Blitzortung.org

A World-Wide Low-Cost Community-Based Time-of-Arrival Lightning Detection and Lightning Location Network

Egon Wanke[*] Richo Andersen Tobias Volgnandt[†]

October 5, 2013

This document will continually change due to further developments and improvements. Please send your remarks to mailto:blitzortung@gmx.org. Most of the receiving stations are currently operating in Europe, but we are emphatically interested to extend our activities to other continents. Please contact us, if you are interested to start the activity in your region.

Blitzortung.org is a world-wide non-commercial low-cost community-based Time-of-Arrival lightning detection and lightning location network. The system is made for private and entertainment purposes. The domain Blitzortung.org is not an official information service for lightning data. A commercial use of data from Blitzortung.org is strongly prohibited, even by the users that send data to our server.

This document can change without further notice. Publishing this document or excerpts of this document on websites not under our control is unwanted.

[*] Düsseldorf, Germany
[†] Roßtal, Germany

Contents

1 Lightning detection hardware

The hardware of our the lightning detection system consists of three components.

- Antenna System

  The VLF-signals are received by wide-band H-Field or E-Field antennas

- Amplifier

  A wide-band VLF-amplifier amplifies the antenna signal such that it can be further processed and analyzed.

- Controller

  The controller digitize the signals and send them together with the reception time and reception location to our server.

Figure 1 shows a complete starter kit.

1.1 Antennas

(will be inserted later)

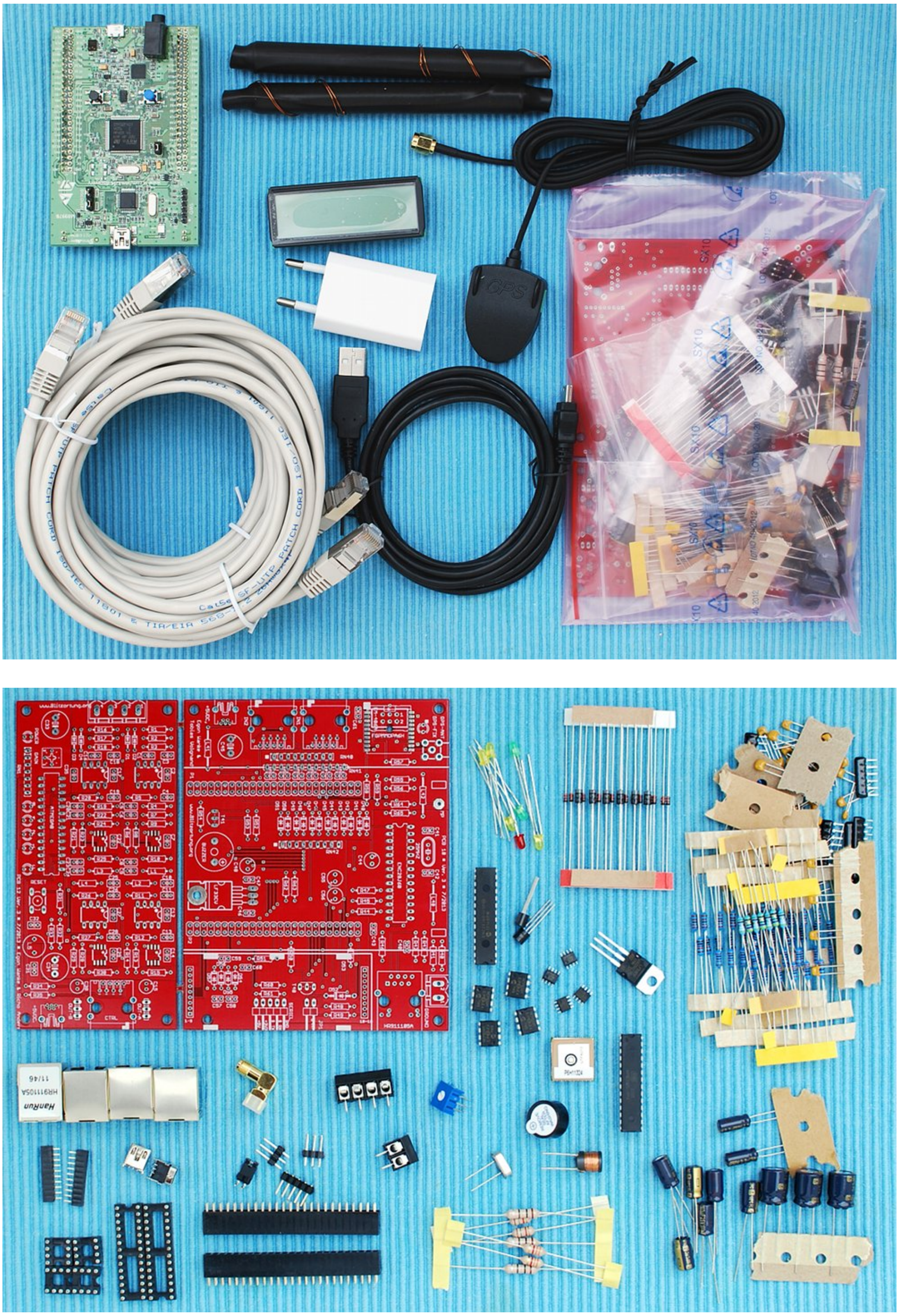


Figure 1: A complete starter kit

1.2 Amplifiers

Every amplifier has its own specific signal characteristic depending on the type of the low-pass, the type of the high-pass filter, the type of the used op-amps and so on. The different types of the circuits and components can cause different signal forms and different time delays. This is the main reason why we strictly recommend that all users should exclusively use the same amplifier circuits, in particular, one of our recommended amplifier circuits. Wrong time stamps can considerably confuse our calculations. Finding the reason for false calculations can be very time consuming.

We offer several different amplifiers for different antenna types. Our standard amplifier for magnetic field antennas (H-field antennas) is Amplifier 12.x . These are for example loop antennas or ferrite rod antennas An amplifier for electric field antennas (E-field antennas) will probably be introduced in the fall of 2013. We will modify the amplifier circuits from time to time to improve its quality for our special application and to reduce the price.

1.2.1 Amplifier 12.x

Amplifier 12.x should be used for all magnetic loop antennas and all ferrite rod antennas. It uses four single supply 5 Volt rail-to-rail op-amp ICs for each channel. The inputs of the amplifier are terminated by damping resistors of 2.2kΩ. This leads to a more realistic current measurement for loop antennas and will reduce self resonances. For larger loop antennas, the damping resistor can be lowered to 100Ω- 300Ωin parallel to the antenna. A coax cable antenna can be terminated by 75Ω.

The first and third stage of the amplifier are realized by programmable gain amplifiers (PGAs) MCP6S91 from Microchip. The gain of these amplifiers is controlled by a serial peripheral interface (SPI) and can be adjusted between 1 and 32 for each MCP6S91. The second stage uses the dual op-amp MCP6292 from Microchip. The two operational amplifiers of this IC realize a Besssel high-pass filter with a cut-off frequency of approximately 1kHz. This filter is realized by two active Sallen-Key filter circuits. The second operational amplifier of the MCP6292 additionally increases the gain by a factor of 4 on version 12.3. After this stage a Besssel low-pass filter with a cut off frequency of approximately 50kHz is placed. This filter is realized by a passive low-pass filter chain of order five. The last stage is realized by the output driver amplifier LMH6642 from Texas Instruments. This stage additionally increases the gain by an factor of 10. The maximal overall gain of the amplifier is about 76dB. The maximal output current is 75 mA.

The block diagram, circuit diagram, printed circuit board, and an assembled circuit board of Amplifier 12.3 are shown in Figure 2, 3, 4, and 5, respectively.

1.2.1.1 Electronic parts

The electronic parts necessary for Amplifier 12.3 are listed in Table 1. The additional necessary mechanical parts can be found in the complete part list, see table **??**.

operational amplifiers	IC1, IC3, IC5, IC7 IC2, IC6 IC4, IC8	MCP6S91 DIP8 MCP6292 SMD SOI8 LMH6642 SMD SOI8
LEDs	Power LED Control LED Gain LEDs 1,2,4,8	LED 3mm (red) LED 3mm (green) LED 3mm (yellow)
Diodes	D1, D2, D3, D4	1N4148
Resistors, metal 1%, 0207	R3, R13, R18, R28	100kΩ
	R6, R21	68kΩ
	R5, R7, R12, R20, R22, R27	22kΩ
	R2, R4, R14, R17, R19, R29, R31	10kΩ
	R8, R23	6.8kΩ
	R1, R9, R16, R24, R32, R33	2.2kΩ
	R10, R11, R25, R26	1kΩ
	R34, R35	330Ω
	R15, R30	47Ω
Resistor network, star, 6-1	RN1	1kΩA102
Potentiometer	P1	50kΩ
Capacitors, electrolytic, 5mm	C33, C34	470 μF
2mm	C15, C30	2.2 μF
Capacitor, ceramic, 2.54mm	C1, C2, C7, C11, C12, C13, C14, C16, C17, C22, C26, C27, C28, C29, C31, C32, C35	100 nF (104)
	C3, C18	22 nF (223)
	C4, C6, C19, C21	10 nF (103)
	C10, C25	6.8 nF (682)
	C5, C20	4.7 nF (472)
	C9, C24	2.2 nF (222)
	C8, C23	470 pF (471)
Inductors, ferrite, 5mm	L5	470/330 μH
Inductors, SMCC	L2, L4	3.3mH
SMCC	L1, L3	1.5mH

Table 1: The list of electronic parts for Amplifier 12 Version 3

1.2.1.2 Connecting an Antenna

The antenna is connected to the terminal block. If you are using ferrite rod antennas with varnished wires, you should solder the stripped wires to bigger copper wires that can be grasped by the terminals screws.

1.2.1.3 Gain Control

The serial peripheral interface (SPI) for setting the gain of the PGAs is controlled by the Atmel ATmega8 micro-controller. The gain can be adjusted either manually by a potentiometer or automatically by the controller board. It is a digital gain control in both cases. If the green LED named C is on, the gain is adjusted by the controller, if it is off, the gain is adjusted manually. The manual gain control operates as follows. A potentiometer is used for setting a variable voltage between 0 and 5 Volt measured by one of the A/D converters of the ATmega8 micro-controller. It determines from the adjusted voltage the corresponding digital gain values and send them to the PGAs via the serial peripheral interface. In this case, the gain can be adjusted in 12 steps displayed by the yellow LED gain index as shown in Table 2.

LED	LED	LED	LED	first	second	gain
8	4	2	1	MCP6S91	MCP6S91	factor
○	○	○	○	1	1	1
○	○	○	●	2	1	2
○	○	●	○	2	2	4
○	○	●	●	4	2	8
○	●	○	○	4	4	16
○	●	○	●	8	4	32
○	●	●	○	8	8	64
○	●	●	●	16	8	128
●	○	○	○	16	16	256
●	○	○	●	32	16	512
●	○	●	○	32	32	1024

Table 2: Gain index of Amplifier 12

1.2.1.4 Connecting the controller

The RJ45 modular connector is used to connect the amplifier to the controller board. You should use a shielded one-to-one CAT 5e network cable. The length of the cable can be up to 30 meters. This allows to place the amplifier near the antennas far away from noisy environments.

1.2.1.5 Output test pins

The output test pins can be used to monitor the output of the amplifier. This is useful if you want to connect an oscilloscope or a sound card to analyze the VLF output without to disconnect the amplifier from the controller board.

1.2.1.6 First hardware check

During normal operation, the controller powers the amplifier through the Ethernet cable. But for the first hardware check, you should not connect the controller but connect the USB jack on the amplifier. After connecting a 5V USB power supply, the red power indicator LED should glow and the four yellow LEDs of the gain index should start flashing four times. The green LED should be off. A few seconds later the four yellow LEDs should display the adjusted gain. If the potentiometer is turned then the new gain value should be displayed, see also Table 2.

The green LED is only on if the amplifier is controlled by the controller board. In this case the yellow LEDs have no meaning. A manual gain adjustment is not possible if the gain is controlled by the controller.

1.2.1.7 Amplifier PCB change log

12.2	First release
12.3	Added additional gain factor 4 in high pass Changed IC2 and IC6 to SMD versions More space between L1/L2 and L3/L4
12.3b	Minor layout changes
12.3c	Changed IC1, IC3, IC5, IC7 to SMD

Table 3: Change log of Amplifier 12

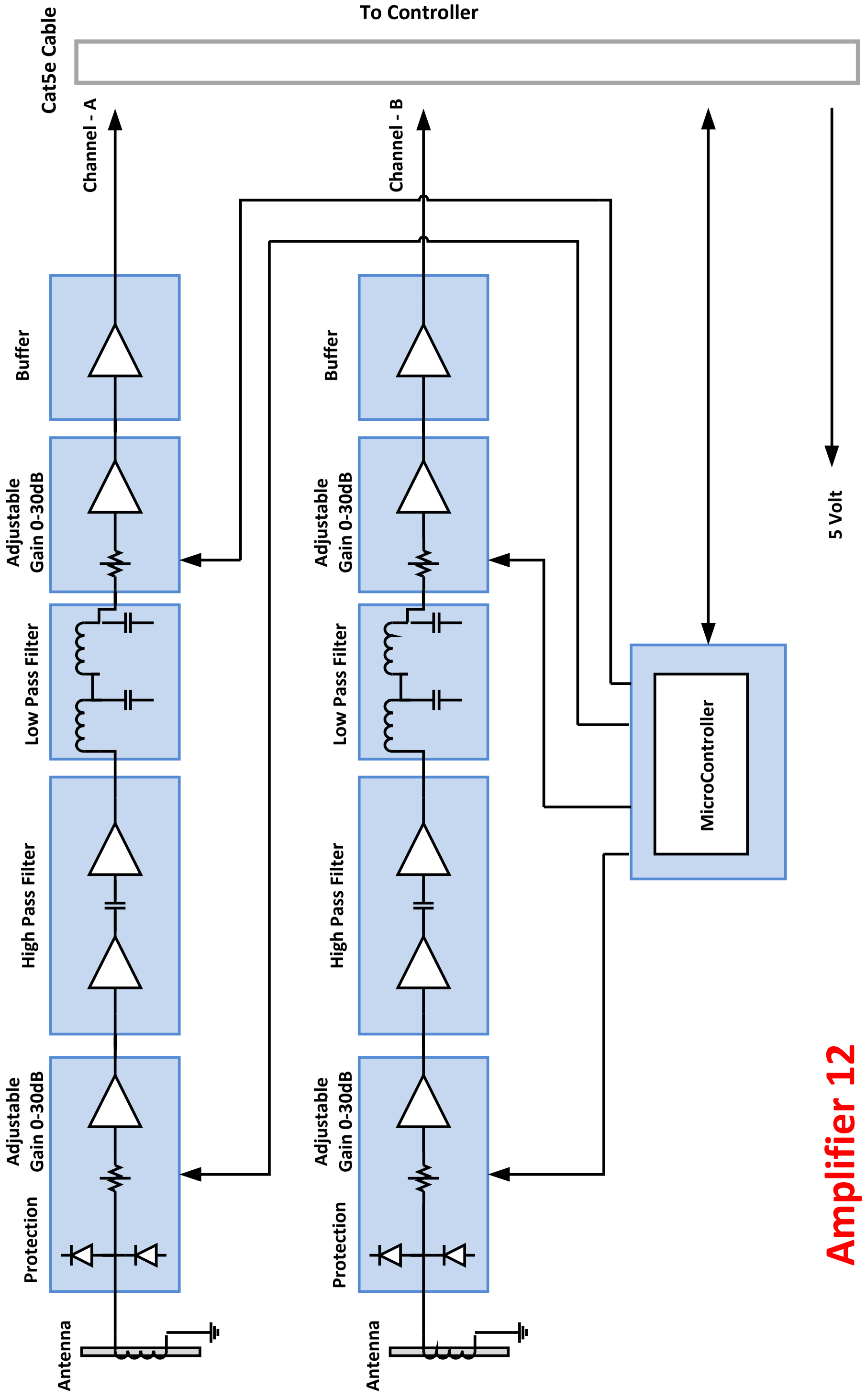


Figure 2: Block diagram of Controller Number 10.3

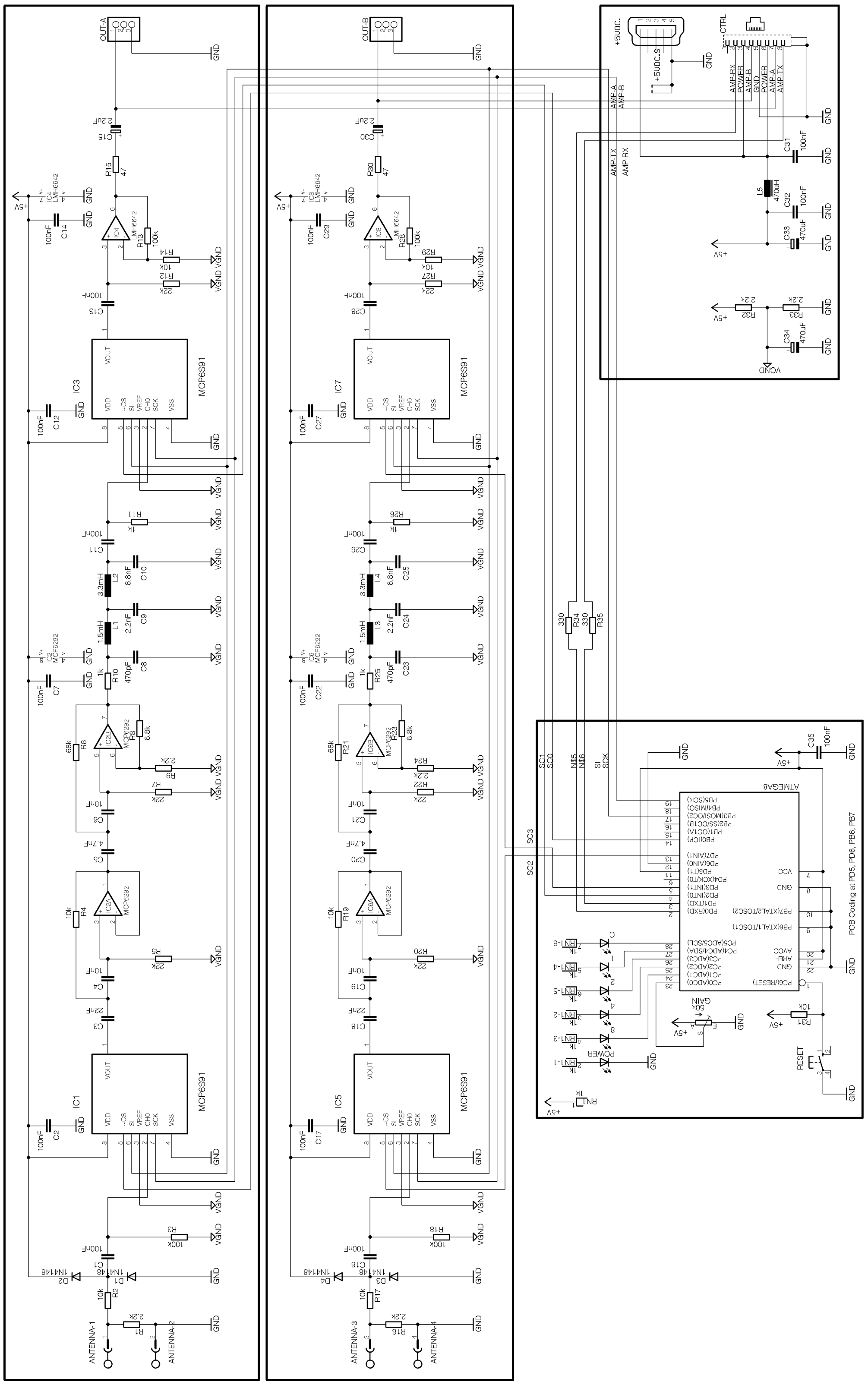


Figure 3: The complete circuit diagram of Amplifier 12.3

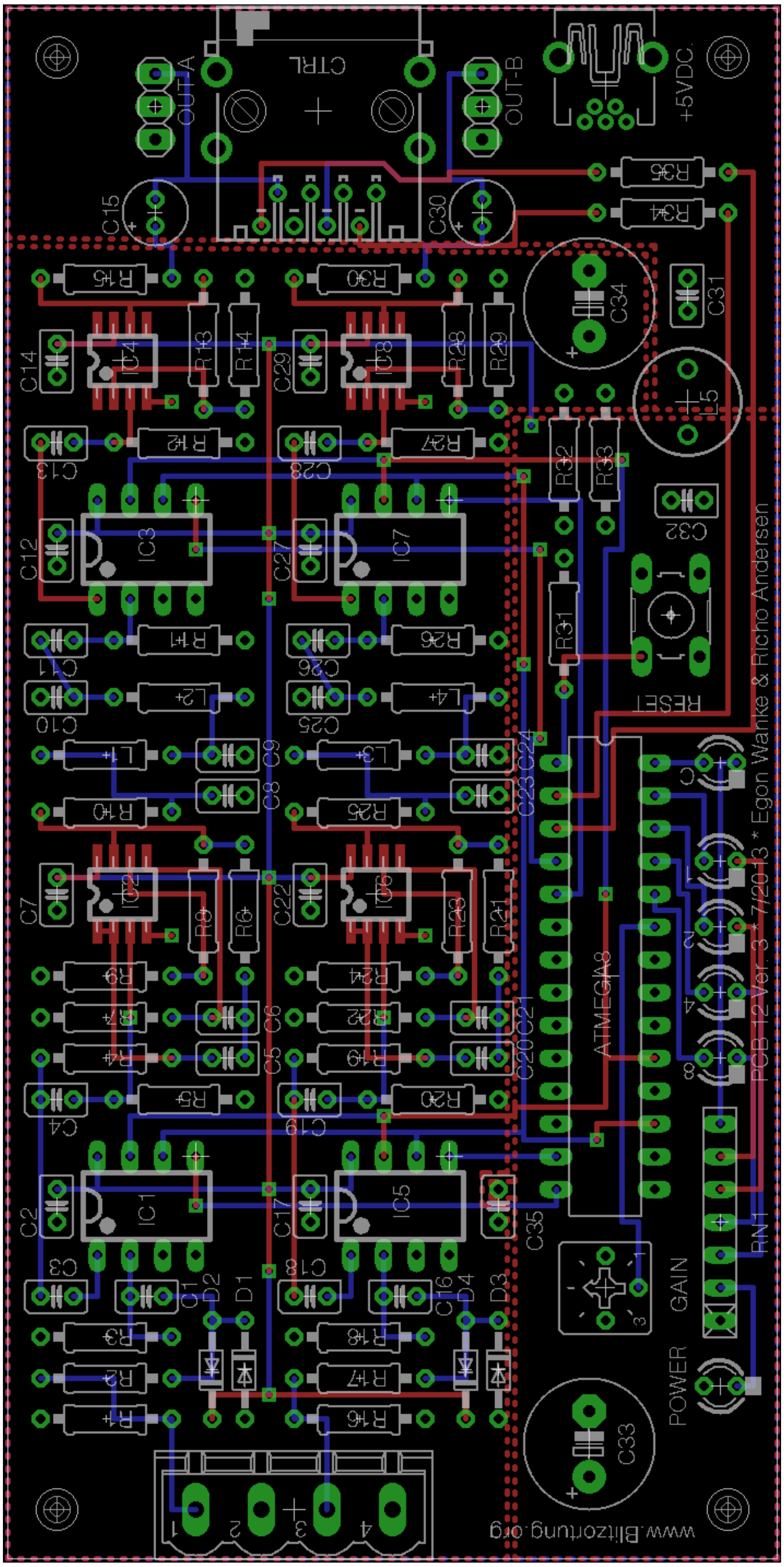


Figure 4: The Printed Circuit Board of Amplifier 12.3

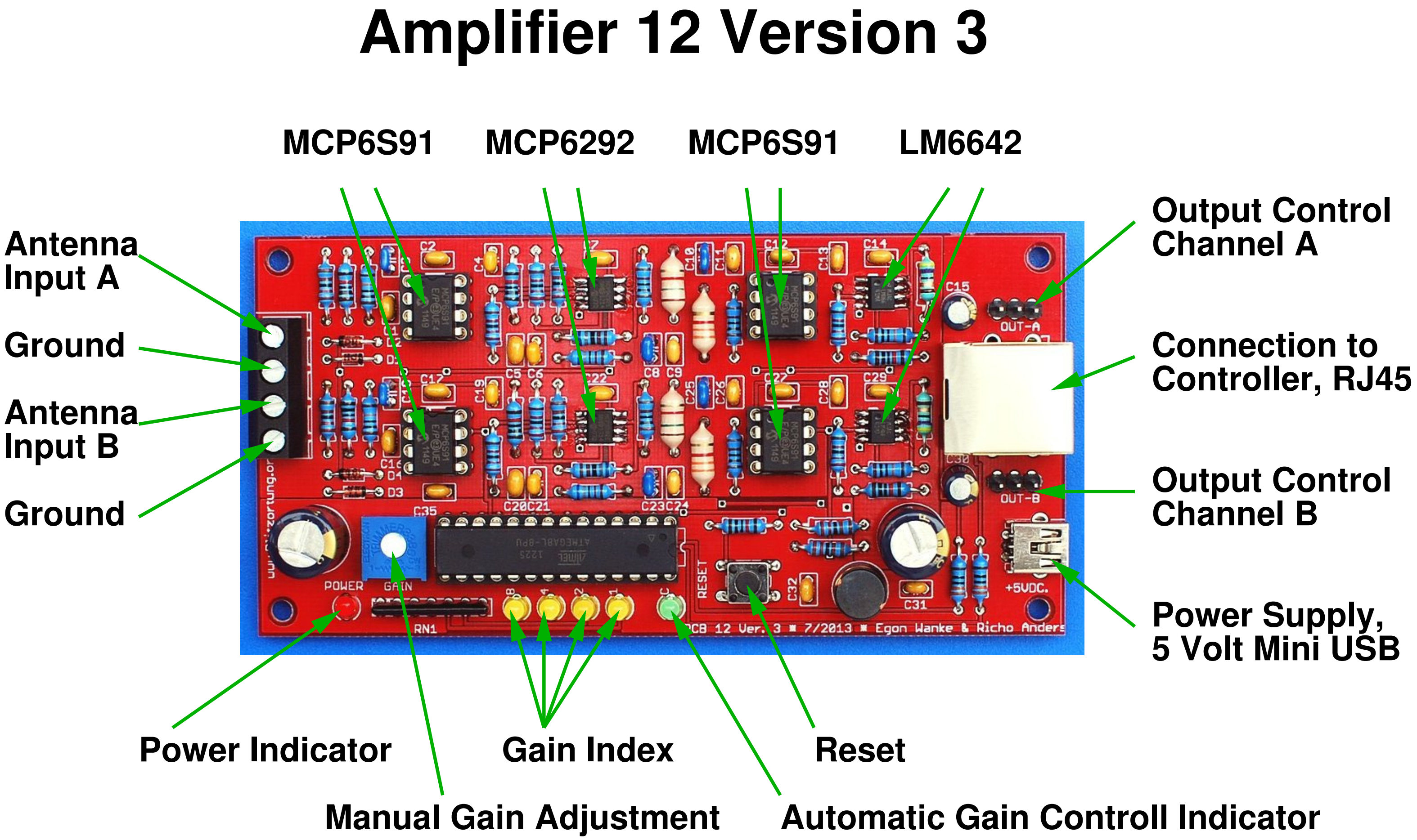


Figure 5: An assembled Amplifier 12 Version 3

1.3 Controller

The heart of our Controller 10.x is an 32-bit ARM Cortex-M4F micro-controller. This controller has 1 MB Flash, 192 KB RAM, and is running with a clock frequency of 168 MHz. Controller 10.x uses the STM32F4DISCOVERY board which also includes an ST-LINK/V2 programmer. The use of this breakout board is only a temporary solution. If the controller works fine with the breakout board then in a later version we will place the ARM Cortex-M4F micro-controller on our controller board.

The ARM Cortex-M4F micro-controller has three 12-bit analog-to-digital converters. These converters are capable to provide a throughput of up to 1,000 ksps= 1,000,000 samples per second even if both input jacks of the controller board are multiplexed. The DMA of the controller is used be store continuously the converted signals into the memory such that we can read out the signal samples stored before the signal reaches a threshold.

The block diagram, circuit diagram, printed circuit board, and an assembled circuit board of Controller 10.3 are shown in Figure 6, 7, 8, 9, and 10, respectively.

1.3.1 Controller 10.x

1.3.1.1 Electronic parts

The electronic parts necessary for Controller 10.3 are listed in Table 4. The additional necessary mechanical parts can be found in the complete part list, see table **??**.

1.3.1.2 STM32F4DISCOVERY board

This board board is placed piggyback on the controller board. It has two push buttons, a black reset button and a blue user defined button. The functionality of the blue push button depends on the state of the program. The four user defined LEDs, LED3 (orange), LED4 (green), LED5 (red) and LED6 (blue) are used to indicate the received signals, the GPS state, the 1PPS signal, the interference mode and other problems.

The other additional hardware of the STM32F4DISCOVERY board is not used by the controller. This especially concerns the motion sensor, the accelerometer, the microphone, and the audio DAC.

1.3.1.3 LEDs

When booting up, the different combinations of the four glowing LED show the steps of the initialization phases. The may be helpful for investigating problems. There are other LEDs on the STM32F4DISCOVERY. One big red/green LED left to the mini-USB connector for the ST-Link firmware programmer and a small red LED on the opposite site as power indicator. Both LEDs can not be controlled by the CPU, so the ST-Link LED is blinking all the time when no USB connection

Breakout board		STM32F4DISCOVERY
Display		LCD 122DIP
GPS module		GlobalTop PA6H (MT3339)
Ethernet Controller		ENC28J60
Voltage transformer		LF33CV
Transistors	T40, T41	BC338
LEDs	GPS fix LED	LED 3mm (green)
Diodes	D40, D41, D42, D43, D44, D45, D46, D47, D48, D49, D50, D51	1N4148
	D52, D53, D54, D55, D56, D57	1N5818
Resistor, metal 1%	R47, R55, R56, R65	10kΩ
	R46, R50, R52, R53	2.2kΩ
	R62, R63, R64	1kΩ
	R40, R41, R57	680Ω
	R54	470Ω
	R58, R59, R60, R61	330Ω
	R42, R43, R44, R45	49.9Ω
	R48	47Ω
	R49, R51	10Ω
Resistor network, star, 8-9	RN42	10kΩ A103
Resistor network, 2 × 3-6	2 × RN41	1kΩ B102
Resistor network, star, 8-9	RN40	100Ω A101
Capacitors, electrolytic, 5mm	C45	470 μF
2.5mm	C44, C48, C55, C56	100 μF
Capacitor, ceramic, 2.54mm	C40, C43, C46, C47, C49, C50, C51, C52, C53, C54, C57, C58, C59, C60, C61	100 nF (104)
	C41, C42	18 pF (18)
Inductances, HBCC	L40, L41, L42	10 μH
Crystal		25 MHz

Table 4: The electronic part list for the controller board Number 10.3

is available. There is another red LED close to the micro USB connector. It can light up when using the analogue user input.

1.3.1.4 Display

The LCD DIP122 graphic display provides a 122×32 dot matrix and is used to show informations about the system. The dot matrix allows to display graphical symbols and graphical informations beside letters and digits. The user can see the state of the network, the state of the GPS module, the state of the amplifiers, and many other useful informations. The display is available with different LED backlight colors. We recommend the most robust and cheapest green display whose size is only 67mm × 27mm. However, the contrast and brightness is more brilliant on the amber display than on the green and blue display. The amber and blue displays have a size of 75mm × 27mm and are a little bit more expensive than the green displays. The green backlight has a higher power consumption than the other ones. The backlight is driven through a transistor, which is connected to a PWM output of the controller. Thus, the brightness can be adjusted by software. A charge pump, which is driven by another PWM of controller, converts 3V to around -5V for the contrast voltage. The correct voltage depends of the display type, supply voltage and temperature. The LCD has an integrated temperature compensation, so the contrast should be stable even when you use the LCD outside.

1.3.1.5 Buzzer

The buzzer of the controller board allows to indicate different states and events by different sound signals. For example, every received signal can be indicated by a beat sound. The buzzer can output an alarm if the system enters the interference mode or if the network connection is lost. The buzzer can also be used as a loudspeaker which allows to output the received VLF signals. This provides a very useful tool to find sources of interferences and thus to find the best place of the antennas in a noisy environment. In any case, the buzzer can also be turned off.

1.3.1.6 GPS Module

The controller needs a GPS device with a serial TTL level interface and a 1PPS signal. We recommend to use the on-board GlobalTop PA6H GPS module. This module allows you to connect an external antenna with an SMA connector. Most external antennas have a connection cable between 2 and 5 meter. The longer the cable the larger the loss. The green LED beside the GPS module stops blinking when the GPS has found its position. Before PCB version 10.3b it will glow steady. Since PCB 10.3b it stays off.

If you do not want to use the GlobalTop PA6H GPS module, we recommend to use the NaviLock or GlobalSat EM-406A GPS module. The pin assignment of the GPS connector on the controller board is shown in Table 5. This pin assignment is identically to the pin assignment of the EM-406A. Although other GPS modules will work, the firmware of the controller currently only supports the MediaTek MT3339 and SiRF chip sets.

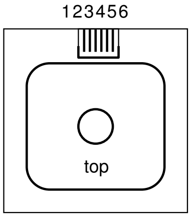
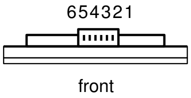
Pin	← EM-406A module		GPS Connector →	Pin
1	ground			1
2	+5 Volt			2
3	serial input TTL			3
4	serial output TTL			4
5	ground			5
6	1PPS output			6

Table 5: Pin assignment of the EM-406A module and its connection to the GPS connector

1.3.1.7 Amplifier connectors

The Controller has two RJ45 connectors for two amplifiers with maximal 3 channels. The firmware automatically detects to type of the connected amplifier. Both connectors can be used independently. Please do not connect old amplifiers version 5.x and before!

1.3.1.8 Ethernet

The controller board is connected to a local area network by a shielded CAT network cable to the RJ45 network connector. This jack has integrated transformers and LEDs. There are only very few compatible jacks, so have a close look into the datasheets when you want to use another connector! Although the ARM Cortex M4 based STM32F4 micro-controllers already have an internal Ethernet controller, they do not include an Ethernet Physical Layer Transceiver (PHY). For experimental reasons, we additionally have placed the Ethernet controller ENC28J60 on the board of Controller 10 .x . The ENC28J60 a very widely used "all-in-one" ethernet controller, but it can only handle 10Mbit/s enthernet connections. It has a relatively high power consumption of araound 150mA. The ENC28J60 is connected to the processor via SPI. In a forthcoming version, we intend to remove the ENC28J60 and to replace it by an Ethernet Physical Layer Transceiver on the controller board. Then we will use the internal Ethernet controller of the processor, which uses DMA and can also handle 100MBit/s connections.

1.3.1.9 Serial interface

The board has a 3.3V TTL level serial interface, which is 5V tolerant. This interface can be used for debug purposes. The connector also provides the power supply for a TTL to RS232 or TTL to USB converter. The pin assignment is printed on the board and can also be found in the schematic.

1.3.1.10 First hardware check

For the first hardware check, do not connect anything. Remove the STM32F4DISCOVERY board and remove the display. After connecting a 5V USB power supply, the green GPS fix LED should start blinking, if you are using the GlobalTop PA6H GPS module. Next remove the power supply and mount the STM32F4DISCOVERY board and the display. Close Jumper 1 only if you are using the green display, otherwise, if you are using the amber or blue display, open Jumper 1. Then connect the power supply again at the mini USB connector of the controller board, not at the mini USB connector of the STM32F4DISCOVERY board. Some of the LEDs on the STM32F4DISCOVERY board should start blinking. The display should not show anything, because the STM32F4DISCOVERY board do not contain our firmware yet. The next step is to upload the firmware. Note that the USB connectors on the STM32F4DISCOVERY board are only used to program the board, not for power supply.

1.3.1.11 Controller PCB change log

10.2	First release
10.3	Added ground connector Added MP joint
10.3b	Changed polarity of GPS LED by request of manufacturer to prevent malfunction

Table 6: Change log of controller 10

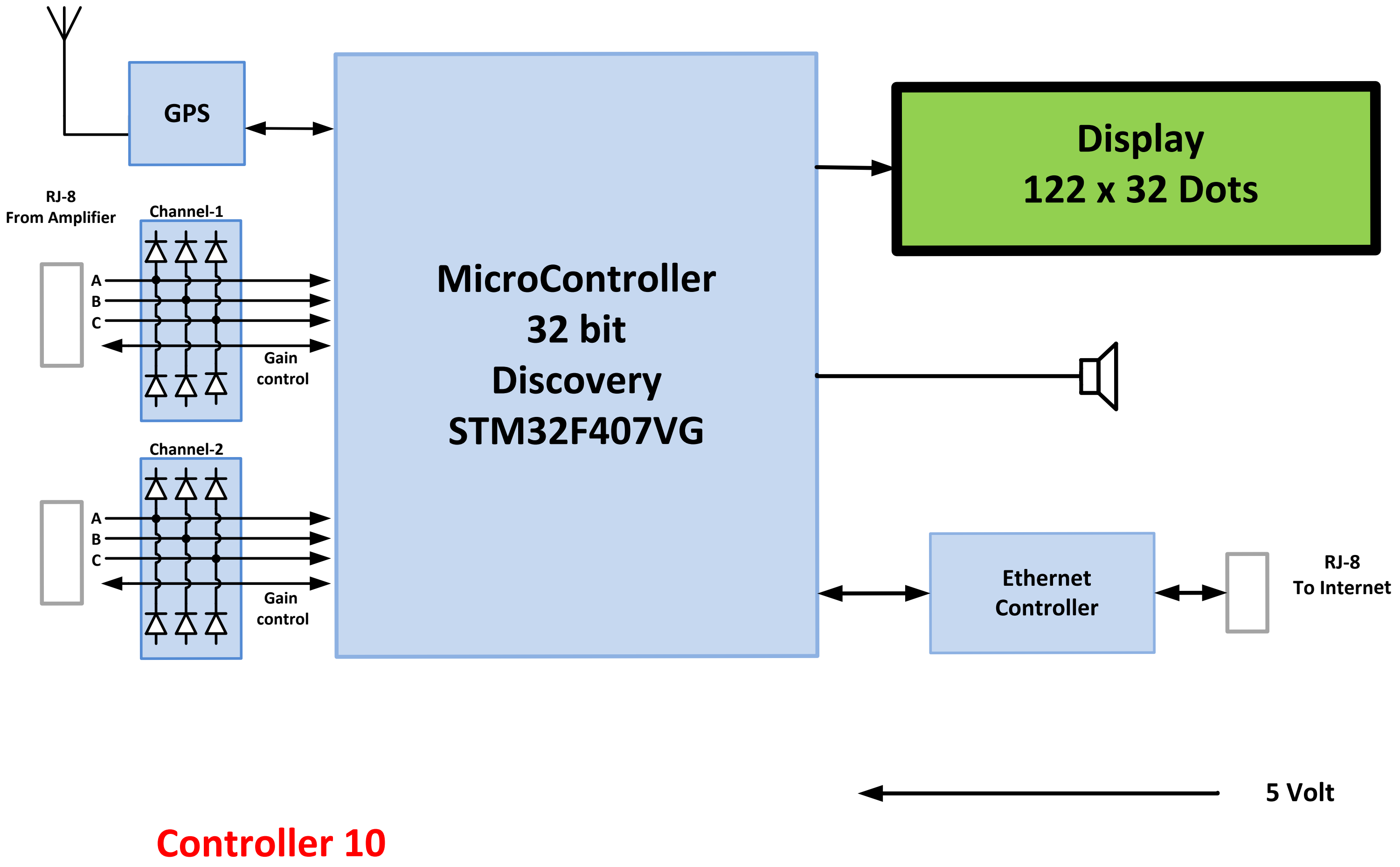


Figure 6: Block diagram of Controller 10.3

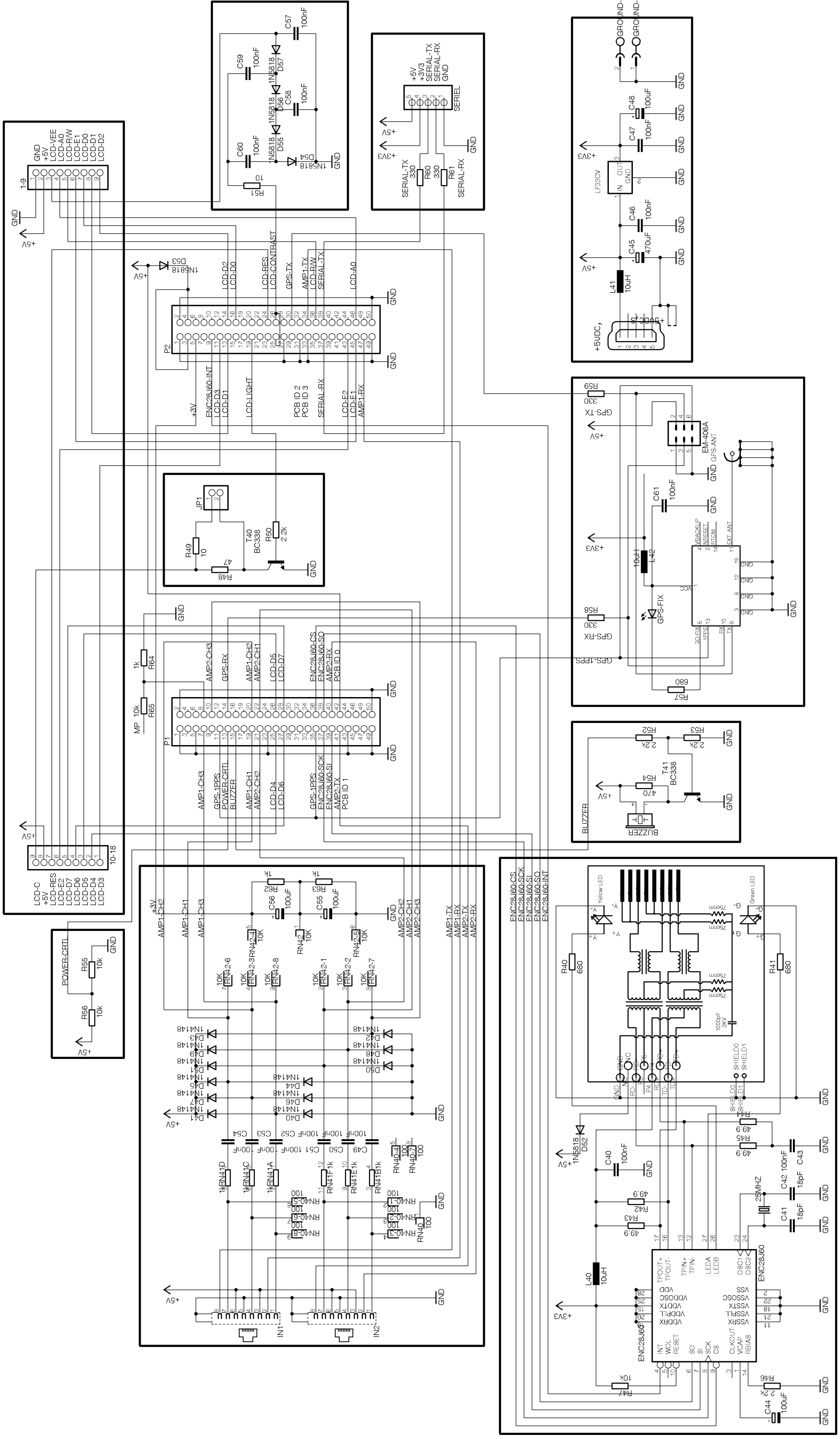


Figure 7: Circuit diagram of Controller 10.3

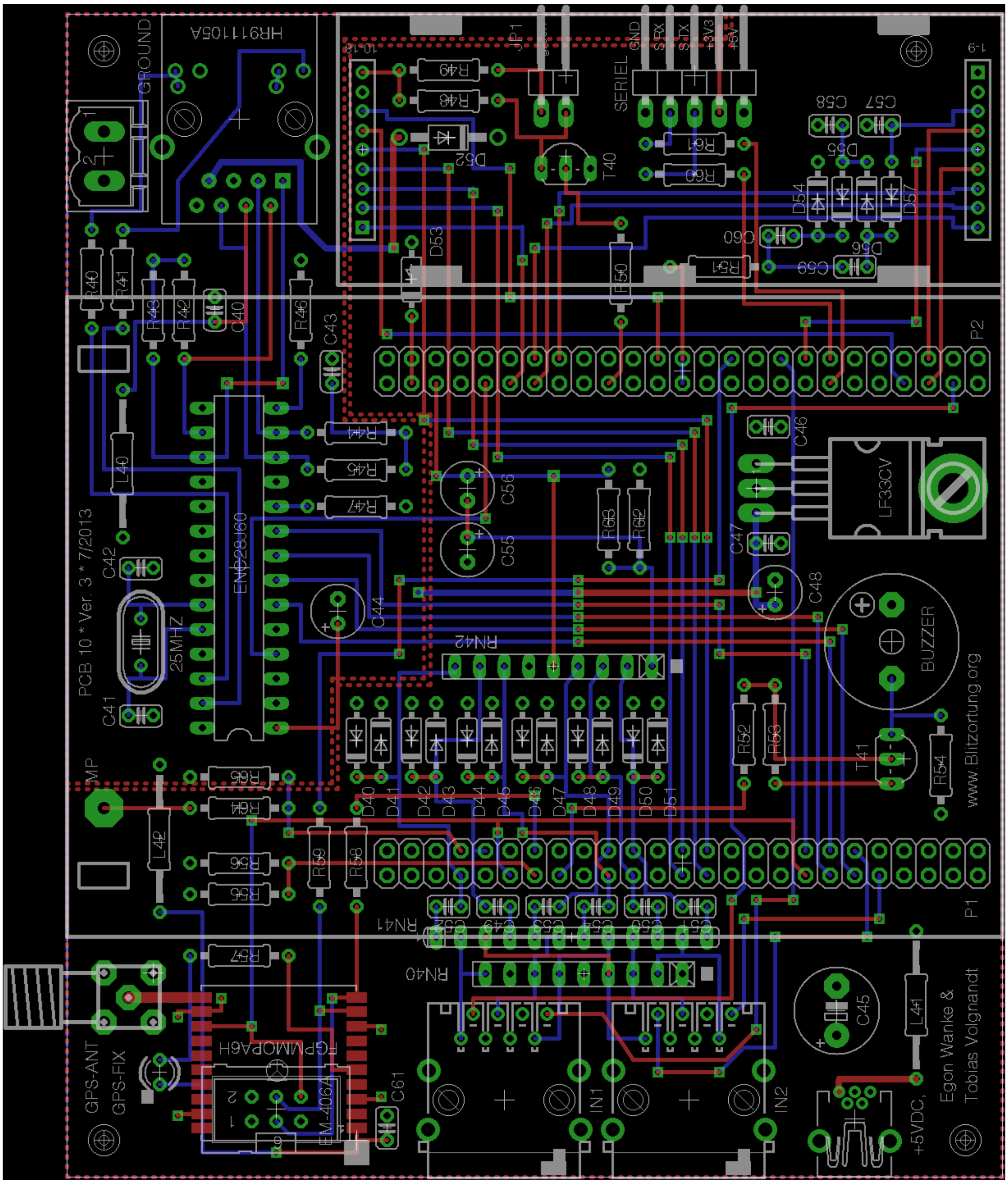


Figure 8: Printed Circuit Board of Controller 10.3

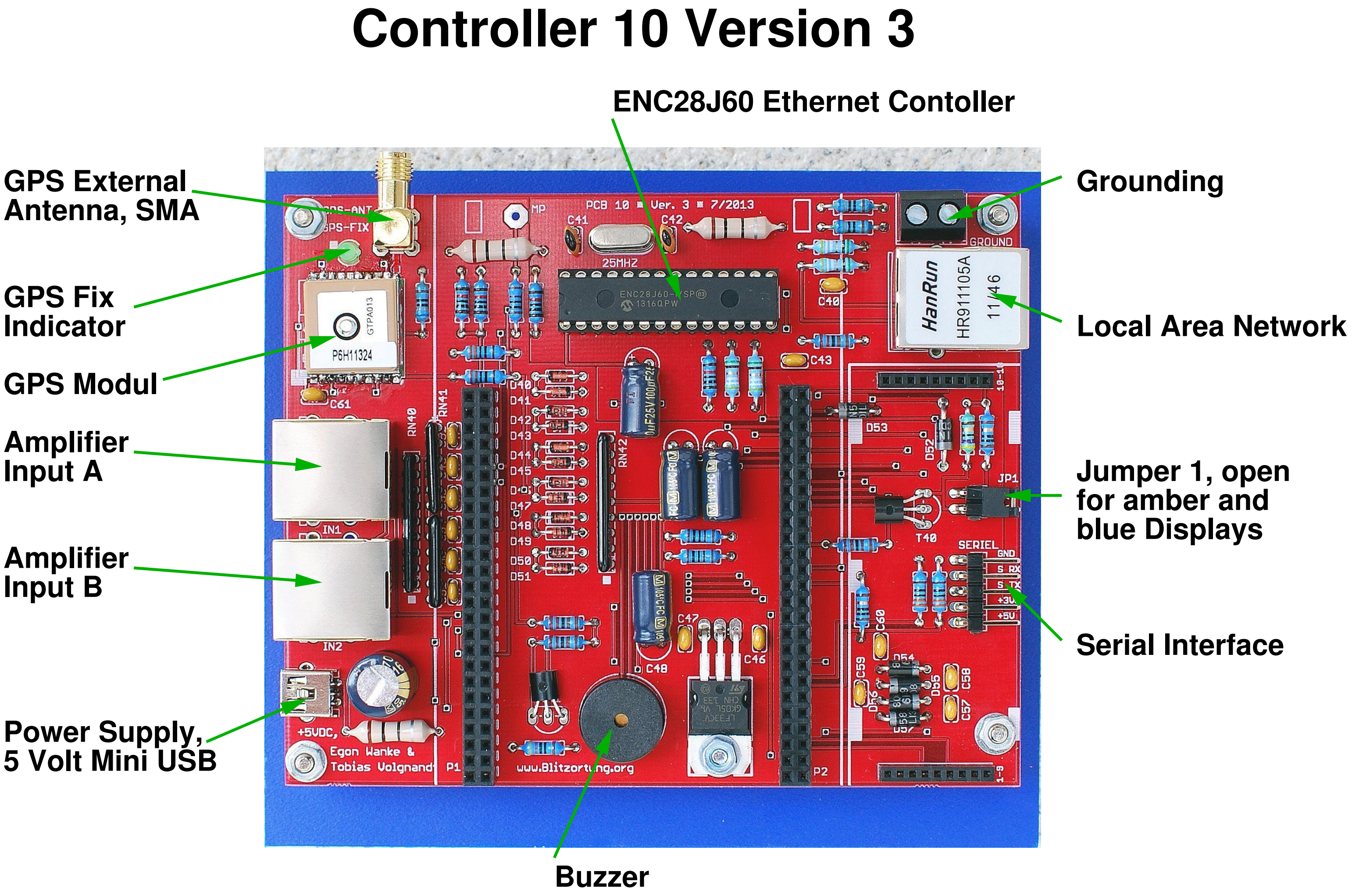


Figure 9: Assembled Controller 10.3

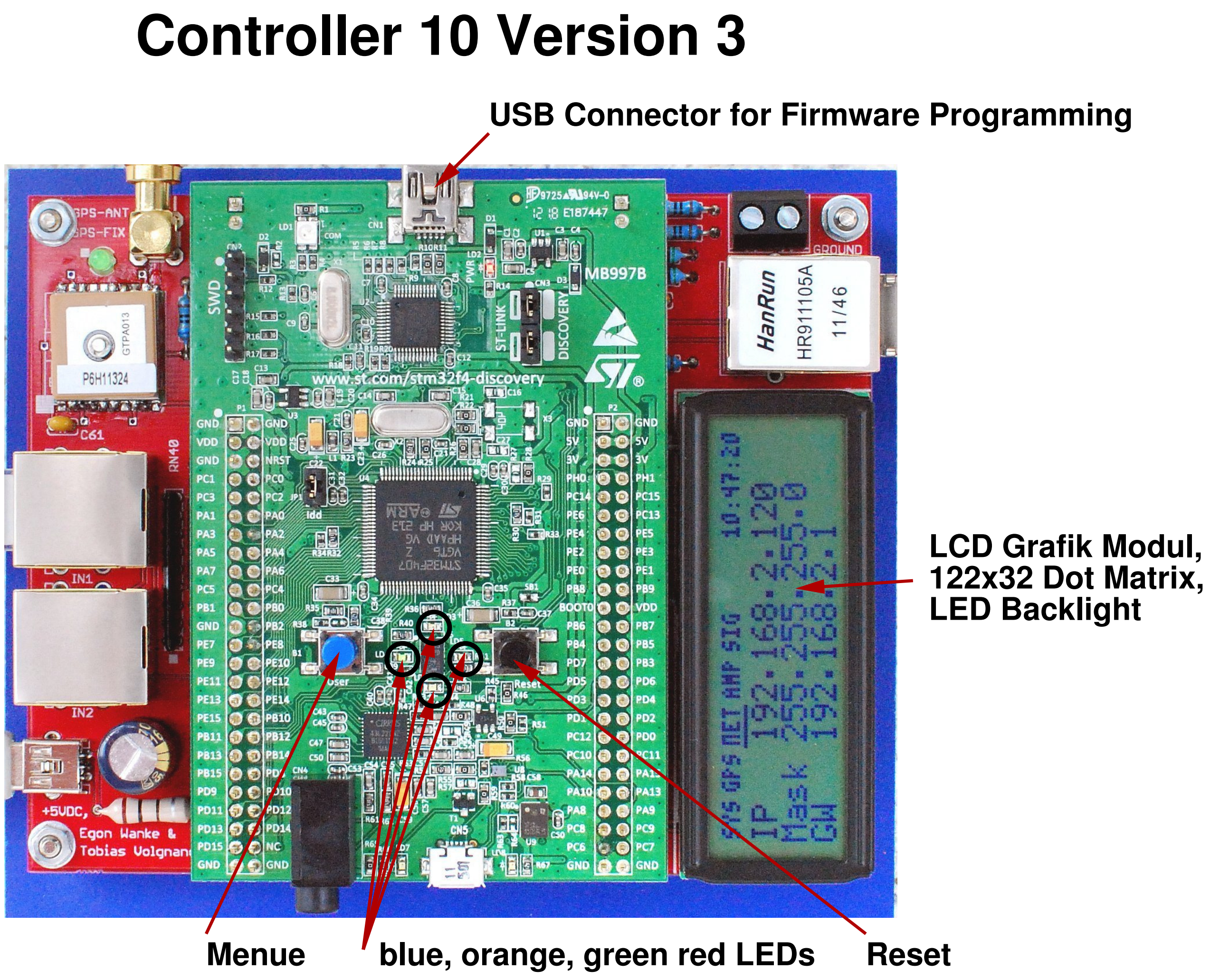


Figure 10: Assembled Controller 10.3 with STM32F4DISCOVERY board and display

1.4 Power supply

The amplifier and the controller can be powered by a 5V USB power supply. Both boards have a mini USB jack. The mini USB jack of the STM32F4DISCOVERY board can not be used to supply the system. This jack is only used for programming the STM32F4DISCOVERY board. The +5V line of both boards are connected via the connection cable. It is neither necessary nor recommended to power the system at both ends. We recommend to power the system at the controller site.

When the boards get power the red power indicator LED on the amplifiers lights up. On the controller board the power on LED of the STM32F4DISCOVERY board and some of the other LEDs will be light on. The power consumption of the amplifier and controller is about 40 mA and 400 mA, respectively.

1.4.1 PoE

The system can also be powered over the Ethernet connection cable if the J0006D01BNL RJ45 Ethernet Jack from Pulse Electronic is used. This Power over Ethernet (PoE) technique use pin 7 for +5V and pin 8 for ground. To inject the power, a simple PoE injector is needed. As only 5V is used as supply voltage, the network cable must not be longer than a few meters. If the voltage drop is too high, you can try to inject up to 6V. Don't use higher voltages! Note: This pseudo PoE is not compatible with the official IEEE standard 802.3af!

1.5 Grounding

The receiving system should have a good ground to operate efficiently. The ground of the amplifier and thus also the ground of the controller board should be connected to a large expanse of metal such as a safety hand rail around a roof or a heating core. Note that the ground of the system should never be connected to the ground wire of the electrical supply system. Usually some grounding is already given if the shielded network cable is connected to the router.

In some situations, grounding can inject even more noise in your system. Grounding on both, controller and amplifier, can also produce some negative side effects. If grounding is needed for your system, we suggest connecting it on the controller only, as it has a special terminal block for that purpose. See also Figure 9.

1.6 Assembling advice

The expected assembling time of the amplifier and controller together is approximately four hours. Most problems are caused by working too fast or too superficial. The parts should always be soldered according to their mounting height.

1.6.1 Parts with a low mounting height

Examples: SMD op-amps, diodes, resistors, ceramic capacitors, axial inductances, the crystal.

These parts should be assembled at first. It is very important that you solder them carefully, because it's hard to do changes there after the other parts have been mounted.

- Pin 1 of the ICs is marked by a dot at the housing. These pins are marked by a white dot at the boards.
- To mount the SMD parts (IC2, IC4, IC6, IC68 and the GlobalTop PA6H GPS module), first solder only one of the four corner pins. Then unsolder and solder this pin repeatedly until the component is properly placed. After that solder the other pins. It is not necessary to use solder paste.
- The cathode $(-)$ of the diodes is marked by a black or white ring at the housing. These sides are marked white on the boards.
- The values of the resistors should be verified by a multimeter, because it is sometimes difficult to distinguish between certain colors, as for example, between red and brown.

1.6.2 Parts with a medium mounting height

Examples: Sockets, mini USB jacks, LEDs, push-button, potentiometer, resistor networks, the GPS module, transistors, the voltage transformer, mount sockets for the display

- The anode $(+)$ of a LED has a longer wire than the cathode $(-)$. The hole for the anode is marked by white squares on the board.
- Every resistor star network has a dot at one side. This side has to be placed at the crossed pin beside the red power LED. Resistor Network RN41 is not a star network. It is either realized by a 6-12 network or by two 3-6 networks. These networks are symmetric can be mounted arbitrarily.
- Both transistors and some of the radial electrolytic capacitors have to be folded down on the controller board such that the STM32F4DISCOVERY board and the display have enough space.
- The two 1×10 mounting sockets for the display have to be shortened to 1×9 by pinching off the last pin with a cutter and smoothing the border with a file. To ensure that the mounting sockets for the display will be correctly aligned, carefully insert the display's pins into the mounting sockets before soldering them to the controller board, see also the hint for soldering the mounting sockets for the STM32F4DISCOVERY board.

1.6.3 Parts with a high mounting height

Examples: Radial inductance, radial electrolytic capacitors, terminal block, buzzer, headers for the output control, board mount sockets for the STM32F4DISCOVERY board, RJ45 jacks.

- The positive pole of an electrolytic capacitor has a longer wire. The hole for this wires is marked by a + symbol.
- The center pin of the 3-pin header for the test pin of the amplifier can be removed. It is not connected.
- To ensure that the double row mounting sockets for the STM32F4DISCOVERY board will be correctly aligned, carefully insert the STM32F4DISCOVERY board's pins into the mounting sockets before soldering the sockets to the controller board.
- On the Buzzer there is sticking a sign marking the positive pin.

2 Operating Instructions

The lightning detector is put into operation as follows:

1. Write a short email request to blitzortung@gmx.org for a new user account if you do not already have one.

   You need only one account independently of the number of station you operate and regions you supply.

2. Install the System**RED** firmware on the STM32F4DISCOVERY board as described in Section 2.1.

   The link to the firmware pages can be found in the services section of www.Blitzortung.org.

3. Connect the controller to a local area network using DHCP (or static IP 192.168.1.235) and use a web browser to open the web interface of the controller. There you can adjust the lightning detector, see Section 2.2.1.

   You find the IP address of the controller at the Display.

4. Assign the station at the `Services→User-data` page of www.blitzortung.org to your user account. That is, insert the Device-ID of your STM32F4DISCOVERY board to the field "Assign a new Station by Device-ID:" and acknowledge by "Save".

   You will find the 24 digits hexadecimal Device-ID on the webpage of the controller at the `Status` page in paragraph `Controller/CPU` paragraph. The station prints this ID also on its LCD, until it is assigned. Do not insert the "-" characters between the 4-digit blocks.

5. Check from time to time that your system is still running and then enjoy your lightning detection.

2.1 Firmware Installation

The firmware on the STM32F4 controller is the heart of the System**RED**. It initializes and controls all peripherals and does the communication with the servers. Compared to the old system, the firmware is very complex but also very powerful. Especially during the first months after start of System**RED**, there will be several updates to improve stability and add new features.

After assembly of the system, you have to install the firmware over the dedicated USB port. There is already a small pre-flashed firmware on your new STM32F4DISCOVERY, which lets the LEDs blink according to accelerometer value. This is only a demonstration and it will be overwritten by our firmware.

Never flash any other 3rd party firmware on the STM32F4DISCOVERY, while it is connected to our PCB! Otherwise parts on both of the PCBs or the CPU might be damaged!

2.1.1 Flash by using on-board programmer

The STM32F4DISCOVERY has an *ST-Link V2* programmer integrated. The flash procedure is easy and takes only some seconds. It is almost impossble to brick the STM32F4DISCOVERY, even if you pull the cable while flashing, because the on-board programmer has direct access to the flash memory. Please never set any user option bits, otherwise the flash might be write protected forever!

There are several tools for flashing the STM32 family on different operating systems. These tools can be used for different hardware types and are not specific for our hardware. You have to download them from their official websites:

- STM32 ST-Link utility
- CooCox CoFlash
- stlink (Linux/Mac)
- *Several others...*

For the following description, we use the official STM32 ST-Link utility (version 3.0.0). First of all, download the tool here. If you don't find the download, then try Google. After download, run the setup. The driver is also included in the package. Follow these steps, to program the flash:

1. USB connection: Connect the mini-USB jack on the STM32F4DISCOVERY with a cable to your computer (see fig. 11). If you do this the first time, the driver will be installed, which can take a while. While connected to the USB port, the CPU gets the necessary power from your computer. It doesn't matter whether the system is powered by external 5V or not while flashing. There is no need to remove the Discovery from the red PCB, as there is a good chance to damage it mechanically or electrically during this procedure. When the connection is established successfully, the big ST-Link LED should stay red. If it is blinking red, then there is a connection or driver problem.

2. Open the programmer window (see fig. 12): Start the *STM32 ST-LINK Utility*. In this tool, select *Target* → *Program...* from menu, click *Browse* and select the Firmware (.bin) file.

3. Start flashing (see fig. 13): Check if the Start address is 0x08000000 and click on *Start*. The data is now programmed into the flash. It takes several seconds.

4. Reset: If the CPU didn't perform a reset after flashing is done, simply press the black button.

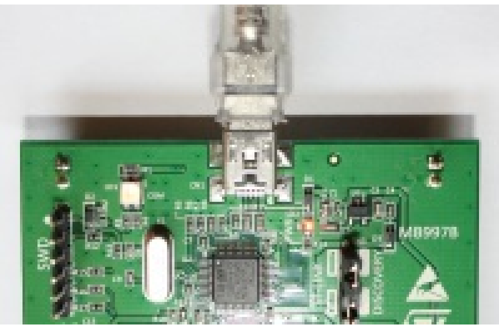
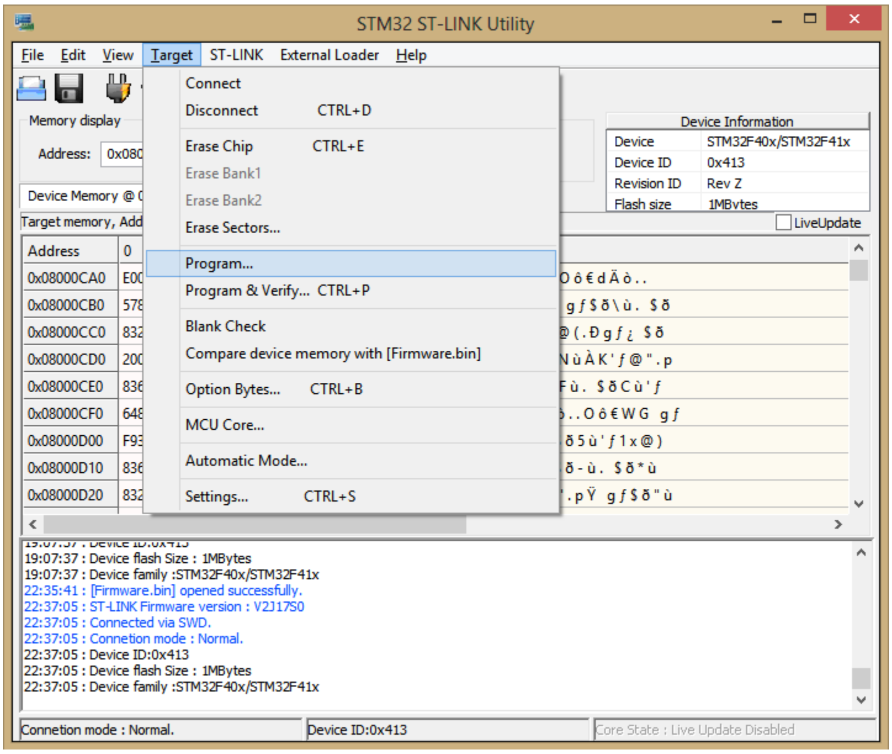


Figure 11: Mini-USB connection



Device Information

Device	STM32F40x/STM32F41x
Device ID	0x413
Revision ID	Rev Z
Flash size	1MBytes

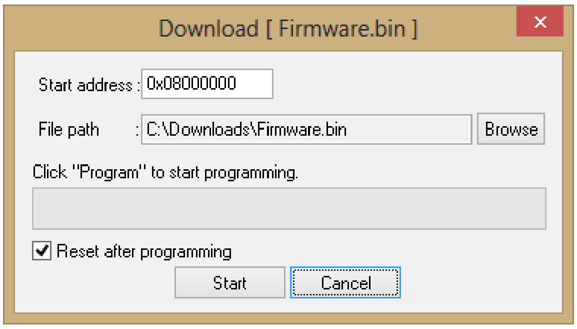
```
19:07:37 : Device ID:0x413
19:07:37 : Device flash Size : 1MBytes
19:07:37 : Device family :STM32F40x/STM32F41x
22:35:41 : [Firmware.bin] opened successfully.
22:37:05 : ST-LINK Firmware version : V2J17S0
22:37:05 : Connected via SWD.
22:37:05 : Connetion mode : Normal.
22:37:05 : Device ID:0x413
22:37:05 : Device flash Size : 1MBytes
22:37:05 : Device family :STM32F40x/STM32F41x
```

Connetion mode : Normal. | Device ID:0x413 | Core State : Live Update Disabled

Figure 12: Open the programmer window

Windows 8 Driver Installation

The driver itself works perfectly with Windows 8 (32 and 64bit), but unfortunately it is not possible to install the driver the usual way, as it doesn't have a valid signature. You must disable this check with the following procedure:



Download [Firmware.bin]

Start address : 0x08000000

File path : C:\Downloads\Firmware.bin

Click "Program" to start programming.

☑ Reset after programming

Figure 13: Start flashing the new firmware

1. Press Windows+I and you will see the Charms Bar
2. Click on power while holding the Shift key and select Restart
3. After some seconds, you should see a troubleshooting page. Select Troubleshoot and then Advanced Options.
4. Then select the Startup Settings and click Restart again. The system performs a reboot.
5. In the boot menu, press 7 or F7 to disable the driver signature check.
6. Now you can install the driver.

This procedure has to be made only once or when changing the driver.

2.1.2 DFU Method

The STM32F4 CPU has an integrated boot loader, which is write proceted. This boot loader allows programming the flash through the serial or USB interface. There is currently only need for using this feature when you have problems with the method above. But we may need this method on future controller boards, where the CPU is integrated on our PCB.

ST provides a tool for USB-DFU mode which you can download here. After installation, connect your Discovery with the micro-USB to your computer, jumper the pins BOOT0-VDD on the STM32F4DISCOVERY and attach the power supply to the controller. Use *DFU File Manager* to create a DFU-file from the Firmware binary and *DfuSe Demonstration* to flash you controller. After flashing, remove BOOT0-VDD and push reset. A more detailed description will be published later.

This method is as robust as the way through the ST-Link programmer. That means, even after a failed flash procedure, the CPU still be programmed with the DFU method.

2.1.3 Flash through the web-interface

Since Firmware version 4.0, there is an integrated firmware update tool available in the web-interface. We can not guarantee for 100% functionality, but it is the easiest and fastest way to update the firmware. It even works through slow or unstable connections. If it fails, you can still use the methods described above. For more information, refer to the guide integrated into the controllers web-interface in the *About* section.

2.2 User interfaces

2.2.1 Web-Interface

To access the web-interface, enter the IP-address shown on the LCD in your browser. In most cases, the IP-address will be assigned automatically by DHCP. However, if the assignment fails or you don't have a DHCP-server running in your network, then the controller uses it's static IP-address. The default one is *192.168.1.235* with a subnet-mask of *255.255.255.0*. To access the controller, your computer must be in the same subnet, i.e. with an address of *192.168.1.2*.

The web-interface supports modern browsers only. If the browser is too old, some features might not work. For web-developers: The webpages use a lot of CSS and some jQuery. You are welcome to improve or add several functions, especially the signal view, but ask us first. Due to lack of flash memory, some static files are compressed.

Important: The firmware uses LwIP, a lightweight IP-Stack for embedded microcontrollers. We can not guarantee for 100% security of this stack or our web-server. There is always the possibility of security holes in such implementations, even when authentication methods are enabled. That means, you shouldn't connect the controller to foreign networks or give access to the controller from the internet with port forwarding. We think the risk of an attack is small, but we would never guarantee that.

2.2.2 LEDs

Every of the four user LEDs has a specific purpose:

- The **Green LED** starts blinking when the controller receives data sentences from the GPS. It will keep glowing, when the GPS locked the position.
- The **Blue LED** flashes every time the controller got a 1PPS signal. That means it should blink exactly every second.

- The **Orange LED** flashes every time the controller receives a signal from the amplifiers that reaches the adjusted threshold.
- The **Red LED** is blinking constant and slowly if the system has entered the interference mode. If signals can not be send, i.e. because there is no network connection or no valid GPS signal, then the red LED blinks at the same time as the orange LED.

During boot sequence, the LEDs will be switched on and off on specific steps. If you experiencing problems during boot, you should tell us which LEDs are glowing.

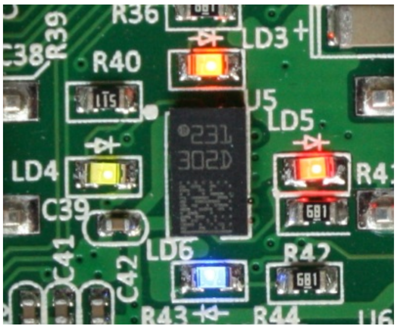


Figure 14: The four SMD LEDs

2.2.3 Buttons

Pushing the **Black Reset Button** results in a hardware reset of the CPU. It is independent from the firmware and works always. The **Blue User Button** has several functions, especially in combination with the LCD. It can also be used to restore the settings to default values, by pushing both buttons together, release the reset button and wait for several seconds until a long beep occurs. The other functions depend on the installed firmware and might be extended in future versions. Refer to the guide in the controllers web-interface for latest information.

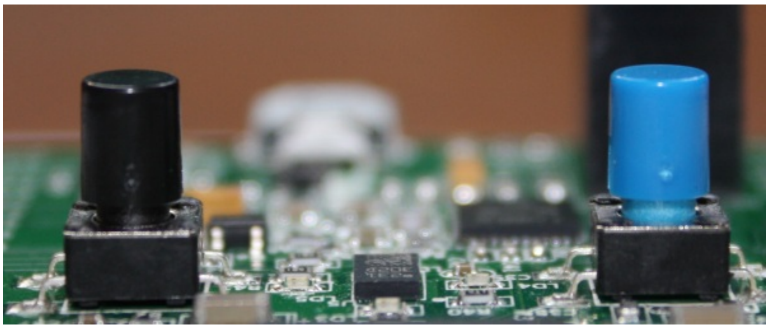
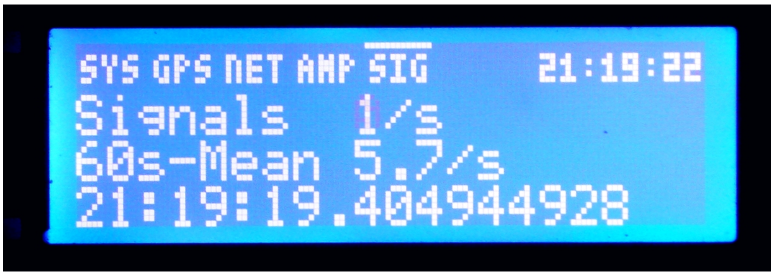


Figure 15: The two buttons

2.2.4 LC display

During boot, the display shows some status information. After that, and while the system is running, there is a status line at the top and according information on the rest of the screen. The status line shows whether there are pending problems.

The main reason for the LCD is to provide an easy installation and a quick overview of the current status. The system would even work without it. There are much more detailed information visible in the web-interface. Changing configuration can not be done through the LCD. For latest information about the display, refer to the guide in the controllers web-interface.



```
SYS GPS NET AMP SIG      21:19:22
Signals   1/s
60s-Mean 5.7/s
21:19:19.404944928
```

Figure 16: The blue LCD showing the signal page

2.2.5 Buzzer

The main purpose of the buzzer is to inform you about status changes. It will beep in high tones on successful actions, on low tones when an error occured and a medium tone during boot. Longer duration of the beep means higher priority.

For example, shortly after the first beep during boot, the buzzer should beep a second time, when an connected amplifier has been identified. If no amplifier is present, it will beep with a low tone. As long as you didn't solve this problem, the controller won't detect any strokes. Another example: Beeps for interference mode are slightly shorter, as they can happen from time to time.

The buzzer is connected to the CPUs digital-analogue-converter DAC. That means, it can even play individual sounds. The firmware uses this feature by sending the recorded signals from the ADC to the DAC (when enabled). Thus, it is possible to hear the atmospheric discharges without additional hardware. This is very useful to find a noise-free place for the antennas. Check the web-interface for detailed settings on that feature.

2.2.6 Analogue user input

The analogue user input is an optional feature, which has nothing to do with lightning detection. When enabled, you can measure voltages from -3 to 30V at the measuring point. You will see them on the web-interface below CPU temperature and on the LCD on the first page. The voltage range can be changed by replacing resistors R64 and R65 with other values. You also have to change those values in the settings section on the web-interface. The default values are 10kΩ for the serial resistor R_s (R65) and 1kΩ for the R_GND (R64).

2.2.7 Debug Log

The controller prints several information in its debug log. This happens during start-up, when configuring peripherals, in case of an error or status changes. If you experience problems, these messages can be very helpful. When contacting the developers, you should try add them to your request. There are two ways to view the messages:

1. Open the debug log on the web-interface and wait for the first messages to appear. Of course it is not possible to view messages when the network connection is not available. Due to technical limitations, some messages might not get transmitted. Note: You have to disable the auto refresh before you can copy the text to clipboard.
2. Connect a serial-to-USB converter to the debug port below the LCD. The baudrate has to be set to 115200. Check 1.3.1.9 for hardware details. You will see all messages, even those from early start phase.

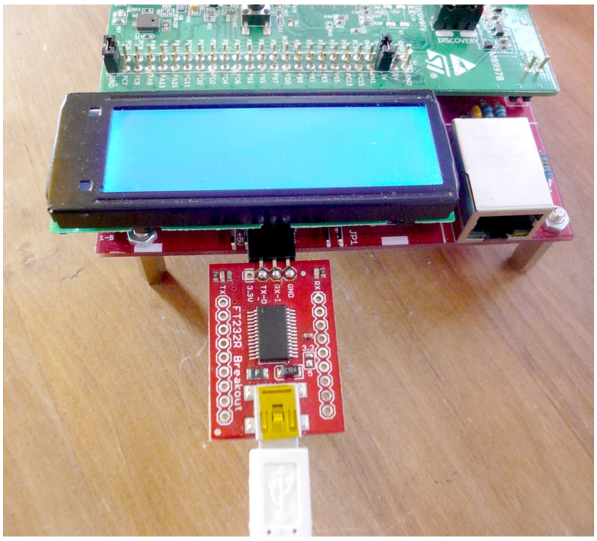


Figure 17: A serial-to-USB adapter connected to the controller

In the web-interface settings you can enabling more verbose logs. You can use that for further investigations. Please don't enable them for normal operation, as it can slow down the system dramatically, even when not using one of the methods above!

2.3 Interference Mode

The controller can not distinguish between electromagnetic signals which have been sent out by lightning discharges on the one hand, and which are a result of local noise on the other hand. There is a variety of such sources which can generate interference, i.e. power-supplies, energy-saving lamps and almost all electric home equipment.

The **interference mode** is used to reduce the amount of improper data send to the computing servers. It will be enabled, when the amount of signals is higher than a defined value. The threshold value for disabling the interference mode is several percents lower, which avoids conditions where the mode toggles between on and off all the time. The thresholds can not be changed by the user.

In detail, there are two interference modes implemented. The first one is very fast and checks the signal rates of a very short time period, i.e. of one second. This is very useful to protect the server against short bursts, which take only some seconds but produces a lot of interference signals. The second one is a slow interference mode, which checks the amount of signals during a longer time span, i.e. 60s. That means, it will be enabled on medium amount of interference signals over a longer time span.

2.4 GPS

The main purpose of the GPS is the 1PPS signal which gives all worldwide station the same absolute time base. The 1PPS-line is directly connected to an I/O pin of the STM32F4 MCU. It is also connected to an ADC input pin to check the quality of the 1PPS-pulse in case of an error.

Position data is sent by the GPS module every second over an serial bi-directional connection to the MCU. The MCU itself can send commands to the GPS, but does that only if needed (i.e. during initialization). The baud rate for that connection is variable and can be set in the web-interface. However, you should avoid slow baud rates as it can take too long for the GPS to send the whole data before the next 1PPS-pulse. We recommend baudrates of 38400 and higher.

In the GPS settings you have the option to enable **SBAS**. This is a service which can improve position accuracy and is provided for different regions of the world (i.e. *EGNOS*, *WAAS*). The accuracy will be improved due to additional information about current status of the ionosphere. The needed data is sent by satellites on geostationary positions. To receive their signals, your GPS antenna needs a good view to them. For the northern hemisphere that means there should be no buildings, trees etc. on the south side of the antenna.

In general, the GPS will receive more satellites when having good view in direction of the equator. That means, if you have only the possibility to place your GPS in-house, you should choose a place on the south side of the building (or north side on the southern hemisphere). This fact is more important, the higher your latitude is.

The GPS module not only transmits position and time, but also several status information about its reception. You can see some of the values on the LCD and on the web-interface. Please note that those values will change very often, because the GPS satellites are moving fast, weather and condition of the ionosphere can change quickly. You should see similar values every 24 hours, as the satellite constellation will be almost the same. The most important values are listed here:

- The signals of **tracked satellites** are used to calculate position and time data. The count of tracked satellites shouldn't fall below 6 too often. A value of four is the absolutely minimum! Higher values are always better.
- **Satellites in view** gives the number of satellites, which could be tracked under perfect conditions (i.e. on a airplane). If the count of tracked satellites should be close to the satellites in view. For example, when you have 15 satellites view and only 4 tracked, then you should think about a better place for the antenna.
- **PDOP** stands for *position dilution of precision*. There are also values for vertical and horizontal position (VDOP, HDOP). The DOPs give the best estimation about the "GPS-quality". The lower the values, the better the position accuracy. Values below 1.0 are almost perfect and those below 3.0 are still far good enough for our purposes. The DOPs give a idea whether the constellation of tracked satellites is good. For example, even if you receive 8 satellites your PDOP could be bigger that 7 (very worse), because the GPS sees only satellites from a part of the sky and can not calculate a good position.

2.5 General mode of operation

The following sections describes the technical background of the hard- and software. You don't have to understand all of them, but you should try to read the important parts.

2.5.1 About the firmware

The firmware is a bare metal microcontroller software. That means, no operating system like Linux or an embedded *real-time-operating-system* (RTOS) is used as a base. This saves space and gives full control over all peripherals. The firmware has been built with the *GNU Tools ARM Embedded* toolchain. There are some 3rd party libraries included, which are listed in the *About* section of the web-interface.

Several peripherals are configured to use **DMA** access: The ADCs which stores all their samples in RAM all the time (up to 150MBit/s), the DAC for playing sounds, GPS USART for receiving the position data and debug USART for sending messages. Those transfers are handled by two DMA controllers without interaction of the CPU.

Other functions are realized with interrupts (**IRQ**). The two most important interrupts are the 1PPS timer capture and the ADC watchdog. The 1PPS-IRQ has the highest priority, which means that a 1PPS impulse from the GPS, immediately will stop the current routine and enters the 1PPS-IRQ function. The ADC-IRQ has "only" the second highest priority. There are a lot of other interrupt sources, where the timing is not so important.

The firmware uses some of the hardware timers for measurements and/or interrupts. The most important one is again the **1PPS timer**, which runs at 84MHz. That means, the timing of the 1PPS signal can be measured with an accuracy up to 12ns - theoretically.

2.5.2 Flash and settings

The flash not only contains the firmware, but also user settings which can be changed through the web-interface. If no settings are present in the flash, then default ones are used.

By design, flash has several limitations. You don't have to care about them, as the firmware already does this. One limitation is, that the whole flash area is divided into sectors. Before writing on a position which already contains data, the whole sector has to be deleted and rewritten. Most sectors in the STM32F407 CPU have a size of 128kB. Thus, a simple change of a small entry in the flash needs a full rewrite which is not so easy possible due to lack of RAM. Additionally, the flash tolerates "only" 10000 erase cycles. This is far enough, even when rewriting flash several times a day.

However, the controller downloads new configuration settings, proposed by the server, several times per hour. Due to reasons mentioned above, those settings won't be stored in flash memory. In the current firmware, all flash operations must be done by the user himself.

2.5.3 Start-up Sequence

After a reset or power-on, the controller initializes itself and peripherals, like CPU-Clocks, LCD, Real-Time-Clock, I/Os, Amplifiers and USARTs. The configuration is loaded from flash too. The LEDs and LCD light up and there should be two beeps: A first start-beep and a second beep for a successful initialization of an amplifier. While this boot-sequence takes less than 3 seconds, some additionally initialization will run in the background.

The GPS-module has to lock position and 1PPS signal, which can take several minutes after a cold start. Additionally, the controller checks for valid communication with the GPS and, if needed, initializes the GPS with the defined baud-rate.

The assignment of a network IP address by your DHCP server can also take some seconds. As soon as the controller has a IP-address assigned, a HTTP-request to the Blitzortung.org-servers will be sent. The response contains information like computing server address or station id.

After all these processes have been finished successfully, the controller starts sending data.

2.5.4 Real-Time-Clock

The MCU has an integrated real-time-clock (RTC). It's only used for internal functions to be independent from GPS time. The time settings stays even when the controller has been resetted. The RTC is adjusted automatically by GPS or remote configuration. It supports UTC time only.

2.5.5 Watchdogs

If the controller hangs due to a software bug or due to external impact, an integrated independent hardware watchdog will reset the CPU. The watchdog can only be disabled by pressing the blue button during start-up. It's not possible to save that to flash.

Both microcontrollers, the STM32F4 on the controller board and the ATMega8 on the amplifier have brown out watchdogs enabled. If there's an undervoltage, a internal reset will be triggered. The brown out watchdogs can not be disabled.

With all those watchdogs enabled, the system should always be accessible.

2.5.6 Server connections

The controller uses two ways to communicate with different servers:

1. For each signal, the controller sends a **UDP packet** with the needed data to at least one computing server. The main information included in such a packet is the time-stamp of the trigger. The signal data itself is included in binary format. The protocol is flexible and data which didn't change since the last packet doesn't need to be transmitted again. That could be sampling settings or gain levels. This saves traffic and processing time on client and server side.

   However, traffic can get high when receiving a lot of signals, i.e. due to high thunderstorm activities or due to interference. We can not give a concrete value, but we've already seen stations with 300MB/day. There are some mechanism implemented to reduce traffic and we are already thinking about further improvements here.

2. A **HTTP connection** is used for bi-directional communication. On the one hand, the controller sends information about itself, like status of GPS, amplifier, CPU or signal statistics to a control server. On the other hand, it gets information from the control server. For example, the controller gets its station-id, which computing server to use and whether a new firmware is available. The server can also change gain, threshold, sampling settings and a lot of other values.

   The control connection has a also fall-back mechanism included: If a HTTP-connection to the control server can not be established, it tries to connect to other servers, until it gets valid information. The servers have no direct access to the controller, but the controller up- and downloads the information from time to time. The overall traffic could be some megabytes per day.

Both different connection methods make the system very flexible and reliable. UDP for sendning the signals reduces overhead a lot and fits perfectly for our needs. We measured a packet loss much lower than 1% even on long distant connections.

2.5.7 Detecting signals

The previous sections gave a look into the technical background. The following list summarizes that and gives a short overview what happens when a signal on a single channel gets received:

1. Electromagnetic fields induce a small current in the **antenna** coil.
2. The signal is **amplified** and filtered several times on the amplifier board. **Gain** settings have already been set before through the on board controller (ATMega8).
3. The signal is transmitted through the cable to the controller board. The reference voltage is $0V \pm 1.5V$.
4. On the controller board, the signal is **shifted** $1.5V$ up.
5. The shifted signal lies on the input of an **A/D-converter** integrated in the MCU. It samples the voltage with about 500,000 times per second (minimum) and stores this value in a register.
6. The DMA controller reads the register values of all three ADCs at once, and saves them one after another in **memory**. This memory block has a size of several kilobytes.
7. A **watchdog** compares the ADC-register value with a threshold. If the threshold is reached, a interrupt is generated.
8. The **interrupt** functions checks the 1PPS timer value and reads the needed ADC-samples from the memory block. This is the first time, where the CPU is needed.
9. Depending on settings, the recorded signals are checked and filtered.
10. The values of the triggered channel and the other channels as well as other information are send within a **UDP** packet to the computing server.

3 Differences to the old hardware

Below is a list of comparisons between technical data of the old **GREEN** and new **RED** systems. The old system produced (and still produces) good results, but the hardware was maxed out. **RED** is much more flexible, precise and has several new features, while staying in a similar price range.

Amplifier performance

GREEN The amplifier had a huge delay. The antennas didn't have damping resistors and thus, also self-resonances. The high and low-pass-filters were simple and had a low cutoff frequency.

RED Reduced delay and antennas with damping resistors. This minimizes self-resonances of the antennas. The filters are better and have a higher cutoff frequency, so we get more information from the signal. All those things are one of the most important improvements on System**RED**!

Gains and Thresholds

GREEN Gains have to be set by jumpers or potentiometer on the amplifier. Threshold for signal trigger were fixed on the controller board.

RED Gains and thresholds can be changed individually for each channel by software. This gives the possibility of automatic gain/threshold configuration by the server to maximize the detection rate of the whole network. There's still a potentiometer on the amplifier board, mainly for testing purposes.

MCU (Microcontroller Unit)

GREEN ATMega644 with 20MHz, 64kB flash, 4kB RAM, no DMA, 2 USARTs. One of the main limitations were the missing DMA capabilities.

RED STM32F4 ARM Cortex M4F technology with 168MHz, 1MB flash, 192kB RAM, several DMA channels, 6 USARTs. This MCU fits perfect for our needs. There are still resources left, for example the integrated DSP. We could use it to examine a signal closer before sending it. The unique processor-ID gives us the possibility no identify each hardware.

A/D Converters (ADC)

GREEN Two independent external 8-bit converters with around 500ksps, which have to be polled by the CPU. There was no other configuration possible.

RED Three independent 12-bit converters integrated in the MCU with up to 2200kspseach, which write their data into memory by themselves (DMA). So we can now see the signal before the trigger, which is important for a better position computing. The A/D-converters can be configured by the CPU in different ways, i.e. sampling time, resolution, threshold. Each A/D-channel has a multiplexer, so one converter can be used for different amplifier channels. All three converters can be joined to sample one input with almost 12000ksps. So the 1PPS signal can be checked for accuracy or the voltages for bad power supplies. The A/Ds are also acting as a hardware monitor by checking voltages and temperature.

Connection

GREEN Needs serial or USB cable connected to a running computer or embedded device, where the tracker has to be installed.

RED Uses direct Ethernet connection, so no other device/computer is needed. Thus, the system runs almost "out-of-the-box". No computer is needed for the installation as all needed information is displayed on the LCD.

Display, Buzzer, Buttons

GREEN Only buttons for a test-signal and a reset-button. Real status could only be checked from connected PC.

RED Buttons, Buzzers and LCD for checking status, easy installation, signal throughput and several other useful things.

Firmware

GREEN Only basic functionality. A firmware update needs special hardware.

RED Uses a much more complex firmware, but with a lot of possibilities. Update is easily possible via web-interface and with two different programmers on-board. New features or bugfixes are easily possible.

Return channel

GREEN Not available. Some trackers (like XLT) had a simple implementation. But there was no possibility to talk to the controller itself.

RED The return channel through HTTP-protocol is a main feature of the new system. It allows automatic remote configuration of the controller and amplifier by the servers. It has several fallback functions, if there are problems with our servers.

Power

GREEN 12V power supply. The system itself had a very low power consumption around 1.5W, but a running PC was needed. If a special router was used instead, the overall power consumption was around 2.5W.

RED 5V power supply through USB jack. This is a common standard today. The voltages are checked by the MCU and can be shown on LCD/web-interface. The power consumption is 2-3W, mainly depending on LCD color and brightness.

4 Support and Help

The System**RED** was designed to be a "user-friendly" system. We've implemented several functions to support you on investigating problems. For examples, the CPU checks the voltages, informs with beeps and text messages on the LCD about errors, writes a lot of debug information and shows them on the web-interface.

In the following sections, we gathered some common problems and frequently asked questions. Before you ask us anything specific you first should inspect the following lists. This will save a lot of time for both of us.

If you did not found your answer there, check our internal forum or other forums related to Blitzortung.org. The chance is high that you get your answer more quickly in a forum than by direct contact via email. You will also help others with similar problems, if you post in a forum. When sending your request by mail, don't forget to attach a screenshot, picture or log output related to your problem.

4.1 Common Problems

Nothing is working, only the two red LEDs near the mini-USB jack of the STM32F4DISCOVERY board are glowing!

This means, the CPU is not working. If this happened after uploading the firmware, then the flash procedure was not successful. Please double check the settings in the flash tool. Is the jumper close to the CPU putted on the pins? If not, then the CPU has no power.

The display has dark bands across it when the board is powered

You are probably using the mini USB jack at the STM32F4DISCOVERY board for the power supply. Please use the mini USB jack of the controller board for power supply. It is not possible to use the mini USB jack of the STM32F4DISCOVERY board to power the controller with 5 Volt. This is a frequently made error.

I don't see any readable information on the LCD!

First check, whether the board is powered via the mini USB jack of the controller, and not via the mini USB jack of the STM32F4DISCOVERY board. Then check, if you accidentally set the LCD contrast to 0 in the settings. Other possible reasons for a not-working display:

- Check the supply voltage. If it's over 5.2V, then the LCD might not work. You should stay below 5.1V. This is a hardware bug in PCB 10.3 and will be fixed in upcoming versions.

- Measure the contrast voltage with a volt meter. It has to be around -4V.
- Carefully check every pin/solder-joint at the display and the controller.

The LCD backlight is not working!

Check, if you accidentally switched the backlight off in the settings. The backlight does not work, when the controller has no firmware.

My system receives a lot of interference.

There could be a lot of different reasons for interferences.

- Search a different place for your antennas.
- Lower your gain. The default values are already very high. Check whether you still receive enough stroke signals by checking the statistics on our web-sites.
- The LCD backlight can send out noise due to PWM dimming. Set it to 100

Under specific conditions, a simple filter could be activated.

LCD reports: "Fatal ERROR. Could not initialize network."

This means, the controller couldn't communicate with the ENC28J60 network chip. Please check if everything soldered correctly and whether the chip is correctly placed in its socket. The crystal must have 25 MHz fundamental oscillation. If it's a overtone type, the ethernet controller might not work!

The controller did a long deep beep and showed a hard fault on LCD. What should I do?

If this had happened only once, then you shouldn't care. On very rare occasions, the on-board programmer can do strange things and interfere with the CPU. Remove the USB cable for programming and also the power supply. After 10 seconds, power up the board again but don't connect the USB on the STM32F4DISCOVERY. You can also remove the two jumpers labeled with ST-Link to disconnect the programmer.

If that didn't help and you've got those faults from time to time, you should contact the developers. If possible, write down the hard fault codes.

PA6H GPS does not work or has very bad reception.

In most cases, this is due to wrong soldering. Carefully check all solder joints. Please be careful when re-solder the joints after the PCB has been fully assembled. It's much more difficult to get access to the GPS solder joints due to some big parts around the GPS module.

If the GPS antenna status shows always *intern* (LCD: *Int*) or a *shortcut* (*ERR*) then you should double check the solder joints from the GPS module to the antenna connector. If you don't find the problem, the you should send us at least two good photos (or post them in the forum) where the solder joints on both sides of the module can be clearly seen (see Fig. 18).

In some very rare occasions, interference can disturb the GPS module. Try a different place and always make some tests outside. Please check 2.4 too!

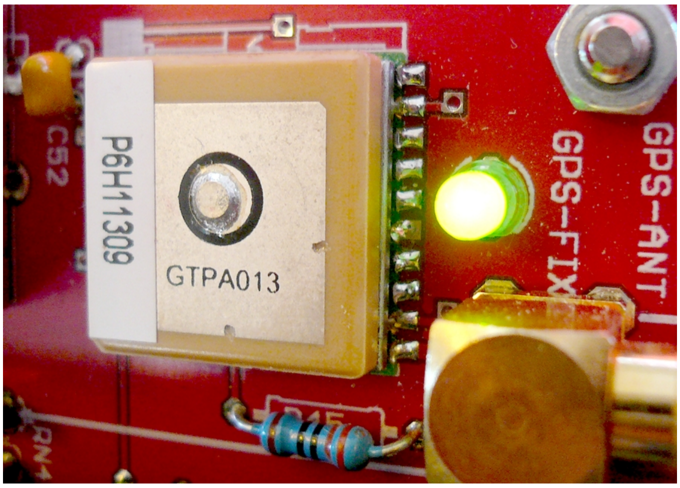


Figure 18: Close photograph of solder joints on one side.

The GPS looses 1PPS when there are a lot of satellites tracked (PA6H only).

This is a bug in the PA6H module which can be observed on very rare occasions. It happens only when the reception quality is good and a lot of satellites are tracked (> 14). This situation can take minutes and the controller shows GPS warnings. If you experience it, check it again on following days at the same time but some minutes earlier on each day. Contact us, if you can observe the behavior on two different days. There's currently no real solution for that. Even a firmware update of the GPS module doesn't help. You can position your antenna so that it does not receive so much satellites. Changing baud rate or SBAS and resetting the GPS can eliminate the symptoms for some minutes but won't solve the problem.

4.2 Frequently asked questions

Where can I get a lightning detector for participating Blitzortung.org and what is the price?

First of all, you have to assemble your lightning detector by yourself. We do not offer complete assembled station. We only offer certain parts or kits. Sometimes participants of Blitzortung.org assemble stations for friend and others. We can not give here any names or addresses of these friendly people. The best place to find them is a forum. You will find some links to lightning location forums on our webpages.

The total price for the necessary parts is less than 200 Euro. The prices will differ from country to country. Note that this is only the price for the real necessary material. It does not include a housing for the assembled boards or the material to mount the antennas under the roof or outside. However, it is not strictly necessary to use a housing or a water proof out-door mounting. Many participants have mounted the components on a plastic board and use ferrite rod antennas in-doors. What you additional need in any case is a wired internet connection, two days of time for assembling and installing such a system, and some experiences in soldering electronic components. Building a lightning detector and its operation is not *plug and play*.

What is the detecting range of a station?

This question is similar to the question "How far can I see?" If it is foggy, you can not see more than 50 meters, but on a starry night you can see stars having a distance of lightyears. If a lightning discharge is very strong and the propagation conditions are well, the signal can be received over several thousands of kilometers even with small antennas. This is the case especially at night over the ocean. If a lightning discharge is very weak, it is difficult to receive it even with large antennas in a distance of only 50 km. When using 20 cm ferrite rod antennas you will receive signals from thunder storms up to 1000 km at day, and over 2000 km at night. If you use larger antennas in more quiet environments, the signal to noise ration will increase, and you can receive signals up to 3000 km. This also depends very much on the location of the antennas. The range will increase dramatically if your antenna is placed in an electromagnetically quiet area outside a city on a hill and it will decrease if it is placed in an urban area inside a building.

Some participants are more interested to receive close lightning discharges instead of distant lightning discharges. They waive the long range fine tuning and concentrate to trim the system for close lightning discharges. The challenge to receive close lightning discharges such that they can be used for the computations is to separate pre- and post-discharges from the main stokes.

What is the detection accuracy of the system?

The answer to that questions is not so easy, as we would have to compare the positions from a lot of observed strikes with those we measured. You are welcome to report strike observations to us, but you must write down the exact time and position of the strike! You can also use a video-camera for this purpose.

There is another method to estimate the accuracy: Some tall structures, like antenna masts, radio towers or wind mills are struck much more often by lightning than other areas. When displaying all detected strokes of a long time period (i.e. one year) on a map, a cloud of strikes can be seen around some of those tall structures. This cloud has a diameter of roundabout $one\ kilometer$ for countries with a high station density. That's all we can say. We currently don't have any authoritative scientific values for that. Please note, that this method doesn't say anything about the accuracy of a single stroke. We also can not tell you anything about the strokes, which are misplaced by several kilometers.

The accuracy varies much in different regions of our network. Higher station density will increase the accuracy and yield for that area. That's why additional stations are needed even in those areas where you already can see lightning data. See also Question 6.

Does the Blitzortung.org network register every stroke?

No, probably not. It depends mainly on station coverage of the specific region. In regions with low station coverage, almost only the strong lightning discharges will be detected, as their signals can be received easily in a distance of several thousand kilometers (see Question 2). These are mainly CG ones (cloud-to-ground). The power of IC lightning (intra-cloud) is lower and the alignment of our antennas is optimized for CG.

Can you differentiate between CG or IC lightning and show polarity and current?

Currently not, but it is planned. The new **RED** stations already provide all needed data to determine the values. Please note, that several estimations have to be made, so these values can be inaccurate. But it's the same for all other lightning detection systems. Even if some commercial networks say, that they can differentiate between CG or IC, there's always a big uncertainty (which they won't tell you).

What are the best distances between the detectors?

Lightning detectors are best placed in a distance between 50km to 300km around the lightning discharges. However, even if there is another station nearby, it is still interesting for the project to place a station beside the existing one. The reason for that is very simple. Only 50 % of all stations will work continuously and only 50 % of the running stations are perfectly adjusted. The TOA lightning location network is based on "Give and Take". Your station best contributes to the computations if it is placed in a distance between 50 km and 300 km from the discharge. If the storm is nearby, your station usually overrides. Even a very low pre-discharge will cause a very high signal strength because of its low distance.

Why do we need a GPS?

The main reason why we use a GPS module is the time measuring. The TOA (time of arrival) lightning location method is based on a high precision time measuring of the lightning discharge with an accuracy of less than $\pm 1\ \mu s$. A time difference of $\pm 1\ \mu s$ corresponds to an distance difference of $\pm$ 300 m. Such a high precision time measurement is only possible with a GPS module that outputs a 1PPS (1 pulse per second) signal. Other techniques like using the DCF77 signal are not adequate for such a high precision time measurement. The additional output of the exact geographic position of the site is a pleasant side effect of GPS devices.

Where should I place the GPS antenna?

GPS antennas should always be placed outside or at least at a window, because frequencies between 1 and 2 GHz can not pass dense matter very well. Some (newer) windows are metalized, which means that GPS signals can not pass them to. You can verify this also with your smartphone with disabled(!) WiFi or test with a car navigation system.

Can I get the source code for the Firmware and the layout data of the PCBs?

In recent years, our lightning detection system growth and become more popular and more successful. Unfortunately, this also increased the number of people who just want to copy our system instead of joining our community. To protect our work in that we have invested several thousands of hours of our free time, we provide the sources of the Firmware and the layout data of the PCBs only for selected familiar members of the project which we already know for a long time. There are several connected projects like *MyBlitzortung* or Smartphone *Apps* which are open source. Every help here would be appreciated.

How should I align the antennas?

The ferrite antennas do not require alignment. However, the two antennas should be placed horizontally (parallel to earth surface) and perpendicular to each other. Such a setup covers all directions equally. The direction of the signals is not determined, only the exact time of the signal arrival is recorded and used during data processing.

Is it possible to disable this big red blinking LED on the Discovery board?

Unfortunately not. It is attached to the on-board programmer and can not be accessed by the CPU.

Can I get an older PCB? I've already bought the parts some time ago.

Sorry, no. It is recommended to wait until you have the PCBs from us and then order the needed parts.

Can I make own PCBs, i.e. with SMD parts?

You should not. Please note, that we've designed our current PCB layout the way it is for a good reason. There are always improvements possible, but not all make sense when regarding the whole process from testing, manufacturing, assembling to support. We release new PCB versions from time to time to improve the system and to add new features. If you have own suggestions, then you are welcome to tell us about them, so we could integrate them in the next release. SMD versions of the PCBs are planned.

Is it possible to build such a hardware with RaspberryPi, Arduino or similar?

Theoretically yes. But you always would have to connect separate (expensive) A/D-converters, because most of such CPUs have only bad or slow ones integrated. Our CPU has them implemented on the same die. Other hardware can have different behavior regarding timing. An 100% equal time-stamping of all stations is very important for a good accuracy. For the same reason, we use a bare metal firmware without any OS. This gives full control over the hardware and its timing.

5 Appendix

5.1 Acknowledgement

We would like to thank all the people who helped building up this Project!

5.2 Translations

We have no time to provide translations to other languages of this document. You are welcome to translate it to another language on your own, but please contact us first.

A German version of this document can be downloaded on Blitzortung.org. It is maintained by motivated participants from wetter-board.de. It is not a 1:1 translation, can have some additional information, but might not be on the same revision level as this document.

5.3 Changes

2013-09-17	**Added:** 2.3 Interference Mode 2.1.2 DFU Method 2.1.3 Flash through the web-interface 2.2 User interfaces 2.2.1 Web-Interface 2.2.6 Analogue user input 2.2.7 Debug Log 2.3 Interference Mode 2.4 GPS 2.5 General mode of operation 2.5.1 About the firmware 2.5.2 Flash and settings 2.5.3 Start-up Sequence 2.5.4 Real-Time-Clock 2.5.5 Watchdogs 2.5.6 Server connections 2.5.7 Detecting signals 3 Differences to the old hardware 4 Support and Help 5 Appendix
	Changed: 2.2.2 LEDs 2.2.3 Buttons 2.2.5 Buzzer

Table 7: Change log of this documentation