



Table of Contents

1.Updating the Mini Lab firmware.	2
1.1. When is it required?	2
1.2. The MCU, STM32F030K6T.	2
1.3. The In-Circuit-Serial-Programming (ICSP) connector.	2
2.Using the ST-LINK/V2 in-circuit debugger/programmer	2
2.1. The ST-LINK/V2	2
3.Connecting the ST-LINK/V2 to the LabBoard	3
3.1. 3 wires required.	3
4.How to get the Firmware from Totem?	3
5.Windows OS with the ST-LINK Utility software.	3
5.1. Where to download it.	3
5.2. Installing the driver.	4
5.3. Starting the STM32 ST-LINK Utility	5
6.Lab Board firmware update on Linux (debian / Ubuntu)	8
6.1. Update your Linux system.	8
6.2. Check if the ST-LINK/V2 has been detected by the OS.	9
6.3. Connect ST-LINK/V2 to PC's USB port and connect wires to the programming connector SV6 on the LabBoard.	10
6.4. Try to connect to the target.	10
6.5. Programming the ST-LINK.	11
7.Updating Mini Lab firmware using ST-LINK/V2 programmer on macOS	13
7.1. Requirements:	13
7.2. Updating procedure	13
7.3. Connect minilab to programmer.	14
7.4. Verify the connection.	14
8.Calibration of the voltage measurement inputs.	15
8.1. Why is calibration needed?	15
8.2. How to find out if your Mini Lab needs calibration?	15
8.3. Connect your voltage inputs for calibration.	15
8.4. Calibration procedure:	16



1. Updating the Mini Lab firmware.

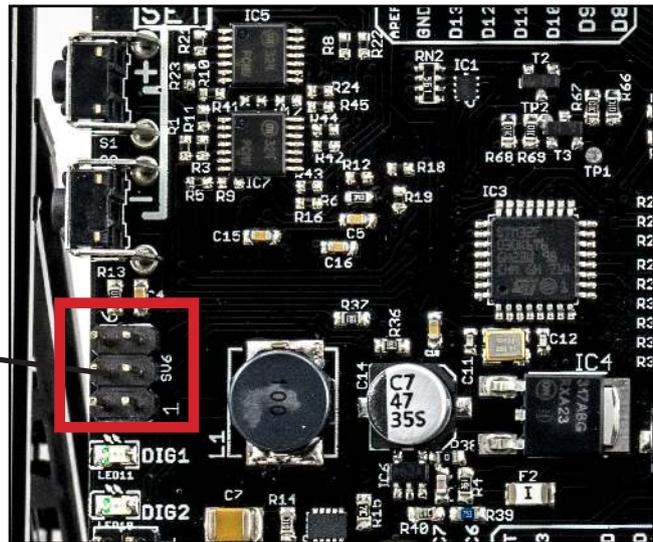
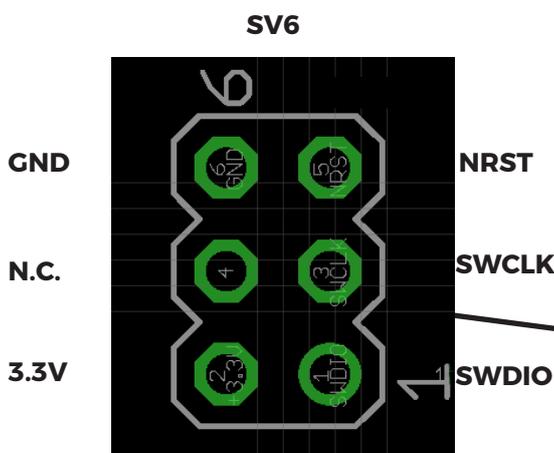
1.1. When is it required?

Normally you will not need to update the firmware to the Lab Board in the Mini Lab. But in case a bug has sneaked in, or a new feature is introduced, it might be handy to be able to upgrade the firmware.

1.2. The MCU, STM32F030K6T.

The micro controller is an ARM Cortex M0, 32 bit, running at 48 MHz, with 32 kB flash program memory and with 4 kB RAM.

1.3. The In-Circuit-Serial-Programming (ICSP) connector.



1. Programming connector.

2. Using the ST-LINK/V2 in-circuit debugger/programmer

2.1. The ST-LINK/V2

First you need to have an ST-LINK/V2 programmer. It is available many places, compatible ones can be bought inexpensively from Totem’s webshop, www.totemaker.net, or on Ebay and AliExpress to mention some.



2. ST-LINK/V2 programmer

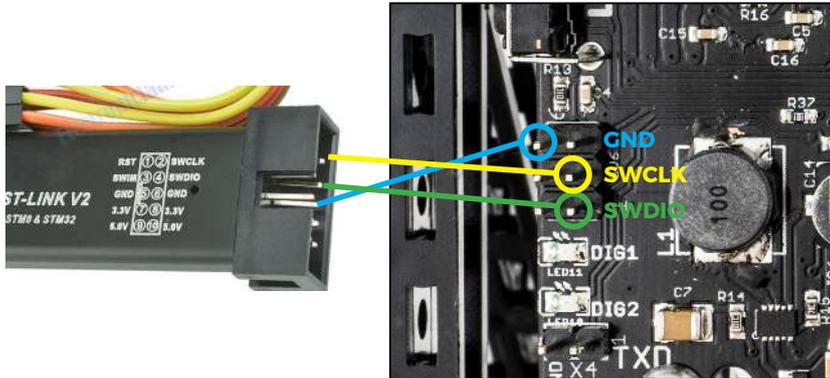


3. Connecting the ST-LINK/V2 to the LabBoard

3.1. 3 wires required.

The 3 signals you need to program the LabBoard is:

GND (Connect this first, it's good practise), SWCLK and SWDIO.



3. Connecting the SV6 connector to the ST-LINK/V2

You are now ready to program the new firmware, but you must have the required software utilities on your computer. The next section will show you how.

OBS ! You must have the Mini Lab connected, so that the LabBoard have power! We are not supplying the power via the ST-LINK, but from the TotemDuino and it's power supply.

4. How to get the Firmware from Totem?

The Firmware is obtainable from www.totemmaker.net/wiki/

When you are on the WIKI section in totemmaker.net, you will navigate yourself to the Mini Lab section, and there you will find a download link to get the latest release of our firmware. The firmware binary file will have a name like this: Labboard_1.5.bin. It will contain a version number.

5. Windows OS with the ST-LINK Utility software.

5.1. Where to download it.

The software you need to program new firmware into the Lab Board is called STSW-LINK004, and is made by STMicroelectronics. From their website www.st.com you can find the STSW-LINK004 software.

http://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-programmers/stsw-link004.html

There is a snag to downloading this utility, you have to register to st.com with some personal information, like an email address etc. But that's expected I suppose.



Here is a description from STMicroelectronics downloading page:

STM32 ST-LINK Utility (STSW-LINK004) is a full-featured software interface for programming STM32 microcontrollers. It provides an easy-to-use and efficient environment for reading, writing and verifying a memory device. The tool offers a wide range of features to program STM32 internal memories (Flash, RAM, OTP and others), external memories, to verify the programming content (checksum, verify during and after programming, compare with file) and to automate STM32 programming. STM32 ST-LINK Utility is delivered as a graphical user interface (GUI) with a command line interface (CLI).

5.2. Installing the driver.

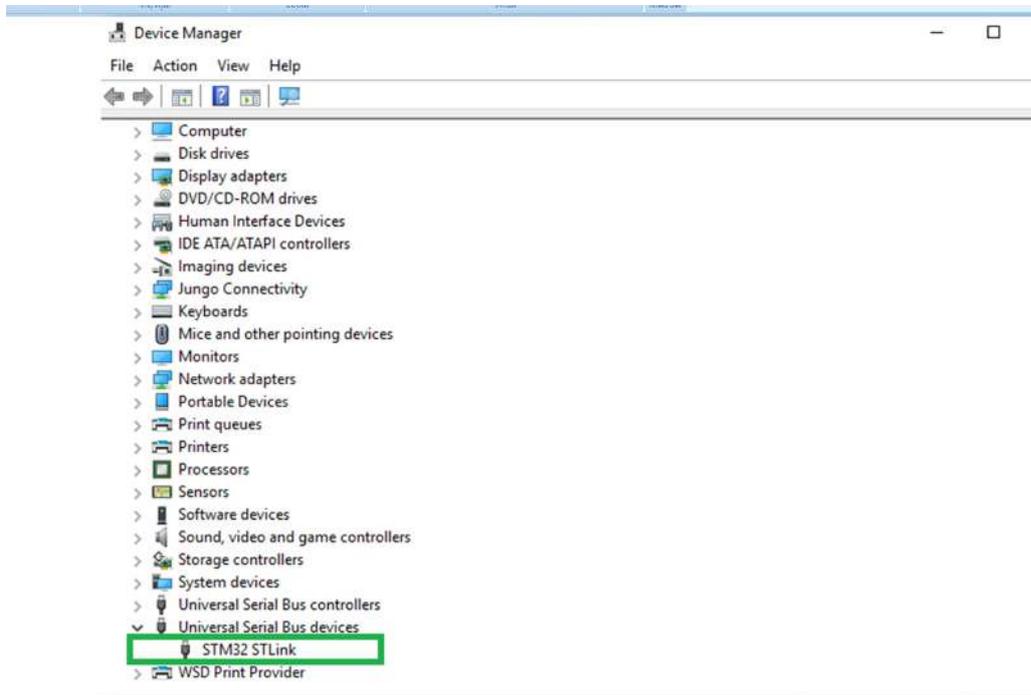
Next step is to install the driver:



After you have installed the STSW-LINK004 software on your computer, you can start the STM32 ST-LINK Utility.

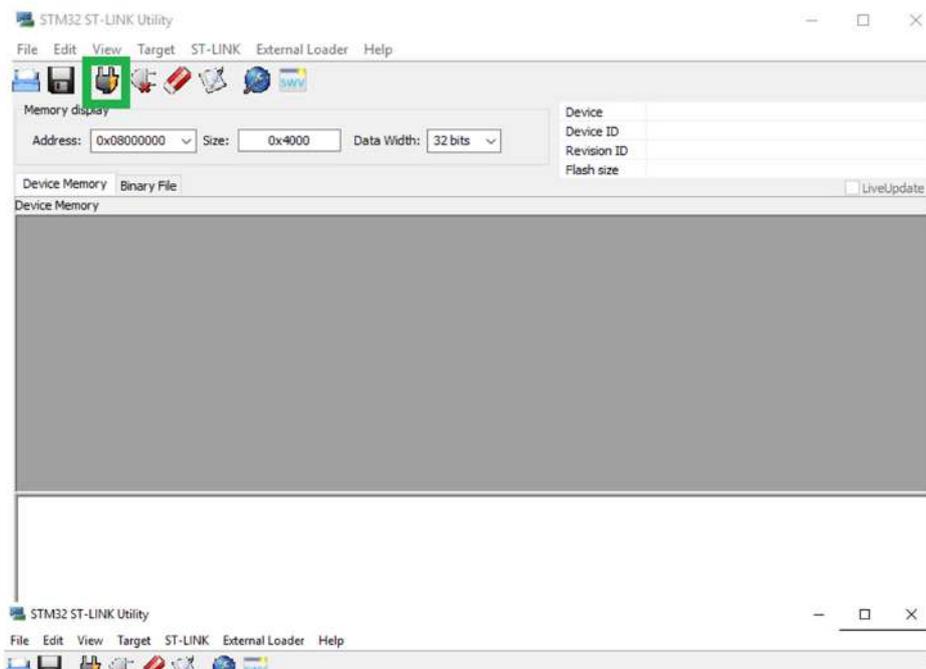


The USB driver will be installed when you put your ST-LINK/V2 into a USB port in your PC. You can check that it is installed by checking the “Device Manager” in the Control Panel in Windows.

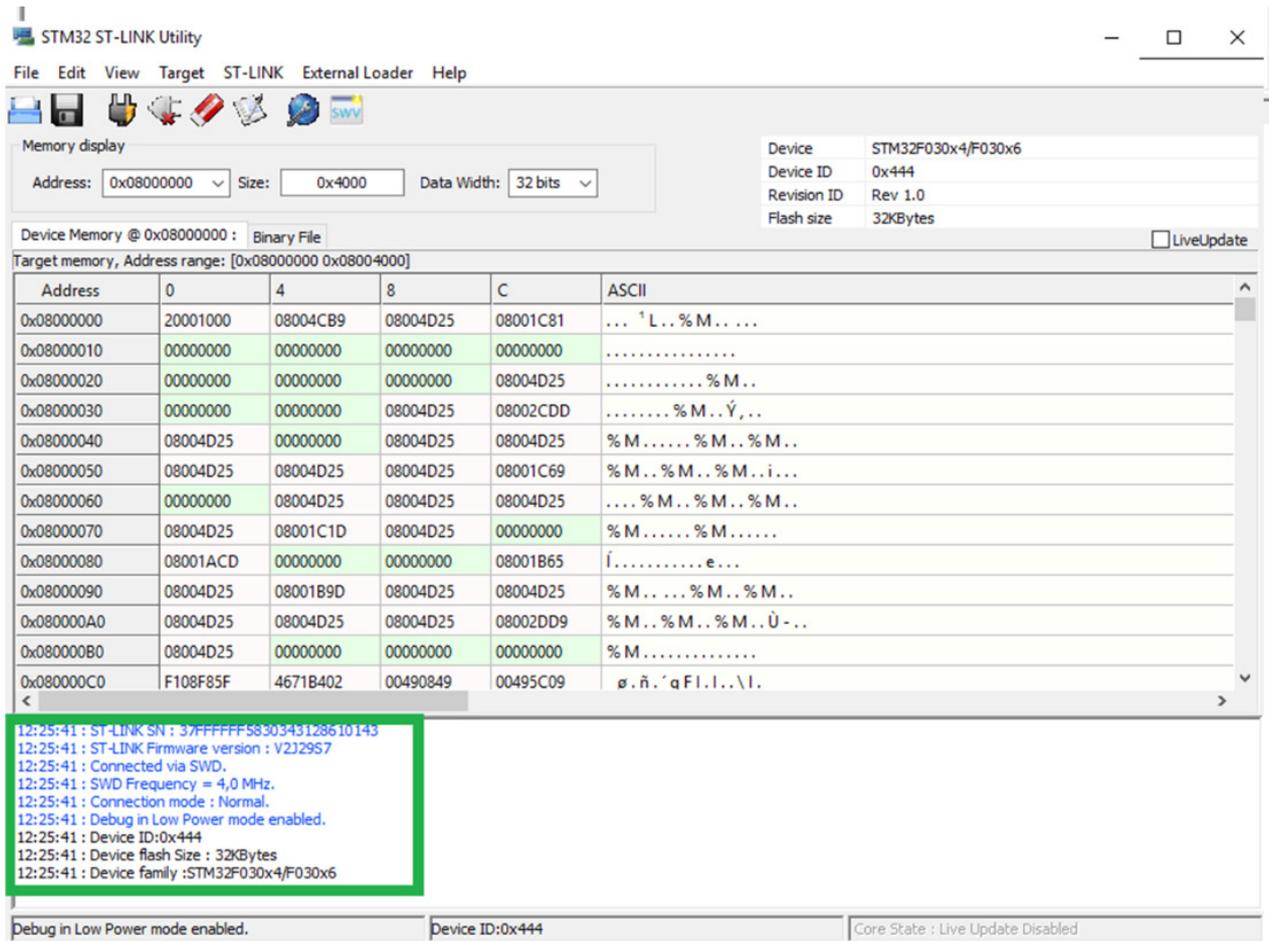


5.3. Starting the STM32 ST-LINK Utility

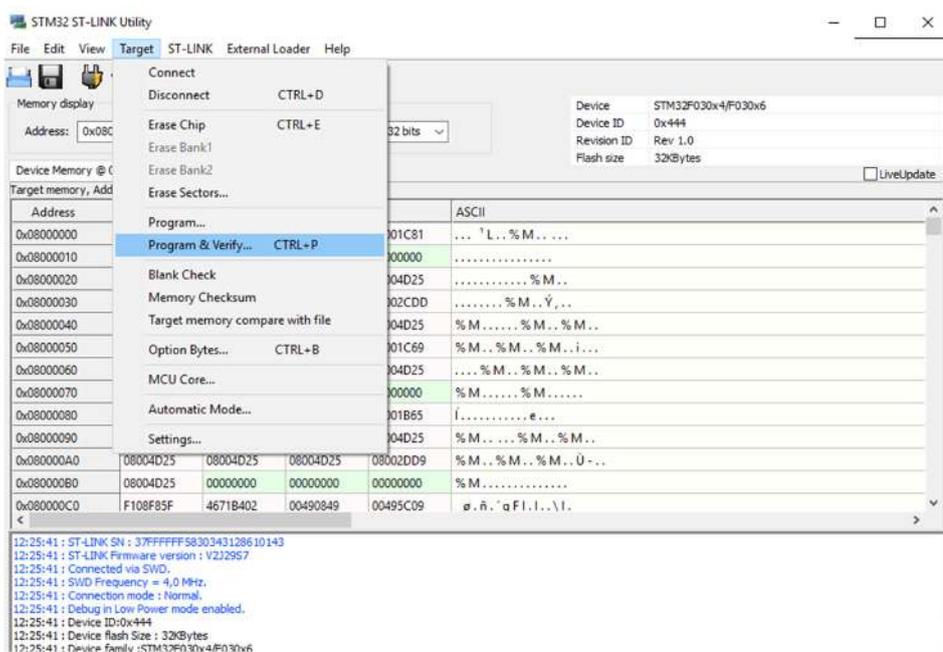
You can type “STM32” in the “Windows Start Menu” in the lower left corner, and see the ST-LINK Utility program, and click on it to start it. You will also see the STM32 ST-LINK Utility User Manual, a PDF file that gives you in depth reference to the software if you want to dig deeper.



Click on the “Connect” icon (marked green above) and you will see if you have a connection to the STM32 via the ST-LINK.



If the programming succeeded, you will see the screen as above. The utility has read the memory of the MCU, and are ready to go.

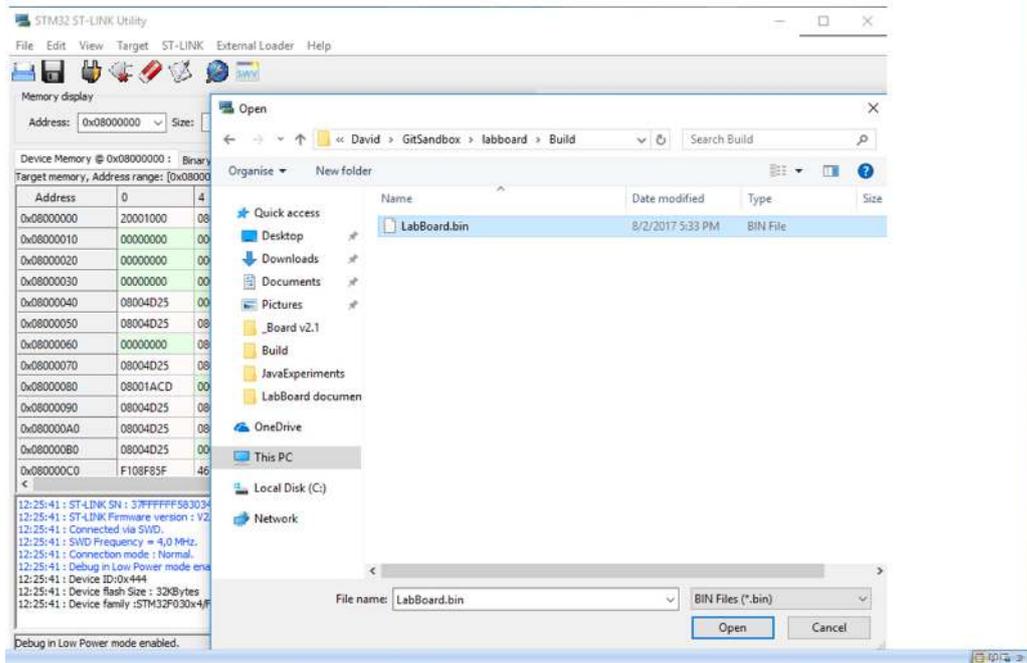


To start programming, select <Target> and then <Program & Verify...>

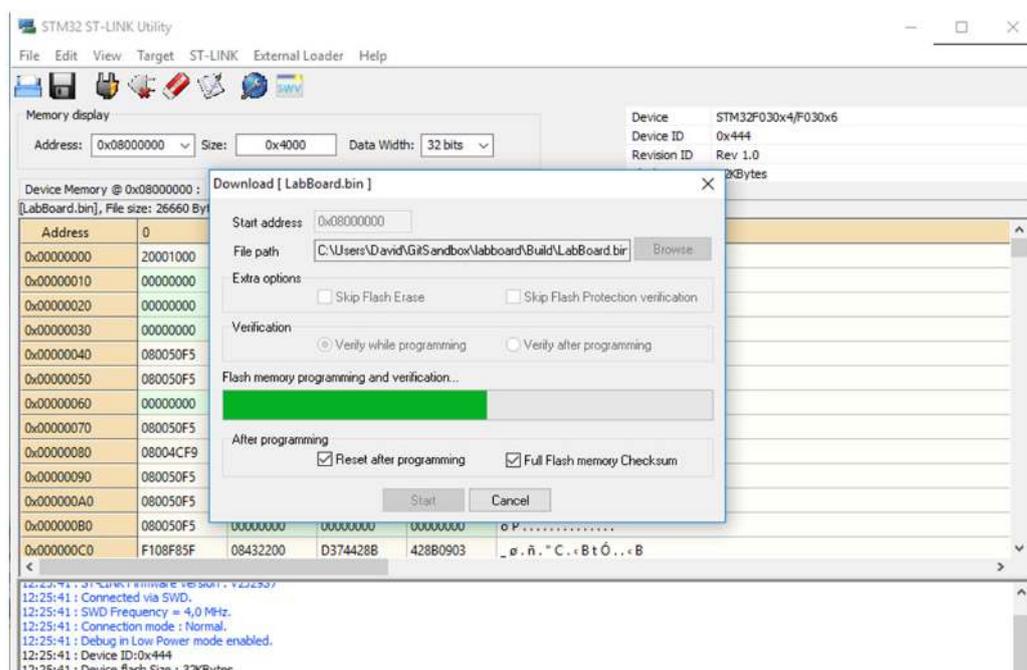


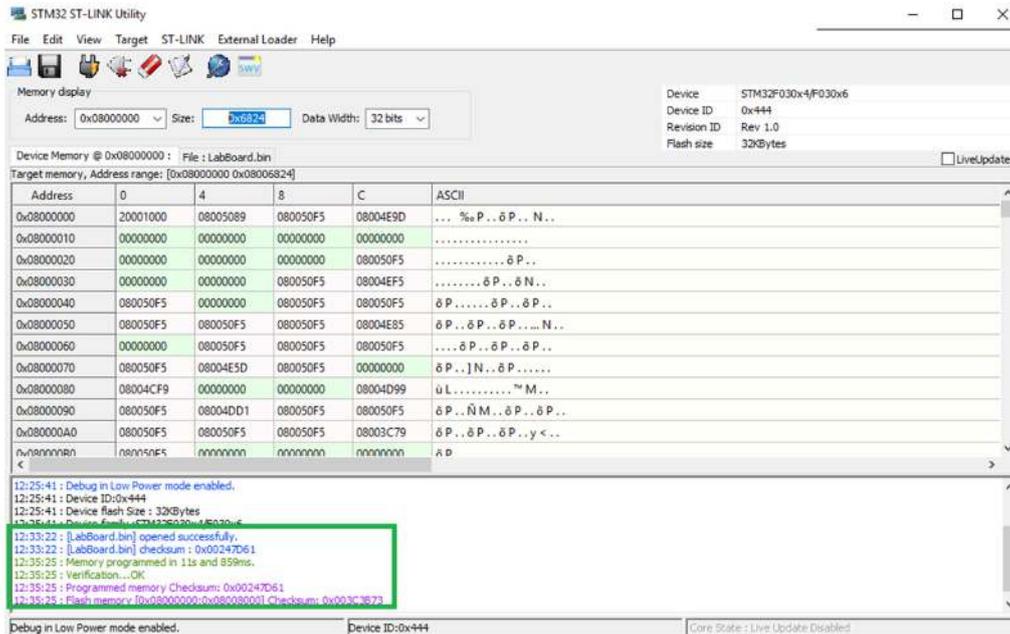
You should have already have downloaded the new firmware binary file from the www.totemaker.net WIKI section. See **chapter 4 How to get the Firmware from Totem?**

Browse to the folder you downloaded the *.BIN file to, and select it in the dialog window like you see in the screen shot below:



And when programming the firmware, you will see this screen:





When a successful programming is done, you will get the message **“Verification... OK”**, and you will know that you have updated the firmware. Now, you will need to restart the LabBoard/MiniLab by switching off and on the power, then check that it starts up normally with the new firmware.. Then you should re-calibrate the LabBoard, because some parameters may have been lost during programming. These parameters store e.g. the zero-points in the voltage metering, so that a higher accuracy is obtained. You can check the calibration section in the last chapter of this document.

6. Lab Board firmware update on Linux (debian / Ubuntu)

6.1. Update your Linux system.

In this section we will show you how to use a Linux system to update the firmware in the Lab Board. We will show it in a debian Ubuntu system, so it might be a bit different in other systems. We assume that users of Linux have a good knowledge of their system, so this chapter is a bit more technical in it's form than the Windows section.

So first you should check for updates for your system, using the **sudo apt upgrade** command.



```
swd55@swd55-K53SC: ~
swd55@swd55-K53SC:~$ sudo apt update
[sudo] password for swd55:
Hit:1 http://lt.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://lt.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://lt.archive.ubuntu.com/ubuntu xenial-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu xenial-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
swd55@swd55-K53SC:~$
```

If there are updates for your system, use the **sudo apt upgrade** command.

6.2. Check if the ST-LINK/V2 has been detected by the OS.

Type **lsusb**

If OS can see ST-LINK/V2 programmer, you can start download the needed software.

```
swd55@swd55-K53SC: ~
swd55@swd55-K53SC:~$ lsusb
\Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 003: ID 0483:3748 STMicroelectronics ST-LINK/V2
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 005: ID 0bda:0139 Realtek Semiconductor Corp. RTS5139 Card Reader
Controller
Bus 001 Device 004: ID 058f:a014 Alcor Micro Corp. Asus Integrated Webcam
Bus 001 Device 003: ID 0cf3:3005 Atheros Communications, Inc. AR3011 Bluetooth
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
swd55@swd55-K53SC:~$
```

You can install st texane flashing tool, but this tool should be build from source (for debian Ubuntu users) and it works only as a flasher.

There is a slightly easier way to install needed software, installing from already existing debian repository.

Firstly we need openOCD software:

sudo apt install openocd

```
swd55@swd55-K53SC: ~
swd55@swd55-K53SC:~$ sudo apt install openocd
[sudo] password for swd55:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
 snap-confine
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
 libhidapi-hidraw0 libjlm0.76
The following NEW packages will be installed:
 libhidapi-hidraw0 libjlm0.76 openocd
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 2.038 kB of archives.
After this operation, 5.7808 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```



Check where openOCD located, as path may differ, it is necessary to check path:

whereis openocd

```
swd55@swd55-K53SC: ~  
swd55@swd55-K53SC:~$ whereis openocd  
openocd: /usr/bin/openocd /usr/share/openocd /usr/share/man/man1/openocd.1.gz /u  
sr/share/info/openocd.info-2.gz /usr/share/info/openocd.info-1.gz /usr/share/inf  
o/openocd.info.gz  
swd55@swd55-K53SC:~$
```

6.3. Connect ST-LINK/V2 to PCs USB port and connect wires to the programming connector SV6 on the LabBoard.

Refer to chapter 3 “Connecting the ST-LINK/V2 to the LabBoard”. Here you will find instructions for how to connect the 3 wires from the ST-LINK/V2 to the LabBoard.

6.4. Try to connect to the target.

Type:

```
sudo openocd -f /usr/share/openocd/scripts/interface/stlink-v2.cfg -f /usr/share/openocd/scripts/target/stm32f0x.cfg
```

```
c33@c33-HyperBot: ~  
c33@c33-HyperBot:~$ sudo openocd -f /usr/local/share/openocd/scripts/interface/  
stlink-v2.cfg -f /usr/local/share/openocd/scripts/target/stm32f0x.cfg  
Open On-Chip Debugger 0.10.0  
Licensed under GNU GPL v2  
For bug reports, read  
  http://openocd.org/doc/doxygen/bugs.html  
Info : auto-selecting first available session transport "hla_swd". To override u  
se 'transport select <transport>'.  
Info : The selected transport took over low-level target control. The results mi  
ght differ compared to plain JTAG/SWD  
adapter speed: 1000 kHz  
adapter_nsrst_delay: 100  
none separate  
Info : Unable to match requested speed 1000 kHz, using 950 kHz  
Info : Unable to match requested speed 1000 kHz, using 950 kHz  
Info : clock speed 950 kHz  
Info : STLINK v2 JTAG v17 API v2 SWIM v4 VID 0x0483 PID 0x3748  
Info : using stlink api v2  
Info : Target voltage: 3.258278  
Info : stm32f0x.cpu: hardware has 4 breakpoints, 2 watchpoints
```

Optionally you may not need full path:

```
sudo openocd -f interface/stlink-v2.cfg -f target/stm32f0x.cfg
```



6.5. Programming the ST-LINK.

Next step is to send commands to do the actual “flashing” of the firmware. To do that you’ll need “**telnet**” which is a default app for many operating systems.

Type:

telnet localhost:4444

```
swd55@swd55-K53SC: ~  
swd55@swd55-K53SC:~$ telnet localhost 4444  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
Open On-Chip Debugger  
> █
```

Now you can send commands for device debugging, memory reading and etc.

First of all, halt device:

reset halt

```
swd55@swd55-K53SC: ~  
swd55@swd55-K53SC:~$ telnet localhost 4444  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
Open On-Chip Debugger  
> reset halt  
target state: halted  
target halted due to debug-request, current mode: Thread  
xPSR: 0xc1000000 pc: 0x08004cb8 msp: 0x20001000  
> stm32f0x mass_erase 0  
stm32x mass erase complete  
> █
```

Now MCU is stopped.

Before writing new firmware, delete the old one in the Lab Board:

stm32f0x mass_erase 0 (see above screen shot)



Now you can flash new firmware.

Firmware flashing:

flash write_image /home/swd55/Desktop/LabBoard.bin 0x08000000

```
swd55@swd55-K53SC: ~
swd55@swd55-K53SC:~$ telnet localhost 4444
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Open On-Chip Debugger
> reset halt
target state: halted
target halted due to debug-request, current mode: Thread
xPSR: 0xc1000000 pc: 0xffffffff msp: 0xffffffff
> stm32f0x mass_erase 0
stm32x mass erase complete
> flash write_image /home/swd55/Desktop/LabBoard.bin 0x08000000
target state: halted
target halted due to breakpoint, current mode: Thread
xPSR: 0x61000000 pc: 0x2000003a msp: 0xffffffff
wrote 20644 bytes from file /home/swd55/Desktop/LabBoard.bin in 0.657397s (30.66
7 KiB/s)
>
```

The filename “LabBoard.bin” may vary, according to versions etc. When you download it, you will see what version it has, we just use this “LabBoard.bin” as a placeholder name. It may look like this: **flash write_image <PATH>/Labboard_1.5.bin 0x08000000**

Reset device :

reset

Now you should re-calibrate the LabBoard, because some parameters may have been lost during programming. These parameters store e.g. the zero-points in the voltage metering, so that a higher accuracy is obtained. You can check the calibration section in the last chapter of this document.



7. Upgrading Mini Lab firmware using ST-LINK/V2 programmer on macOS

This document guides through steps required to upgrade firmware in the Mini Lab, using a macOS PC. Upgrading firmware can lead to better performance and functionality bug fixes. See chapter : **4. How to get the Firmware from Totem?**

When you have downloaded the latest version of the firmware, you can continue with the steps below:

Before starting, make sure that you meet all necessary requirements:

7.1. Requirements:

- **macOS v 10.10 or higher**
- **SWD compatible programmer (ST-LINK/V2)**
- **Jumper cables to connect programmer to Mini Lab**

SWIO, SWCLK and GND signals will be used for flashing. If you're using different programmer from the one used in this example, refer to that programmer documentation for correct pinout. For the flashing software, we'll be using stlink application. The recommended way of getting it is using homebrew package manager. Refer to homebrew installation instruction on how to install it on your computer.

7.2. Updating procedure

1. Prepare software. After installing homebrew, stlink can be installed from Terminal.app, by writing command shown below:

brew install stlink



7.3. Connect the ST-LINK/V2 to the Lab Board's programming port..

Referring to pinout schematics below, connect all three needed pins using jumper cables: You can also read chapter 3 in this document : "Connecting the ST-LINK/V2 to the LabBoard" that explains in more detail.

7.4. Verify the connection.

Power on minilab and plug in programmer into your computer. Successful connection and installation of stlink can be verified by issuing this command in your Terminal.app:

st-flash reset

If programmer was able to establish connection to the LabBoard, information about the chip should be printed: Flash the updated firmware. Using command below, flash the updated firmware package into minilab:

```
regina:~ karolistarasauskas$ st-flash reset
st-flash 1.4.0
2018-01-11T13:48:13 INFO src/common.c: Loading device parameters....
2018-01-11T13:48:13 INFO src/common.c: Device connected is: F0 small device, id 0x10006444
2018-01-11T13:48:13 INFO src/common.c: SRAM size: 0x1000 bytes (4 KiB), Flash: 0x8000 bytes
(32 KiB) in pages of 1024 bytes
regina:~ karolistarasauskas$ █
```

st-flash write <file.bin> 0x08000000

<file.bin> should point to the exact file path in your computer. The recommended way of doing this is to drag the file into the terminal window, and the path will be filled automatically, only the finishing address number needs to be input. The actual filename of the binary firmware file may be more like: **Labboard_1.5.bin**

```
regina:knob karolistarasauskas$ st-flash write /Users/karolistarasauskas/Desktop/update.hex.
txt 0x08000000
st-flash 1.4.0
2018-01-11T13:56:33 INFO src/common.c: Loading device parameters....
2018-01-11T13:56:33 INFO src/common.c: Device connected is: F0 small device, id 0x10006444
2018-01-11T13:56:33 INFO src/common.c: SRAM size: 0x1000 bytes (4 KiB), Flash: 0x8000 bytes
(32 KiB) in pages of 1024 bytes
2018-01-11T13:56:33 INFO src/common.c: Attempting to write 3166 (0xc5e) bytes to stm32 addre
ss: 134217728 (0x8000000)
Flash page at addr: 0x08000c00 erased
2018-01-11T13:56:33 INFO src/common.c: Finished erasing 4 pages of 1024 (0x400) bytes
2018-01-11T13:56:33 INFO src/common.c: Starting Flash write for VL/F0/F3/F1_XL core id
2018-01-11T13:56:33 INFO src/flash_loader.c: Successfully loaded flash loader in sram
4/4 pages written
2018-01-11T13:56:33 INFO src/common.c: Starting verification of write complete
2018-01-11T13:56:33 INFO src/common.c: Flash written and verified! jolly good!
regina:knob karolistarasauskas$ █
```

That's it, firmware has been updated. Now your Mini Lab should be reset by cycling it's power back on.

Then you should re-calibrate the LabBoard, because some parameters may have been lost during programming. These parameters store e.g. the zero-points in the voltage metering, so that a higher accuracy is obtained. You can check the calibration section in the next chapter of this document.



8. Calibration of the voltage measurement inputs.

8.1. Why is calibration needed?

The device voltmeter circuit can be affected by the environment. After some time the device may lose a little accuracy. Normally the voltmeter inputs are floating within a small range. If the floating measurements is out of range, you can very simply re-calibrate the voltage inputs. Also if you uploaded new firmware, you should calibrate again.

8.2. How to find out if your Min Lab needs calibration?

1.To check if you need calibration, connect 50/5/0.5 V channels to ground(GND) with some short patching cables, and read voltage measurement values.

Abnormal conditions:

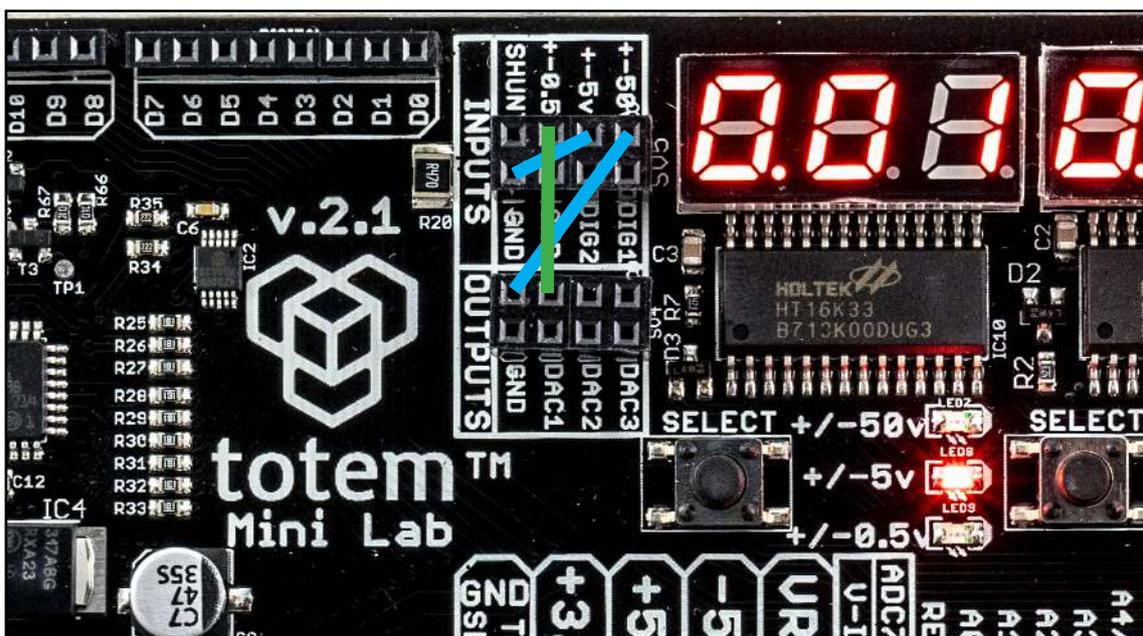
+/-50V channel value floats more than 0.2V when connected to ground(GND).

+/-5V channel value floats more than 0.1V when connected to ground(GND).

+/-0.5V channel value floats more than "005" (0.005V->5mV) when connected to ground(GND).

Note: 0.5 V channel is very sensitive and when floating (when nothing is connected) it may show values more than 5mV. It picks up statics etc.

If you found any of above conditions, then calibration could be needed.



5. How to connect voltmeter inputs for calibration.

8.3. Connect your voltage inputs for calibration.

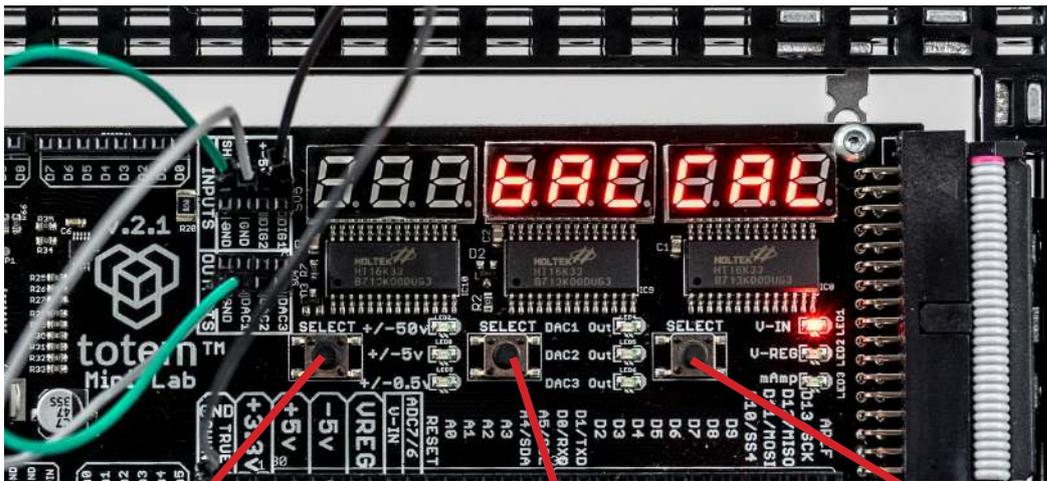
The above picture shows how to connect the Lab Board for calibration. You simply take 3 short patching cables, and connect the +/-50v and the +/-5v to a GND header. You will normally find the closest GND headers. In figure 6, these patching cables are shown in blue color. The +/- 0.5v input should be connected to the DAC 1 output.



It is illustrated in the figure 6 as the green patching cable. The +/-0.5v input will use also a +0.5v output from the DAC 1, so it gets a 2-point calibration. This gives it an even better calibration than using only GND, as the 2 other inputs use.

8.4. Calibration procedure:

You may wait for 2-3 minutes until the circuit's temperatures stabilizes.

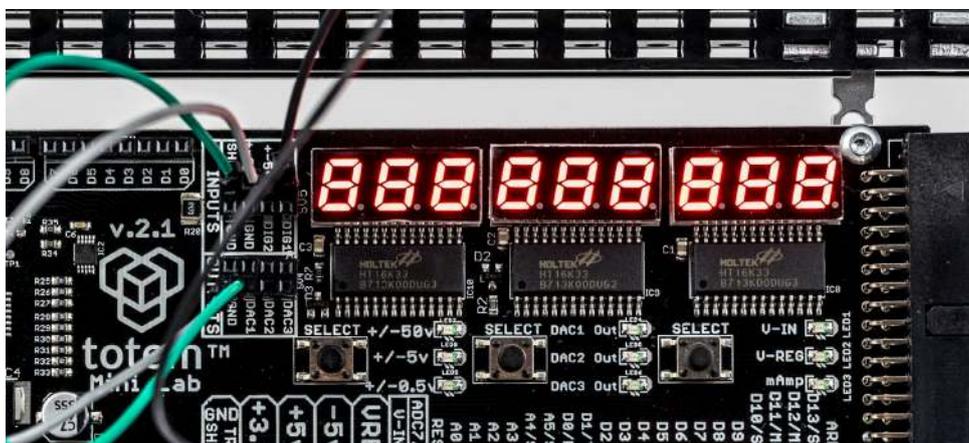


6. To start the calibration routine, press the SELECT voltage display button for more than 5 seconds.

7. "bAC" means "back", and simply escapes the calibration routine. Nothing is done or stored if you escape with this SELECT button.

8. Pressing the SELECT button to the right, under the "CAL" display, will start the calibration.

So, pressing the rightmost SELECT button will start calibration. For a moment "88888888" will appear on the display when the device calibrates. After that, the device will restart itself.



Calibration is then done. Now you can then check if the floating values are better.