connector	30436.1	2 Outlets	-
31	1	Circuit board connector	30436.2	3 Outlets	-
32	1	LockStar Back Plane	-	-	Has to be ordered extra. This is the plane to wich the 4 LockStars will be connected to.
33	1	Compression cable lugs (blue)	30273.5	-	-
34	1	Screw (black)	20132.28	M4 x 8	-
35	1	Screw nut	20189.2	M4	-

Shop: https://lager.phys.ethz.ch

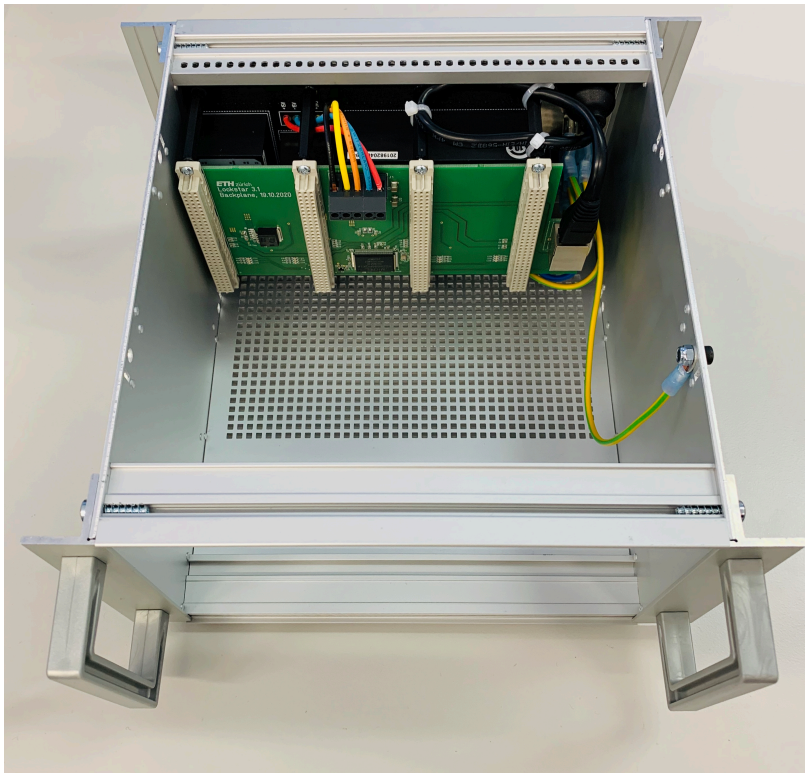
Tools for assembling

Steps to assemble

Follow the instructions in the following document (“Z:\Library\Electronics\QO-EL-0097 MCU Lock aka LockStar\v3\HOWTOBUILDALOCKSTARRACK.pdf”). Be aware of the following steps, which are missing in this Manual:

1. Use the items list above, since several parts are missing in the manual
2. Solder the two circuit board connectors to the LockStar Back Plane
3. Before mounting the Power Supply on top of the Insulator Plate, make sure to solder the 5 cables to the Power supply (since after mounting this becomes quite hard.)

Final result



Test the rack

Attention, before using the rack in any way (even plugging-in) the DIN/VDE 0701 check has to be performed.

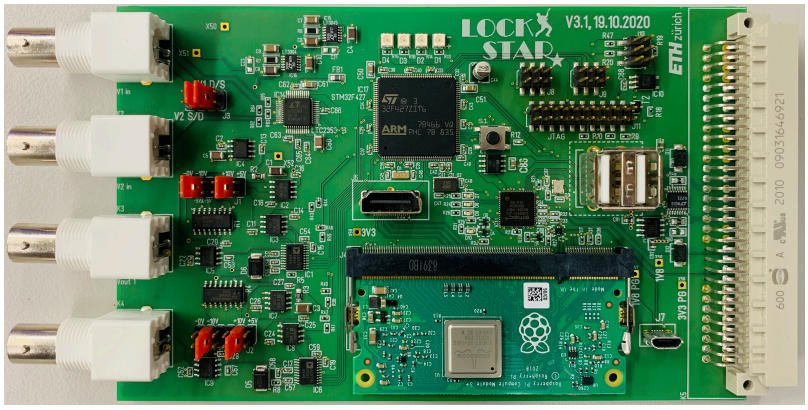
Week 13

07.12.2020 - 11.12.2020

Build new LockStar

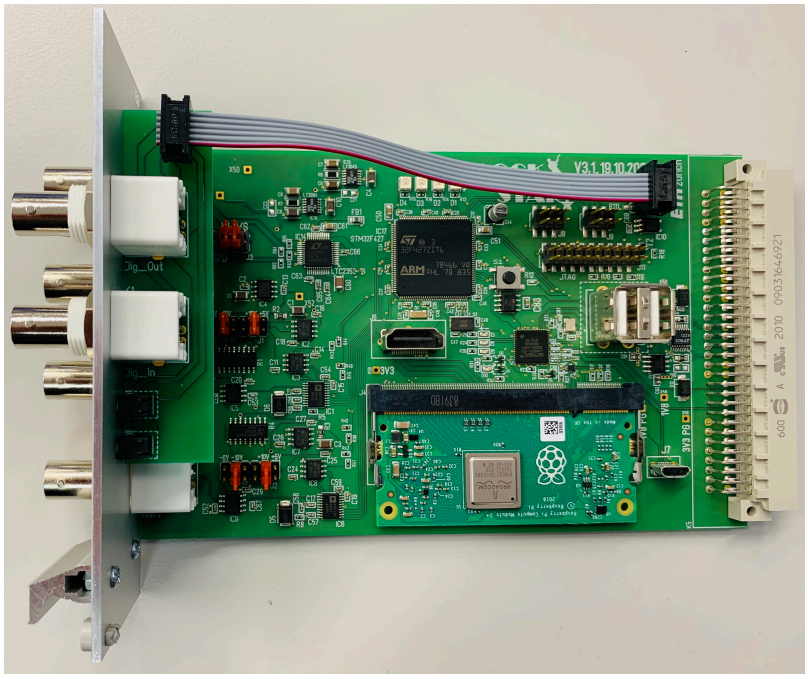
Minimum requirements (without the Digital-In&Out):

1. Connect 6 x Jumpers (see image)
2. Connect Raspberry Pi (Compute Module 3+ 32GB)



Full LockStar (with Digital-In&Out and Frontpanel):

1. Solder the 2 x BNC, 2 x LED and 1 x Header to the separate digital-platine
2. Connect this platine using the 1 x cable and 2 x Cable connector
3. Mount the Frontpannel-handle with 2 x screws and 2 x screw nuts
4. Mount the front panel to the LockStar-Card and the Digital-In&Out Platine



Flash the Raspberry Pi on the LockStar

This step requires the following prerequisites:

1. Fully assembled LockStar-Power-Rack
2. Fully assembled LockStar - card (can be without the Digital-In&Out)
3. Windows Computer

The following steps have to be performed:

1. Install RPiBoot.exe on the Windows Computer (download from here: https://www.raspberrypi.org/documentation/hardware/computemodule/cm-emmc-flashing.md)
2. Install Win32DiskImager on the Windows Computer (download from here: https://sourceforge.net/projects/win32diskimager/)
3. Connect Micro-USB to J7 on the LockStar Card and to the Windows Computer
4. Insert LockStar Card into the Rack
5. THEN connect the power cable to the LockStar-Rack
6. Execute the RPiBoot.exe and wait for it to finish (should take a couple of seconds)
7. In case a Window pops-up which asks to format the Raspberry Pi - just ignore (this is done via the Win32DiskImager)
8. Use Win32DiskImager to finally flash the Raspberry Pi with a new image (Select 2020-03-23-backup.img as image, select the right device and click write)
9. The flashing might take a while (several minutes) ... after finishing disconnect the MicroUSB cable and unplug-replug the rack

Note: Do not use the 2020-03-23-raspbian-buster-full.img as image, since this is only the empty RPI image without any required Packages (like for the SPI connection: openocd, or for the Remote

Desktop connection ... etc)

Week 14

14.12.2020 - 18.12.2020

LockStar - Powersupply Error

The current version (the ones ordered in Oktober 2020) of the LockStar (PCB) contains a power supply error, which prevents the RPi to boot.

The solution for the problem (after extensive testing) was found to be the following:

1. Change R56 to 8kOhm
2. Add to C95 a capacitor with 100nF

After these changes the RPI boots as expected.

LockStar - Final Power Supply Checkup

For the final checkup of the LockStar and to make sure the new Power Swap Controller (PSC) works the following (short) test was performed:

1. Plug in (at least) two LockStars into the Rack and connect the Rack to power.
2. Plug out one of the LockStars and in parallel monitor the others if they are affected by this unplugging
3. Plug the LockStar back in and watch out for the same

Both (plugin in and out) did not affect the behaviour of the running LockStars which means the new PSC works as expected!

LockStar - Final Tests & Current Status

The current (assembling) status of the new LockStars is the following:

- 2x fully assembled & working LockStar Racks
- 2x fully assembled & flashed & working LockStar Cards (with the changes on the PCB)

The final test was then to just output certain voltages on the DACs (and also use the DigitalOut) and measure with a Picoscope wether the voltages are correct- which was the case

2021

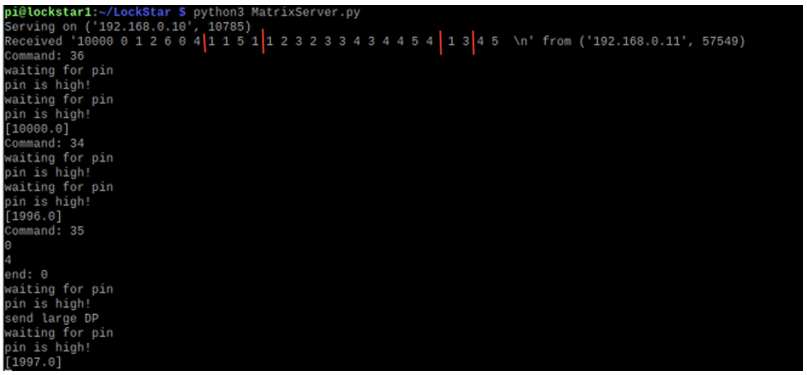
January & February

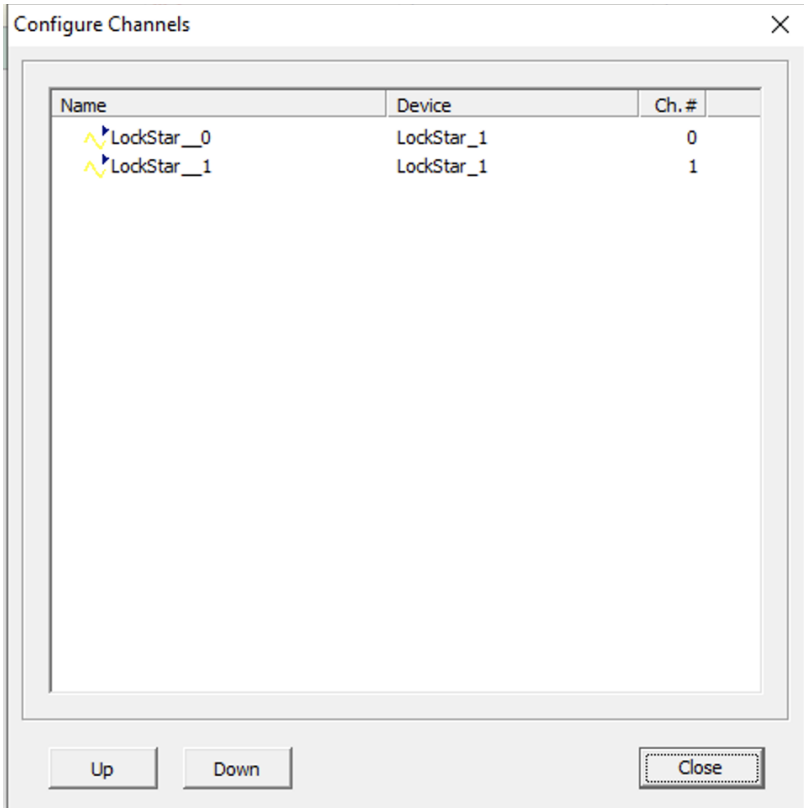
- External triggering of LockStar via DigitalInput works (The TTL must have a voltage of 5V to be detected by the LockStar (not 1V as tested before))
- Successful test of the PID Function on the LockStar using a Frequency generator. Note: The PID Loop is connected to the digital input. The output of the PID Loop is sampled by the rate of the TTL signal.
- Code Cleanup of the Matrix-related Code on the Microcontroller (as separate Class)
- Assembled & Flashed two more LockStar cards
- Updated Matrix on the Impact Lab Computer
- Connected & installed the first full working LockStar Rack (with the new design) in the Lab
- Tested Timing of the LockStar
- Added second channel for the LockStar in the Matrix
- Added Sample and Hold Feature in the Matrix
- Resolve timing issue of the LockStar

In order to mount the shared folder on the LockStar run the command:

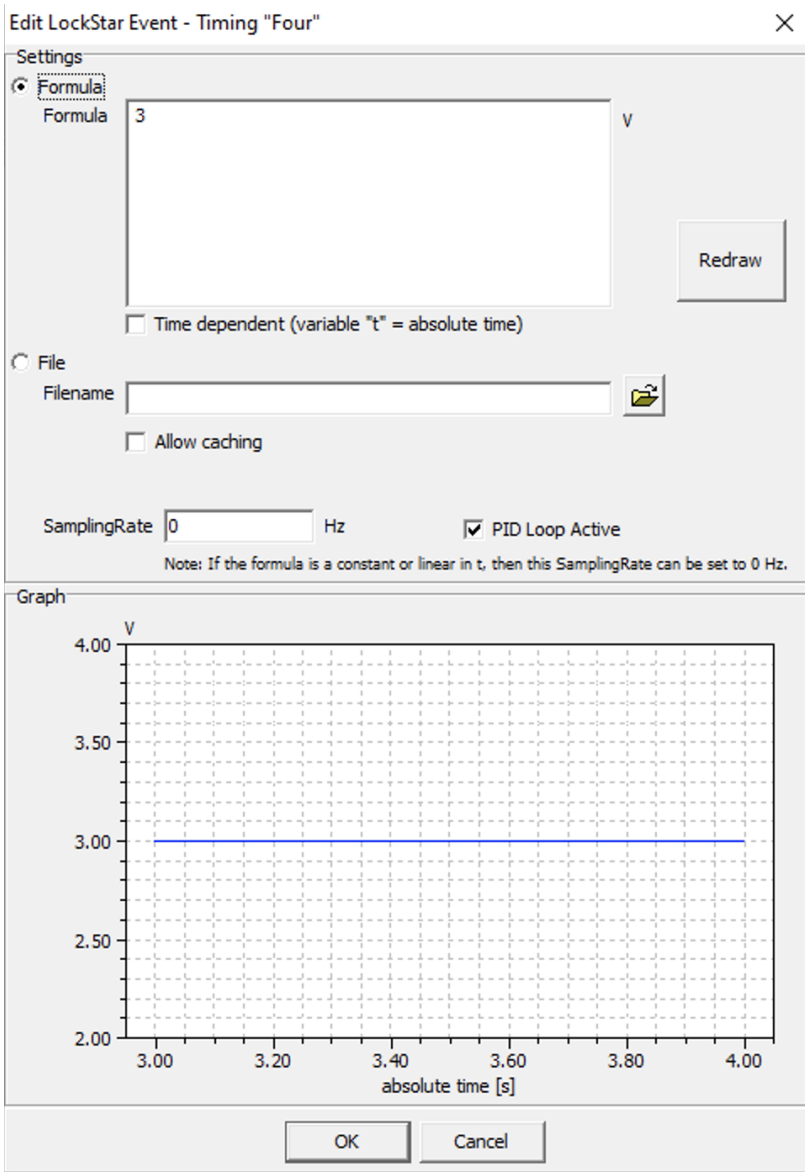
“sudo mount -t cifs -o username=qo,password=…. share-folder destination-folder”

Two Channels





Sample and Hold Feature



Timing Issue

The timing issue consisted of the following problems:

1. Non-linear time delay (eg. in the range 0-11,5s the time delay fluctuated between ±several ms)
2. Time delay 'jitters' and is not constant for every sequence
3. Different time delay for different samplingRates

All of these issues could be resolved:

1.) Use a uint32_t counter (instead of a float variable) to measure the time

2.) Reset the counter of the internal MC clock every time a sequence starts (when the MC get the Digital Trigger)

3.) Add a '-1' in the StartTimer function. This leads to a constant time delay for every samplingRate as shown in the table below.

t	Δt (in µs)

—-	————
0	+ 6
1	-70
2	-150
3	-230
4	-310
5	-390
6	-470
7	-550
8	-630
9	-710
10	-790

The minus means that the LockStar clock is too fast. To be precise, after (approx) 12500 clock cycles the counter is exactly one step ahead. So therefore resetting the counter by one after the 12500 clock cycles decreases the time delay to a minimum, which is the given samplingRate respectively.

The following graph represents an example output of two LockStars with two channels active each. Different kind of functions can are realised (constant, ramplin(), sin()).