Diplomarbeit

Zusammenbau und vorläufige Evaluierung eines low-cost Blitzortungssensors

Assembling and preliminary evaluation of a low-cost lightning detection sensor

Hans-Gottfried Ritzer

January 5, 2015

Examiner
Univ. Prof. Dr.-Ing. Wolfgang Gawlik

Supervisor
Dr.techn. Dipl.-Ing. Gerhard Diendorfer
Dipl.-Ing. Hannes Pichler

Abstract -

Lightning discharges are daily phenomena, yet, certainly incredibly powerful and fascinating. Fascinated by this powerful phenomenon, many scientists have engaged in understanding lightning discharges, in lightning protection and in lightning detection, in the course of time. In particular, the detection of lightning discharges, knowing the quantity, knowing the impact time and knowing the source location, seems to arouse the interest of many scientists and engineers again and again. Beside professional and commercial lightning detection systems, there are also private and non-commercial lightning detection systems. For all types of lightning locating systems there is always the question about their location accuracy and their detection efficiency in terms of the percentage of the detected lightning relative to the lightning that occurred in nature.

In this thesis a low-cost lightning detection sensor, developed and operated by Blitzortung.org, was assembled, tested and analysed regarding the sensor performance. Blitzortung.org is a private internet community operating a low-cost and non-commercial lightning detection network.

The sensor cannot be purchased but needs to be assembled by the members of the community. The main and specific components are offered as a starter kit to the members by the community. In a first step we have organized all necessary components, assembled the sensor and performed initial technical tests of functionality. Obviously, assembling the sensor requires relatively advanced technical and electronic skills, much more than you can expect from a typical layman.

After completion of the sensor hardware we have compared and analysed in a second step the sensor reported lightning data with the data of the Austrian lightning detection and information system ALDIS. Both systems are GPS time synchronized and we could search for time matched events based on the GPS timestamp. We have developed tools necessary for this task, such as a tool to readout and convert the data stream sent by the sensor to the central server in hexadecimal format.

The detection range of the sensor was up to a distance of 900 km, but only for about 20 % of the strokes located by ALDIS we could find a time correlated message delivered by the sensor. However, the sensor performance depends on the gain and threshold settings and the thunderstorm activity (distance of the storm to the sensor and flash rate). Within a distance range of 50 to 250 km from the sensor site the sensor reported with best performance and stroke-reports for about 60 % of the strokes located by ALDIS were received. Most of the reported strokes were low-current strokes with peak currents between 5 and 20 kA.

In addition to the sensor evaluation we have also made a direct comparison of the resulting lightning stroke data provided by Blitzortung.org and ALDIS for one thunderstorm

day in Austria. Although the overall lightning activity displayed on a geographical map looked more or less similar for both networks, we observed significant differences in the two data sets when comparing them stroke by stroke. Only for 11 % of the ALDIS reported strokes (CG and CC) we could find a time correlated stroke in the Blitzortung.org data set.

Zusammenfassung -

Blitze zählen zu den alltäglichsten, aber auch eindrucksvollsten und faszinierendsten Naturereignissen. Von diesem gewaltigen Naturphänomen in den Bann gezogen, beschäftigten sich im Laufe der Zeit viele Naturwissenschaftler damit, Blitze zu verstehen, sich effektiv vor den Auswirkungen von Blitzeinschlägen zu schützen und Blitze zu orten. Besonders die Blitzortung, das Wissen über die Anzahl, den Zeitpunkt und den Ort, scheint immer wieder die Begeisterung von Wissenschaftlern und Ingenieuren zu wecken. Neben professionellen und kommerziellen Blitzortungssystemen gibt es auch private und nicht-kommerziell betriebene Blitzortungssysteme. Bei allen Blitzortungssystemen stellt sich dabei immer die Frage nach deren Ortungsgenauigkeit und deren Ortungseffektivität, d.h. inwieweit tatsächlich alle Blitzentladungen oder nur ein prozentualer Anteil geortet werden.

In dieser Arbeit wurde ein kostengünstiger Blitzortungssensor, entwickelt und betrieben von Blitzortung.org, zusammengebaut und auf seine Leistungsfähigkeit getestet. Blitzortung.org ist eine via Internet vernetzte Gruppe privater Personen, die ein nicht-kommerzielles Blitzortungssystem betreibt.

Der betreffende Sensor kann nicht als fertiges Produkt erworben werden. Die als Bausatz angebotenen Teile müssen von den Teilnehmern dieses Ortungsnetzwerks selbst zusammengebaut werden. In einem ersten Schritt wurden also alle zum Bau des Sensors notwendigen Bauteile und Komponenten beschafft, der Sensor aufgebaut und schließlich auf seine grundsätzliche Funktion getestet. Es hat sich gezeigt, dass für den kompletten Aufbau des Sensors schon ein unerwartet hohes Maß an technischem und handwerklichem Geschick erforderlich ist.

Nach Vorliegen eines technisch funktionsfähigen Sensors wurden in einem zweiten Schritt die mit dem Sensor aufgezeichneten Blitzdaten mit den Blitzdaten des österreichischen Blitzortungs- und Blitzinformationssystems ALDIS verglichen und analysiert. Da beide Systeme mittels GPS zeitsynchronisiert sind, ist eine direkte und eindeutige Zuordnung der einzelnen Blitzentladungen (Strokes) grundsätzlich möglich. Die dazu notwendigen Softwarewerkzeuge (z.B. Mitlesen und Konvertieren der Sensormeldungen) wurden im Rahmen dieser Arbeit entwickelt.

Der Sensor erfasste zwar Blitze bis zu einer Entfernung von 900 km, lieferte jedoch nur Meldungen für etwa 20 % der von ALDIS georteten Blitze. Die Leistungsfähigkeit des Sensors hing natürlich von der Einstellung des Sensors und der Gewitteraktivität

(Entfernung und Intensität) ab. Der Sensor arbeitete in einer Entfernung von 50 bis 250 km zum Standort am besten. In diesem Entfernungsbereich wurden ca. 60 % der von ALDIS georteten Blitze auch vom Testsensor erfasst. Die meisten georteten Blitze waren relativ stromschwach mit einer Stromstärke zwischen 5 und 20 kA.

Neben einer direkten Gegenüberstellung der Sensordaten mit den ALDIS Ortungsdaten wurde auch ein erster Vergleich der von Blitzortung.org georteten Entladungen und den von ALDIS georteten Entladungen für einen Gewittertag in Österreich gemacht. Dabei zeigte die graphische Darstellung der gesamten Gewittersituation auf den ersten Blick kaum Unterschiede. Erst beim Vergleich der einzelnen zeitlich korrelierten Entladungen waren deutliche Unterschiede feststellbar. Nur 11 % der ALDIS Entladungen konnte eine zeitgleiche Entladung in den Daten von Blitzortung.org zugeordnet werden.

Contents

Introduction

Lightning discharges are daily phenomena, yet, certainly incredibly powerful and fascinating. Although many scientists have engaged in this topic in the course of time, this natural phenomenon is not fully understood and is still a relevant topic for research. It is not only the initiation of the discharge itself and the exact mechanisms that are of interest, but also the knowledge of the quantity of discharges and the lightning parameters.

Knowing the lightning parameters is important for lightning protection, for example for laying-up components of the protection system like surge protection devices (SPD). Besides, data on the quantity of lightning discharges are used to determine the local flash density, which in turn is the basis for risk calculations.[5] In order to obtain these data, lightning detection systems and measurement stations are required.

The professional and commercial lightning detection network, called EUCLID, is a collaboration of national lightning detection networks and detects lightning discharges all over Europe. One of these national networks is ALDIS, the Austrian Lightning Detection and Information System. ALDIS has been detecting lightning for more than 20 years. ALDIS possesses much operating experience and is considered as a well-established detection system.[3]

In Germany another and non-commercial lightning detection system, called 'Blitzortung.org', has been operating for about ten years. This system is based on volunteers and anybody can participate in this project. Blitzortung.org detects lightning all over the world and offers lightning information for free, but only for private and entertainment purposes and not for commercial use.[8]

Recently, ALDIS has noticed that the lightning data provided by Blitzortung.org differs from the lightning data provided by ALDIS. Due to these differences it is interesting for ALDIS to understand the principle and performance of the sensors used by Blitzortung.org and to analyse the lightning data messages of such a sensor and the computed strokes of Blitzortung.org.

1. Assignment of tasks

The aim of this thesis is to evaluate a low-cost lightning detection sensor of Blitzortung.org.

ALDIS already owns a fully assembled old version of the detection sensor of Blitzortung.org, which is called 'System Green'. So the first sub-goal is to become acquainted with the existing sensor and to get an overview of Blitzortung.org and their recent detection hardware.

The current detection device from Blitzortung.org is called 'System Red' and is obviously more powerful than the old one that is owned by ALDIS. Blitzortung.org only provides all parts for a detection device, but not a fully assembled sensor. Therefore, the second sub-goal is to organize the components and to assemble the new detection device. The assembling requires soldering the electronic parts and setting the boards up in a casing.

The final and main sub-goal is to record lightning discharges with the old and the new sensor and to analyse both, the recorded data of the devices and the computed stroke data from Blitzortung.org in comparison to the lightning data of ALDIS. In that process, the lightning data of the sensors and the two networks are matched based on the GPS timestamp.

2. Blitzortung.org

All information about the network of Blitzortung.org, their method of measurement and their detection hardware given in this chapter refers to the manual of Blitzortung.org [8].

2.1. Network

Blitzortung.org is a low-cost, world-wide lightning location network, which is based on a high number of receiver sites spaced close to each other. The ideal distance between two devices is between 50 km and 250 km. Figure 2.1 shows the station map of Blitzortung.org in Austria.

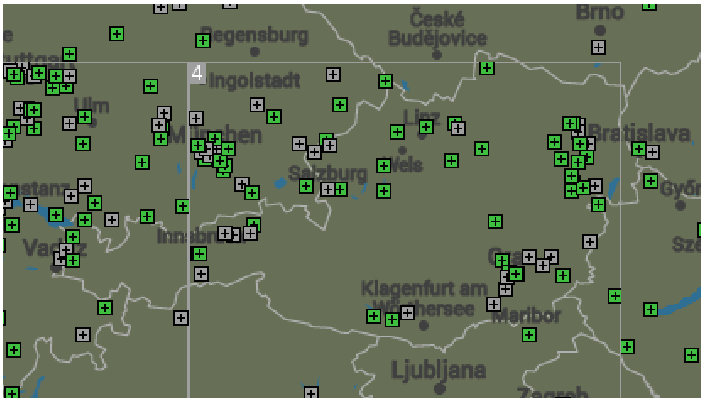


Fig. 2.1.: Station map of Blitzortung.org in Austria ©Blitzortung.org

The stations transmit their data in short time intervals over the internet to a central server, where the stroke locations are computed by the Time-of-Arrival method (TOA). Every stroke data sentence contains the precise time of arrival of the received lightning strike impulse and the geographic position of the receiver site.

All station operators are volunteers and members of the Blitzortung.org community. They have free access to the computed data in raw format, as long as their stations are sending signals. The raw data can be used by the members for all non-commercial purposes.

2.2. Time-of-Arrival method

The Time-of-Arrival method is based on the time of arrival of the emitted electromagnetic wave of a lightning discharge. The emitted electromagnetic wave is propagating at the speed of light. When the electromagnetic wave reaches a receiver site, the received signal gets a unique and precise time stamp. The difference of the time of arrival at two receiver sites and the position of these receiver sites define a hyperbolic curve. The possible impact location of the lightning discharge has to be somewhere on that hyperbolic curve. The hyperbolic curves of three such pairs of receiving stations intersect in the unique source location. Thus at least four reporting receivers are required to define an unambiguous intersection point.[1][2]

2.3. Detection Hardware

The Blitzortung.org detection device consists of three main components: the antennas, the amplifier and the controller. There are two kinds of antennas, a magnetic-field and an electric-field antenna. Since the electromagnetic signals have the form of an impulse and are emitting waves over a wide range of frequencies (3 - 30 kHz), a wide-band antenna is required. The amplifiers are also wide-band VLF-amplifiers, which amplify the antenna signals for further processing and analysing. The controller digitizes the signals and sends the curve values together with the time of arrival and the GPS position of the sensor site to the Blitzortung.org server. By transmitting the GPS position with each stroke data sentence, Blitzortung.org is able to control possible changes in location of the sensor.

There are two versions of the detection device. The old version is called 'System Green', because of the green colour of the printed circuit boards. And the new version is called 'System Red', because of the red colour of the printed circuit boards. The most important difference between the two versions is that 'System Red' can record more information for one signal as a result of the reduced delay and the higher cut-off frequency of the filters. Another important difference is that the gain and the threshold can be remote-controlled by the computing server, receiving the sensor data.

3. Assembling of the sensor

Blitzortung.org does not ship fully assembled boards, but either basic kits or complete starter kits. A basic kit only contains the most important components, like the printed circuit boards (PCB), the GPS module, the operational amplifiers and the programmed microcontrollers. The other electronic components, like the resistors, the capacitors or the diodes, and the small parts, like the sockets or the screw terminals, are not included. In contrast, a complete starter kit consists of all components that are needed to solder the respective parts of the detection hardware.

There are separate basic kits and complete starter kits for the controller, the magnetic field amplifier and the electric field amplifier. For example, a complete starter kit for the magnetic field amplifier consists of the printed circuit board, the programmed microcontroller and all electronic components. In addition to the starter kits, ferrite rod antennas, the discovery board of the microcontroller, an external GPS antenna and another LCD display with a different backlight colour can be ordered. In addition to the green display, which is included in the complete starter kit for the controller board, an amber or blue display can be ordered. Figure 3.1 shows the components of the complete starter kit for the magnetic field amplifier and the controller board. Ordering a complete starter kit saves time and is a little bit cheaper than ordering a basic kit and the further needed electronic components individually.

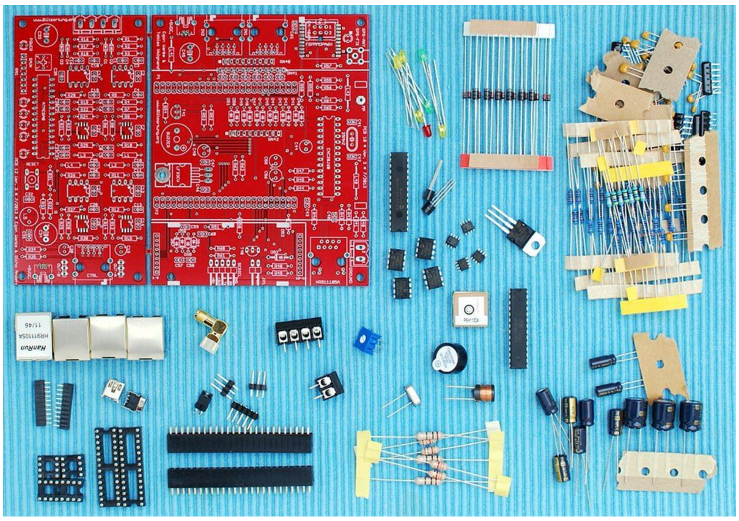


Fig. 3.1.: Complete starter kit for the magnetic field amplifier and the controller [8]

All technical information of the detection hardware given in this chapter refers again to the manual of Blitzortung.org [8].

3.1. Soldering

The first step of assembling the device is to solder the electronic components in the boards. For soldering some experience and good equipment is needed. It is necessary to use a soldering station, where the temperature is adjustable. The printed circuit boards are multilayer boards. In order to transfer enough heat to the solder joint, a soldering iron with a chisel tip should be used. Furthermore, a desoldering braid, a pair of curved forceps and a diagonal cutter for electronics are needed. Besides, a soldering mounting help ('Third Hand') is useful.

3.1.1. Magnetic field amplifier

Figure 3.2 shows the amplifier board with the soldered SMD operational amplifiers. Soldering the operational amplifiers, a lot of care is needed in order to avoid solder bridges and not to overheat and thus not to destroy them.

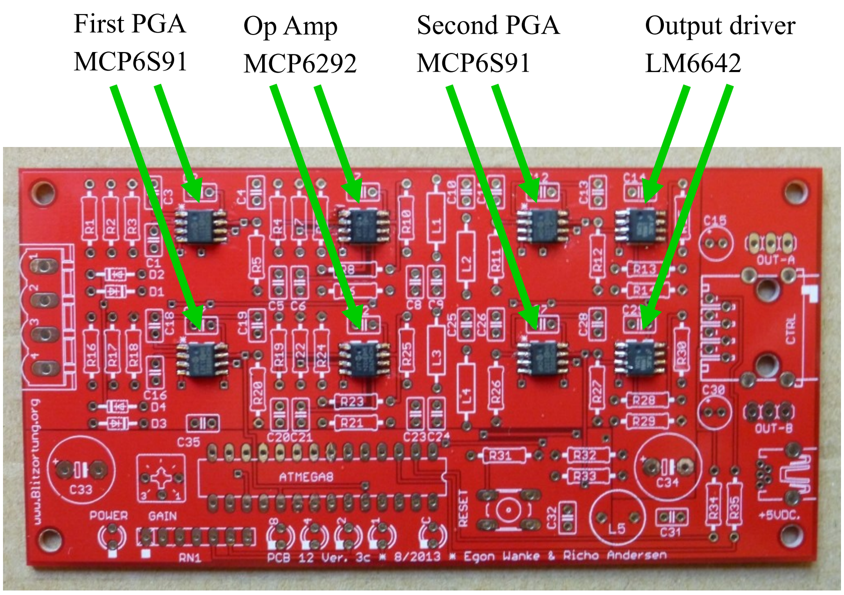


Fig. 3.2.: H-field amplifier board with operational amplifiers only

The upper four operational amplifiers in figure 3.2 realize the amplifier stage for channel one and the lower four ones realize the amplifier stage for channel two. Each amplifier stage is realized by two programmable gain amplifiers (MCP6S91), a Bessel high-pass filter with a cut-off frequency of 1 kHz, a Bessel low-pass filter with a cut-off frequency of 50 kHz and an output driver amplifier (LM6642). The Bessel low-pass filter is realized by a passive filter chain of order five, which is not yet soldered in figure 3.2.

Figure 3.3 shows the about half assembled board. The yellow LEDs that are marked with a green ellipse indicate the gain adjustment. There are three options to adjust the gain: (1) manually by a potentiometer, (2) manually over the web-interface of the controller or (3) automatically by the server of Blitzortung.org. If the green LED on the right-hand side of the yellow LEDs is on, the gain is adjusted by the controller board. The potentiometer shown in figure 3.4 is disabled.

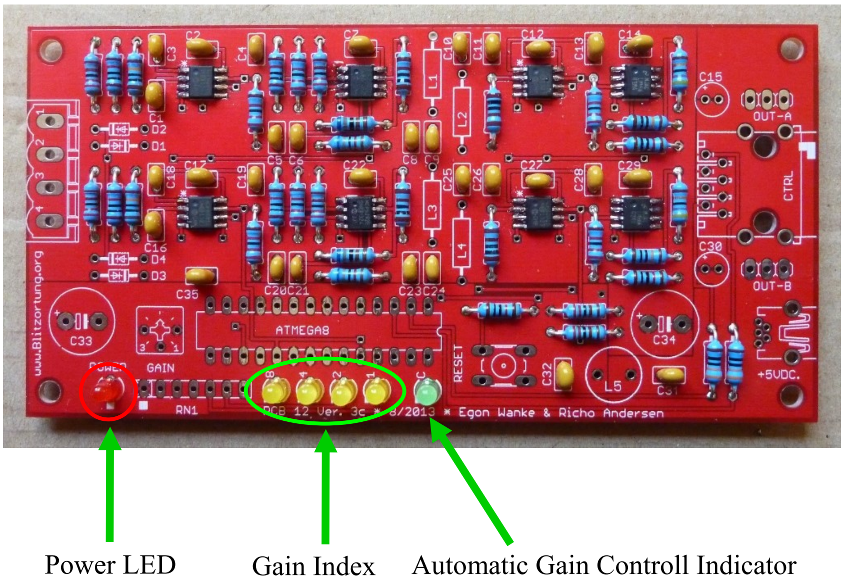


Fig. 3.3.: Half assembled H-field amplifier board

The exact gain index that is given by the yellow LEDs is shown in table 3.1. The first and the second programmable gain amplifier MCP6S91 of each channel (see figure 3.2) can be adjusted from 1 to 32. The gain factor is the product of the first and the second MCP6S91 and has to be multiplied by a factor of 40 in order to get the final gain factor. This factor of 40 comes from the operational amplifier MCP6292 and the output driver amplifier LM6642. So the lowest gain is 40 and the highest gain is 40960.

The gain adjustment determines the sensitivity of the sensor. The signals that are received from the antennas are filtered and amplified with the adjusted gain factor on the amplifier board. If a signal is higher than a pre-set trigger threshold, it will be sent to the server. The higher the gain the more signals exceed the threshold and are detected. Thus a high gain increases the sensitivity and a low gain decreases it. On the other hand local noise at the sensor site restricts to use a too high gain.

Tab. 3.1.: Gain index of the amplifier [8]

LED 8	LED 4	LED 2	LED 1	first MCP6S91	second MCP6S91	gain 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

The fully assembled board is shown in figure 3.4. The potentiometer for the manual gain adjustment is marked with a green circle. The amplifier board is connected to the controller board with a network cable via the RJ45 jack that is marked with a yellow dot.

The ferrite rod antennas (see figure 3.15) are connected to the amplifier board by the screw terminal, marked with a white oval. The two wires of the used antennas, which have a knot, have to be connected to the positive inputs of the screw terminal (marked with a white dot). For that purpose the very thin wires of the ferrite rod antennas should be soldered to bigger copper wires in order to have a good mechanical connection.

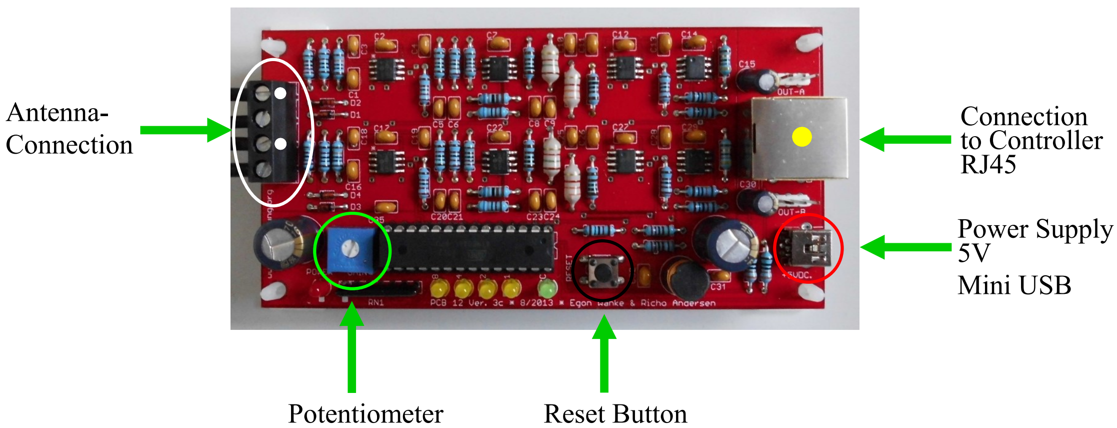


Fig. 3.4.: Fully assembled H-field amplifier board

The black button (marked with a black circle) is for resetting the amplifier board. The amplifier board can either be powered with 5 V from the Mini USB jack, which is marked with a red circle, or through the connection to the controller board.

3.1.2. Electric field amplifier

The conventional amplifier system of Blitzortung.org is the magnetic field amplifier system. However, since May 2014 an electric field amplifier system is also available. The advantages and the disadvantages of an electric field antenna according to Blitzortung.org [8] are listed below.

+ Only one antenna is needed
+ The received signals are very clean and free of any resonance frequencies
+ The polarity of the lightning discharge can be determined, if the lightning discharge was not too far away
+ Easy construction for a very low price
− High sensitivity to field electric noises
− Positioning outside or at least on an attic is needed

The magnetic field and the electric field amplifier system can be used simultaneously, if the environmental conditions allow. Due to the fact that the electric field amplifier system is highly sensitive to interference, it has to be placed outside or at least in the attic. In contrast the magnetic field amplifier system can also be placed inside a building.

The electric field amplifier system is composed of an amplifier and a pre-amplifier. The amplifier consists of four programmable gain amplifiers (MCP6S91), a high-pass filter, three different low-pass filters and three output driver amplifiers (LM6642). Figure 3.5 shows the amplifier board with the soldered SMD operational amplifiers. The high-pass filter is realized by the operational amplifier MCP6292. After the high-pass filter, three different low-pass filters are realized: a 50 kHz Bessel filter, a sharper 44 kHz elliptic filter and an 18 kHz elliptic filter. The low-pass filter filters out disturbances due to various radio transmitters. Because of the fact, that “there is no ideal filter that suppresses all types of disturbances equally well“[8] three different types of low-pass filters are realized.

“The filtered signals can be used for different purposes. The signal after the 50KHz [sic] Bessel filter is optimal for computing characteristic time stamps. The signals after the 44KHz [sic] and 18KHz [sic] elliptic filters can be used to trigger in disturbed regions and for a more simple computaton [sic] of the polarity.“[8]

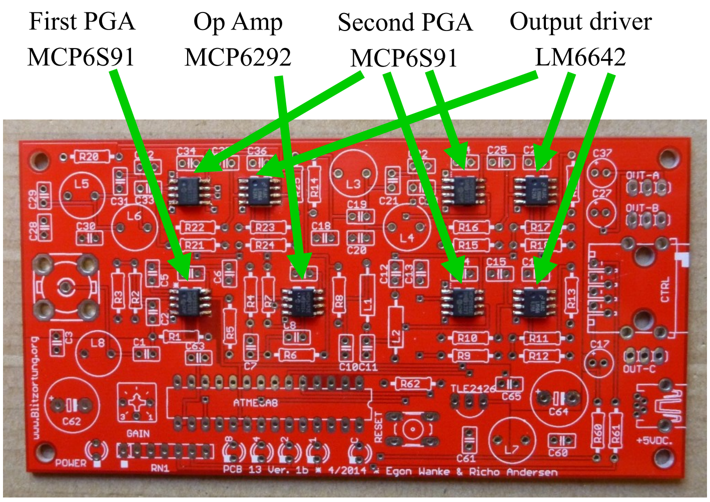


Fig. 3.5.: E-field amplifier board with operational amplifiers only

Figure 3.6 shows the fully assembled electric field amplifier board. The gain index of the electric field amplifier is the same as the gain index of the magnetic field amplifier that is shown in table 3.1. The connection to the controller board and the power supply is also analog to the magnetic field amplifier.

The pre-amplifier board is connected to the F-connector of the amplifier board, which is marked with a green ellipse, with a 75 Ohm coaxial cable. The length of the coaxial cable can be up to 300 m.

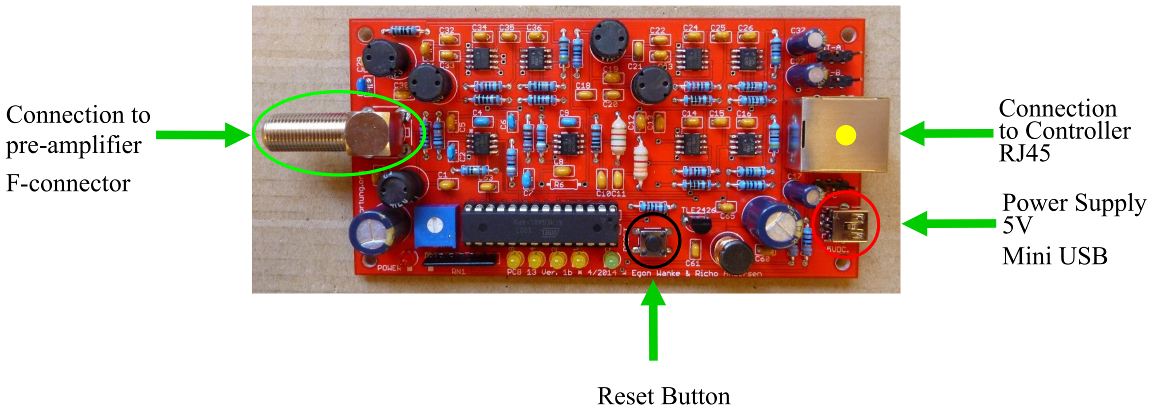


Fig. 3.6.: Fully assembled E-field amplifier board

The pre-amplifier is a high-pass filter. The cut-off frequency of the combined high-pass filter of pre-amplifier and amplifier is 5 kHz and filters out interference due to power supply with frequencies of around 50 Hz and their harmonics.

The fully assembled pre-amplifier board is shown in figure 3.7. The antenna for the electric field can be a vertically orientated wire in the simplest case. It is connected to the positive input of the screw terminal that is marked with a white dot.

Fig. 3.7.: Fully assembled E-field pre-amplifier board

3.1.3. Controller board

Figure 3.8 shows the partly assembled controller board with the GPS module and the serial interface. The GPS module is an on-board module, which allows connecting an additional external antenna. It provides a TTL level interface and a one pulse per second (PPS) signal. This signal is needed for the high precision time measurement that is required for the TOA technique. Soldering the GPS module care is needed to avoid soldering bridges, because the solder pads are very small. And after soldering the other components it is difficult to make any changes there.

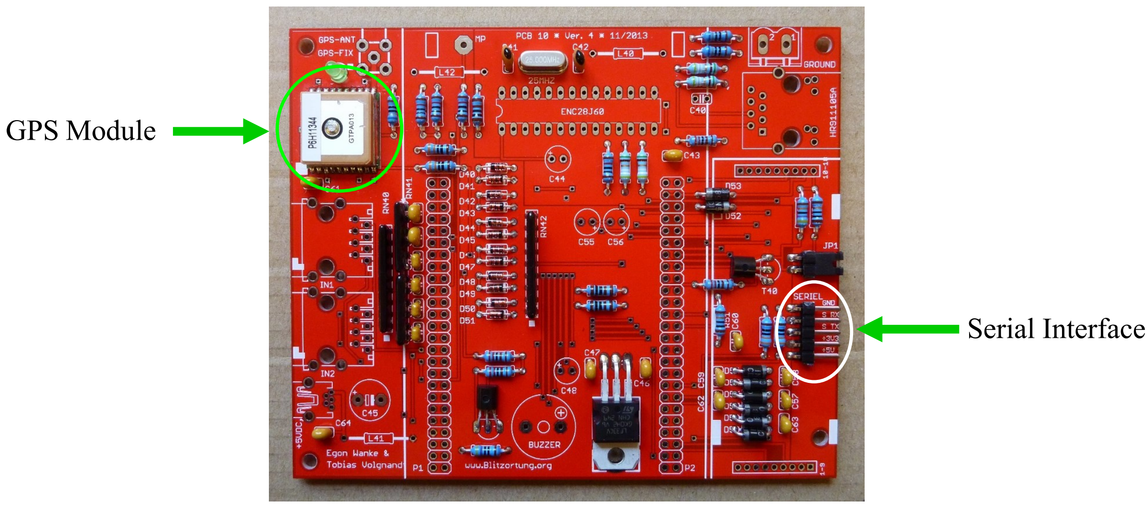


Fig. 3.8.: Controller board with GPS module and serial interface

The serial interface that is marked with a white ellipse in figure 3.8 can be used for debug purposes. The interface provides a 3.3 V TTL-level, so you have to use a TTL to USB serial converter(TTL-232R-PCB of FTDI), which is shown in figure 3.9 and which is not a part of the starter kit, and a terminal program such as HTerm. Figure 3.9 shows that a cable is soldered to the TTL to USB serial converter. At the other end of the cable a plug is mounted to connect the converter to the pin strip of the controller board.

Fig. 3.9.: TTL to USB serial converter cable

The pin assignment of the interface is printed on the controller board and the pin assignment of the converter cable is given in table 3.2. For communication the options for data transmission in the terminal program has to be set to 115200 baud rate, eight data bits, one stop bit and no parity bit.

Tab. 3.2.: Pin assignment of TTL to USB serial converter cable

pin	**wire**
Rx	brown
Tx	red
Ground	blue

The controller board without the discovery board and the LCD display is shown in figure 3.10. The buzzer that is marked with a green circle is used to indicate a received signal, the interference mode and the broken network connection. It can also be used to make the received signal audible and thus to find a good place for the antennas without any interferences.

The controller board is connected to a local area network over an Ethernet jack (marked with a white dot) and a network cable. The magnetic field amplifier is connected to the RJ45 jack that is marked with a red dot and the electric field amplifier to the RJ45 jack that is signed with a black dot. The external GPS antenna is connected to the SMA connector marked with a yellow ellipse. When the green LED next to the GPS module is off, the GPS module has found a valid signal.

The screw terminal on the board is for grounding. If any grounding for the system is required it should be done on the controller board. Grounding can be realized by connecting the ground wire to a large expanse of metal, like a radiator or by connecting it to an earth peg when recording outside in the field. But the ground wire should never be connected to the ground wire of the power supply system. Some grounding is already given by the connection of the controller board to the router via the shielded network cable.

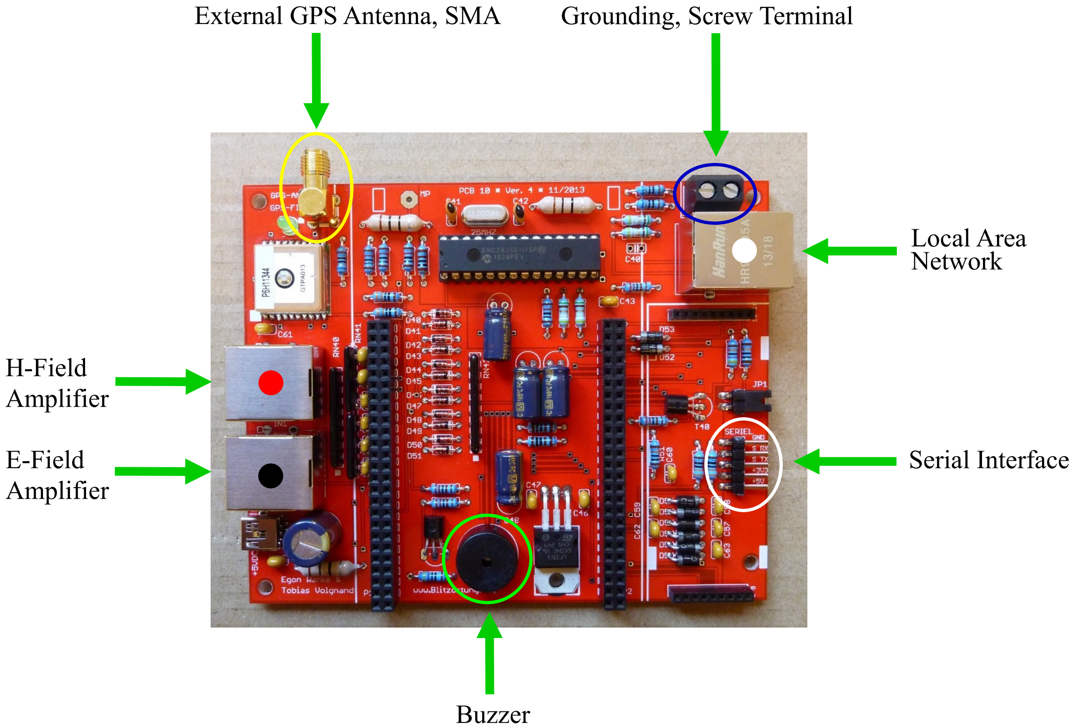


Fig. 3.10.: Controller board without the discovery board and the LCD display

The fully assembled controller board is shown in figure 3.14. The mini USB port (marked in figure 3.14 with a white circle) on the discovery board of the microcontroller is used to flash the firmware of Blitzortung.org, which is available for all members of Blitzortung.org on their web-account. When registering to the community this web-account is created by Blitzortung.org.

For flashing the firmware to the controller the tool STM32 ST-Link Utility (Version 3.4 [1]) is used. After connecting the controller via the mini USB jack to the personal computer, the STM32 ST-Link Utility has to be opened. During firmware installation the microcontroller receives the necessary power from the personal computer. In order to open the download window the button 'Target→Program' in the menu bar has to be selected (see figure 3.11). The open window (see figure 3.12) appears and the bin file of the firmware has to be selected. After opening the equivalent bin file the download window appears. Before flashing the firmware on the controller by clicking on the start button, the start address has to be set to 0x08000000 (see figure 3.13).

[1] http://www.st.com/web/en/catalog/tools/PF258168

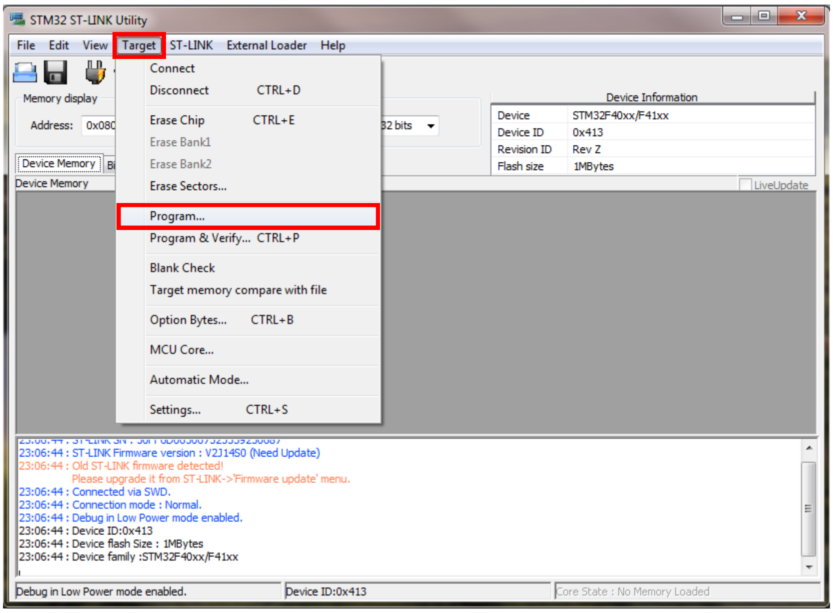


Fig. 3.11.: Menu bar of the tool STM32 ST Link Utility

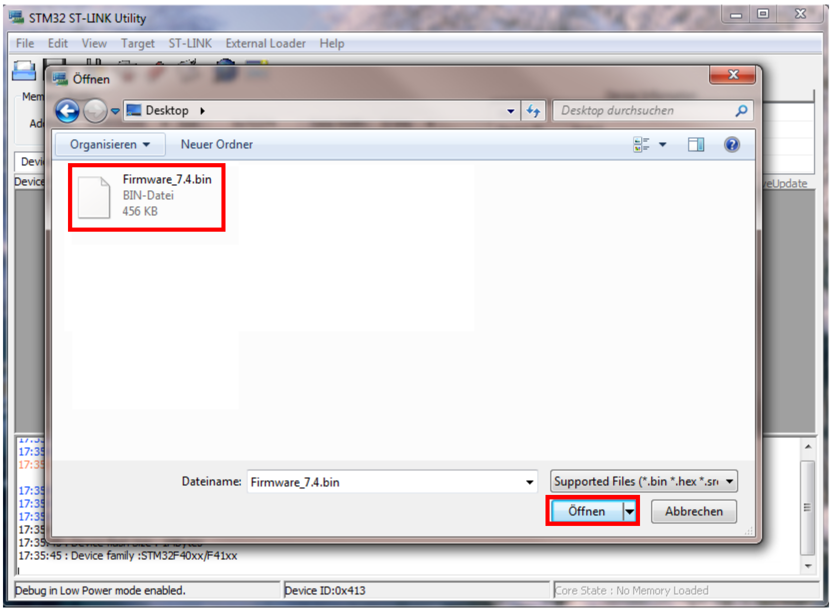


Fig. 3.12.: Open window for opening the bin file of the firmware

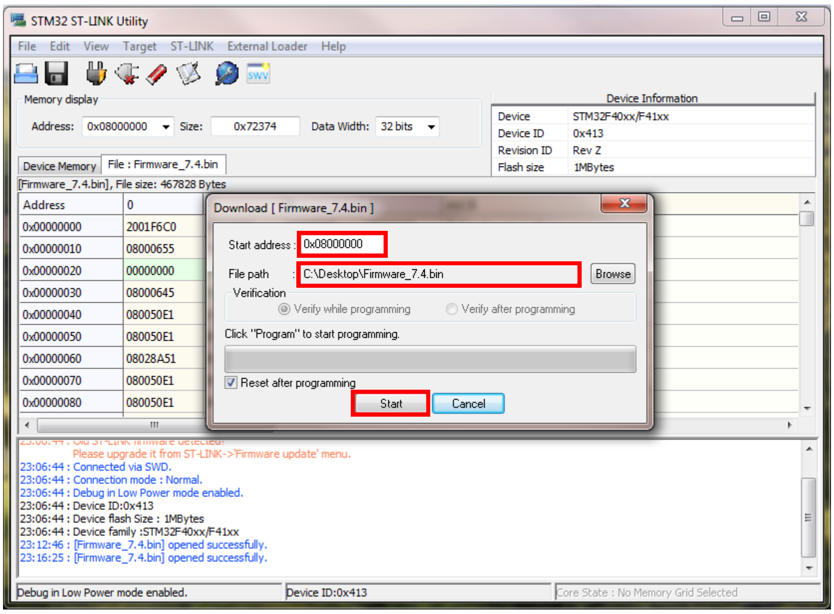


Fig. 3.13.: Download window for flashing the firmware on the controller

On the discovery board, shown in figure 3.14, there are four user defined LEDs in four different colours to show some status information. Table 3.3 gives an overview of this information.

Tab. 3.3.: Status information of 'Sensor Red' given by the four user defined LEDs

LED	**state**	**information**
green	periodic blinking	GPS signal found
green	permanent on	valid GPS signal locked
blue	periodic blinking	1PPS signal
orange	flashing	lightning signal received
red	constant blinking	interference mode
red	flashing	received signal can't be send

The LCD display is used to show different status information of the system such as the state of the GPS module or the state of the amplifiers. The blue button is used to step through this information and the black button is used to reset the system.

The power supply for the entire system is provided through the mini USB jack of the controller board that is marked with a black oval. The power supply is 5 V DC, the current consumption of the amplifier board and the controller board is about 40 mA and 400 mA, respectively.

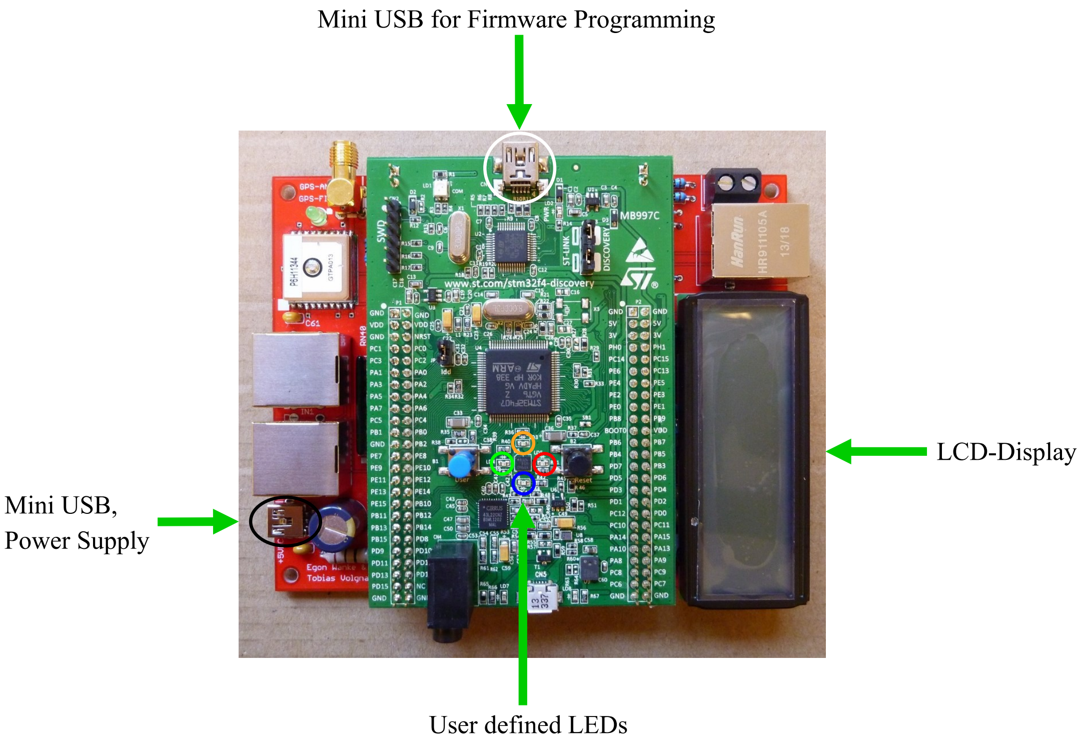


Fig. 3.14.: Fully assembled controller board

3.2. Housing

3.2.1. Magnetic field amplifier and controller

In this thesis two ferrite rod antennas are used for the magnetic field amplifier system. Figure 3.15 shows ferrite rod antennas, which are shrink sleeve fixed, in three different lengths. The used two ferrite rod antennas shipped by Blitzortung.org have a ferrite rod diameter of 10 mm, a resonance frequency of more than 190 kHz and a length of 20 cm [4]. But there is also the possibility to order individually two ferrite rod antennas with a different length. The length of the used antennas should be entered in the web-account of Blitzortung.org.

Fig. 3.15.: Ferrite rod antennas in three different lenghts [4]

The two ferrite rod antennas of equal length are mounted perpendicular to each other and together with the magnetic field amplifier board in a waterproof box with the class of protection IP 65 (See figure 3.16). For this reason the box can be also placed outside in the rain for recording. It should be mounted at least half a meter above ground. The antennas don't have to be aligned to north/south or east/west, but they have to be placed horizontally, because the magnetic field of a lightning discharge is orientated horizontally. The network cable between the amplifier box and the controller board is 15 m long.

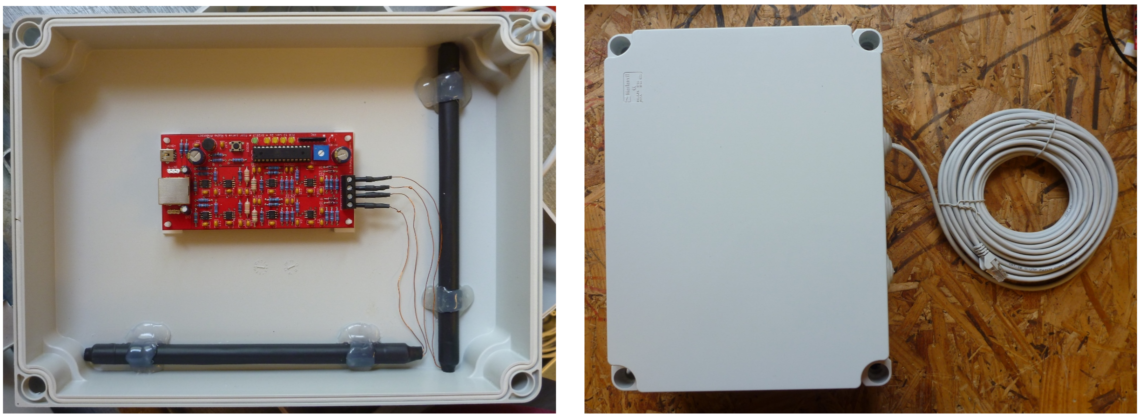


Fig. 3.16.: Magnetic field amplifier box

The magnetic field amplifier box is put together with the controller board in an aluminium case (see figure 3.17). For this purpose the controller board is mounted on a flagstone, which is attached to the case. Between the lid and the bottom of the case some cable bushings are filed for the power supply, the amplifier box, the GPS antenna and the network connection.

For recording the magnetic field amplifier box should be placed outside the aluminium case, because of its shielding effect. Figure 3.18 shows a possible recording setup for the magnetic field amplifier system. The antennas are placed in the attic and the aluminium case with the controller board is set up inside the house.

Hence, the aluminium case with the controller board and the amplifier box form a portable recording device for the magnetic field of a lightning discharge. For recording, 'only' a power supply and the access to a local area network are necessary.

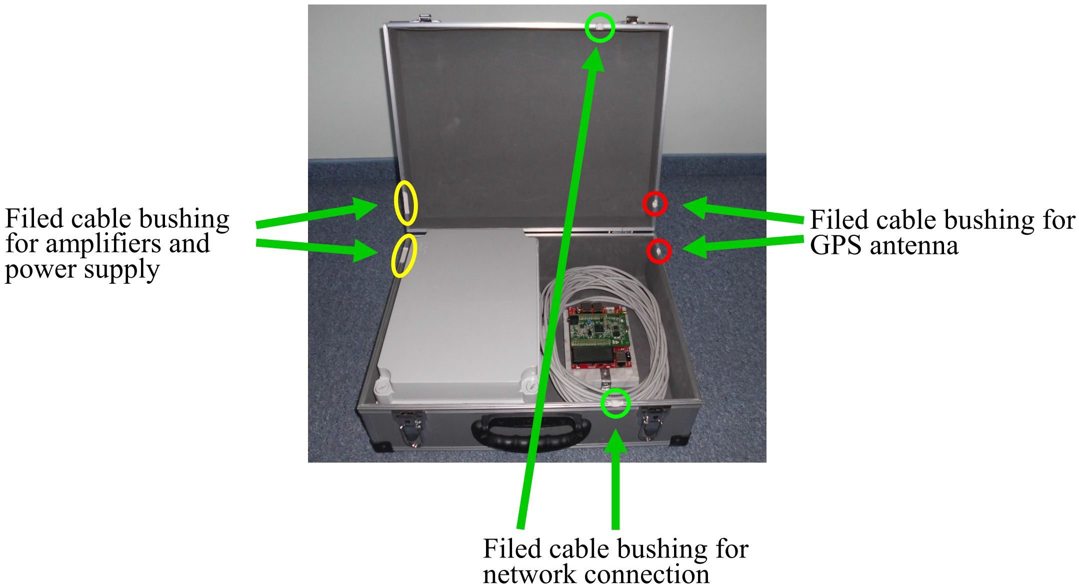


Fig. 3.17.: Aluminium case for the magnetic field amplifier box and the controller

Fig. 3.18.: Possible recording setup for the magnetic field amplifier system
left: amplifier box placed in the attic
right: aluminium case with the controller board placed inside

3.2.2. Electric field amplifier

The electric field amplifier board is also mounted in a waterproof box with the class of protection IP 65. In the box there are two cable bushings, one for the coaxial cable to the pre-amplifier and the other for the connection to the controller (see figure 3.19). The pre-amplifier and the antenna are put into a plastic tube in order to make it waterproof (see figure 3.20).

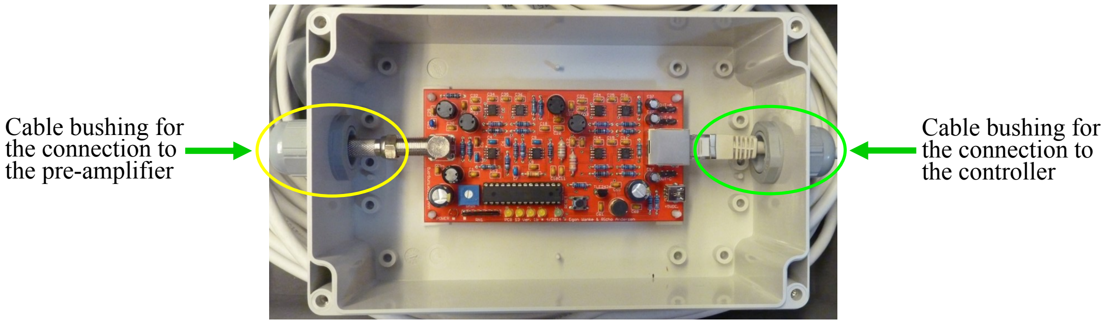


Fig. 3.19.: Housing of the electric field amplifier

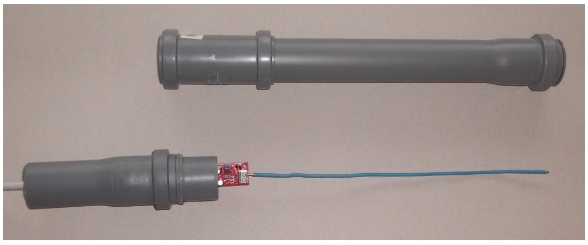


Fig. 3.20.: Housing of the electric field pre-amplifier

The aluminium case serves as transport case for the electric field amplifier system (see figure 3.21 and 3.22) as well as bracket for the antenna (see figure 3.23). For transport the plastic tube is fixed with elastic bands to the lid of the case and for recording it is fixed with a pipe clip to the side of the case. The coaxial cable is about 20 m long and the network cable is 5 m long. So the antennas can be placed for example outside in the field where the interference level is low.

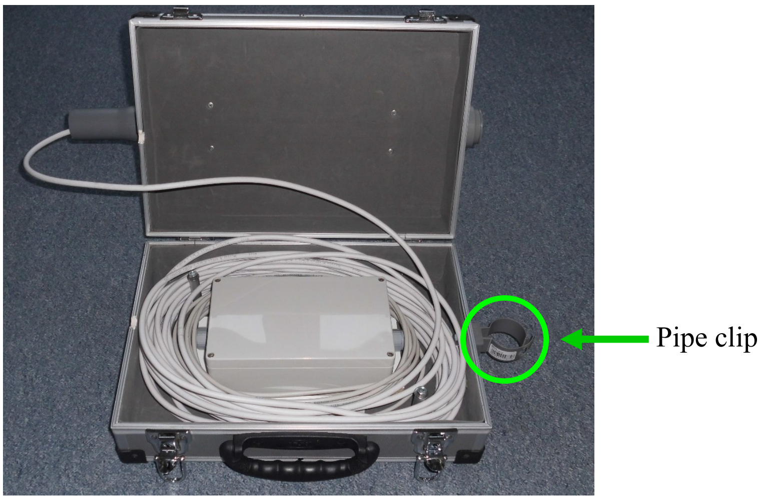


Fig. 3.21.: The electric field amplifier and the cables placed inside the aluminium case

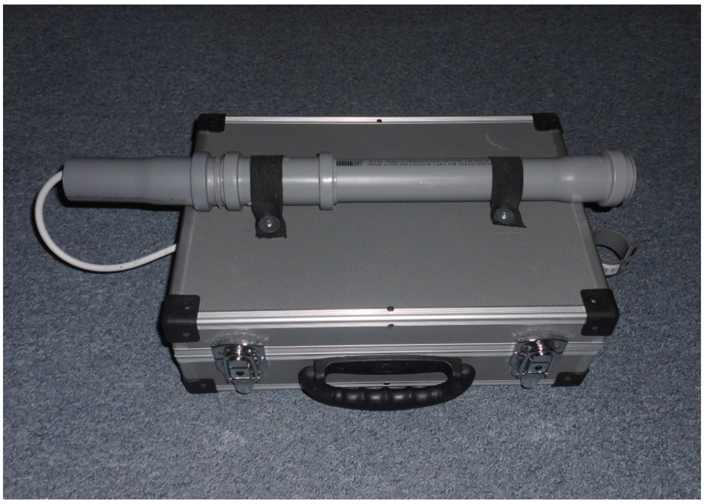


Fig. 3.22.: The antenna and the pre-amplifier fixed outside the aluminium case for transportation

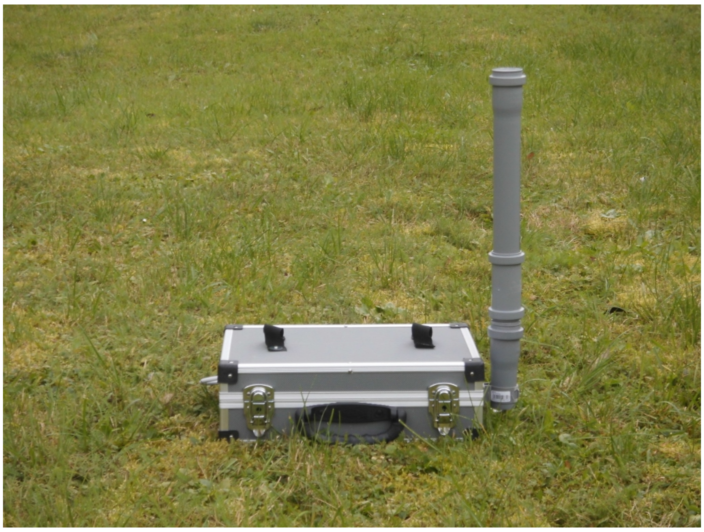


Fig. 3.23.: Aluminium case as bracket for the antenna (recording setup)

4. Recording

4.1. Recording Setup

4.1.1. Sensor Green

The old sensor ('Sensor Green') that was assembled some time ago consists of two 20 cm ferrite rod antennas, a magnetic field amplifier, a GPS module, a headphone and an evaluation board (controller). Figure 4.1 shows the aluminium case with the respective parts as mentioned above. The two ferrite rod antennas and the magnetic field amplifier are set up in the amplifier box analogous to the amplifier box of 'Sensor Red' (see chapter 3.2.1). The headphone is used to make the received signal audible and thus to find a good place for the antennas with regards to acceptable interferences. It is connected to the respective jack on the magnetic amplifier board.

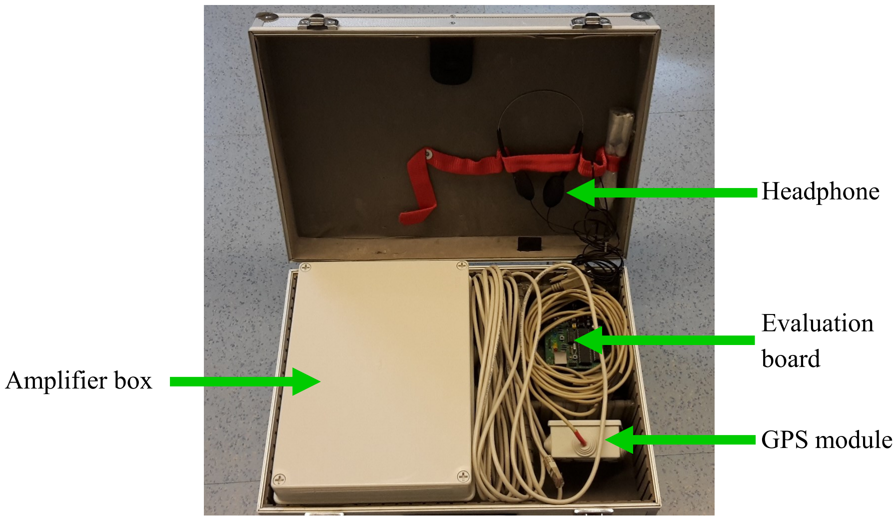


Fig. 4.1.: Aluminium case with the components of 'Sensor Green'

Figure 4.2 shows the evaluation board (PCB 6 Version 5) and the connection assignment of the respective parts. First, the RS232 interface should be connected to the local personal computer in order to guarantee a good electrical grounding. The power supply for the old system (12 V DC) should be connected as the last step. The status information given by the four LEDs, shown in figure 4.2, is shown in table 4.1.

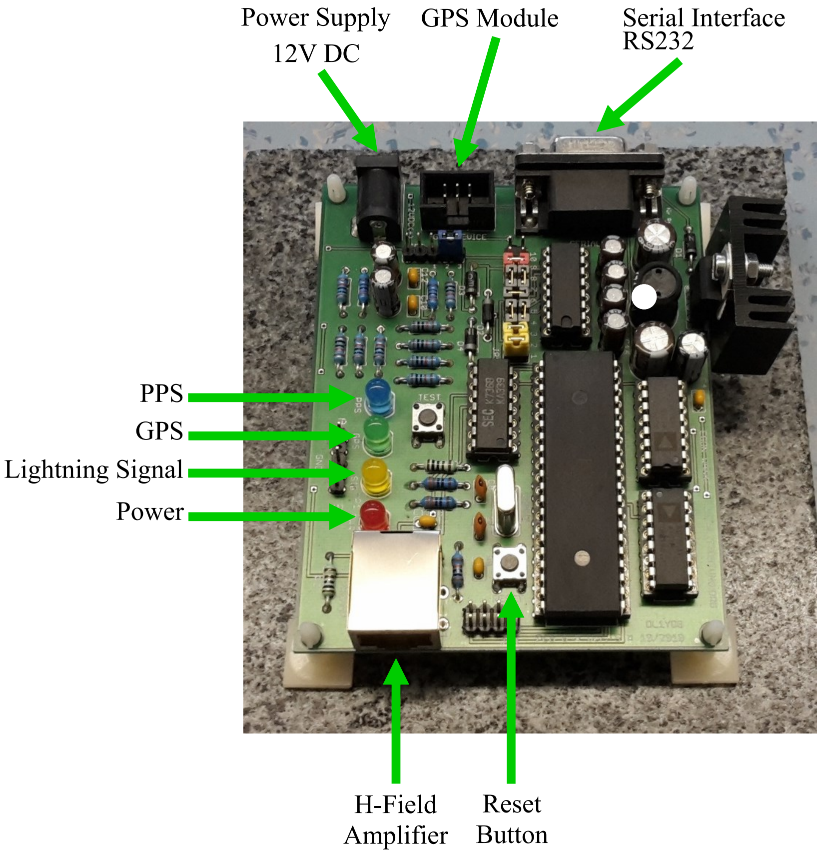


Fig. 4.2.: Evaluation board of 'Sensor Green'

Tab. 4.1.: Status information of 'Sensor Green' given by the LEDs [7]

LED	**state**	**information**
green	blinking	GPS signal found
green	glowing	valid GPS signal locked
blue	blinking	1PPS signal
orange	flashing	lightning signal received
red	glowing	power on

The magnetic field signals that are received by the antennas are filtered and amplified. The amplifier transmits the analogue signal to the controller. The controller converts the analogue signal into a digital signal and assigns an unique timestamp to that signal. The controller receives this timestamp from the GPS device. The digital data sentence containing the strike impulse information, the timestamp and the GPS position of the receiver site is transmitted to a local personal computer over a RS232 interface. On the local personal computer a tracker programme, which is written by Blitzortung.org, is used to transmit the data of the lightning discharge to the server of Blitzortung.org via Internet. Operation of 'Sensor Green' requires permanent running of the local personal computer and a permanent Internet connection. Figure 4.3 shows the schematic of the recording setup for the old device.[7]

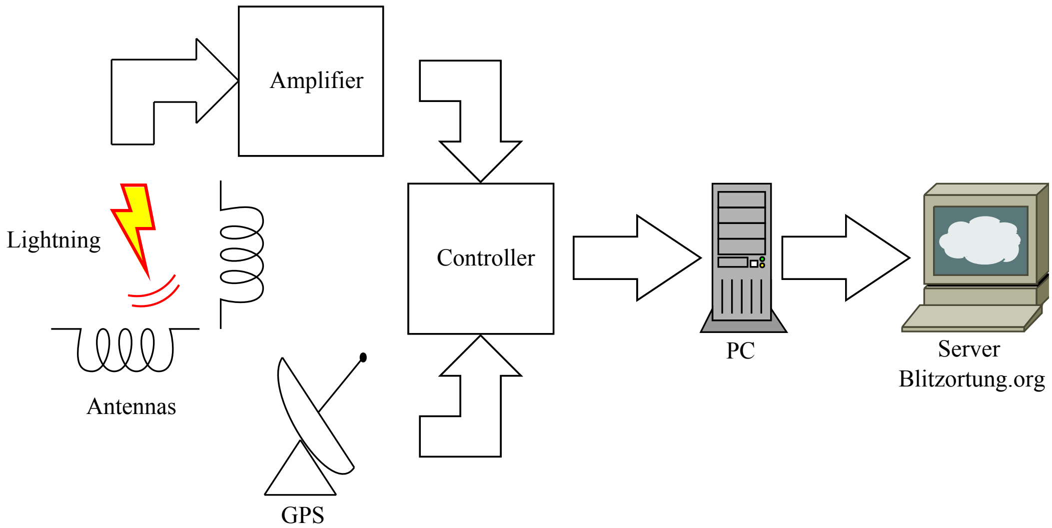


Fig. 4.3.: Recording setup for 'Sensor Green'

4.1.2. Sensor Red

In this thesis only the magnetic field amplifier was used for recording with the new device, because first, this amplifier system is the conventional and most used system. Second, during initial tests no good gain adjustment for the electric field amplifier system was found for the sensor site when the electric field antenna was placed in the attic. Since the electric field amplifier is highly susceptible to interference, the sensor was permanently in interference mode. In this mode no signals are sent to the central server of Blitzortung.org. Both in the automatic mode and the manual mode the gain and the threshold could not be set to a value to ensure that the sensor operated regularly. Due to the fact that the focal point of this thesis was on the magnetic amplifier system and due to the few occasions for recording lightning discharges in good conditions no further time was spent on adjusting the electric field amplifier system. One possibility to solve the interference problem is to ground the system effectively. Another possibility to test the electric field amplifier system is to record lightning data outside in the open field, where the interference level is low.

The recording setup from the antennas to the controller for the new device is similar to the recording setup of the old one. The controller also receives the timestamp and the geographic position from the GPS module and the strike data from the amplifier. But in the case of the new sensor the communication between the controller and the amplifier is bidirectional. The controller can also send commands to the amplifier, for example the controller can adjust the gain.

The controller communicates directly with the server via Internet. For that purpose, the controller is connected to a local area network over Ethernet. The used Ethernet controller ENC28J60 can handle 10 Mbit/s [8]. On the one hand the controller transmits the lightning data to the server, and on the other hand the server is able to adjust the gain and the threshold of the amplifier. The captured UDP [2] data upload from the controller on a day with a medium thunderstorm activity (about 22000 strokes located by ALDIS within a distance up to 500 km around the sensor) was about 300 MB. So the average transfer rate on this day was about 3,46 kB/sec or 0,028 Mbit/sec.

The controller communicates with the operator by means of two kinds of user interfaces, the unidirectional and the bidirectional interface. The unidirectional interfaces, such as the LCD display or the LEDs, are used to give some status information. The bidirectional web-interface is used to give status information in more detail, to show live-lightning data and to set the controller [8]. Figure 4.4 shows the recording setup for the new device.

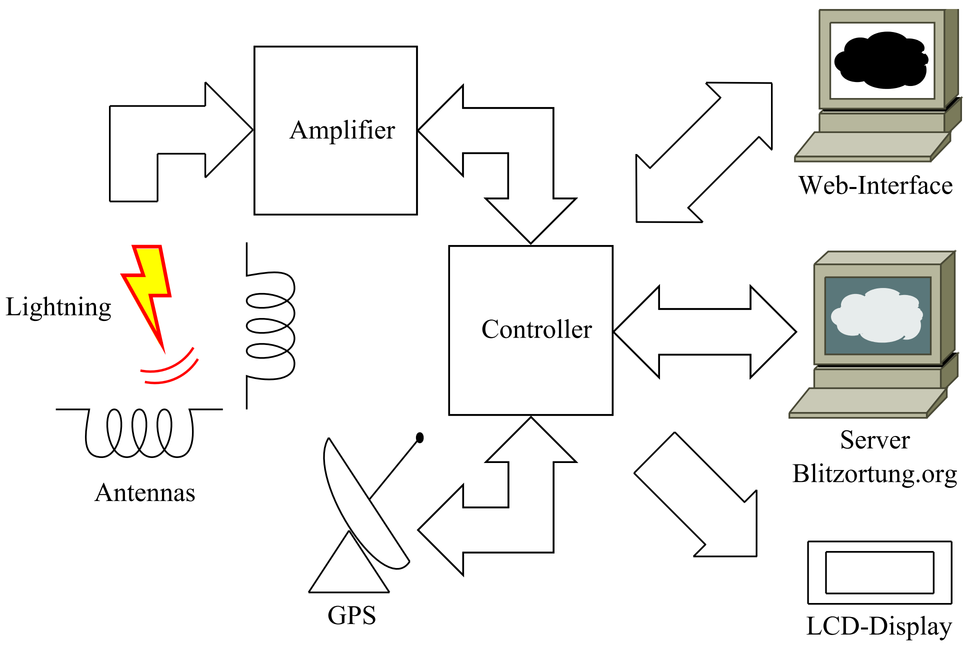


Fig. 4.4.: Recording setup for 'Sensor Red'

[2] User Datagram Protocol

4.2. Data collection

4.2.1. Sensor Green

In order to collect the lightning data of the old sensor ('Sensor Green') for the validation procedure in this project, we have recorded the data, which is transmitted from the controller over the RS232 interface, by the RS232 data logger of Eltima Software. For that purpose, the baud rate has to be set to 4800 and the other options for the data transmission have to be set to 8 data bits, 1 stop bit and no parity bit. No data are forwarded to the central server of Blitzortung.org.

There are two types of data sentence: the 1PPS data sentence beginning with $BS and the stroke data sentence beginning with $BD. The 1PPS data sentence is transmitted each second and the stroke data sentence each time when a lightning signal is received. Figure 4.5 shows the 1PPS data sentence and figure 4.6 the stroke data sentence.

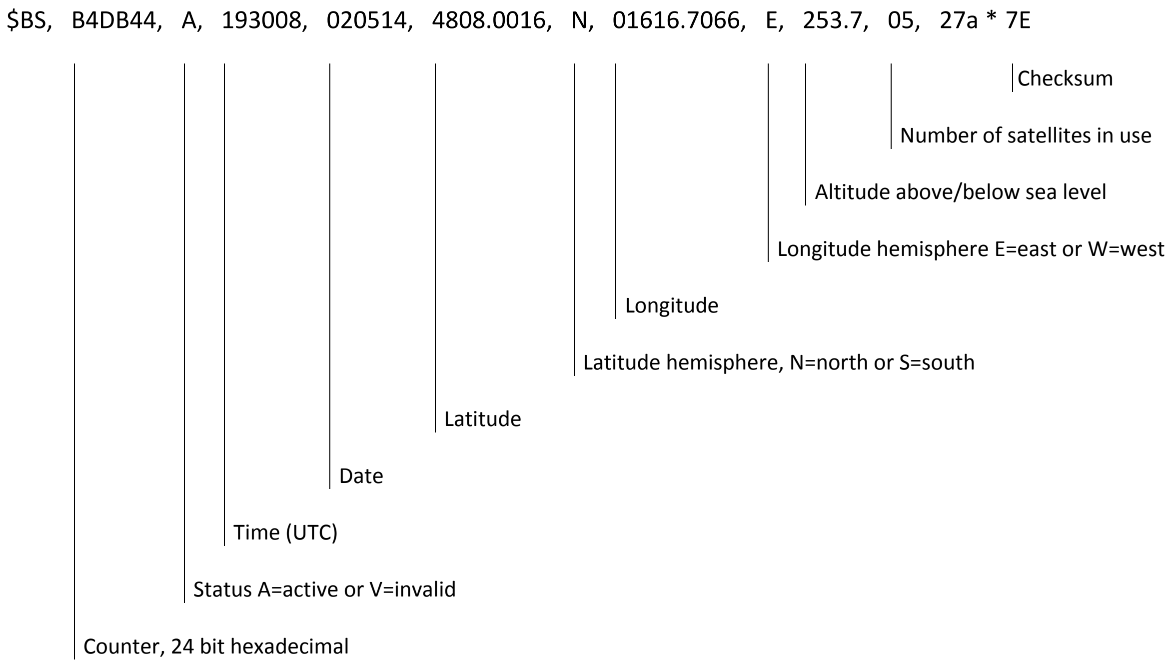


Fig. 4.5.: 1PPS data sentence of 'Sensor Green' [7]

The stroke data sentence contains the counter and the stroke data in hexadecimal characters for both channels. The first two characters are the first byte of channel one, the second two characters are the first byte of channel two and so on. There are 64 bytes for each channel.[7]

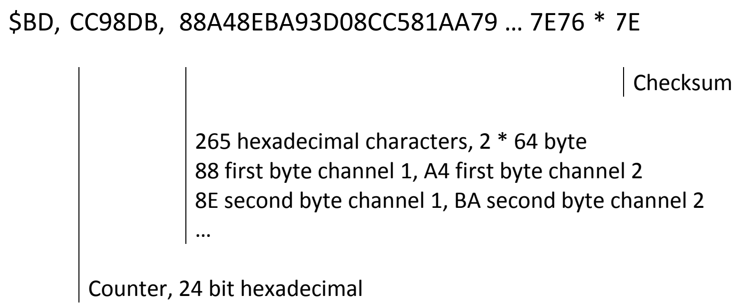


Fig. 4.6.: Stroke data sentence of 'Sensor Green' [7]

The counter is given in a 24 bit hexadecimal format and runs continuously from zero to FFFFFF (16.777.216 decimal). When the counter reaches the maximum value of 16.777.216, it starts again from zero. The counter is transmitted both with the PPS data sentence and the stroke data sentence. The counter difference between two PPS signals is 2.500.000 and thus one counter unit corresponds to 400 ns [7]. The nanoseconds of the timestamp of the lightning signal are calculated from the difference between the counter of the PPS signal and the counter of the stroke signal. Figure 4.7 shows the running of the counter and illustrates the facts explained above.

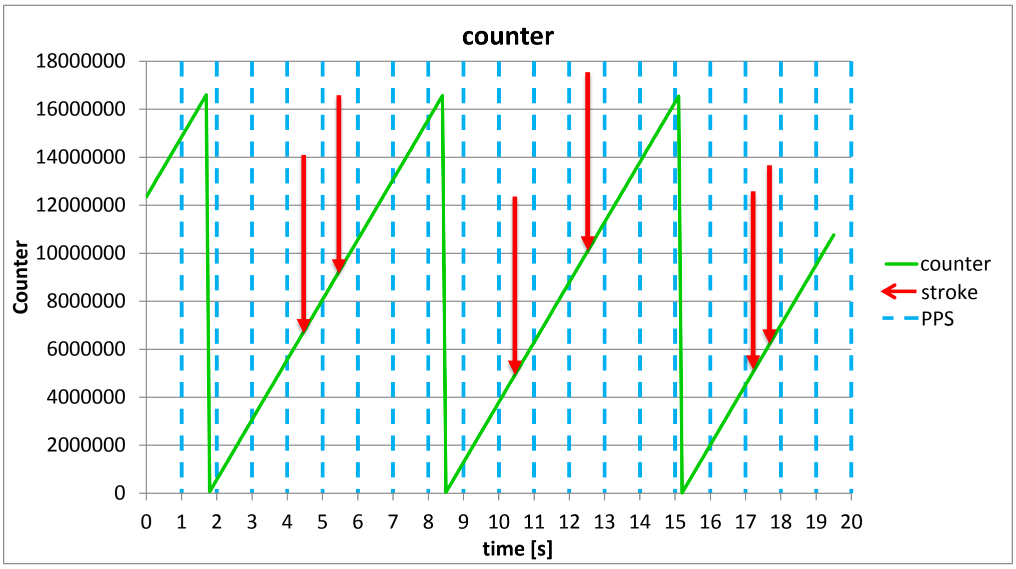


Fig. 4.7.: Running of the counter

The stroke data in hexadecimal format is separated into the bytes for channel one and the bytes for channel two. The bytes in hexadecimal format are converted to decimal format and plotted over time. Due to a sampling rate of 320 kS/s the 64 bytes captured of each channel correspond to a time span of 200 μs [7]. The time difference between two bytes is about 3,13 μs. Figure 4.8 shows the characteristic of a captured stroke data. Due to the fact that only 64 bytes are recorded the curve is quite ragged. The beginning phase of the lightning discharge is not completely recorded.

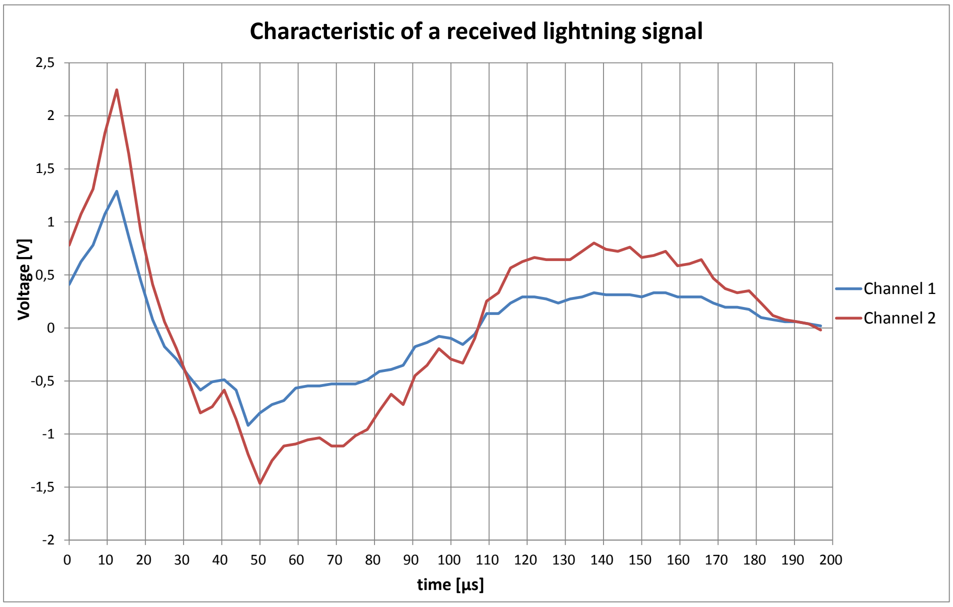


Fig. 4.8.: Characteristic of a lightning signal captured by the H-field antenna of 'Sensor Green'

4.2.2. Sensor Red

Since the new recording device communicates directly with the server of Blitzortung.org over an Ethernet controller, the stroke data is not delivered to the serial interface during normal operation. If the data of a received signal of a lightning discharge would be written to the serial interface, this would slow down the system significantly [8]. Therefore it is not possible to use the debug interface to log the lightning data during an active storm.

Thus, the communication between the device and the server is monitored by a so called sniffer programme. The sensor and the local personal computer are connected to a local area network by a hub. The sniffer programme Wireshark (version 1.12.2) runs on the local personal computer and captures the communication between the sensor and the server. Figure 4.9 shows the setup for the recording with the sniffer programme.

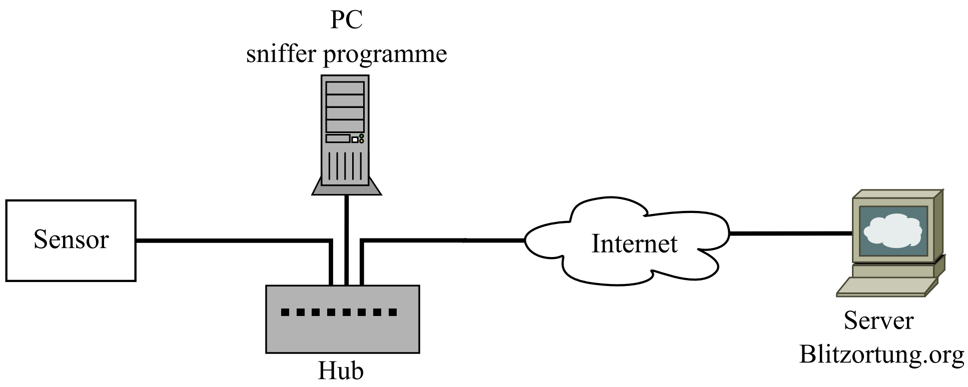


Fig. 4.9.: Setup for recording with the sniffer programme Wireshark

All data traffic is filtered by the IP-address of the detection device and the UDP packets. So only the UDP packets, which are sent from the detection device to the server, are recorded. For that purpose, select the LAN-connection in the dialogue box 'Capture Options' of the sniffer programme Wireshark (see figure 4.10), and enter the capture filter. The capture filter has the following format: `host IP-address and udp`. Instead of 'IP-address', insert the IP-address of the sensor.

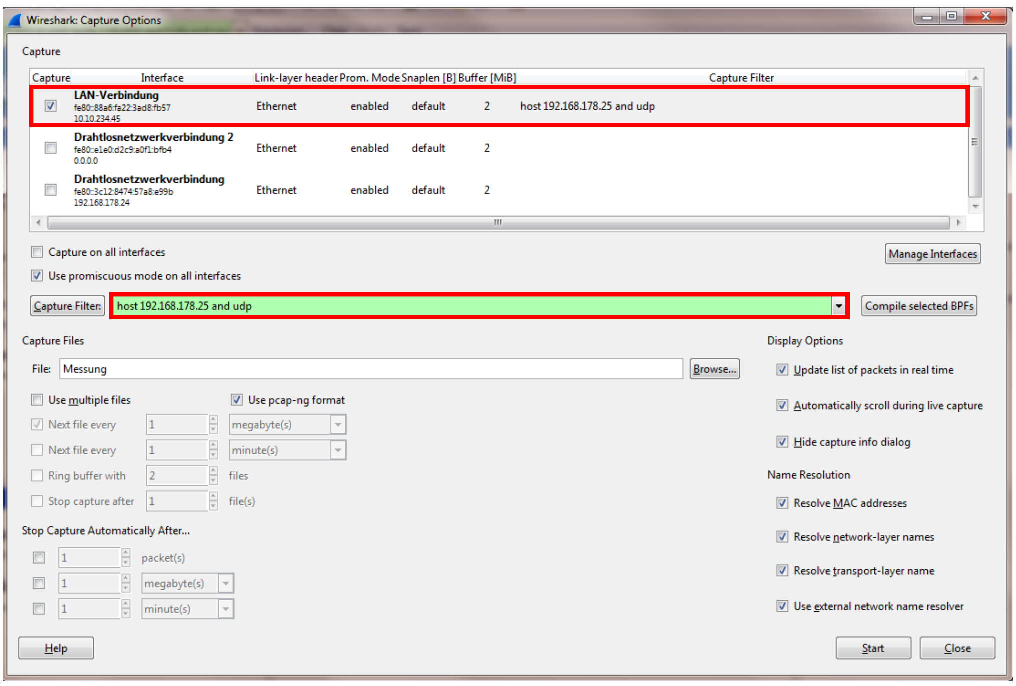


Fig. 4.10.: 'Capture Options' dialogue box of the sniffer programme Wireshark

To export the captured UDP data as plain text file select the field 'File → Export Packet Dissections → as 'Plain Text' file…'. In the dialogue box 'Export File', choose the checkbox 'Packet Bytes' in the section 'Packet format'. Figure 4.11 shows the 'Export File' dialogue box.

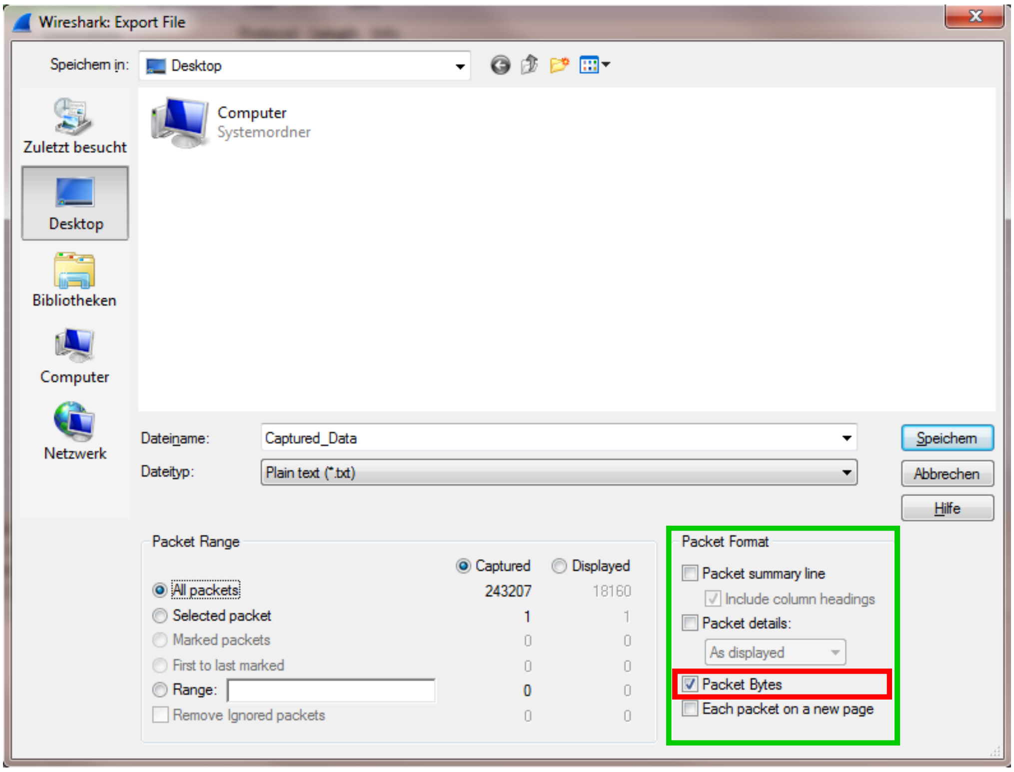


Fig. 4.11.: 'Export File' dialogue box of the sniffer programme Wireshark

The captured data is stored in hex dump format, because the stroke data sentence contains ASCII code as well as hexadecimal code. In hex dump format the data is stored both as ASCII code and as hexadecimal code. The data sentence in ASCII or raw format displays the station number, processor ID, date, time and the geographic position of the sensor, but not the stroke data in hexadecimal characters. For further data analysis the timestamp is cut out from the ASCII code and the stroke data is cut out from the hexadecimal code (see chapter 4.4.1.2). Figure 4.12 shows the format of the text file and figure 4.13 shows the schematic of the stroke data sentence of 'Sensor Red'.

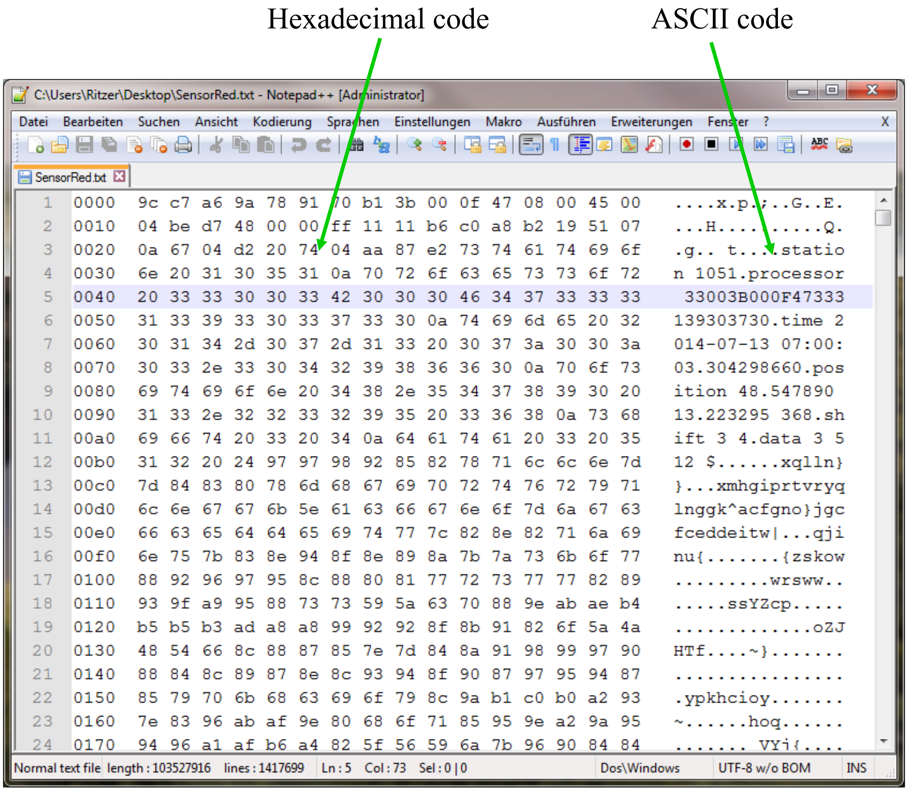


Fig. 4.12.: Format of the plain text file of the captured data of 'Sensor Red'

In the web-account the processor ID can be assigned to the station number that is given by Blitzortung.org when creating the web account. The transmitted timestamp contains the date and the time including the nanoseconds. The geographic position is given in latitude, longitude and altitude. In figure 4.13 the stroke data in hexadecimal characters of channel one is written after the $ character after 'data 3' and the data of channel two is written after the $ character after 'data 4'. The stroke data of one channel consists of 512 bytes. Two characters form one byte.

```
station 1051
processor 33003B000F47333139303730
time 2014-07-13 07:00:03.304298660
position 48.547890 13.223295 368
shift 3 4
data 3 512 $ 97 97 98 92 85 82 … 63 63 60 8c 0a
shift 4 4
data 4 512 $ 8b 81 73 6a 65 69 … 2f 52 75 b3 0a
```

Fig. 4.13.: Stroke data sentence of 'Sensor Red'

The stroke data in hexadecimal format of each channel is converted to decimal format and plotted over the time. Due to a sampling rate of 618 kS/s, the 512 bytes captured of each channel correspond to a time span of 828 μs. The time difference between two bytes is about 1,62 μs. Figure 4.14 shows the characteristic of a captured stroke data. Due to the fact that 256 bytes are stored before the trigger and due to a sampling duration of 828 μs the lightning discharge is completely recorded.

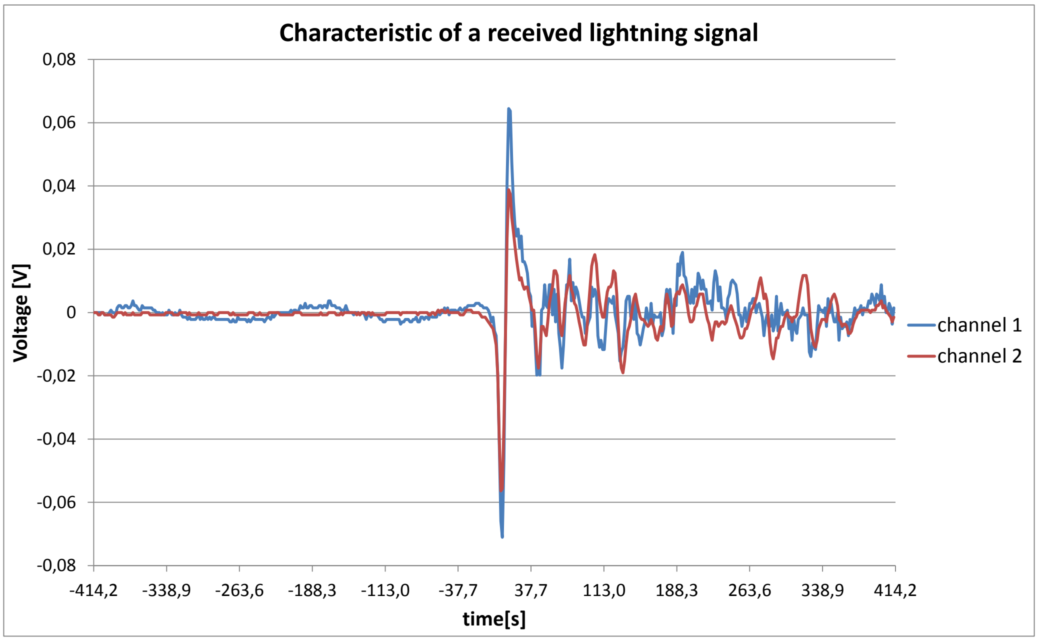


Fig. 4.14.: Characteristic of a lightning signal captured by the H-field antenna of 'Sensor Red'

4.3. Recordings

Table 4.2 gives an overview of the recordings of real lightning data in summer 2014, which were made with 'Sensor Green' and 'Sensor Red' and which are incorporated into this thesis. The number of strokes detected by ALDIS (N_{ALDIS}) and the number of strokes detected by Blitzortung.org (N_{BO}) are given as function of the distance to the sensor site.

Tab. 4.2.: Data recordings with 'Sensor Green' and 'Sensor Red'

sensor	**date**	**time (UTC)**	**location**	**sensor site**	**gain**	$\mathbf{N_{ALDIS}}$	$\mathbf{N_{BO}}$	**distance [km]**
Sensor Green	02.05.2014	19:21:50 - 20:57:32	Liesing (Wien)	inside	low	2086	741	500
Sensor Red	13.07.2014	07:00:03 - 09:00:00	Ortenburg (Bayern)	inside	automatic	6017	3653	1000
Sensor Red	20.07.2014	15:19:53 - 15:59:59	Ortenburg (Bayern)	inside	automatic	11054	6381	1000
Sensor Red	09.09.2014	19:47:28 - 21:44:48	Ortenburg (Bayern)	inside	high	6150	862	500
Sensor Red	11.09.2014	13:32:15 - 16:40:47	Ortenburg (Bayern)	inside	low	4524	1040	500
Sensor Red	22.09.2014	15:10:02 - 18:30:01	Ortenburg (Bayern)	inside	medium	10377	2543	1000

4.4. Data Analysis

The recorded data of both devices, 'Sensor Red' and 'Sensor Green', and the computed lightning data of Blitzortung.org are compared to the computed lightning data of ALDIS. For that purpose two Excel-programmes (Excel 2010) are developed, one for 'Sensor Green' and the other one for 'Sensor Red'. The particular programme imports the respective lightning data, compares the data in consideration of the timestamp, and analyses the time matched signals as a function of peak current and distance of the lightning discharge. The difference between the two programmes is the import procedure of the recorded signals, due to the different format of the data sentences provided by 'Sensor Green' and 'Sensor Red', respectively.

4.4.1. Data Import

4.4.1.1. Sensor Green

In the worksheet 'Blitzortung_Sensor' of the Excel programme for 'Sensor Green' the captured sensor data can be imported. By clicking on the button 'data sensor' the dialogue box 'Import data' appears (see figure 4.15). The button '. . . ' is for selecting the text file to be inserted. The maximum text file size that can be imported, is 50 MB. If a text file is larger than 50 MB, the text file has to be split into several files, which are smaller or equal to the limit. One option to split the text file is to use a text editor like Notepad++. In order to import the data into the programme at once, the ending of the first text file name has to be '_1', the ending of the second file has to be '_2' and so on. The total number of text files, which should be imported for the analysis, has to be inserted into the field 'number of files'. If no text file is selected and if the field 'number of files' is empty, the programme will exit. With the checkbox 'delete data' it is possible to choose whether the already existing data in the Excel sheet should be deleted or whether the new data should be imported and appended at the end of the already existing data.

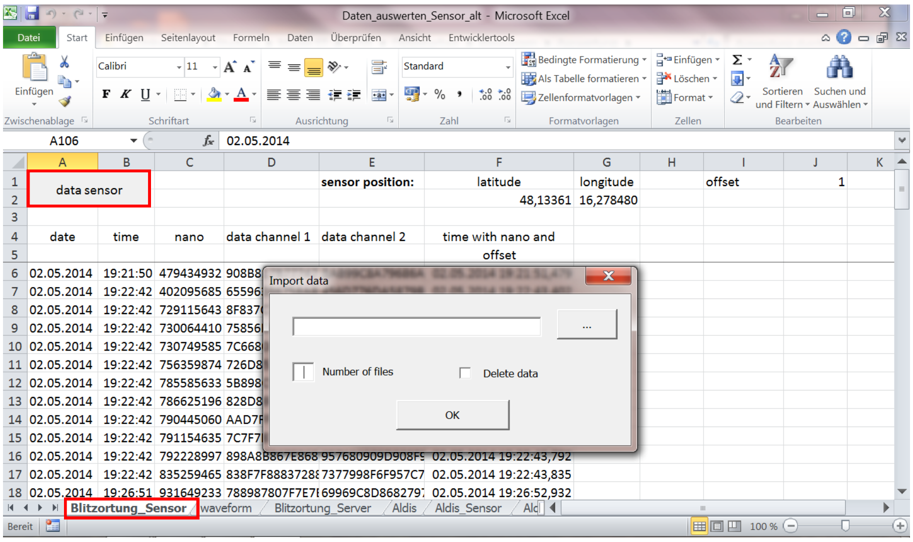


Fig. 4.15.: Dialogue box 'Import data'

Beside the recording duration and the thunderstorm activity, the file size depends on the format of the stroke data sentence, which is the number of bytes taken per channel, and on the format, in which the lightning data is stored. The file size of the lightning data stored in hex dump format is bigger than the file size of the one in ASCII format. For 'Sensor Green' it is usually not necessary to split the text file, because the number of captured bytes for each channel is only 64 bytes and the captured lightning data is stored in ASCII format. For this reason the file size is smaller than 50 MB in most cases, except the case when the recording lasts several days and the thunderstorm activity is very high. This means that for a recording that lasts 24 hours, about 160,000 stroke data sentences could be recorded to reach the maximum file size of 50 MB.

By clicking on the button 'OK' the visual basic (VBA) macro opens the selected text file, searches for the stroke data sentences and imports the relevant data into the Excel sheet (see figure 4.15). The nanoseconds are calculated from the counter difference between the 1PPS data sentence and the stroke data sentence (see chapter 4.2.1). The stroke data are divided into the information of channel one and channel two and inserted separately in the Excel sheet. Finally, the timestamp consisting of date, time, nanoseconds and offset, is calculated. The offset is the time difference between the timestamp of the captured stroke data and the timestamp of the computed data (see chapter 5). If more than one text file is to be imported, the macro opens the next file.

4.4.1.2. Sensor Red

The procedure for the data import of 'Sensor Red' is similar to the procedure for the data import of 'Sensor Green'. By clicking on the button 'data sensor' in the worksheet 'Blitzortung_Sensor' in the excel-programme for 'Sensor Red' the dialogue box 'Import data' opens, as explained in chapter 4.4.1.1.

Since the text file size of the captured stroke data of 'Sensor Red' is in most cases bigger than the maximum file size of 50 MB, the text file of one recording session has to be split up in several smaller files. An easy way to create text files with a size of less than 50 MB is to create respective Wireshark capture files during the capturing period. For that purpose the option 'Use multiple files' in the 'Capture Options' dialogue box (see figure 4.10) has to be selected and the option, that the next capture file shell be created every 10 MB, has to be set. A capture file with a size of 10 MB corresponds to a text file with a size of about 50 MB. Another simple way to create such text files is to export only a certain range of the captured data packets in Wireshark. About 8900 UDP packets correspond to a text file size of 50 MB. Therefore, insert the range of packets, for example 1-8900, in the section 'Packet Range' in the 'Export File' dialogue box (see figure 4.16) of Wireshark.

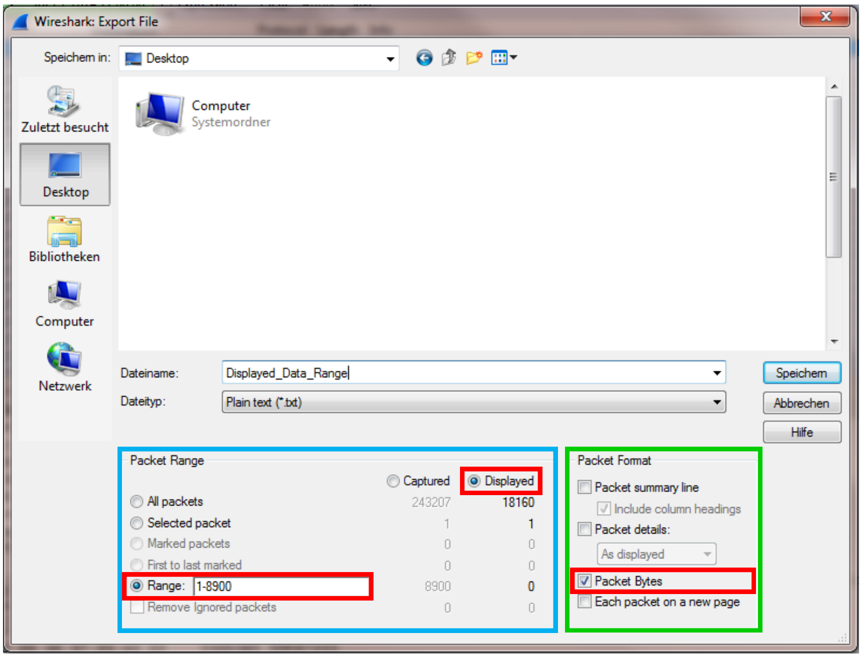


Fig. 4.16.: Options in the dialogue box 'Export File' for exporting a packet range

In combination with the display filter function the captured data can be filtered by the frame time in such a way, that only a certain and relevant time span is displayed and saved. The display filter for filtering by time has the following format:

```
(frame.time >= "Jul 13, 2014 09:00:00") && (frame.time <= "Jul 13, 2014 11:00:00")
```

By clicking on the button 'OK' the VBA macro opens the selected text file, and stores the hexadecimal numbers and the ASCII characters in separate variables. The timestamp and the nanoseconds of the respective signals are cut out from the ASCII stream and the stroke data is cut out from the hexadecimal numbers. Finally, the data are printed in the Excel sheet and the timestamp including the offset is calculated.

4.4.1.3. Data Import Blitzortung.org

By clicking on the button 'data server' in the worksheet 'Blitzortung_Server' in both Excel programmes the dialogue box 'Select File To Be Opened' appears (See figure 4.17). Both, a text file and a CSV file can be imported.

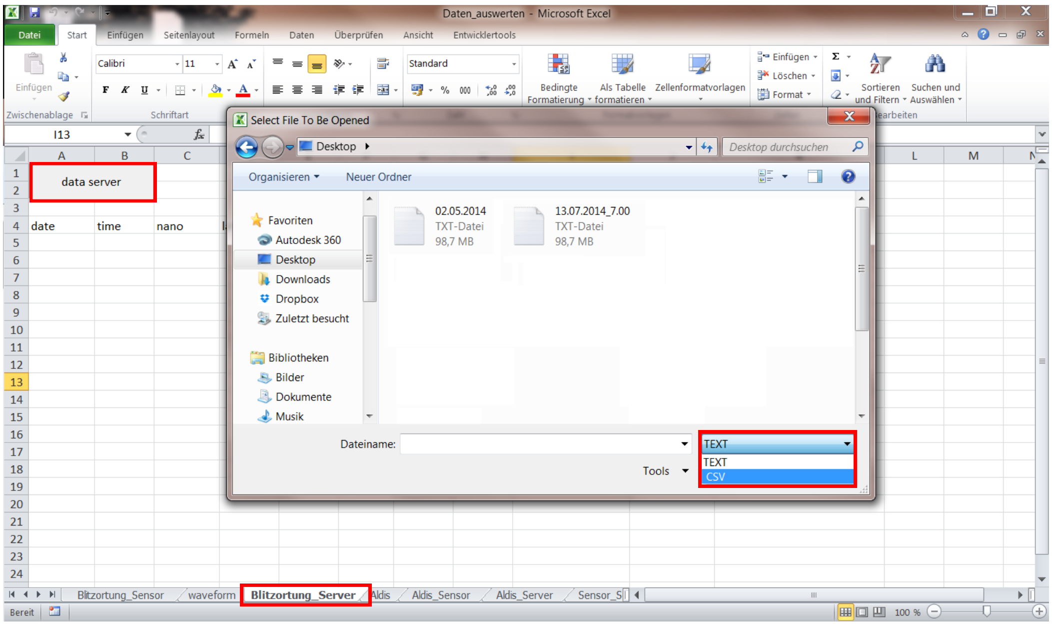


Fig. 4.17.: Dialogue box 'Select File To Be Opened'

To import the computed data of Blitzortung.org the text file has to be edited in a text editor. The computed lightning data of Blitzortung.org is available for all participants on their web-account. The original format and the required format of the text file of the lightning data of Blitzortung.org are shown in figure 4.18.

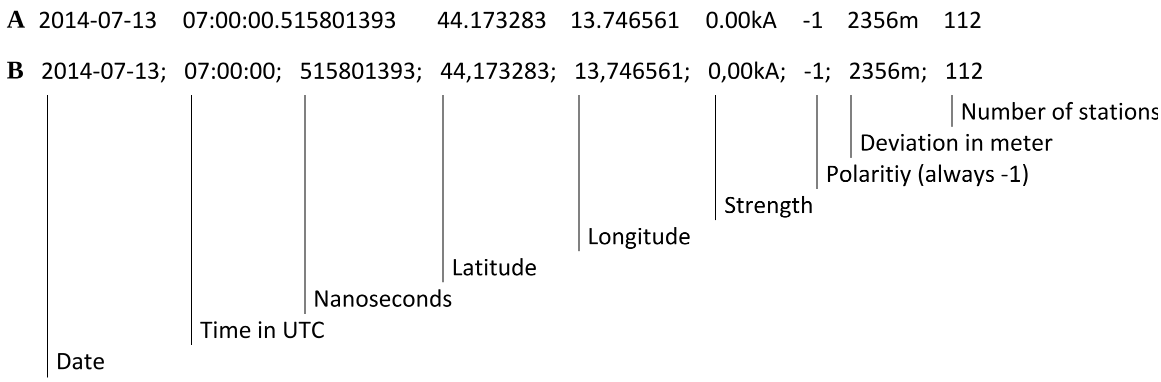


Fig. 4.18.: Original and converted data format of the computed lightning data of Blitzortung.org
A: original format **B:** required format

After inserting the text file, the distance between the lightning discharge and the recording device is calculated based on spherical coordinates. The timestamp including nanoseconds and propagating time is computed. The propagating time is the time that the electromagnetic wave takes to reach the receiver site. ALDIS and Blitzortung.org provide the actual time of occurrence of the lightning strike. Therefore, the propagating time has to be added to the timestamp in order to compare the computed lightning data with the recorded sensor data.

4.4.1.4. Data Import ALDIS

The procedure for the data import of the lightning data of ALDIS is similar to the procedure for the data import of the lightning data of Blitzortung.org. By clicking on the button 'data aldis' in the worksheet 'Aldis' the dialogue box 'Select File To Be Opened' also appears (see figure 4.17), where the CSV file of the computed data of ALDIS must be selected.

As a result of the ALDIS lightning database query, a CSV file is created. Figure 4.19 - 4.22 show the options for the database query using the ALDIS internal tool CATS EXPERTISE. All strokes, including the Intra-Cloud ones, are interrogated. The fields in the dialogue box 'List of fields' in figure 4.21 determine the format of the CSV file. To import the computed data of ALDIS, the CSV file has to be edited in a text editor. Figure 4.23 shows the original and the required format of the CSV file of the computed lightning data of ALDIS. After inserting the CSV-file, the timestamp including nanoseconds and propagating time is computed.

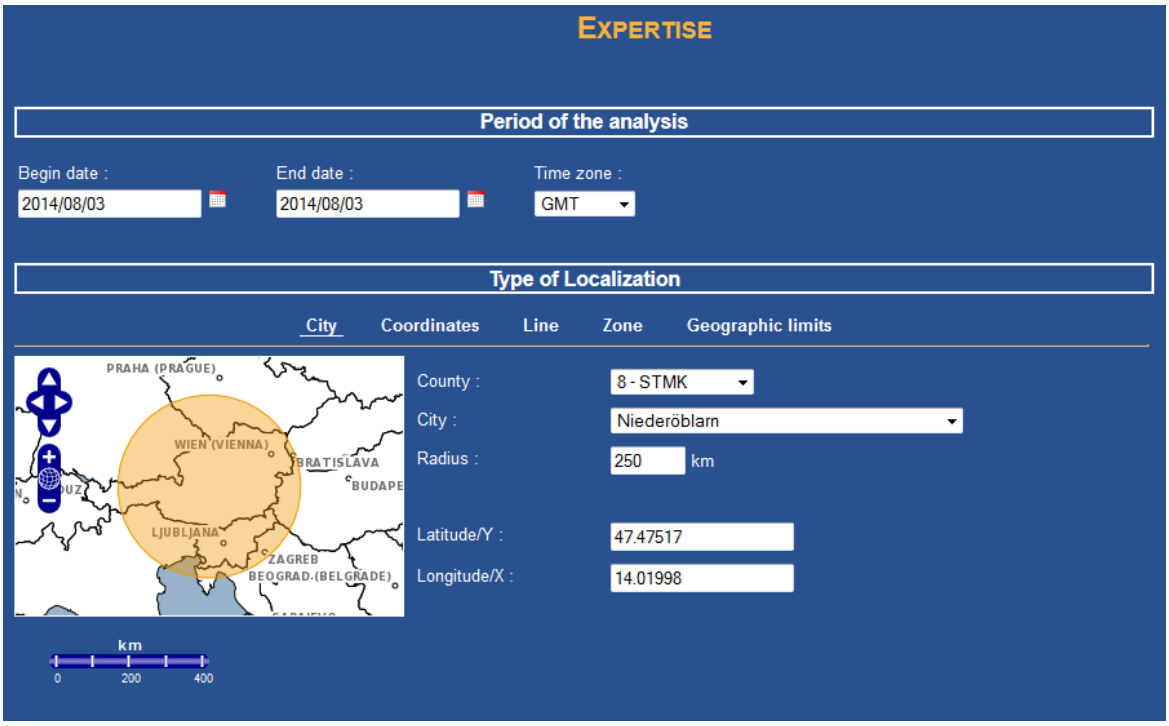


Fig. 4.19.: Options for ALDIS lightning database query - time period and localisation

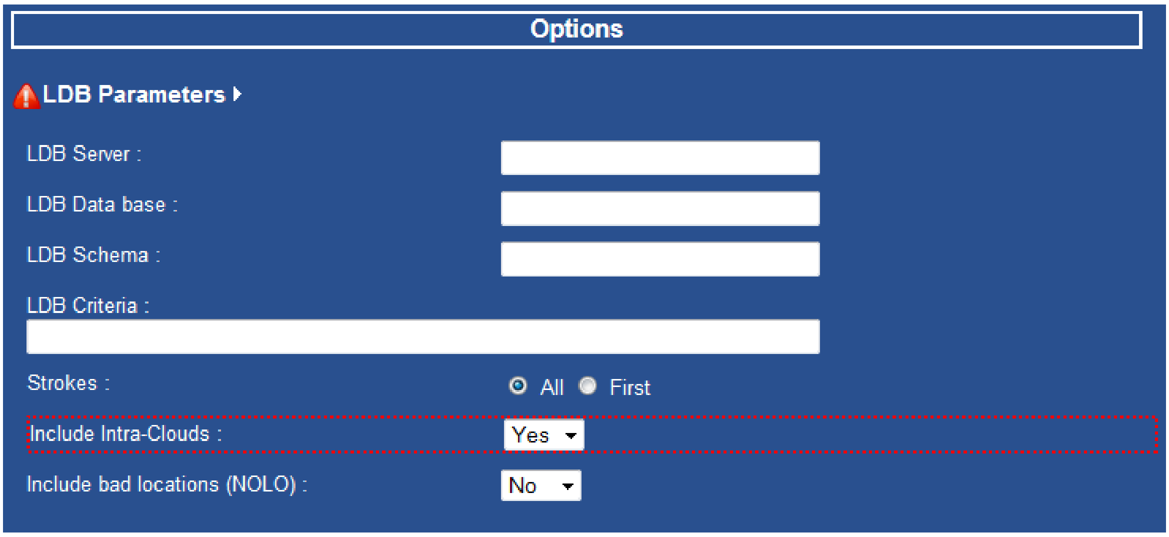


Fig. 4.20.: Options for ALDIS lightning database query - stroke options

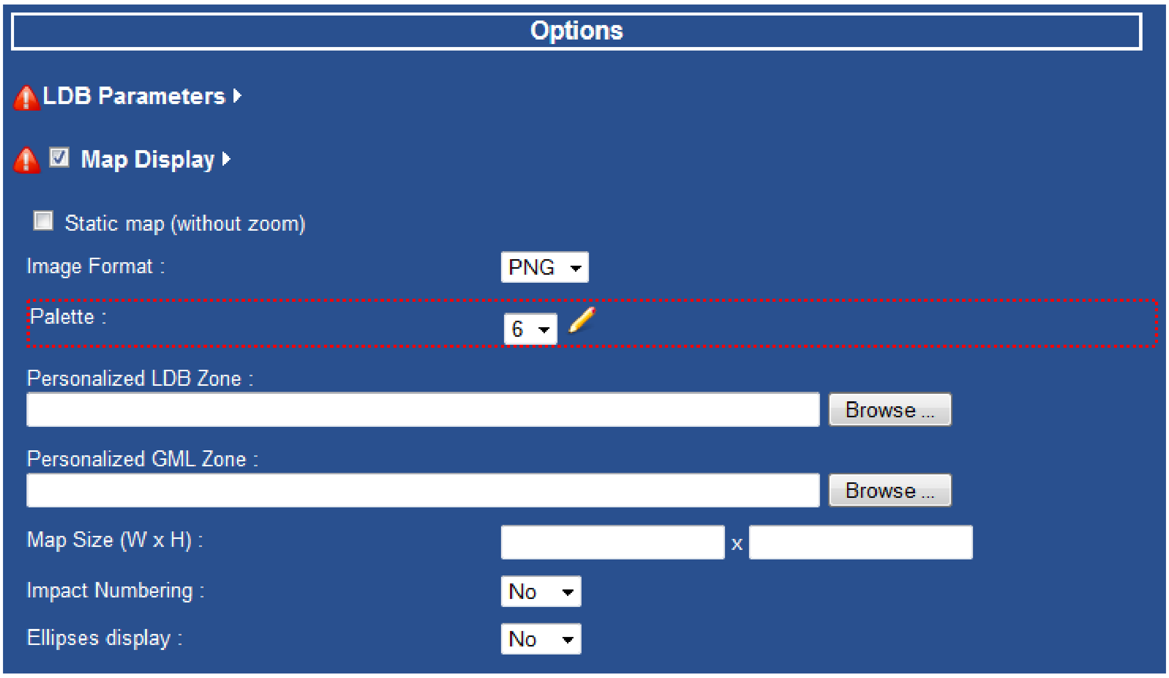


Fig. 4.21.: Options for ALDIS lightning database query - map options

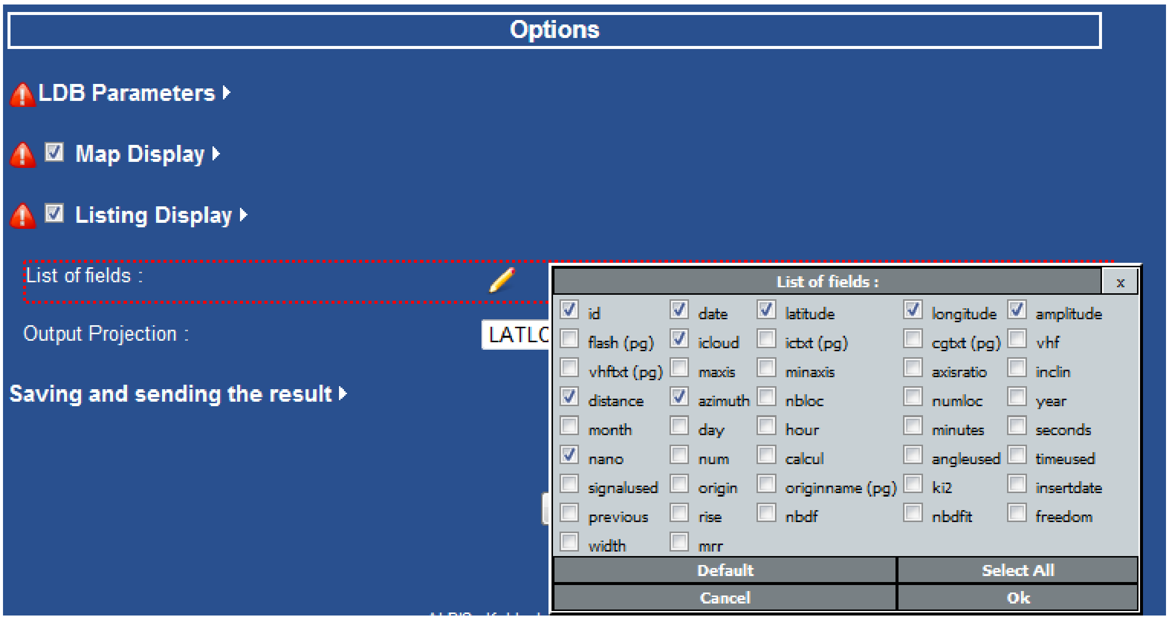


Fig. 4.22.: Options for ALDIS lightning database query - listing options

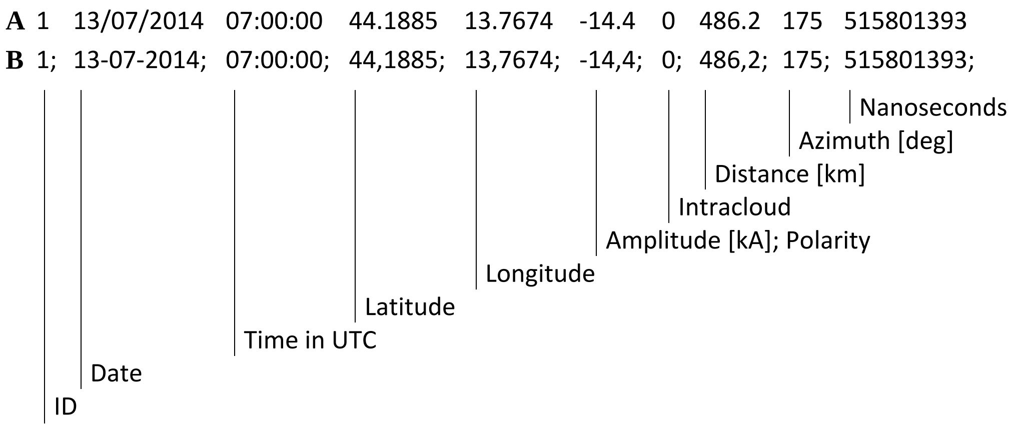


Fig. 4.23.: Original and converted data format of the computed lightning data of ALDIS
A: original format **B:** required format

4.4.2. Data Comparison

In the worksheets Aldis_Sensor (ALDIS vs Sensor) and Aldis_Server (ALDIS vs Server BO) the recorded raw data of the test sensors and the computed lightning stroke data of Blitzortung.org are compared to the lightning stroke data of ALDIS in consideration of the GPS timestamp. In the worksheet 'Sensor_Server' (Sensor vs Server) the recorded data of the sensors and the computed lightning data of Blitzortung.org are compared. Figure 4.24 gives an overview of the comparison of the different data.

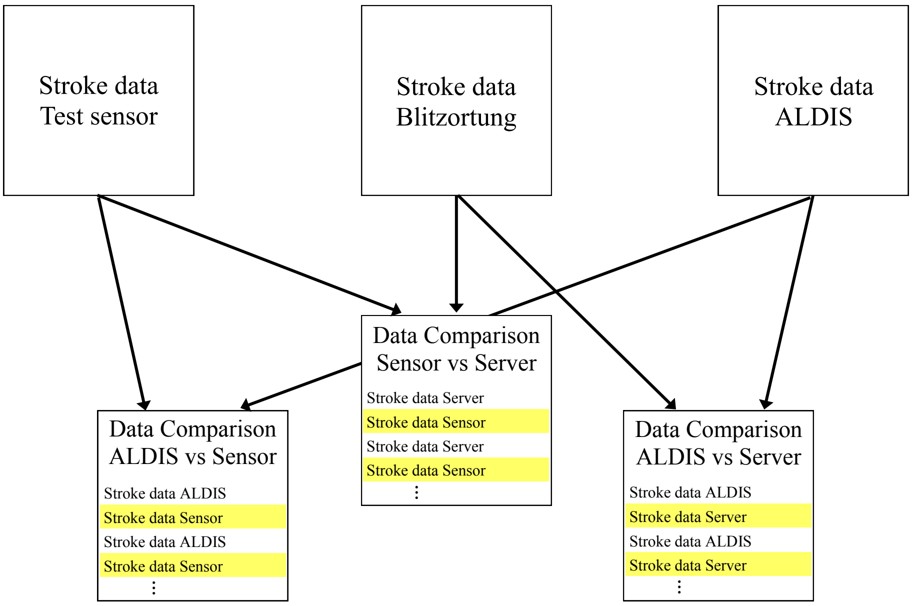


Fig. 4.24.: Schematic of the data comparison procedure

By clicking on the command button in the respective worksheet the timestamps of the measured data or the computed data are inserted in the worksheet. In the worksheets 'Aldis_Sensor' (ALDIS vs Sensor) and 'Aldis_Server' (ALDIS vs Server BO) the data of the server and the sensor of Blitzortung.org are marked yellow and the data of ALDIS is marked white. In the worksheet 'Sensor_Server' (Sensor vs Server) the data of the sensor is marked yellow and the data of the server is marked white. Additionally relevant information, such as amplitude, distance or classification, is imported.

All the data are correlated according to the timestamp. For this purpose both data sets combined are sorted according to the timestamps and the time difference between two consecutive signals is calculated. If the time difference between a received signal of the sensor and a located lightning discharge of ALDIS is less than 30 microseconds, the signal is classified as correlated lightning discharge and the flag is set to one. Finally, the data are filtered by the flag so that only the time matched signals are displayed. Figure 4.25 shows the Excel sheet 'Aldis_Sensor' (ALDIS vs Sensor) with the time matched signals between sensor and ALDIS.

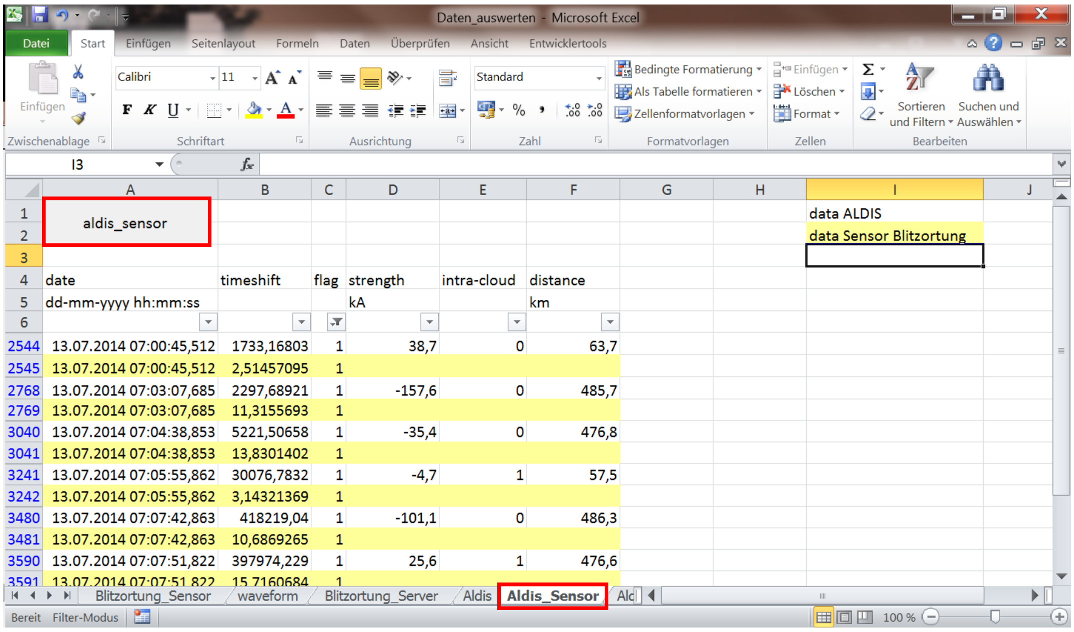
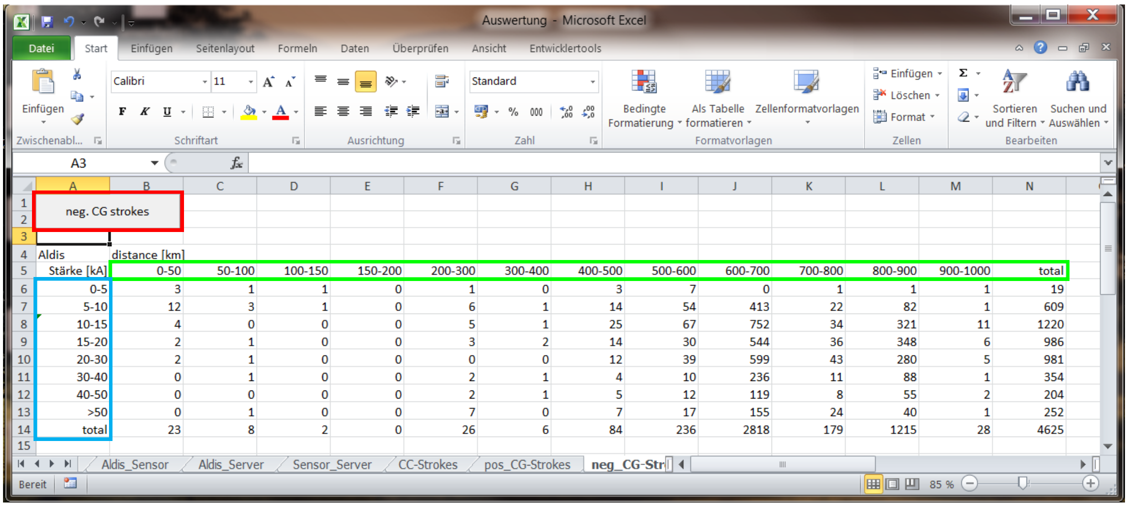


Fig. 4.25.: Worksheet 'Aldis_Sensor'

4.4.3. Data Analysis

The time matched signals are classified in Cloud-Cloud, positive Cloud-Ground and negative Cloud-Ground strokes according to the classification of ALDIS. In the worksheets 'CC-Strokes', 'pos_CG-Strokes' and 'neg_CG-Strokes' the time matched signals are analysed as a function of current intensity and distance. The range of distance is between 0 - 1000 km. Figure 4.26 shows the resulting matrix table of this analysis, in which the divisions for the peak current (blue rectangle) and the distance (green rectangle) are depicted.


neg. CG strokes

Aldis	distance [km]												
Stärke [kA]	0-50	50-100	100-150	150-200	200-300	300-400	400-500	500-600	600-700	700-800	800-900	900-1000	total
0-5	3	1	1	0	1	0	3	7	0	1	1	1	19
5-10	12	3	1	0	6	1	14	54	413	22	82	1	609
10-15	4	0	0	0	5	1	25	67	752	34	321	11	1220
15-20	2	1	0	0	3	2	14	30	544	36	348	6	986
20-30	2	1	0	0	0	0	12	39	599	43	280	5	981
30-40	0	1	0	0	2	1	4	10	236	11	88	1	354
40-50	0	0	0	0	2	1	5	12	119	8	55	2	204
>50	0	1	0	0	7	0	7	17	155	24	40	1	252
total	23	8	2	0	26	6	84	236	2818	179	1215	28	4625

Fig. 4.26.: Matrix table for analysis of negative CG-strokes

By clicking on the command button the frequency of a lightning discharge with a particular peak current within a particular distance to the position of the device is calculated using matrix formulas. The frequency is calculated for the time matched signals between ALDIS and the sensor and the time matched signals between ALDIS and Blitzortung.org. For example, the frequency of a lightning discharge with a current intensity of 5 kA in a distance of less than 100 km is computed. The results are displayed in corresponding graphs.

5. Results

The following discussion of the results of the analysis of the time matched lightning strokes is limited to the negative Cloud-Ground (CG) strokes, because the antennas of the detection devices of Blitzortung.org are designed for CG strokes [8] and the negative CG strokes are the most frequent in nature. The results of analysis for the Cloud-Cloud (CC) strokes and the positive CG strokes are displayed in the appendix.

The recorded data of both sensors, 'Sensor Green' and 'Sensor Red', have a time offset of exactly one second to the computed lightning data of Blitzortung.org and ALDIS. The reason for that time offset is a bug in that part of the firmware of the controller that interprets the 1PPS signal of the GPS device and computes the timestamp. The time offset is corrected by the central server of Blitzortung.org.[personal communication with Frank Dahlslett on 10.09.2014 [3]]

5.1. Sensor Green

The recording of data that is analysed in this section was done on the 02.05.14 (see table 4.2) while a thunderstorm was passing over. The sensor was placed inside the house in an urban area, which is a normal operation of the sensor according to the specifications of Blitzortung.org. The gain of the amplifier was set to very low. Figure 5.1 shows the lightning stroke map of the recording period according to ALDIS and figure 5.2 shows the lightning stroke map according to Blitzortung.org up to a distance of 500 km to the sensor site (green house). The thunderstorm activity in the entire distance range during the recording period was rather low.

During that time span ALDIS has detected 2086 lightning strokes at a distance up to 500 km and Blitzortung.org has detected 741 lightning strokes. The number of the time matched signals between 'Sensor Green' and ALDIS is 41 and between Blitzortung.org and ALDIS is 405. Blitzortung.org does not differentiate between Cloud-Cloud and Cloud-Ground strokes. Therefore the classification of the time matched strokes is done according to the classification of ALDIS. Figure 5.3 shows the number of all strokes located by ALDIS and the number of time matched strokes of 'Sensor Green'

[3]Frank Dahlslett also found this time offset of 1 s and communicated with Blitzortung.org about this offset on 09.02.2009

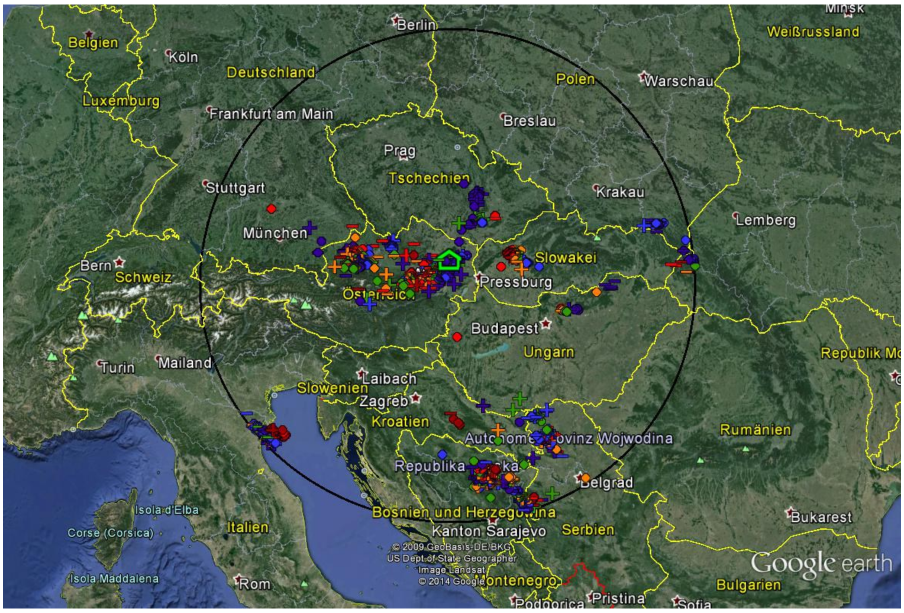


Fig. 5.1.: Lightning stroke map of the data recording period on 02.05.2014 by ALDIS green house indicates sensor site; area radius is 500 km; $N_{ALDIS} = 2086$

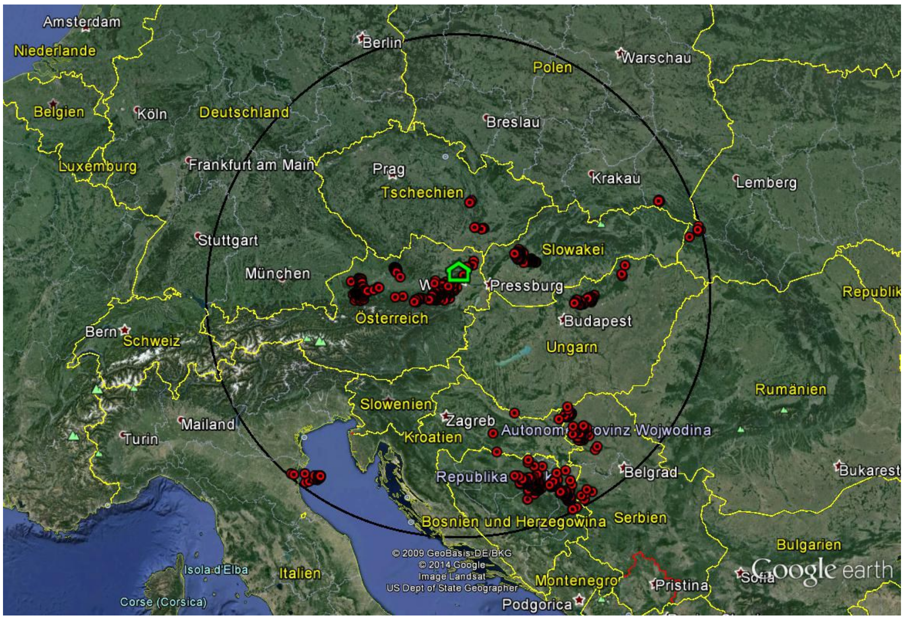


Fig. 5.2.: Lightning stroke map of the data recording period on 02.05.2014 by Blitzortung.org
green house indicates sensor site; area radius is 500 km; $N_{BO} = 741$

and of Blitzortung.org, according to their classification as Cloud-Cloud (CC), positive Cloud-Ground (CG+) and negative Cloud-Ground (CG−) strokes.

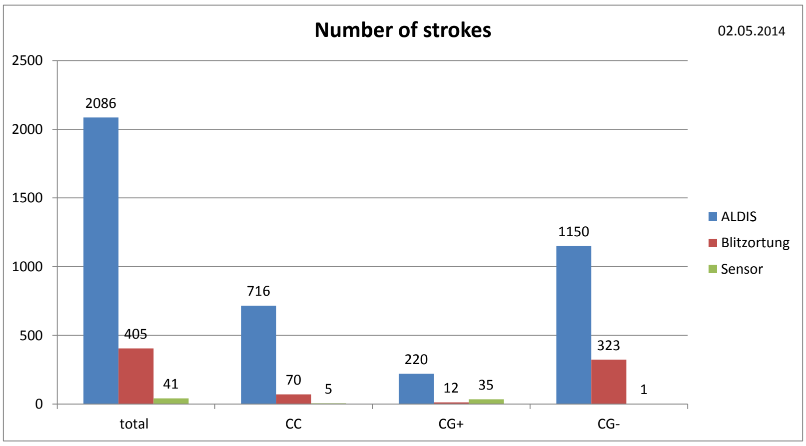


Fig. 5.3.: Number of located strokes on 02.05.2014
ALDIS: All strokes located by ALDIS
Blitzortung: Time matched strokes of Blitzortung.org and ALDIS
Sensor: Time matched stroke reports of 'Sensor Green' and ALDIS

The number of all negative Cloud-Ground strokes detected by ALDIS as a function of peak current and distance is shown in figure 5.4. About 60 % of the located negative CG-strokes are low-current strokes with a peak current in the range of 5 to 20 kA. Most strokes have a peak current between 10 and 15 kA and are located in a range of 400 to 500 km from the sensor site.

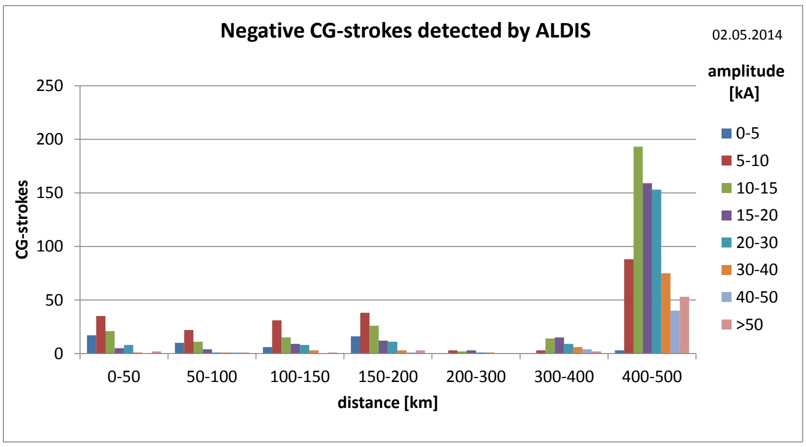


Fig. 5.4.: Negative CG-strokes detected by ALDIS on 02.05.2014 as a function of distance and peak current

Figure 5.5 shows the number of the time matched negative Cloud-Ground strokes of 'Sensor Green' and ALDIS in percent relative to all strokes detected by ALDIS.

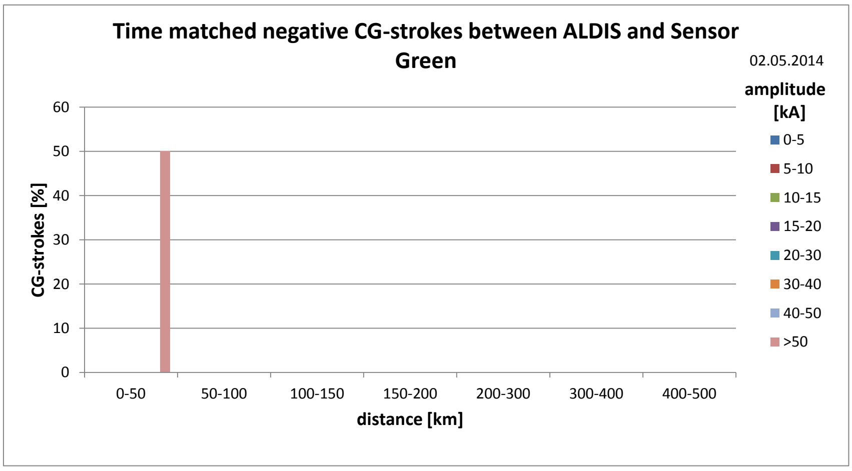


Fig. 5.5.: Time matched negative CG-strokes of 'Sensor Green' and ALDIS on 02.05.2014 as a function of distance and peak current

Due to the very low gain, the sensor only reports the high-current strokes at relatively small distances. Out of the 1275 negative Cloud-Ground strokes only one out of two high-current strokes with a peak current of 95 kA is reported in a distance of 32.5 km. On the other hand the sensor reports 36 positive Cloud-Ground strokes, but only high-current ones. Figure 5.6 shows the signal of a positive Cloud-Ground stroke with a peak current of 86 kA in a range of 3.2 km, that the two H-field antennas received. The sensor is in saturation and the signal is not completely recorded.

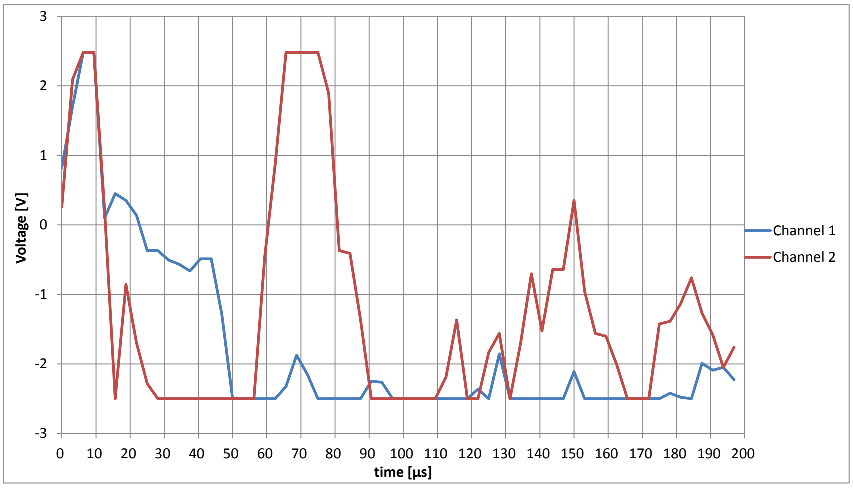


Fig. 5.6.: Received signal of a positive CG-stroke ($I_P = 86$ kA) by the H-field antennas

5.2. Sensor Red

The following discussion of the results of the analysis of 'Sensor Red' is based on several recordings, which are listed in table 4.2. The two modes of operation, automatic mode and manual mode, are analysed separately.

5.2.1. Automatic mode

The automatic setting of the gain and the threshold of the amplifiers by the computing server of Blitzortung.org is at an early beta stage. Below, the data recordings of 13.07.2014 and 20.07.2014 with the automatic gain setting are discussed.

The recording of 13.07.2014 was performed while a storm was passing over the sensor site and covered a time interval of two hours from 7.00 am to 9.00 am (UTC). The storm took about half an hour to pass over the sensor site. The sensor was placed inside the house in a rural area. Figure 5.7 shows the lightning stroke map of the recording according to ALDIS and figure 5.8 shows the lightning stroke map of Blitzortung.org up to a distance of 1000 km to the sensor site (red house). The thunderstorm activity in the whole distance range during the recording was rather low. There was a small thunderstorm cell nearby the sensor site up to a distance of 100 km, but the most active thunderstorm cell was at a distance between 600 and 700 km at the Adriatic Sea.

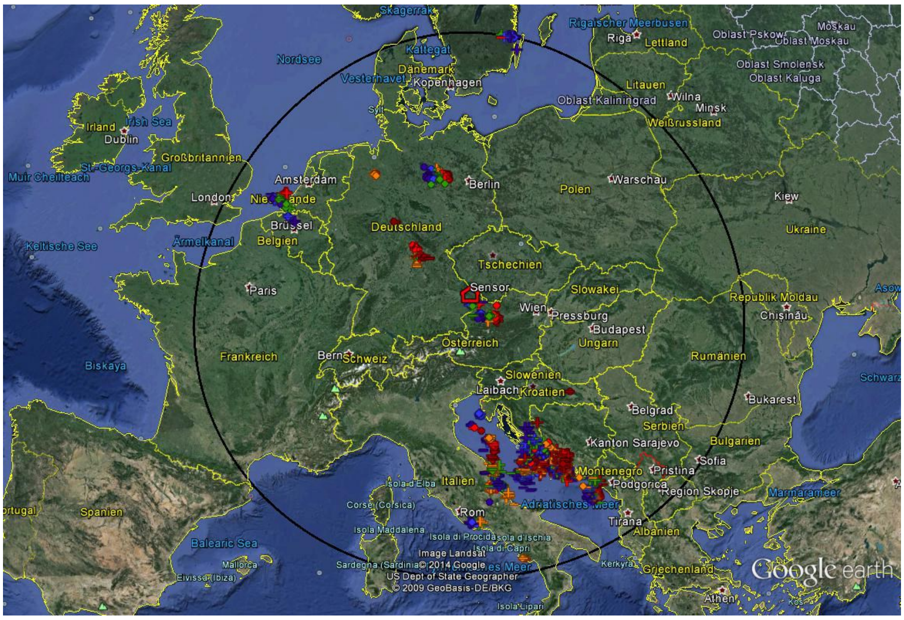


Fig. 5.7.: Lightning stroke map of the data recording period on 13.07.2014 by ALDIS red house indicates sensor site; area radius is 1000 km; $N_{ALDIS} = 6017$

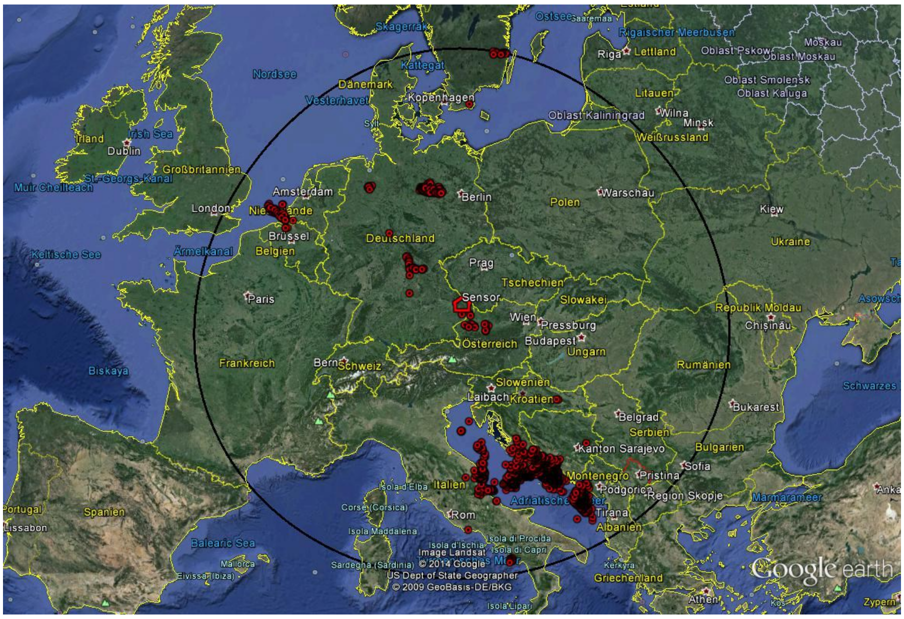


Fig. 5.8.: Lightning stroke map of the data recording period on 13.07.2014 by Blitzortung.org
red house indicates sensor site; area radius is 1000 km; $N_{BO} = 3653$

During the time interval 07:00 - 09:00 (UTC) ALDIS has detected 6017 lightning strokes within a distance of 1000 km to the sensor site and Blitzortung.org has detected 3653 strokes. The number of the time matched strokes between the sensor and ALDIS is 332 and between Blitzortung.org and ALDIS is 1119. Figure 5.9 shows the number of all strokes located by ALDIS and the number of time matched strokes of 'Sensor Red' and of Blitzortung.org, according to the ALDIS classification as Cloud-Cloud (CC), positive Cloud-Ground (CG+) and negative Cloud-Ground (CG-) strokes.

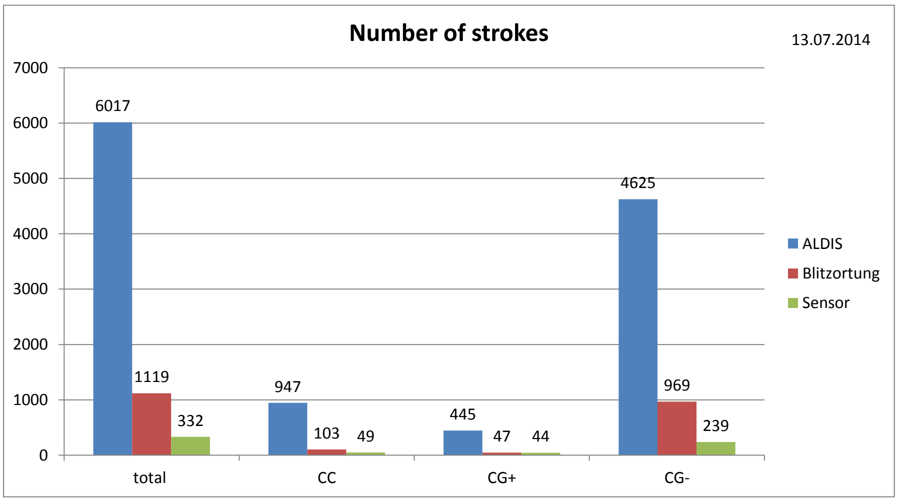


Fig. 5.9.: Number of located strokes on 13.07.2014
ALDIS: All strokes located by ALDIS
Blitzortung: Time matched strokes of Blitzortung.org and ALDIS
Sensor: Time matched stroke reports of 'Sensor Red' and ALDIS

Figure 5.10 shows all negative Cloud-Ground-strokes that are located by ALDIS. Most located strokes are low-current ones with a peak current between 10 and 15 kA. At a range of 600 to 700 km to the sensor most strokes are located and at the range of 150 to 200 km no stroke is located.

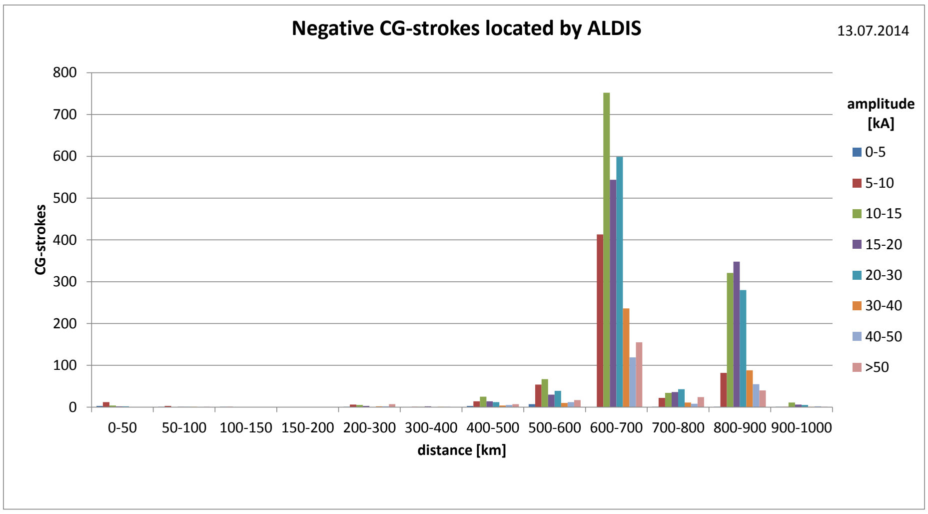


Fig. 5.10.: Negative CG-strokes detected by ALDIS on 13.07.2014 as a function of distance and peak current

Figure 5.11 shows the number of the time matched negative Cloud-Ground strokes of 'Sensor Red' in percent of all strokes located by ALDIS as a function of distance. Due to the low gain and the low thunderstorm activity within a range of up to 50 km the sensor reports about 70 % of all strokes in this range, which is 16 strokes out of 23 strokes. With increasing distance the locating frequency in percent decreases. The detection range of the sensor reaches to a distance of 900 km, because of the long range tuning of the system [8].

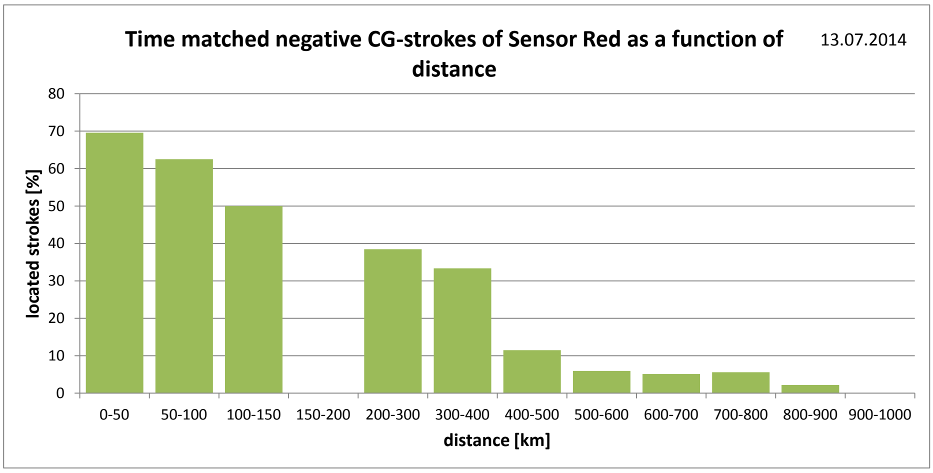


Fig. 5.11.: Time matched negative CG-strokes of 'Sensor Red' on 13.07.2014 as a function of distance
Between 150 and 200 km no stroke is located by ALDIS

Figure 5.12 shows the number of the time matched negative Cloud-Ground strokes of 'Sensor Red' in percent of all strokes located by ALDIS as a function of distance of the recording of 20.07.2014 (see table 4.2). Due to the higher thunderstorm activity at a distance up to 50 km from the sensor site the location performance is decreasing in this range. Out of the 155 negative Cloud-Ground strokes within this range, the sensor reports 49. During the recording time of 40 minutes ALDIS has located 6757 Cloud-Cloud and Cloud-Ground strokes up to a distance of 200 km. Due to the high thunderstorm activity at this distance range the gain is set very low by the server of Blitzortung.org (automatic mode), and therefore the detection range is decreasing to 600 km. At a distance between 200 and 400 km ALDIS has located no strokes. Figure 5.13 shows the lightning map of this recording period, based on the strokes located by ALDIS.

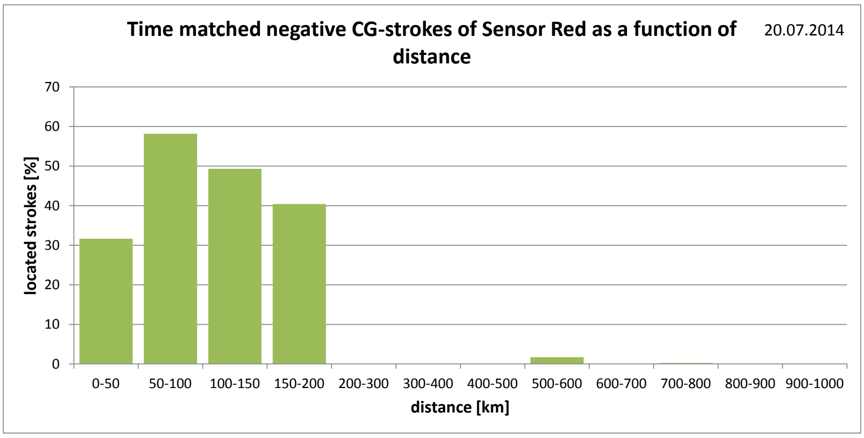


Fig. 5.12.: Time matched negative CG-strokes of 'Sensor Red' on 20.07.2014 as a function of distance
Between 200 and 400 km no stroke is located by ALDIS

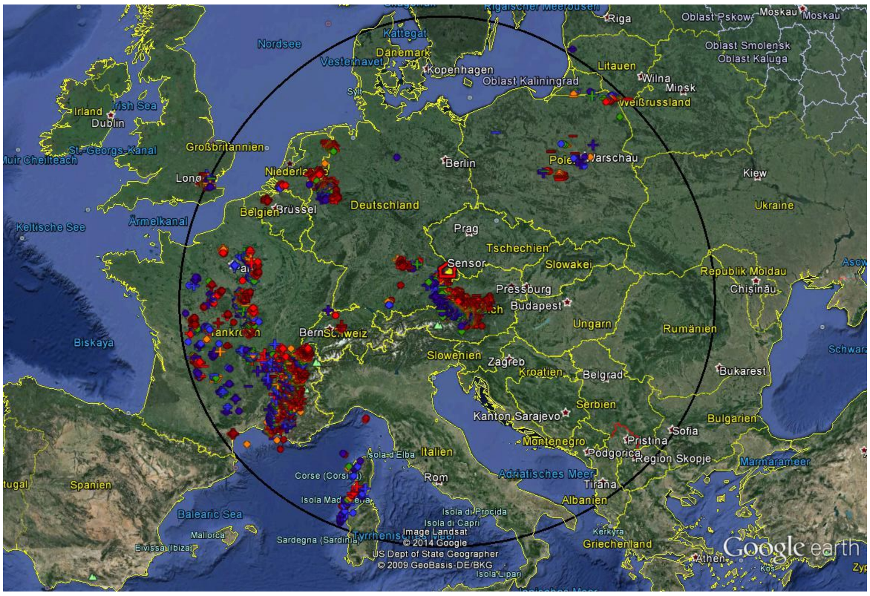


Fig. 5.13.: Lightning stroke map of the data recording period on 20.07.2014 by ALDIS red house indicates sensor site; area radius is 1000 km; $N_{ALDIS} = 11054$

Figure 5.14 shows the number of time matched negative Cloud-Ground strokes of 'Sensor Red' as a function of distance and peak current of the recording of 13.07.2014. At a distance above 500 km mainly the high-current strokes are reported. Within a distance of 100 km from the sensor site almost every stroke with peak current between 15 and 40 kA is reported. At a distance of 150 to 200 km no stroke is located by ALDIS.

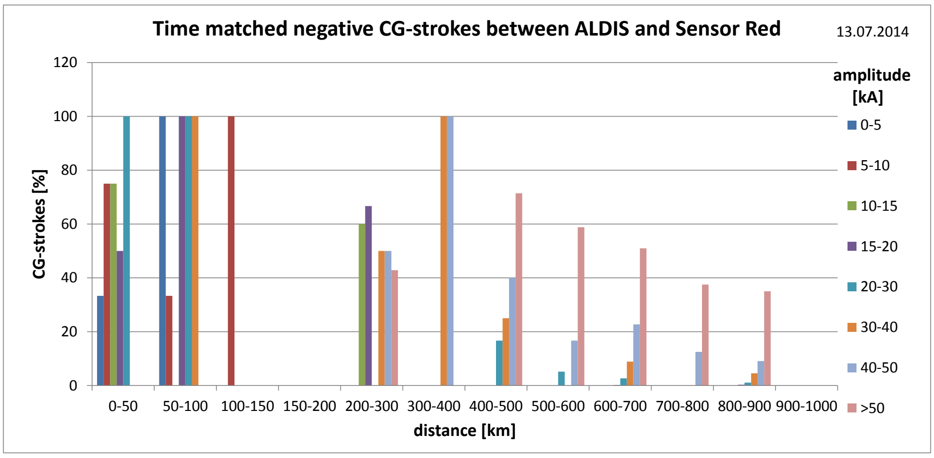


Fig. 5.14.: Time matched negative CG-strokes of 'Sensor Red' on 13.07.2014 as a function of distance and peak current

5.2.2. Manual Mode

In the manual mode the gain and the threshold of the amplifier are set manually by the station operator. The range of gain and threshold settings is so wide, that in this thesis the gain adjustment is divided up into three rough classes: low, medium and high gain. The gain and threshold are set in a manner, that the ratio of stroke signals to noise is good. If the amount of received signals per second exceeds the pre-set value of 20 signals per second, the sensor goes into interference mode. The sensor is adjusted, so that about 4 signals per second are received. The manual gain setting can possibly last several days in order to find an optimal setting for the respective sensor site. Table 5.1 gives an overview of the gain and threshold settings of three different data recordings.

Tab. 5.1.: Data recordings of 'Sensor Red' with manual gain setting

date	**time (UTC)**	**gain channel 1**	**threshold channel 1**	**gain channel 2**	**threshold Channel 2**	**class**
09.09.2014	19:47:28 - 21:44:48	10240	400 mV	10240	725 mV	high
11.09.2014	13:32:15 - 16:40:47	640	60 mV	400	75 mV	low
22.09.2014	15:10:02 - 18:30:01	5120	300 mV	6400	700 mV	medium

Table 5.2 shows the results of the analysis of the recordings explained in table 5.1. The total number of strokes located by ALDIS (N_{ALDIS}), the total number of strokes located by Blitzortung.org (N_{BO}) and the time matched strokes between 'Sensor Red' and ALDIS classified according to ALDIS as Cloud-Cloud strokes, positive and negative Cloud-Ground strokes are displayed. The time matched strokes are displayed in absolute numbers and in percent to all strokes located by ALDIS. A zone of 500 km

around the sensor is regarded. The different thunderstorm activity during the three different recordings renders a direct comparison not possible. But the general impact of the gain settings on the location performance can still be seen.

Tab. 5.2.: Results of analysis of the recordings with manual gain setting

date	gain	N_{ALDIS}	N_{BO}	CC	CC [%]	CG+	CG+ [%]	CG-	CG- [%]	Total [%]
09.09.2014	high	6150	862	452	12,6	111	26,9	320	14,8	14,4
11.09.2014	low	4524	1040	140	4,8	54	27,0	133	9,6	7,2
22.09.2014	medium	9121	1798	416	6,6	181	36,9	473	20,1	11,7

Regarding all time matched strokes, the high-gain recording is the most effective, but regarding only the negative Cloud-Ground strokes, for which the antennas are designed for, the medium-gain recording is best. Figure 5.15 shows the time matched negative CG-strokes as a function of distance. At a distance up to 50 km in the recording of 09.09.2014 and 11.09.2014, no strokes are located by ALDIS. As a result of the low-gain setting the sensor operates best up to 200 km, but above 300 km the detection efficiency decreases dramatically. In contrast the high gain setting causes a poor sensor performance nearby the sensor up to a distance of 100 km, but it also causes a better sensor performance at a distance above 300 km. The medium gain setting causes ordinary detection efficiency nearby the sensor as well as above a distance of 300 km.

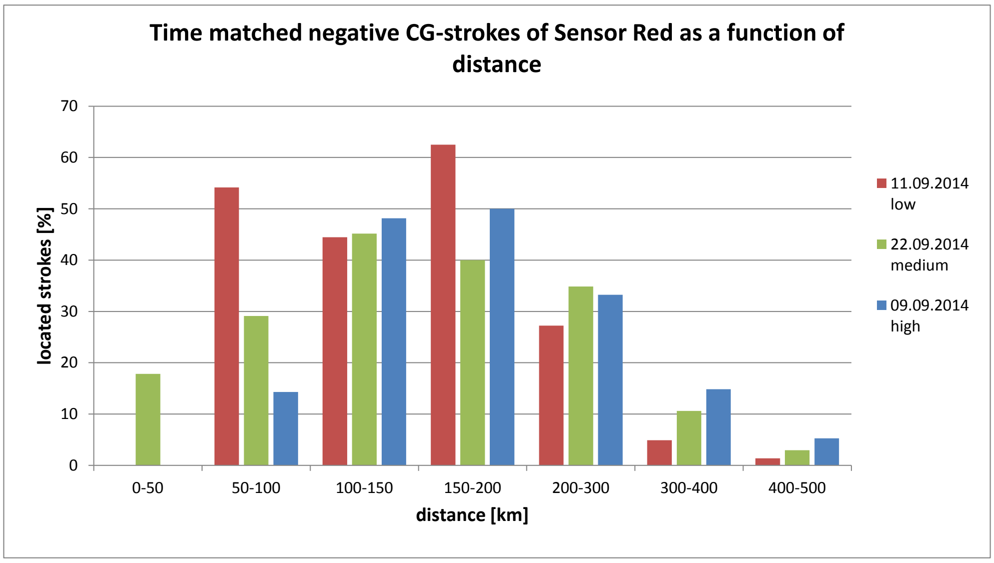


Fig. 5.15.: Time matched negative CG-strokes of 'Sensor Red' as a function of distance of the recordings with manual gain

Next, the medium gain recording of 22.09.2014 (see table 4.2) is analysed in detail up to a distance of 1000 km around the sensor site. During recording, which covered about three hours, an active thunder cell was nearby the sensor. The sensor was placed in the attic of a house in a rural area. The thunderstorm activity over the whole distance range during the recording was medium. Figure 5.16 shows the lightning stroke map of the recording up to a distance of 1000 km to the sensor site (red house) according to ALDIS.

During that time interval ALDIS has detected 10377 lightning strokes within a distance of 1000 km to the sensor site and Blitzortung.org has detected 2543 strokes. The number of time matched strokes between 'Sensor Red' and ALDIS is 1226 and the number of time matched strokes between Blitzortung.org and ALDIS is 614. Figure 5.17 shows the number of all strokes located by ALDIS and the number of time matched strokes of 'Sensor Red' and of Blitzortung.org, according to the ALDIS classification as Cloud-Cloud (CC), positive Cloud-Ground (CG+) and negative Cloud-Ground (CG−) strokes.

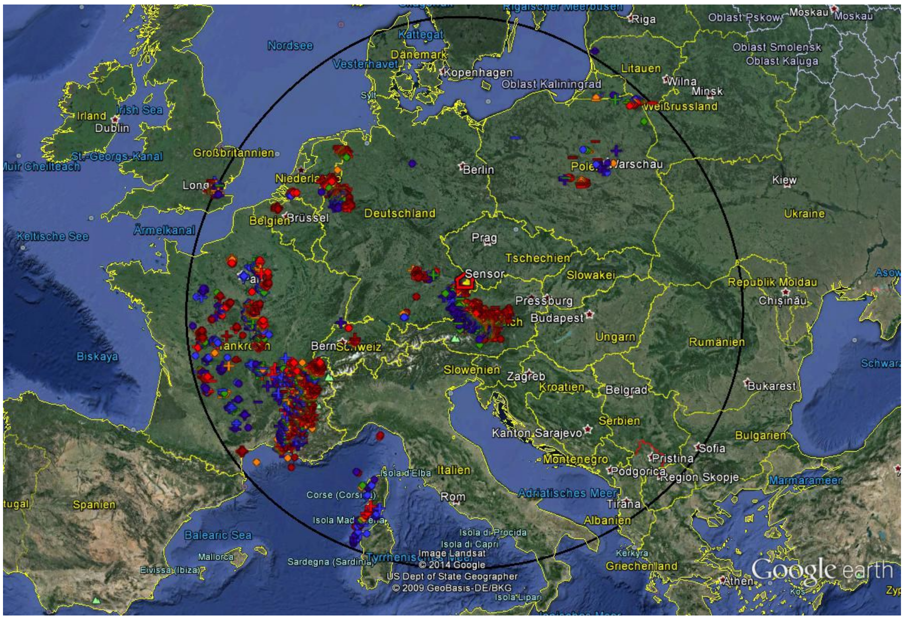


Fig. 5.16.: Lightning stroke map of the data recording period on 22.09.2014 by ALDIS red house indicates sensor site; area radius is 1000 km; $N_{ALDIS} = 10377$

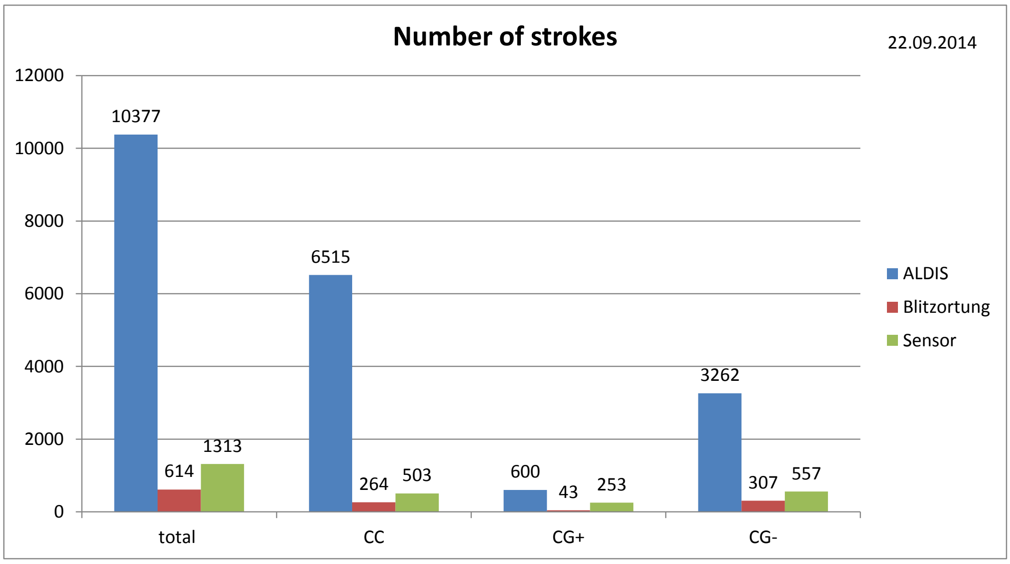


Fig. 5.17.: Number of located strokes on 22.09.2014
ALDIS: All strokes located by ALDIS
Blitzortung: Time matched strokes of Blitzortung.org and ALDIS
Sensor: Time matched stroke reports of 'Sensor Red' and ALDIS

Figure 5.18 shows the negative Cloud-Ground-strokes that are located by ALDIS. Most located strokes are low-current ones with a peak current between 5 and 10 kA. Most strokes are located in a range from 300 to 400 km to the sensor .

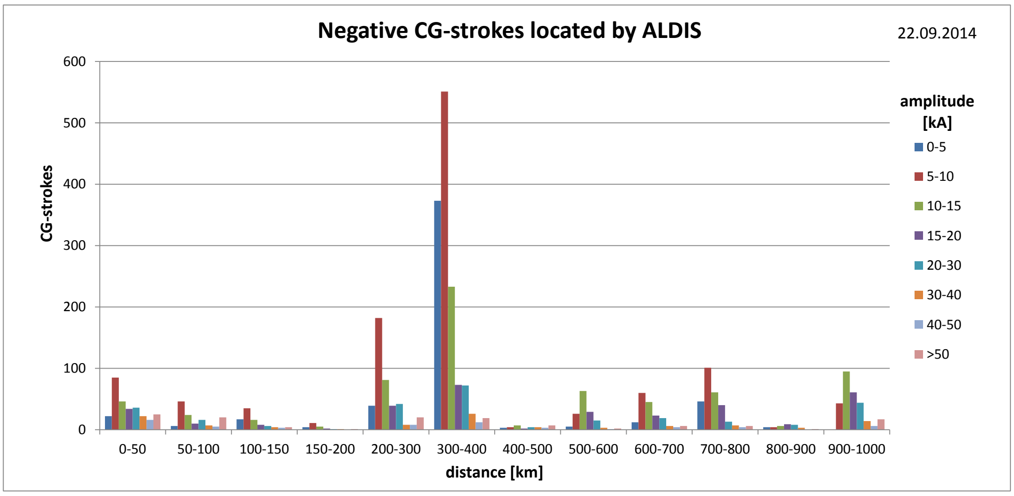


Fig. 5.18.: Negative CG-strokes detected by ALDIS on 22.09.2014

Figure 5.19 shows the number of the time matched negative Cloud-Ground strokes of 'Sensor Red' in percent of all ALDIS strokes as a function of distance. Due to the medium-gain setting and the medium thunderstorm activity within a range of 50 km, the sensor reports only about 18 % of the strokes in this range. The sensor performance increases up to a distance of 150 km, but above it first decreases slowly and above 300 km sharply. It is astonishing that above a distance of 500 km the sensor performance increases again. The detection range of the sensor reaches to a distance of 900 km, because of the long-range tuning of the system. Up to a distance of 500 km mainly the groundwave of the very low frequency signals, emitted by the lightning discharge, is received, but above mainly the ionospherical reflection is received. The delay time between ground wave and sky wave depends on the range; with increasing range the delay time decreases [6].

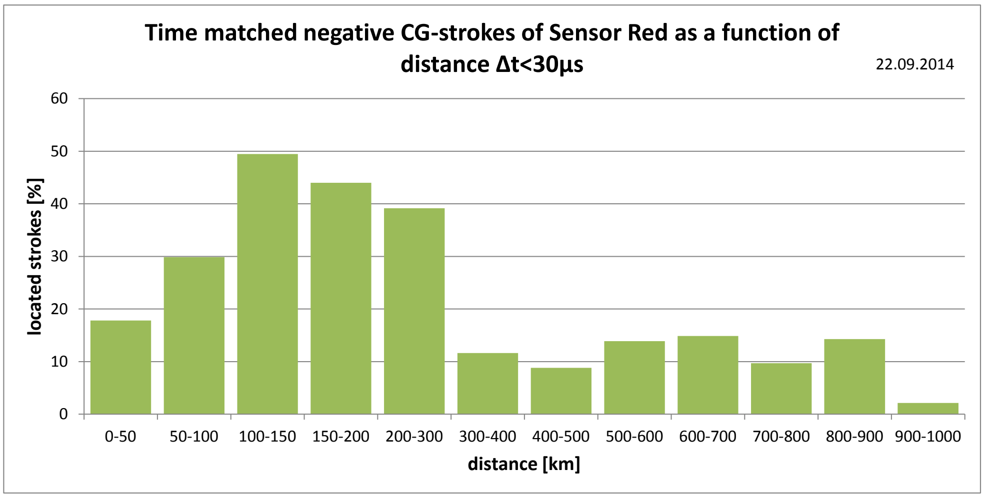


Fig. 5.19.: Time matched negative CG-strokes of 'Sensor Red' on 22.09.2014 as a function of distance

Figure 5.20 shows the number of time matched negative Cloud-Ground strokes of 'Sensor Red' as a function of distance for various time differences Δt that are allowed to determine the time matched events. When comparing the lightning data of 'Sensor Red' with the data of ALDIS, the time difference to match both data is varied. With increasing time difference the number of strokes at a distance above 500 km is increasing due to the detection of the sky wave. The number of strokes within 500 km is almost constant because in this range mainly the ground wave is reported.

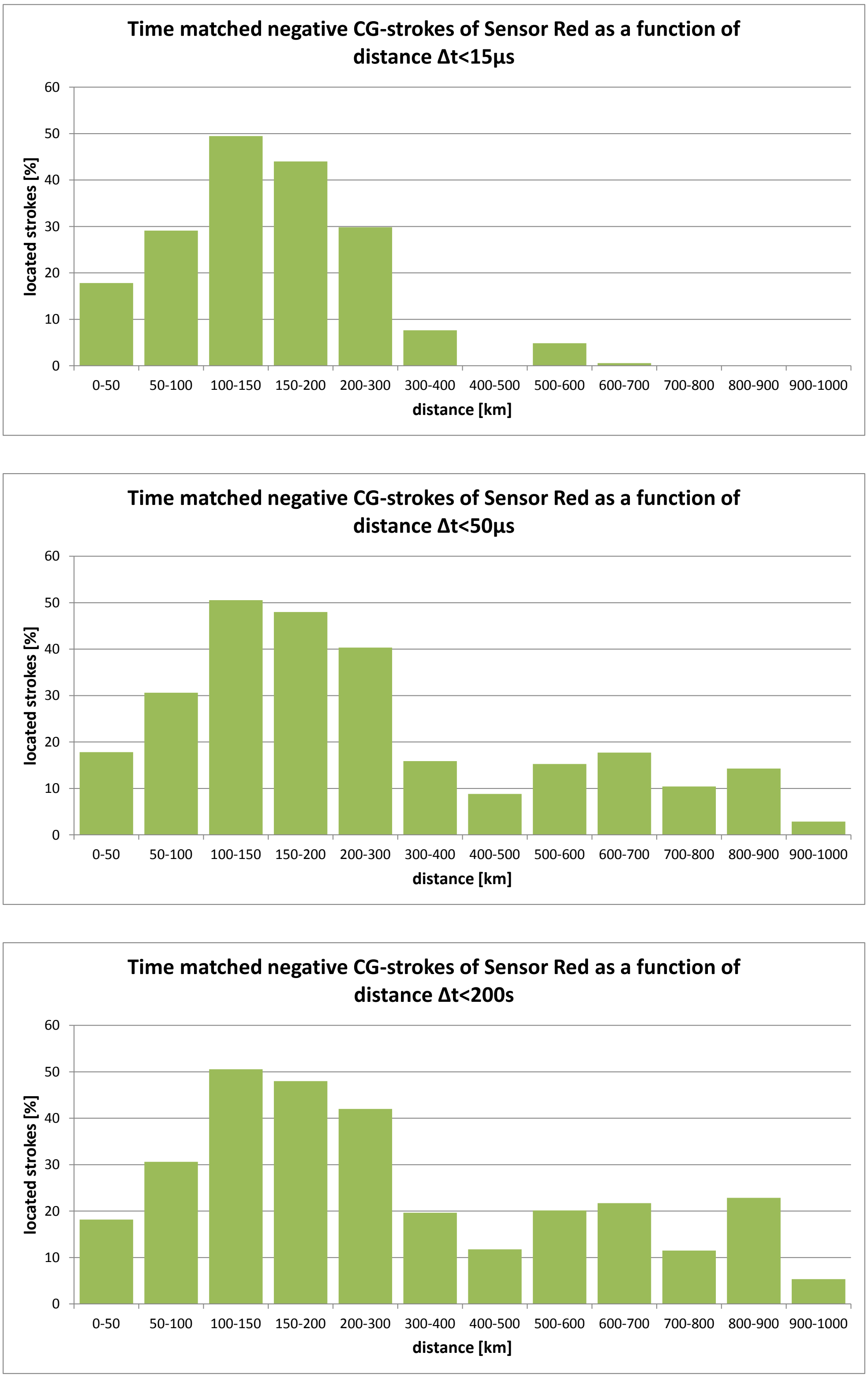


Fig. 5.20.: Effect of the allowed time difference to determine time matched strokes

Figure 5.21 shows the number of time matched negative Cloud-Ground strokes of the sensor as a function of distance and peak current of the analysis with a time difference of 30 μs. At a distance above 500 km mainly the high-current strokes are reported with a good efficiency up to 900 km. Nearby the sensor, up to a distance of 200 km, mainly low-current strokes are reported; out of the low-current strokes with a peak current between 5 and 10 kA almost 50 % are reported by the sensor. Within a distance of 500 km the best detection range is from 50 km to 200 km and above the best range is up to 900 km.

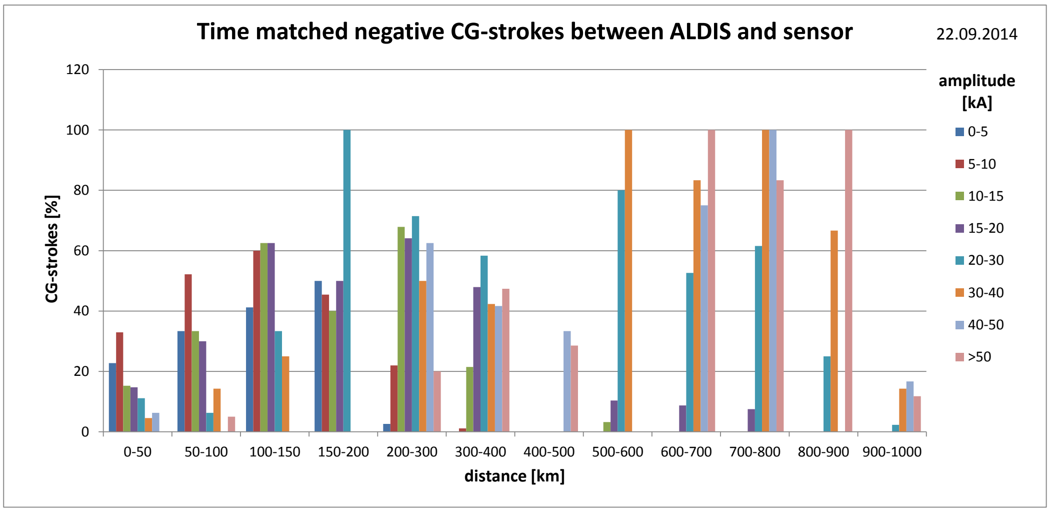


Fig. 5.21.: Time matched negative CG-strokes of 'Sensor Red' on 22.09.2014 as a function of distance and peak current

5.3. Blitzortung.org versus ALDIS

The computed lightning data of Blitzortung.org was compared with the computed lightning data of ALDIS of the Austrian area on a day with high thunderstorm activity and thus with a high number of lightning discharges. A circular area with a radius of 250 km centered at Niederöblarn (latitude 47.47517°, longitude 14.01998°) was observed on 03.08.2014 for the whole day (24 h). Figure 5.22 shows the lightning stroke map according to the strokes located by ALDIS and figure 5.23 shows the lightning stroke map according to the strokes located by Blitzortung.org.

On this day ALDIS has detected 77087 lightning strokes and Blitzortung.org has detected 19914 lightning strokes. The number of time matched strokes between Blitzortung.org and ALDIS is 8575. This means that Blitzortung.org has detected about 26 % of the number of lightning strokes located by ALDIS, and out of this

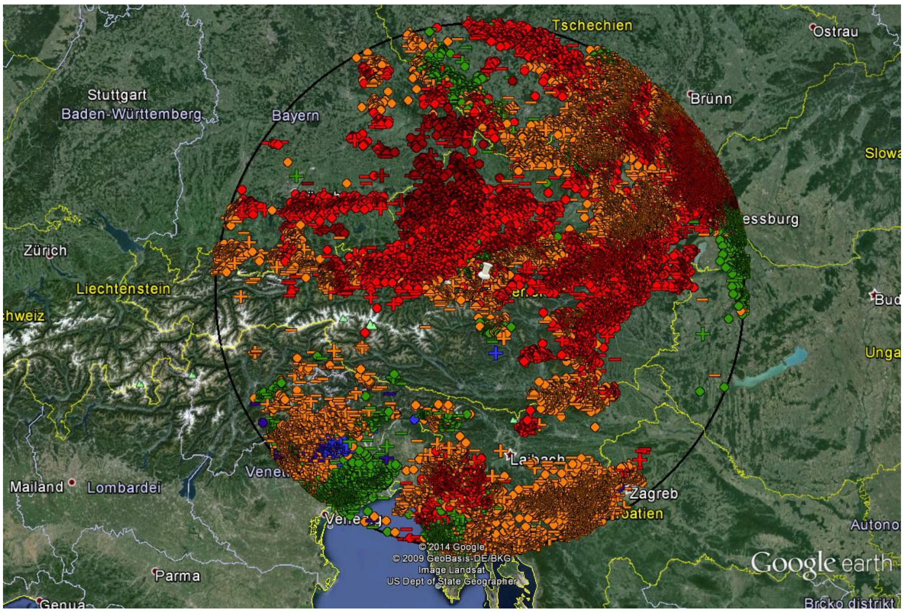


Fig. 5.22.: Lightning stroke map on 03.08.2014 by ALDIS
White pin indicates centre; area radius is 250 km; $N_{ALDIS} = 77087$

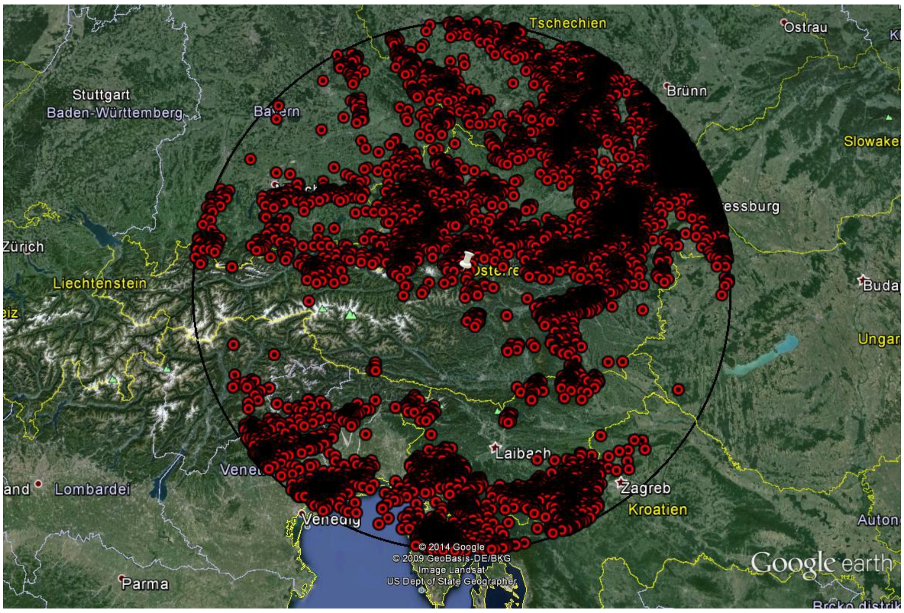


Fig. 5.23.: Lightning stroke map on 03.08.2014 by Blitzortung
White pin indicates centre; area radius is 250 km; $N_{BO} = 19914$

percentage about 42 % are time matched events. Consequently, Blitzortung.org has detected only 11 % of the 77087 strokes detected by ALDIS. Figure 5.24 shows the number of all strokes located by ALDIS and the number of time matched strokes of Blitzortung.org, according to the ALDIS classification as Cloud-Cloud (CC), positive Cloud-Ground (CG+) and negative Cloud-Ground (CG-) strokes.

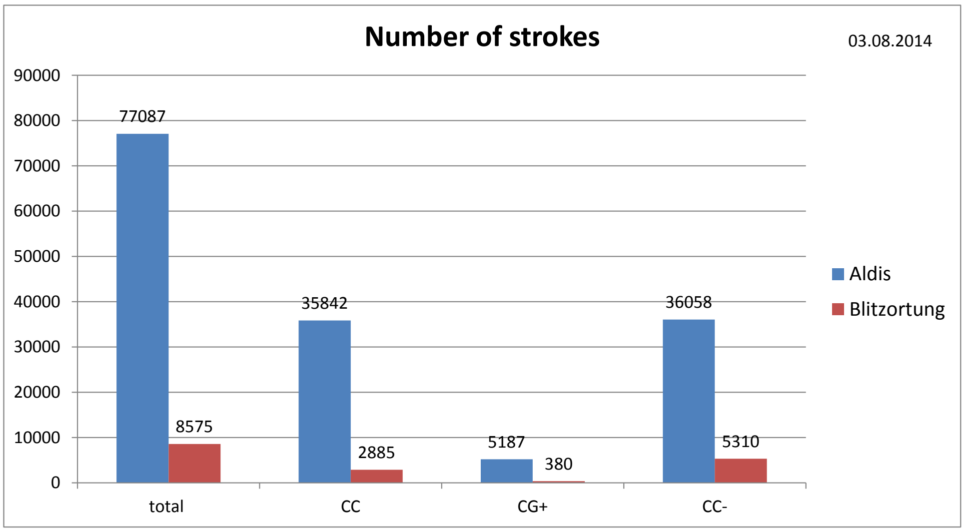


Fig. 5.24.: Number of located strokes on 03.08.2014
ALDIS: All strokes located by ALDIS
Blitzortung: Time matched strokes of Blitzortung.org and ALDIS

Figure 5.25 shows all negative CG-strokes located by ALDIS as a function of distance and peak current and figure 5.26 shows the time matched negative CG-strokes between Blitzortung.org and ALDIS as a function of distance and peak current. Blitzortung.org does not classify the strokes into Cloud-Cloud or Cloud-Ground ones and does not provide peak current information. Therefore, the time matched strokes are grouped according to the ALDIS classification.

In general, the characteristics of the detection of ALDIS and Blitzortung.org look similar, except for the range between 200 and 250 km. In this range, most strokes located by ALDIS are low current strokes with a peak current between 5 and 10 kA. In contrast Blitzortung.org located most strokes with a peak current between 10 and 15 kA. Regarding the time matched strokes between Blitzortung.org and ALDIS as a percentage of all strokes located by ALDIS, the 5 to 20 kA strokes are most located by Blitzortung.org over the whole area. Within a distance of up to 50 km to the centre of the observed area the location performance is much lower than in the rest of the observed area. The general increase of the detection efficiency versus distance is a result of the increasing area with distance. Figure 5.27 shows the number of time matched negative Cloud-Ground strokes between Blitzortung.org and ALDIS as a function of distance and peak current in percent of all negative Cloud-Ground strokes located by ALDIS.

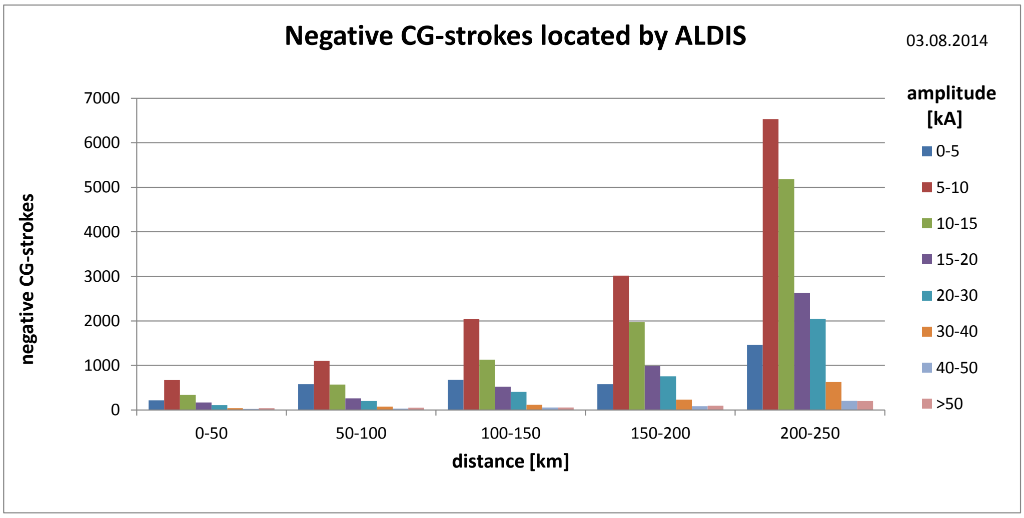


Fig. 5.25.: All negative CG-strokes detected by ALDIS on 03.08.2014 as a function of distance and peak current

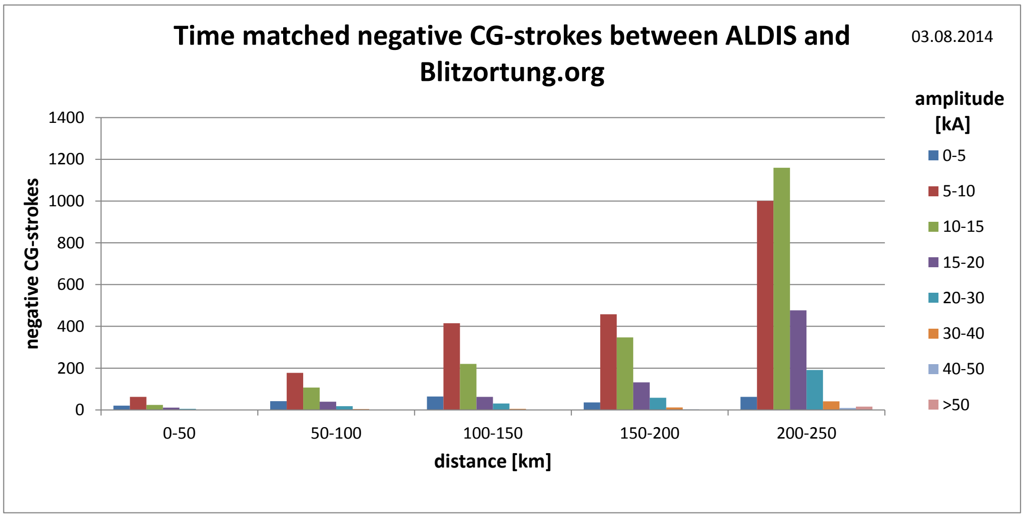


Fig. 5.26.: Time matched negative CG-strokes of Blitzortung.org and ALDIS on 03.08.2014 as a function of distance and peak current

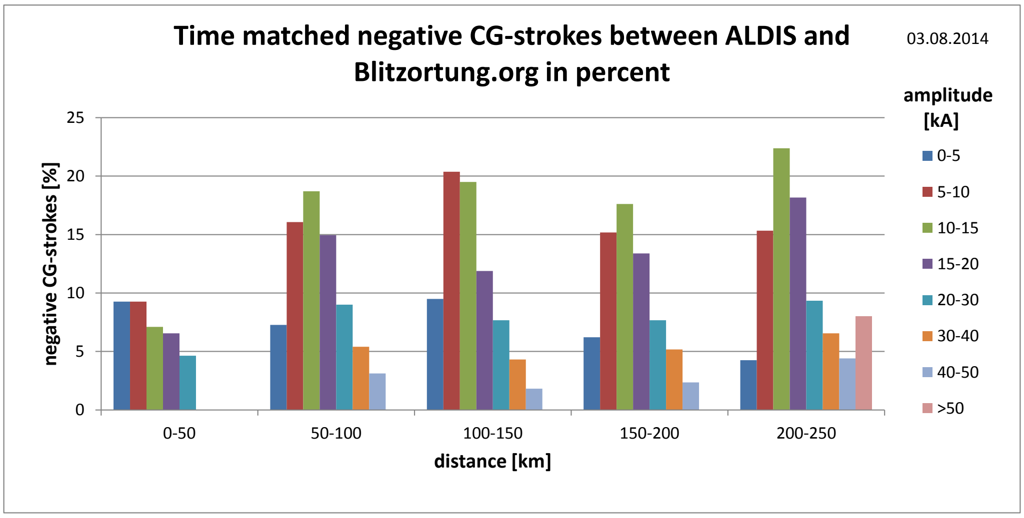


Fig. 5.27.: Time matched negative CG-strokes (in percent) of Blitzortung.org and ALDIS on 03.08.2014 as a function of distance and peak current

6. Summary and Discussion

As the main task of this thesis the most recent version ('Sensor Red') of a low-cost lightning detection sensor offered by the Blitzortung.org community was assembled and tested. The detection efficiency of 'Sensor Red' was analysed as a function of peak current and distance from the sensor site. Recordings with different gain settings were made and their impact on the detection efficiency was analysed.

The gain setting of the sensor has decisive impact on the sensor performance. A low-gain setting results in a good sensor performance up to a distance of 200 km from the sensor site, low-current and high-current strokes are detected with about equal performance, but it also leads to a decrease in detection efficiency at distances of more than 300 km, where mainly high-current strokes are reported. A high-gain setting results in low detection efficiency at distances up to 100 km from the sensor, and leads to an increase of the sensitivity to distant low-current strokes at distances larger than 500 km. A medium-gain setting leads to an increase of the detection efficiency near the sensor up to a distance of 100 km as well as to an increase for distances above 300 km.

In a circular area with a radius of 500 km centred at the sensor site, 'Sensor Red' is able to report up to 15 % of all the strokes located by ALDIS (CC and CG) depending on gain setting and thunderstorm activity. Considering only the negative Cloud-Ground strokes the sensor is able to report - again depending on gain settings and thunderstorm activity - about 20 % of all negative Cloud-Ground strokes located by ALDIS. The sensor performance for low-gain setting is best at a distance between 50 and 200 km and for medium-gain and high-gain setting at a distance between 100 and 300 km. In this case, up to 60 % percent of all negative Cloud-Ground strokes located by ALDIS are reported. Most reported strokes are low-current strokes with peak currents between 5 and 20 kA.

Besides, the detection efficiency of 'Sensor Green' (the previous version) was analysed as a function of peak current and distance from the sensor site. This initial analysis was performed only for recordings with low gain setting in an urban area. In this setting the sensor only reported about 2 % of the strokes located by ALDIS and only high-peak current strokes up to a distance of 100 km to the sensor site. But the low-gain recording and its results of the analysis are not sufficient to adequately evaluate the sensor performance of 'Sensor Green'. A more comprehensive analysis is to be performed during the next thunderstorm season using the sensor and the analysis tools developed and described in this thesis.

The detection efficiency of the Blitzortung.org network was also analysed for a 250 km circular area, which covers the Austrian area on a day with high thunderstorm activity. The number of strokes located by Blitzortung.org is only 26 % of the number of strokes located by ALDIS. The percentage of the time-matched signals with a time difference of 30 μs between Blitzortung.org and ALDIS is about 11 %. This means that Blitzortung.org only locates 11 % of the strokes located by ALDIS. This corresponds to Cummins et al. [1] discussing the detection efficiency of long range VLF systems. Only a small fraction of the lightning discharges is detected by long range VLF systems. Most detected strokes are low-current strokes with peak currents between 5 and 20 kA.

Since the Blitzortung.org network is based on a high but non-uniform distributed number of sensors and the gain can be adjusted either automatically or manually, there is a wide range of possible gain settings of the respective sensors and thus also a considerable range in the expected detection efficiency. The setting of each receiver has an impact on the detection efficiency of the system. In areas with low station coverage this impact is higher than in areas with high station coverage. For example, if a sensor in an area with low station coverage is not adjusted well, it reports less lightning signals or goes into interference mode, and may cause that several strokes are not located. The same sensor in an area of high sensor density may have no effect, as there is sufficient redundancy for appropriate location of lightning discharges in this area.

Overall, the preliminary data analysis showed that there are significant differences when doing a stroke-by-stroke comparison of the Blitzortung.org network and ALDIS.

7. Evaluation and Outlook

In this thesis, the main goal was to assemble the most recent version of a low-cost detection sensor of Blitzortung.org and to perform preliminary test recordings with the most recent version ('Sensor Red') and the previous version ('Sensor Green'). The focus was on the recent version of the detection hardware. The recordings with the old detection hardware were only performed with a low-gain setting. For the future it would be also interesting to analyse the results of recordings with medium-gain and high-gain settings. The newly built sensor has shown its basic functionality and a more detailed sensor analysis and comparison with ALDIS data should be done during an upcoming lightning season in order to run the sensor at various settings and different thunderstorm situations.

Furthermore, recordings with both detection devices should be conducted simultaneously, in order to compare and evaluate both sensors on the basis of the same lightning data. Up to now, the recordings with the new device, 'Sensor Red', were only performed with the magnetic field amplifier because the electric field amplifier is more susceptible to interference and thus the sensor was permanently in interference mode for the selected sensor site in this thesis. For recordings with the electric field amplifier system the effect of the grounding of the system should be evaluated and the recordings should be performed at a place with low interferences, like outside on an open field and not inside a building.

In future studies it is also recommended to analyse the effect of the sensor location (sensor is placed inside or outside a house) and the surrounding (e.g. other metal objects) on the sensor performance.

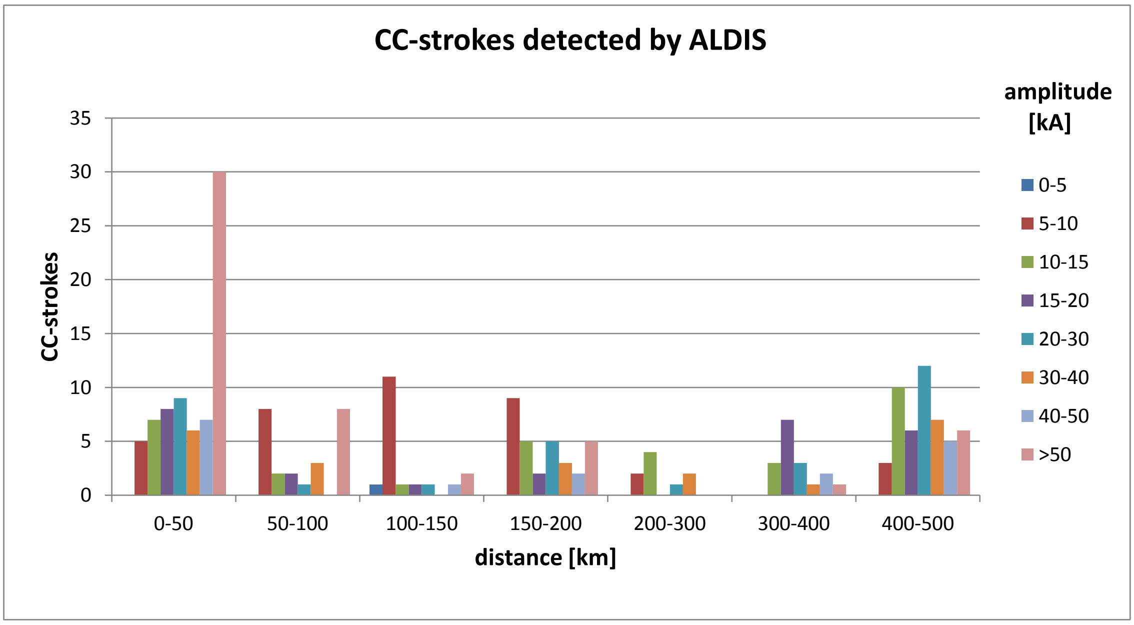
List of Figures

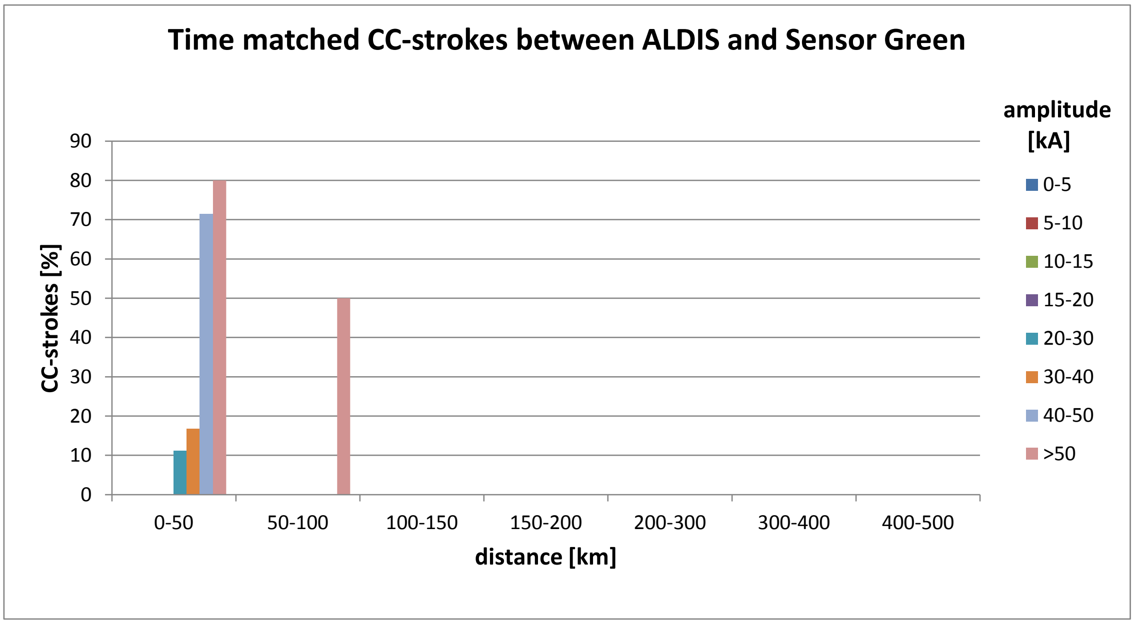
List of Tables

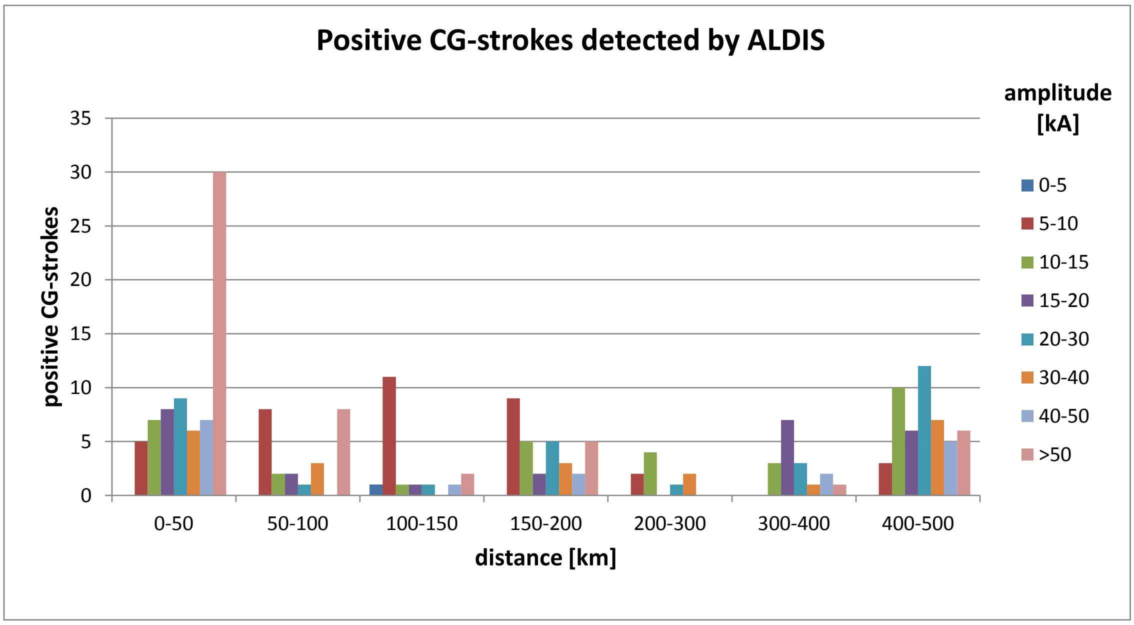
A. Appendix

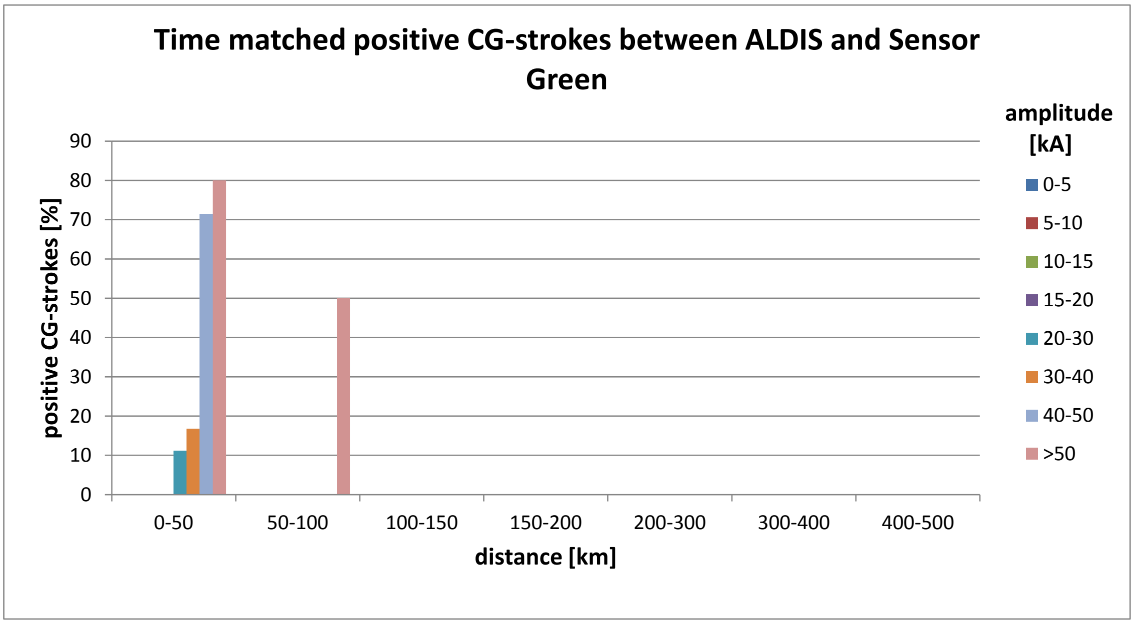
A.1. Results

A.1.1. Sensor Green – 02.05.2014

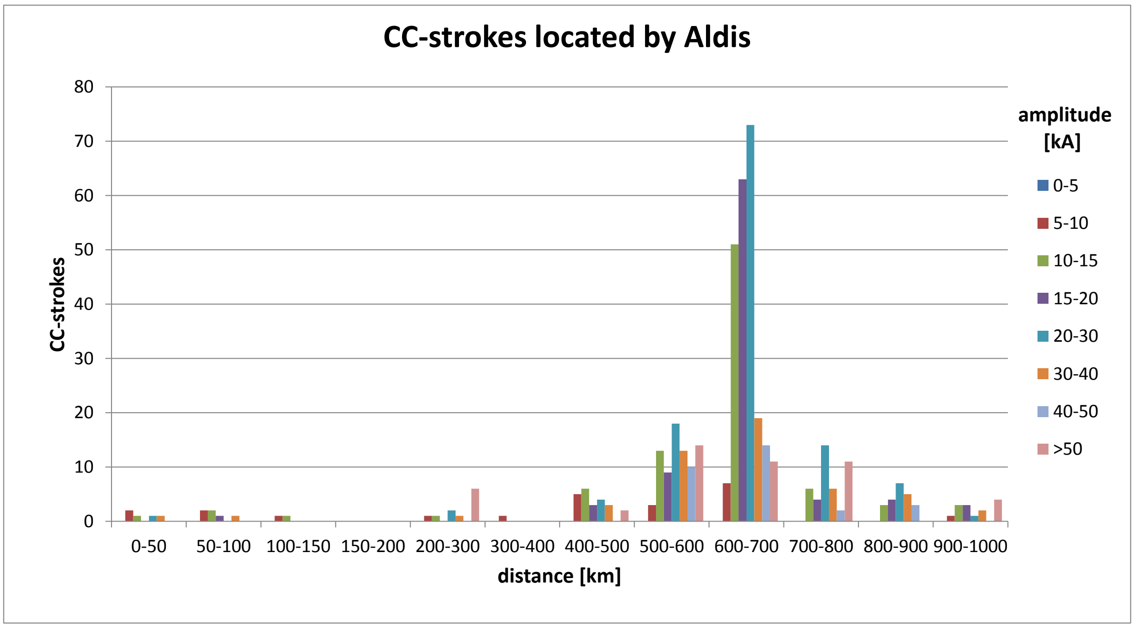


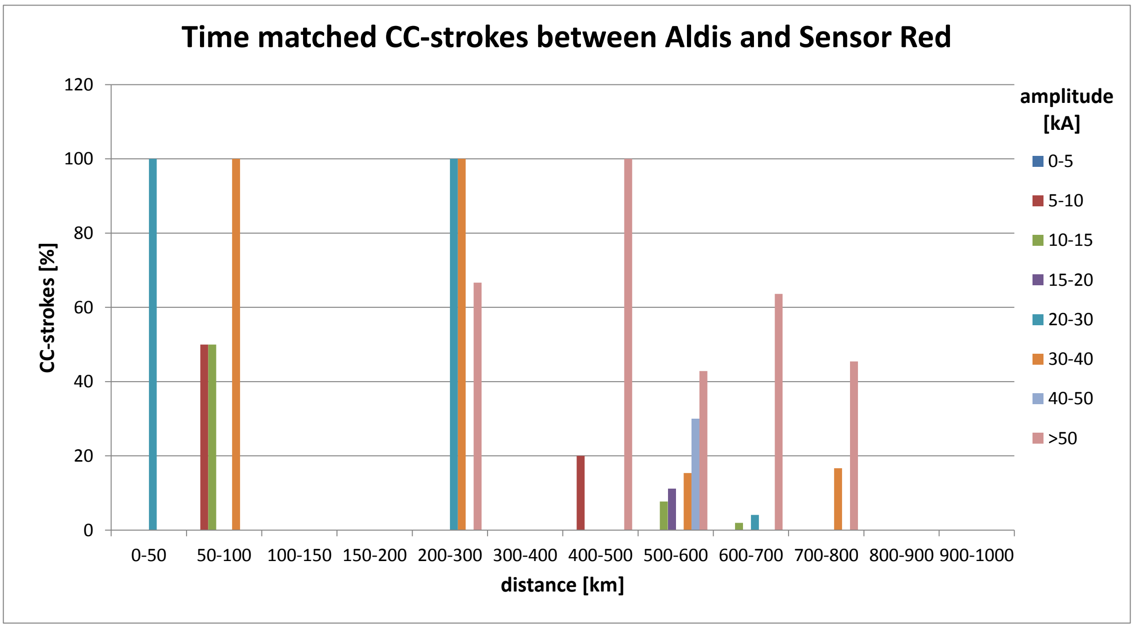

Time matched CC-strokes between ALDIS and Sensor Green
CC-strokes [%]
0
10
20
30
40
50
60
70
80
90
0-50
50-100
100-150
150-200
200-300
300-400
400-500
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

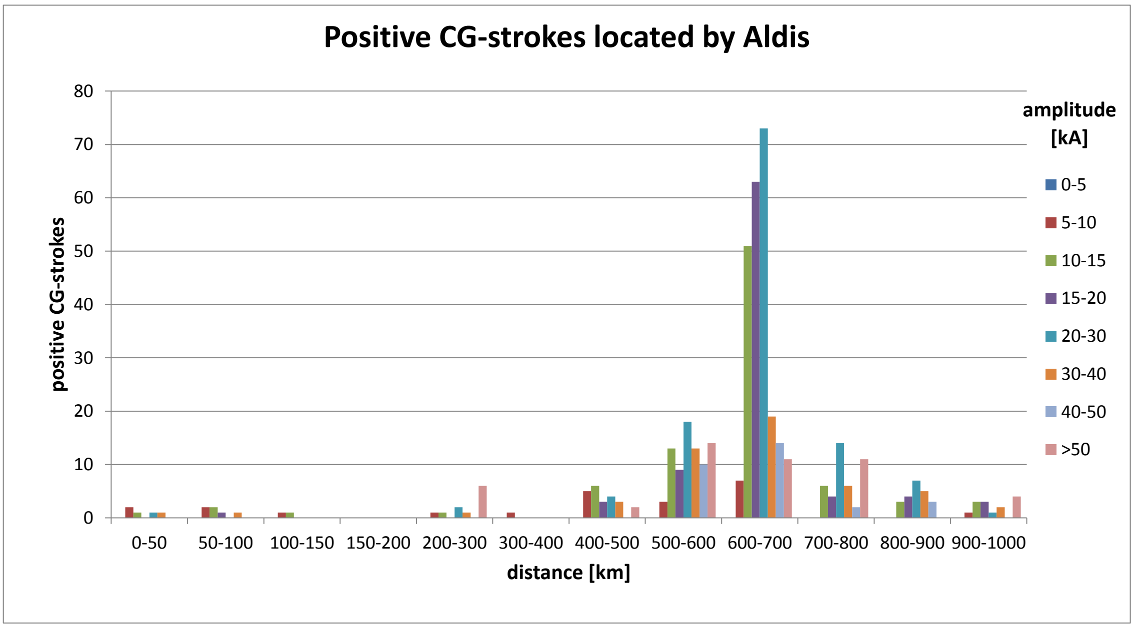

Positive CG-strokes detected by ALDIS
positive CG-strokes
0
5
10
15
20
25
30
35
0-50
50-100
100-150
150-200
200-300
300-400
400-500
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

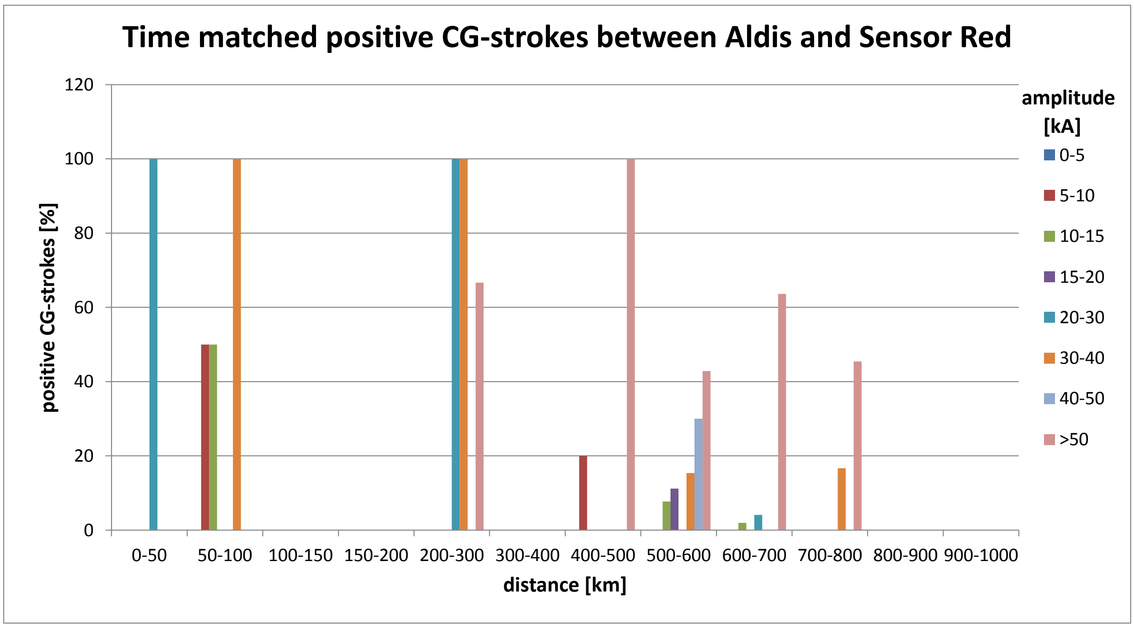


A.1.2. Sensor Red – 13.07.2014

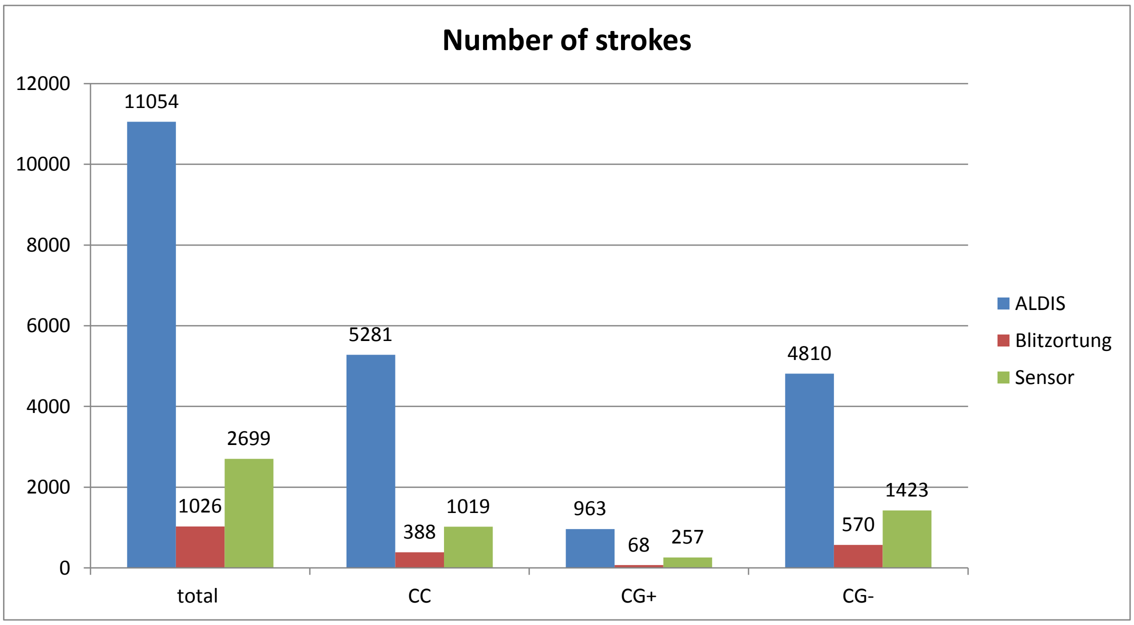


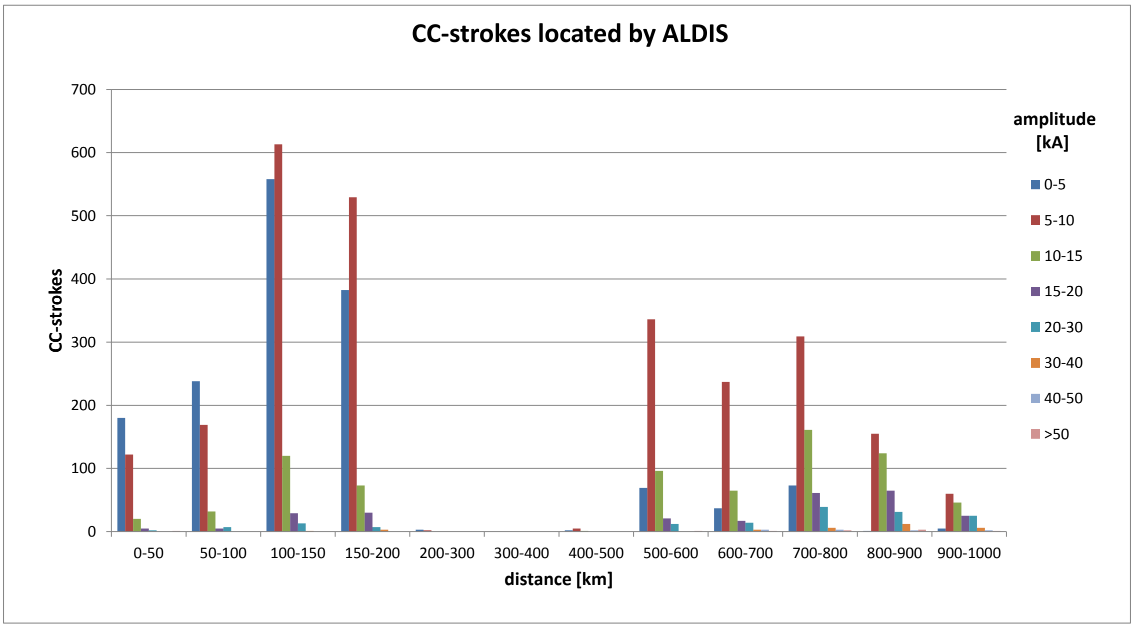

Time matched CC-strokes between Aldis and Sensor Red
CC-strokes [%]
0
20
40
60
80
100
120
0-50
50-100
100-150
150-200
200-300
300-400
400-500
500-600
600-700
700-800
800-900
900-1000
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

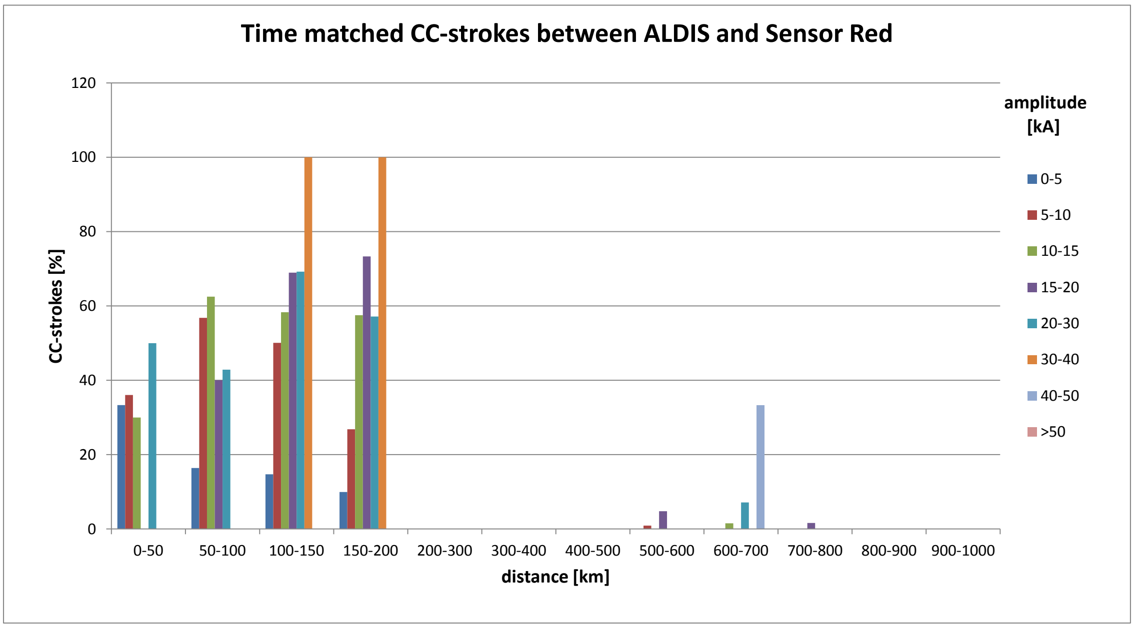

Positive CG-strokes located by Aldis
positive CG-strokes
0
10
20
30
40
50
60
70
80
0-50
50-100
100-150
150-200
200-300
300-400
400-500
500-600
600-700
700-800
800-900
900-1000
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

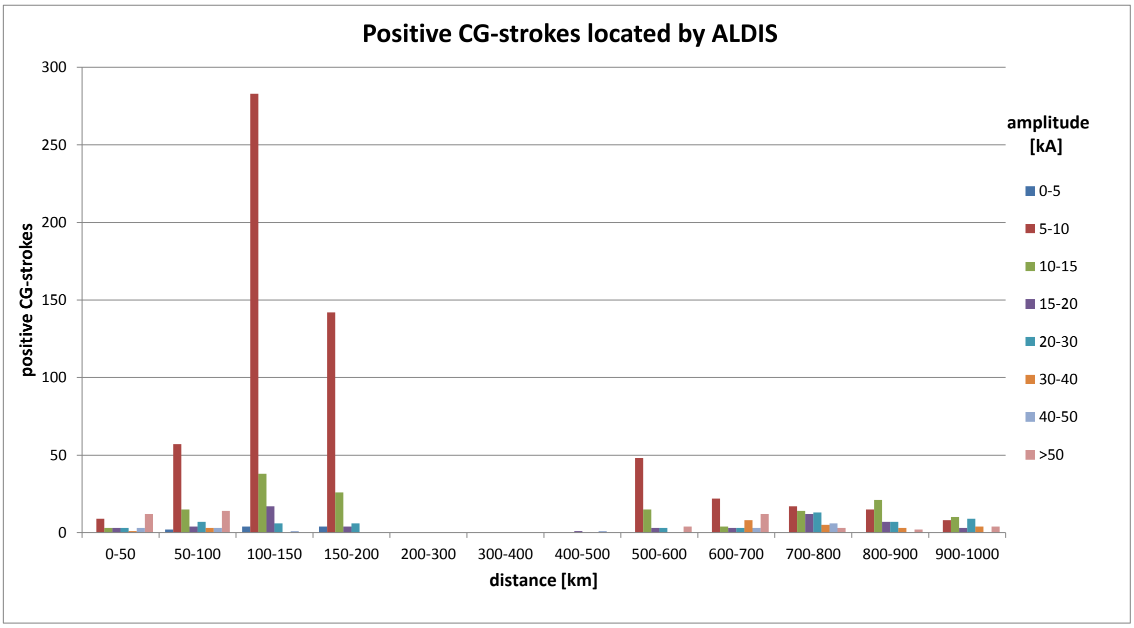


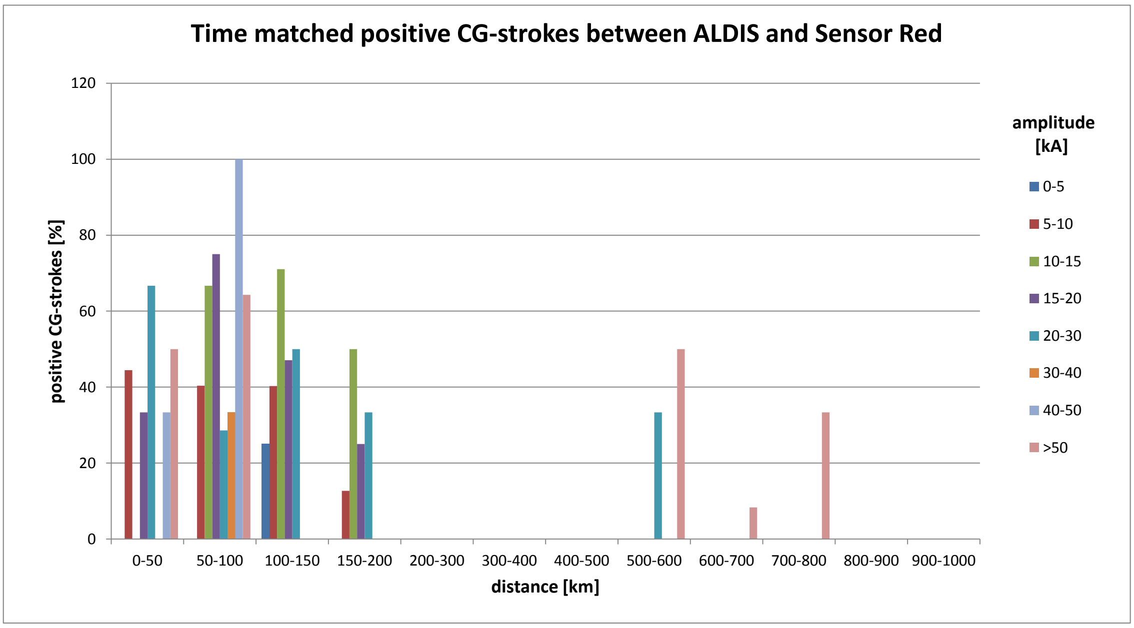
A.1.3. Sensor Red – 20.07.2014



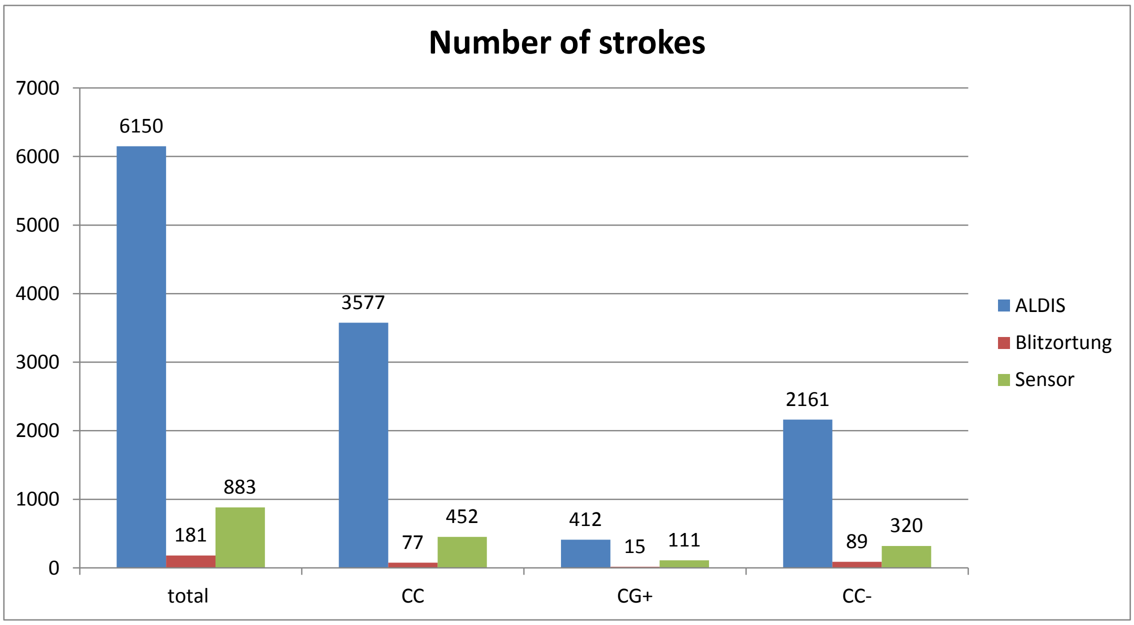

CC-strokes located by ALDIS
700
600
500
400
300
200
100
0
CC-strokes
0-50
50-100
100-150
150-200
200-300
300-400
400-500
500-600
600-700
700-800
800-900
900-1000
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

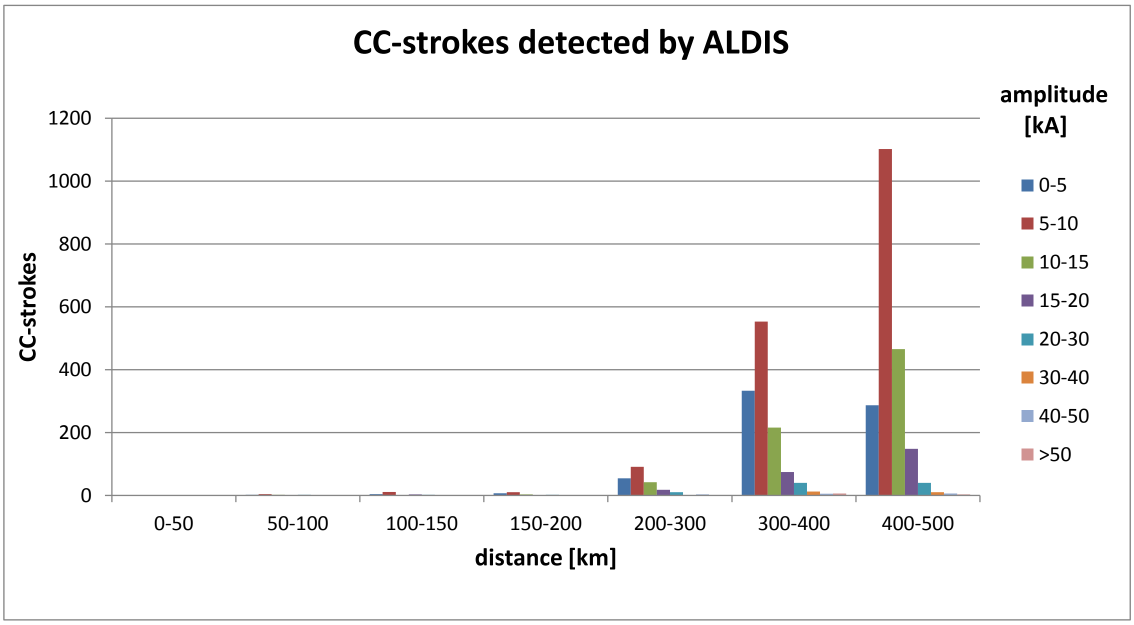

Time matched CC-strokes between ALDIS and Sensor Red
120
100
80
60
40
20
0
CC-strokes [%]
0-50
50-100
100-150
150-200
200-300
300-400
400-500
500-600
600-700
700-800
800-900
900-1000
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

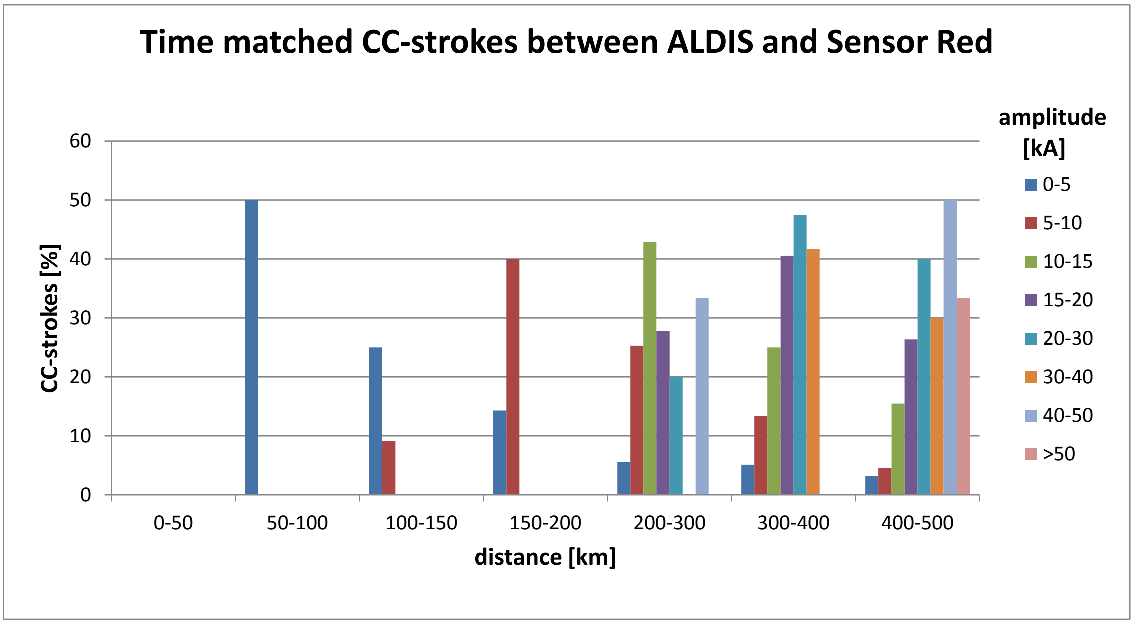

Positive CG-strokes located by ALDIS
positive CG-strokes
300
250
200
150
100
50
0
0-50
50-100
100-150
150-200
200-300
300-400
400-500
500-600
600-700
700-800
800-900
900-1000
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50


Time matched positive CG-strokes between ALDIS and Sensor Red
positive CG-strokes [%]
120
100
80
60
40
20
0
0-50
50-100
100-150
150-200
200-300
300-400
400-500
500-600
600-700
700-800
800-900
900-1000
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

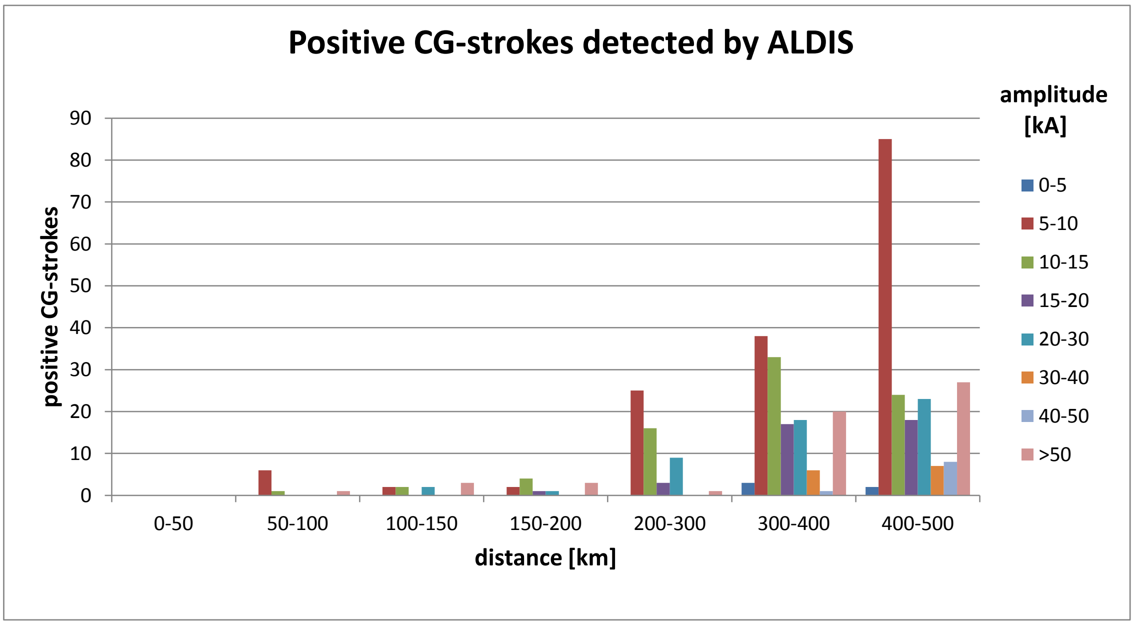
A.1.4. Sensor Red – 09.09.2014



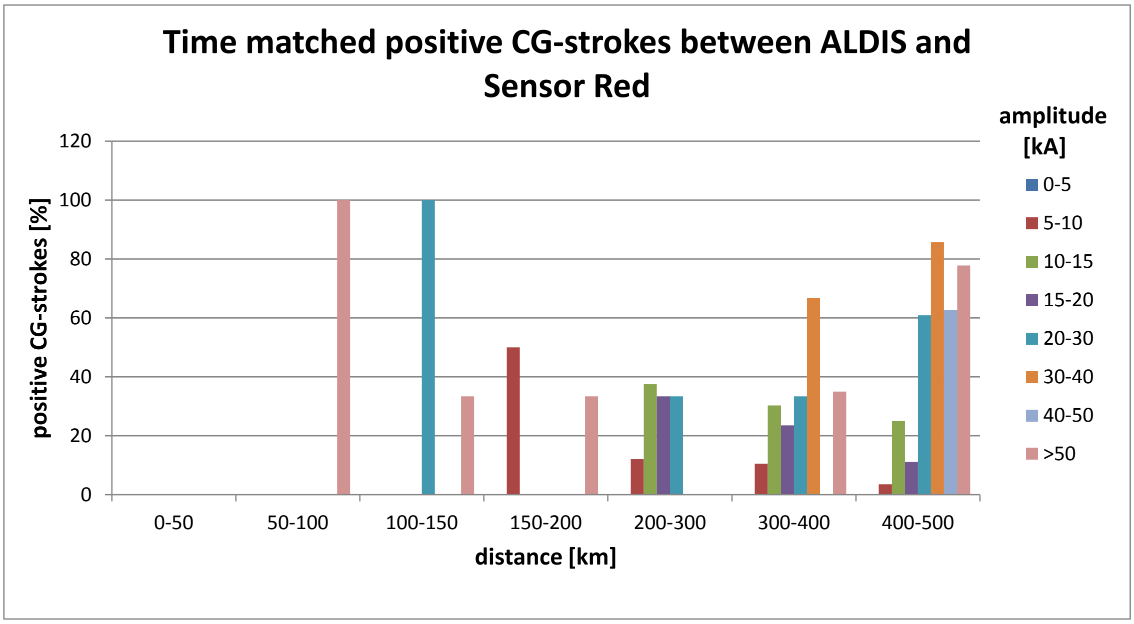




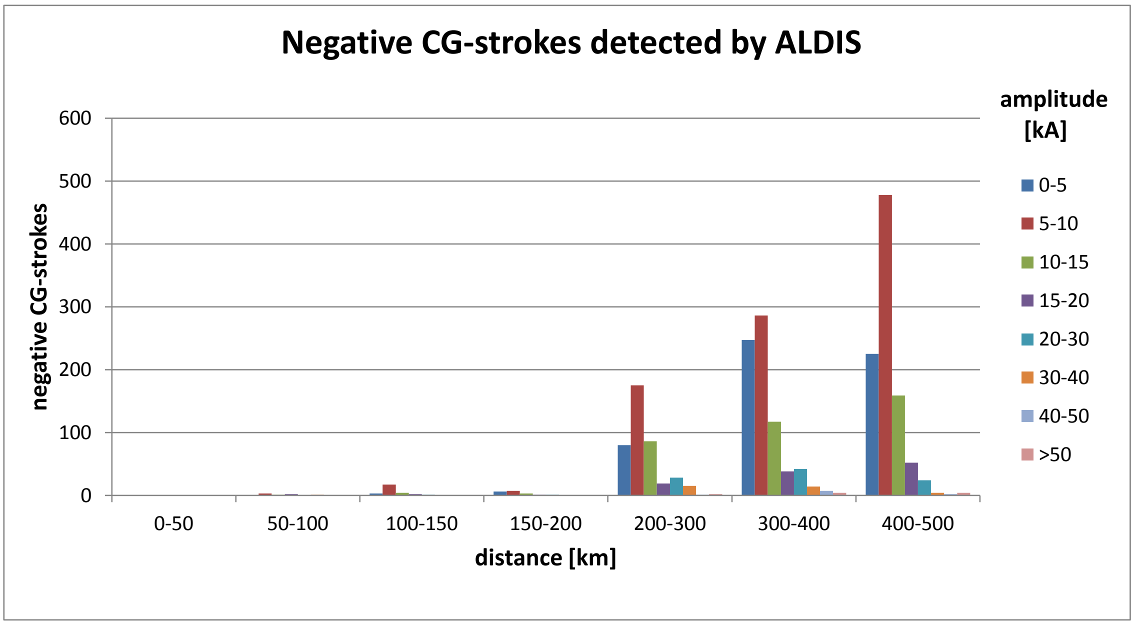
Time matched CC-strokes between ALDIS and Sensor Red
CC-strokes [%]
0
10
20
30
40
50
60
0-50
50-100
100-150
150-200
200-300
300-400
400-500
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50



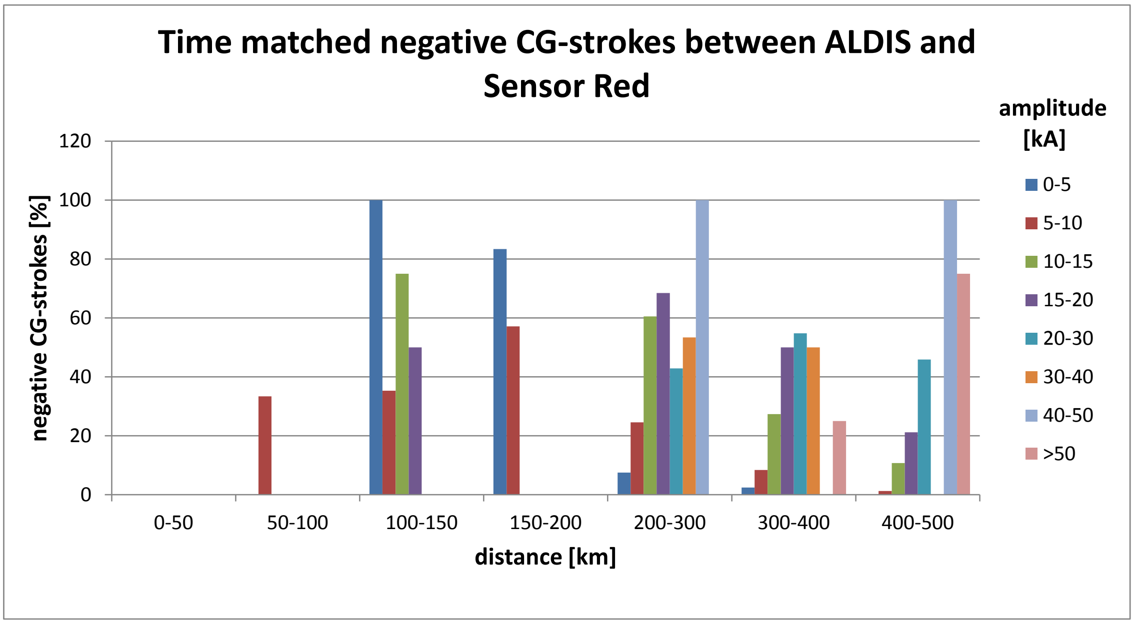
Positive CG-strokes detected by ALDIS
positive CG-strokes
0
10
20
30
40
50
60
70
80
90
0-50
50-100
100-150
150-200
200-300
300-400
400-500
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50



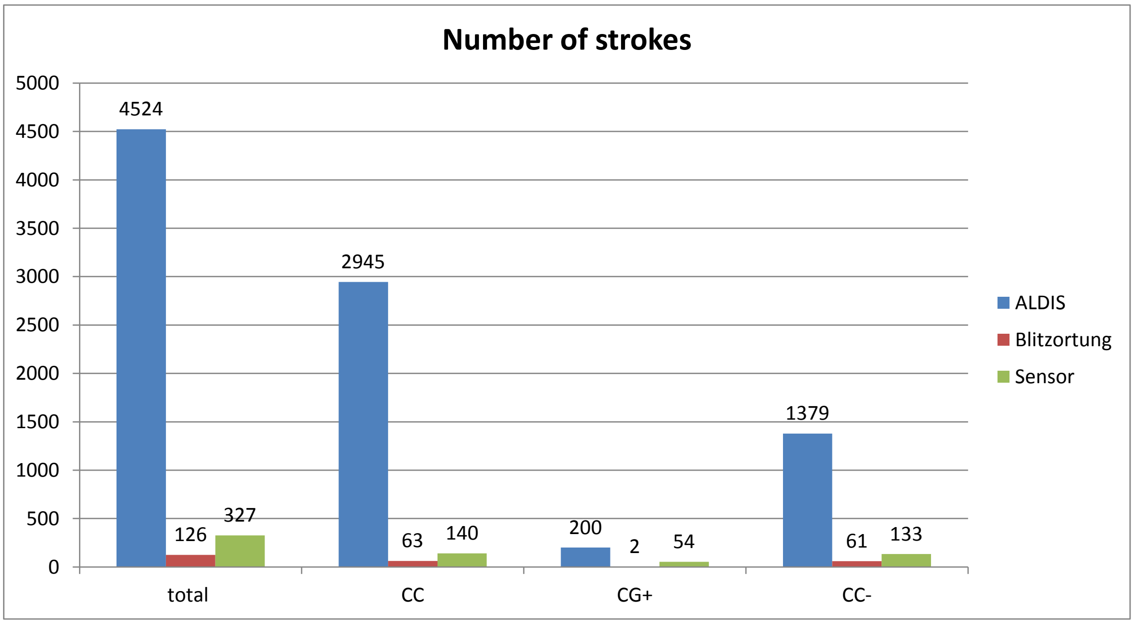
Time matched positive CG-strokes between ALDIS and Sensor Red
positive CG-strokes [%]
120
100
80
60
40
20
0
0-50
50-100
100-150
150-200
200-300
300-400
400-500
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

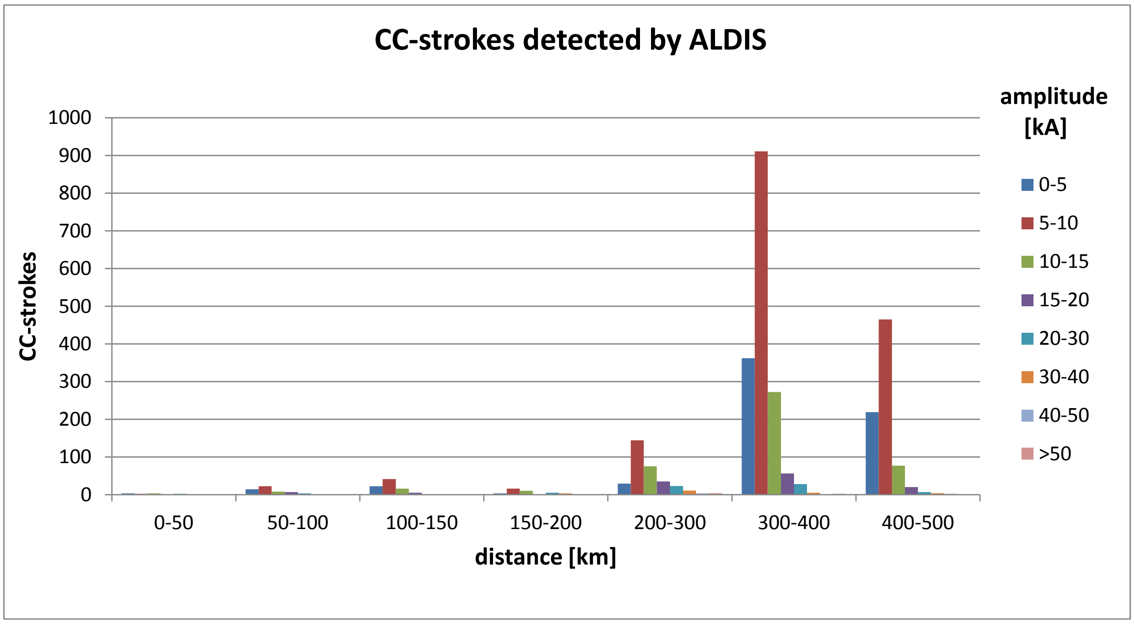


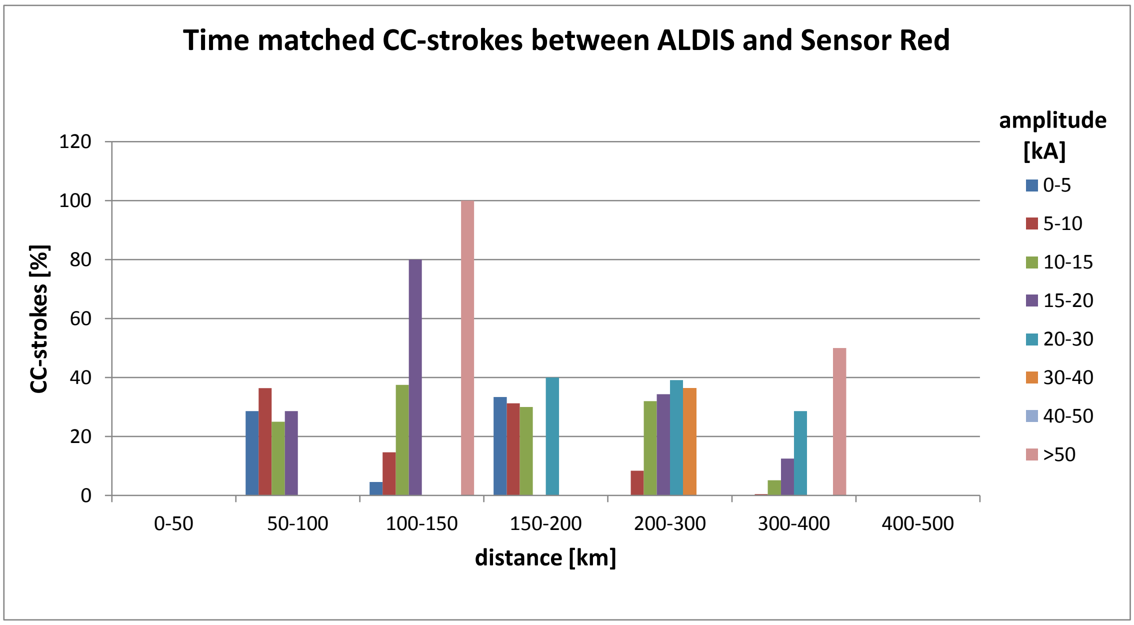
Negative CG-strokes detected by ALDIS
negative CG-strokes
600
500
400
300
200
100
0
0-50
50-100
100-150
150-200
200-300
300-400
400-500
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

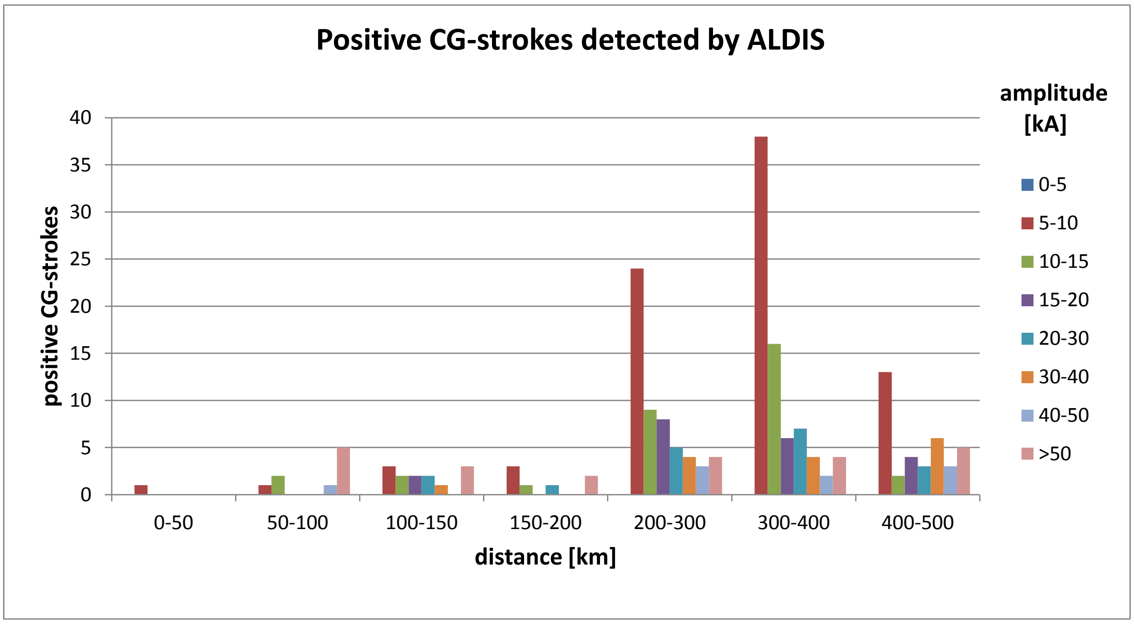


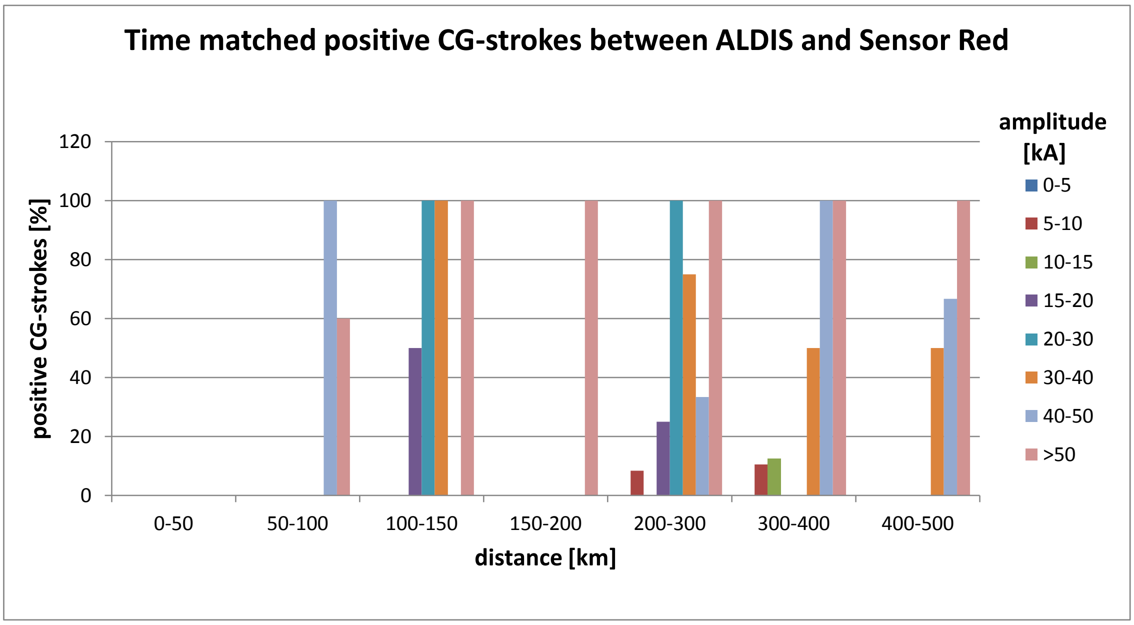
A.1.5. Sensor Red – 11.09.2014

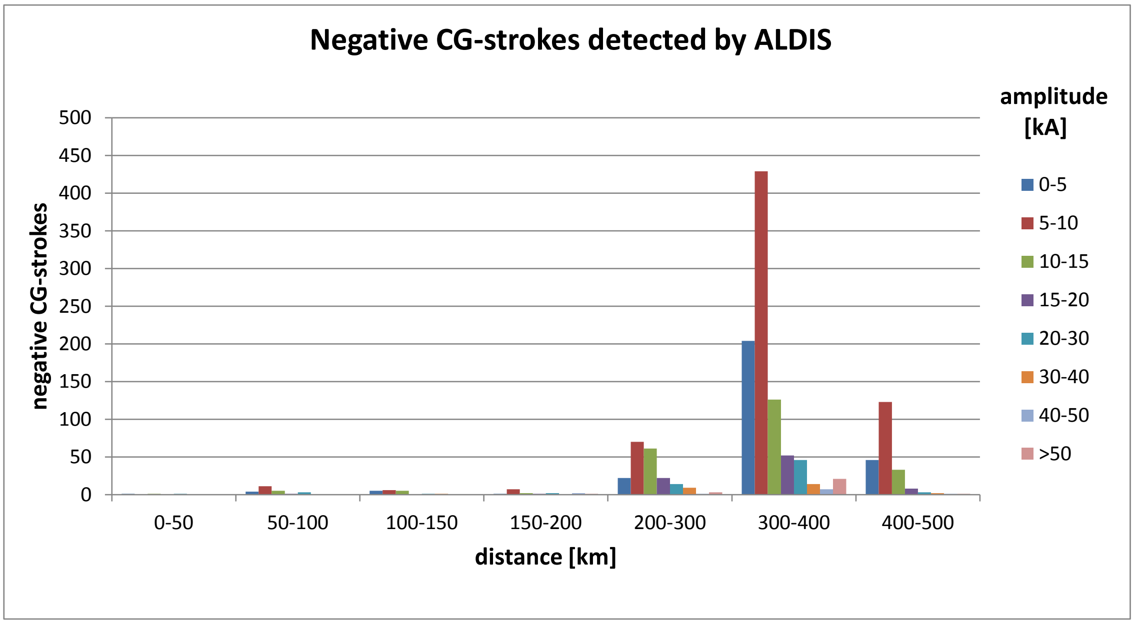


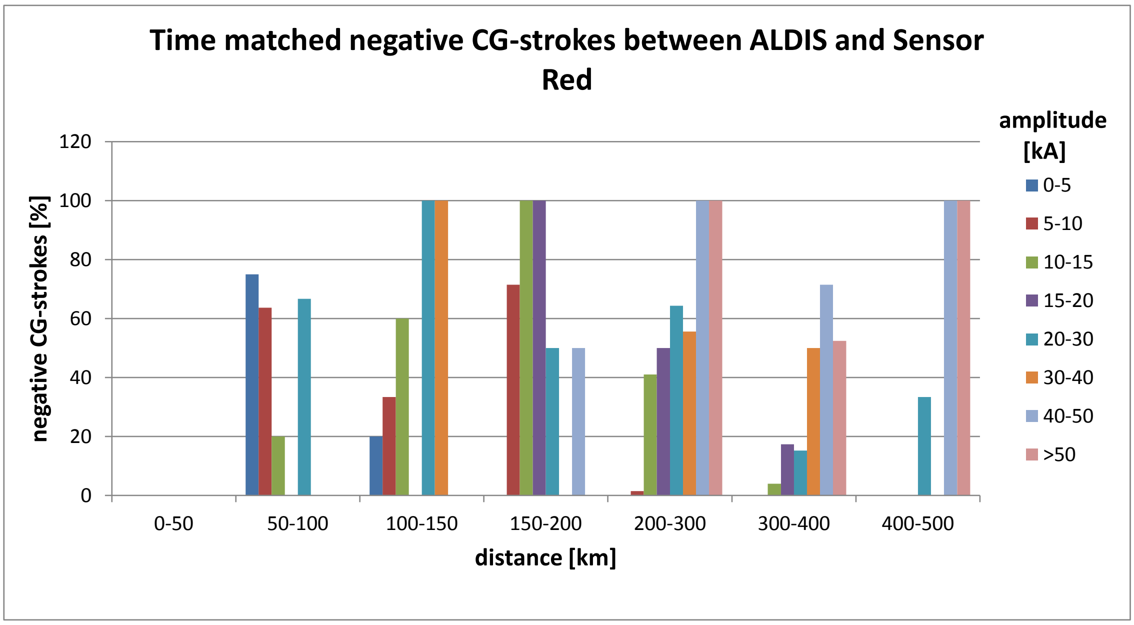

CC-strokes detected by ALDIS
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50
CC-strokes
1000
900
800
700
600
500
400
300
200
100
0
0-50
50-100
100-150
150-200
200-300
300-400
400-500
distance [km]


Time matched CC-strokes between ALDIS and Sensor Red
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50
CC-strokes [%]
120
100
80
60
40
20
0
0-50
50-100
100-150
150-200
200-300
300-400
400-500
distance [km]

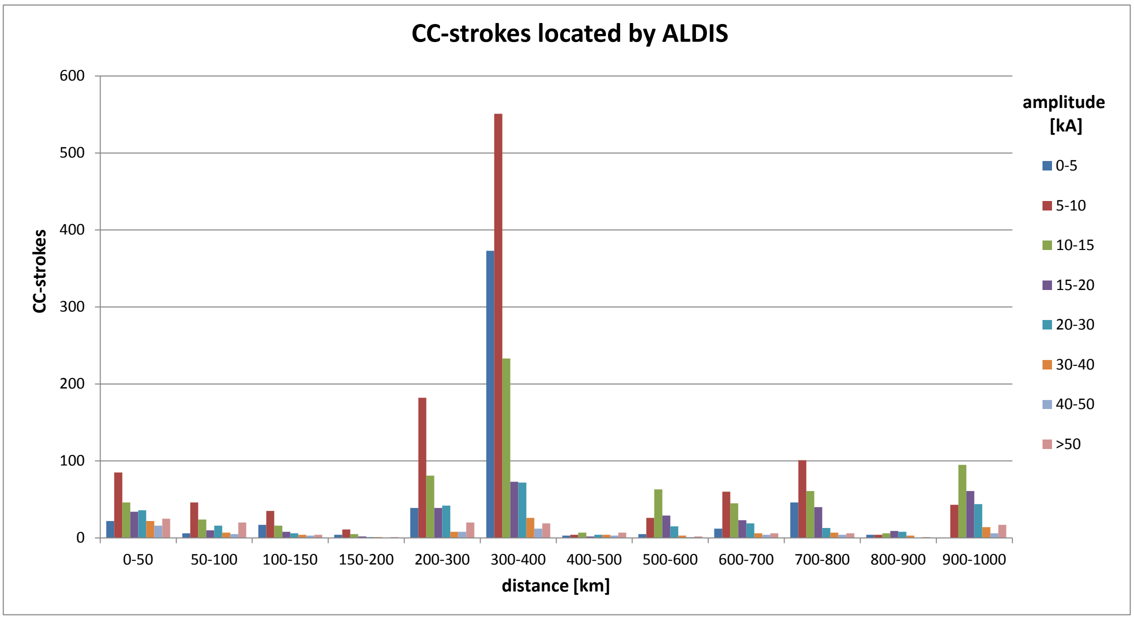

Positive CG-strokes detected by ALDIS
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50
positive CG-strokes
0
5
10
15
20
25
30
35
40
0-50
50-100
100-150
150-200
200-300
300-400
400-500
distance [km]

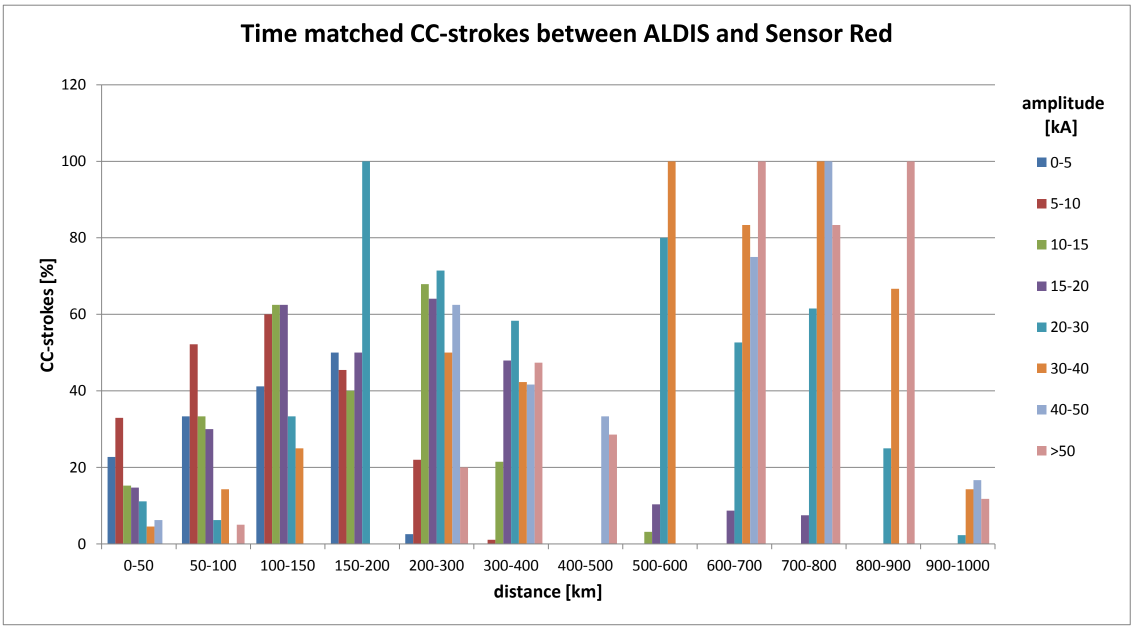

Time matched positive CG-strokes between ALDIS and Sensor Red
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50
positive CG-strokes [%]
0
20
40
60
80
100
120
0-50
50-100
100-150
150-200
200-300
300-400
400-500
distance [km]

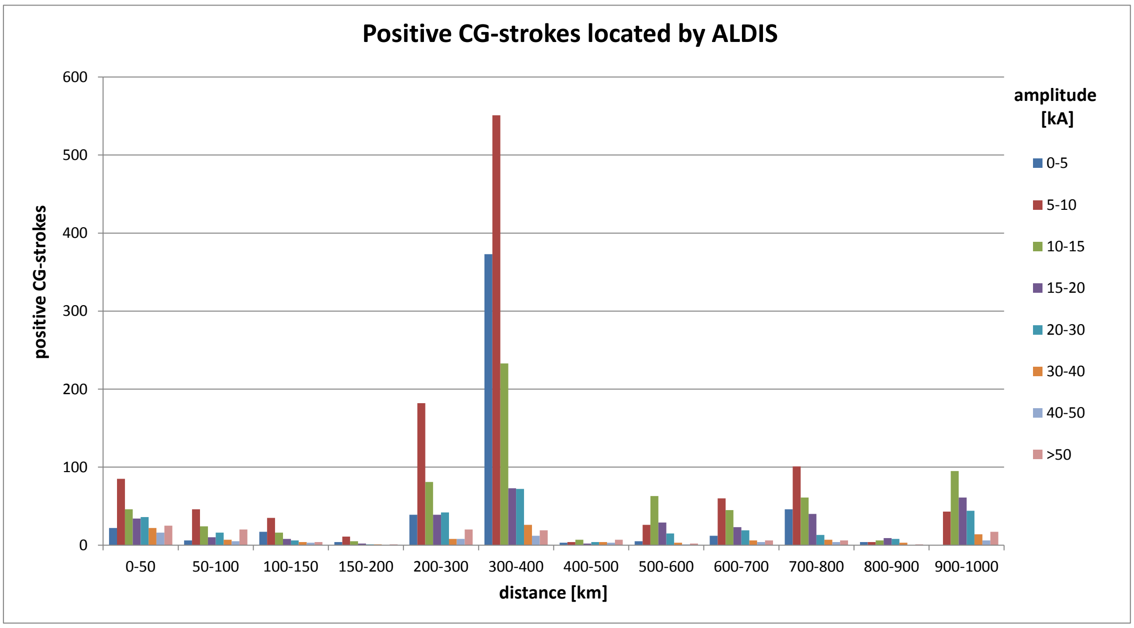

Negative CG-strokes detected by ALDIS
negative CG-strokes
500
450
400
350
300
250
200
150
100
50
0
0-50
50-100
100-150
150-200
200-300
300-400
400-500
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

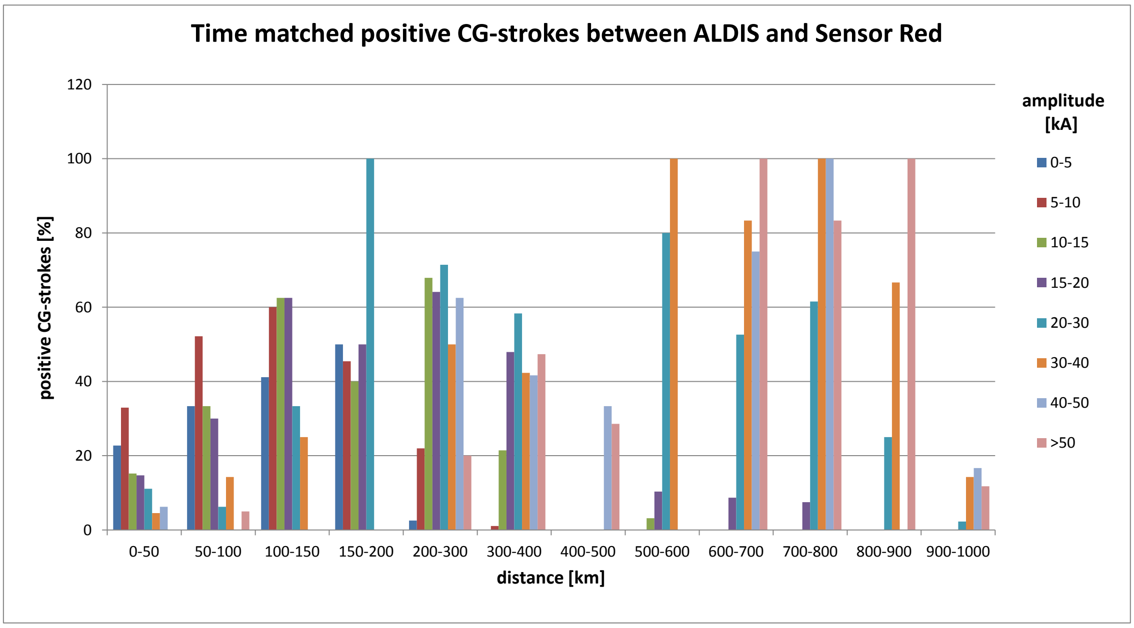

Time matched negative CG-strokes between ALDIS and Sensor Red
negative CG-strokes [%]
120
100
80
60
40
20
0
0-50
50-100
100-150
150-200
200-300
300-400
400-500
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

A.1.6. Sensor Red – 22.09.2014

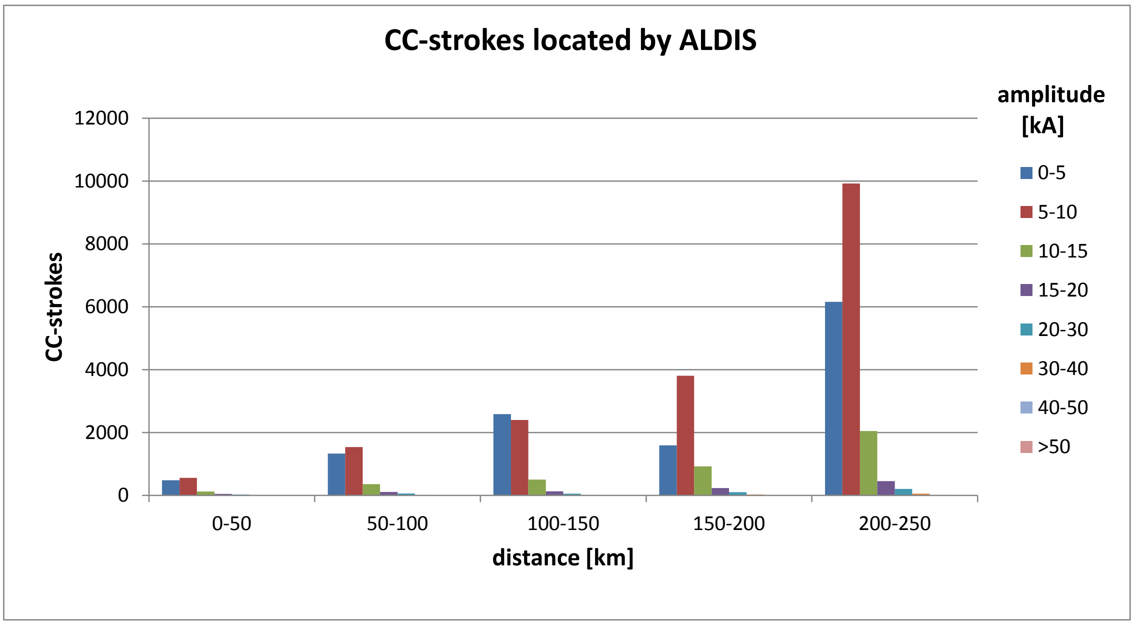


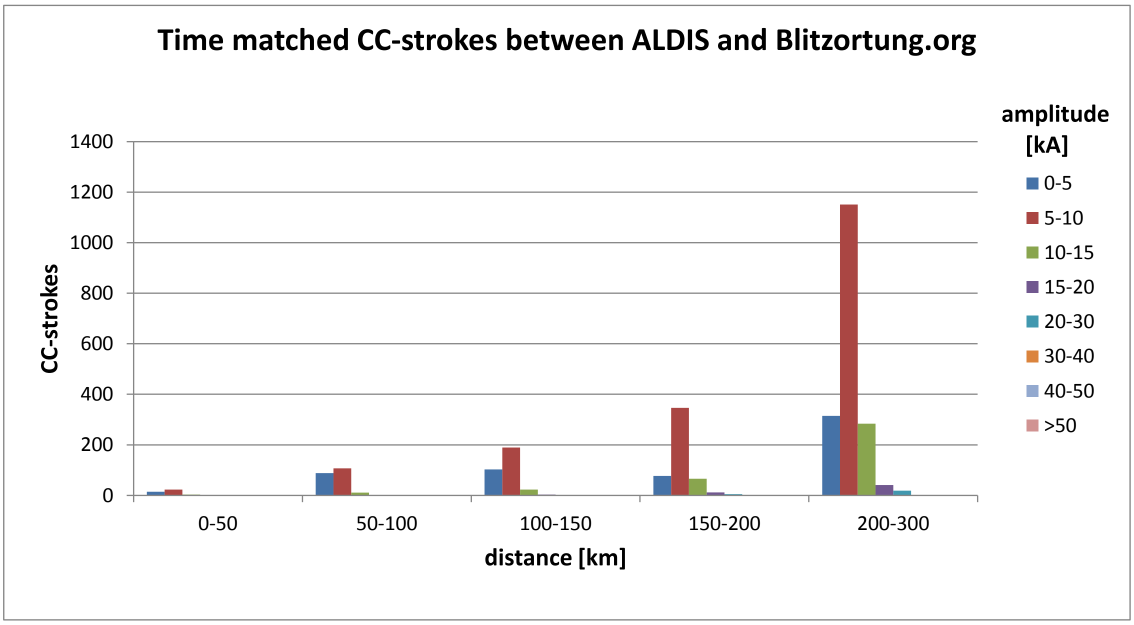


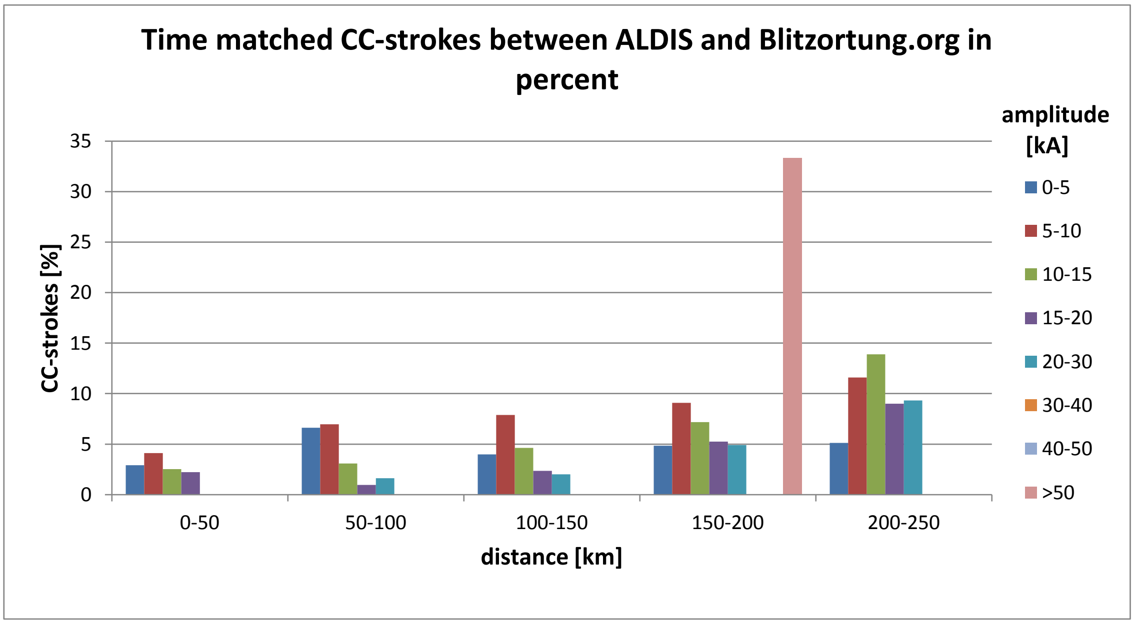

Positive CG-strokes located by ALDIS
positive CG-strokes
0
100
200
300
400
500
600
0-50
50-100
100-150
150-200
200-300
300-400
400-500
500-600
600-700
700-800
800-900
900-1000
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

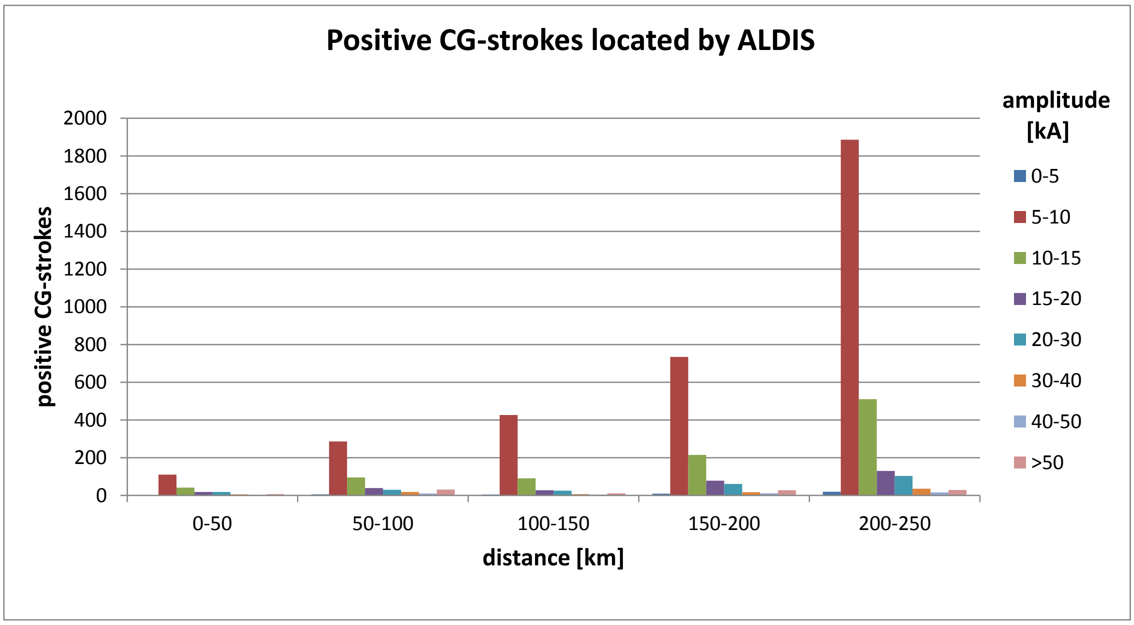

Time matched positive CG-strokes between ALDIS and Sensor Red
positive CG-strokes [%]
0
20
40
60
80
100
120
0-50
50-100
100-150
150-200
200-300
300-400
400-500
500-600
600-700
700-800
800-900
900-1000
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

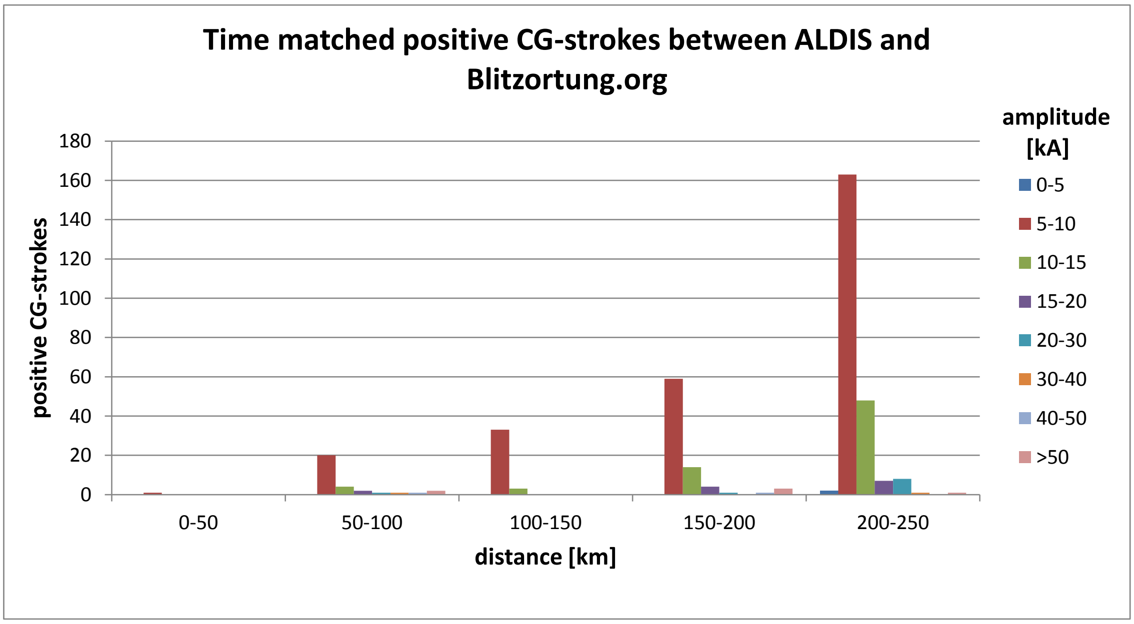
A.1.7. Blitzortung.org versus ALDIS – 03.08.2014

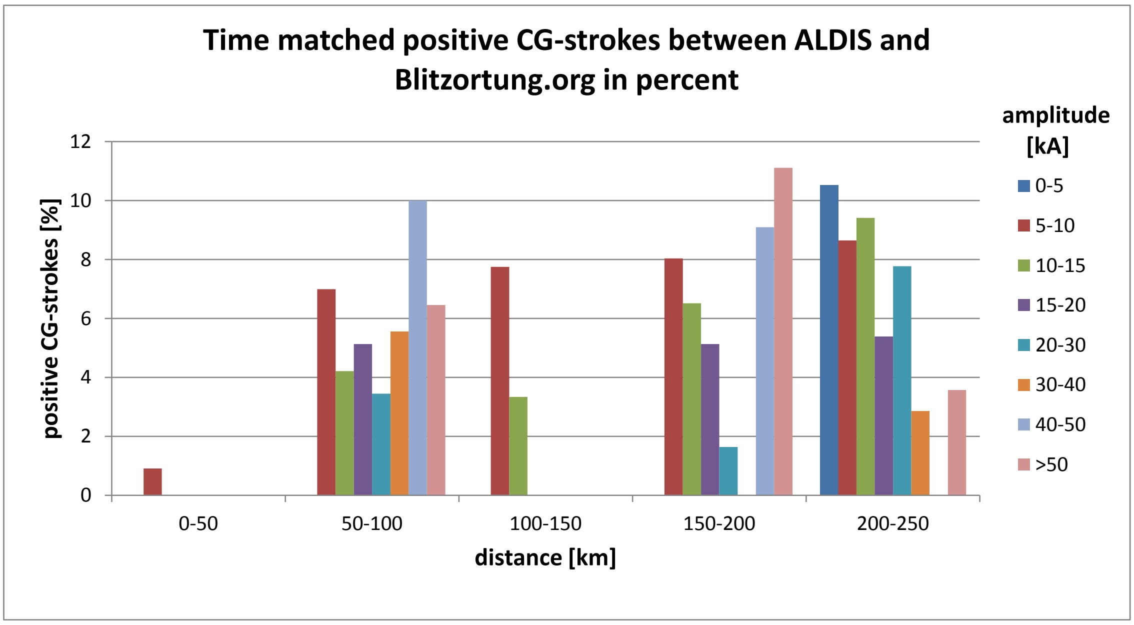





Time matched CC-strokes between ALDIS and Blitzortung.org in percent
CC-strokes [%]
35
30
25
20
15
10
5
0
0-50
50-100
100-150
150-200
200-250
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50


Positive CG-strokes located by ALDIS
positive CG-strokes
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0-50
50-100
100-150
150-200
200-250
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50


Time matched positive CG-strokes between ALDIS and Blitzortung.org
positive CG-strokes
180
160
140
120
100
80
60
40
20
0
0-50
50-100
100-150
150-200
200-250
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50


Time matched positive CG-strokes between ALDIS and Blitzortung.org in percent
positive CG-strokes [%]
12
10
8
6
4
2
0
0-50
50-100
100-150
150-200
200-250
distance [km]
amplitude [kA]
0-5
5-10
10-15
15-20
20-30
30-40
40-50
>50

A.2. Source code

A.2.1. Data Import Sensor Green

```
'Daten Einlesen von Sensor Green

Option Explicit
Public deletedata As Integer
Public fnumber As Integer
Public fNameAndPath As Variant
Public UFclose As Boolean


Public Sub DatenLesenSensor()


Dim Arr
Dim FSO
Dim Datei
Dim L As Long
Dim I, J As Integer
Dim Str_String As String
Dim str_Hex As String
Dim sPosStrike, sPosPulse As Long
Dim KeyStrike, KeyPulse As String
Dim arr_Data
Dim FoundIt As Boolean
Dim begin As String
Dim row As Long
Dim pointer, counter, diff_counter, counter_corr As Long


UFclose = 1 'globale Variable zum Schließen der Userform mit dem Schließenkreuz auf 1 setzen

import_data.Show vbModal 'userform zum datenimport öffnen
If UFclose = 0 Then Exit Sub 'exit wenn die Userform geschlossen wird, kein Dateipfad oder keine Dateianzahl eingegeben wird

'Daten markieren und löschen

    If deletedata = -1 And ActiveWorkbook.Worksheets("Blitzortung_Sensor").Range("A6").Value <> "" Then
        Range(Cells(6, 1), Cells(1048576, 16384).End(xlDown)).Select
        Selection.Delete Shift:=xlUp
        begin = "A6"
    ElseIf ActiveWorkbook.Worksheets("Blitzortung_Sensor").Range("A6").Value = "" Then
        begin = "A6"
    Else
        Range("A6").Select
        Selection.End(xlDown).Select
        ActiveCell.Offset(1, 0).Select
        begin = ActiveCell.Address
    End If

    row = 0


'####################
```

```
'Textdatei auslesen

I = 1

'Einlesen der Textdatei in ein Array
'(Das zeilenweise Einlesen der Datei und das anschließende Bearbeiten benötigt zu viel Zeit oder führt zum Absturz von Excel)
Do While I <= fnumber
    If Dir(fNameAndPath) <> "" Then
        Set FSO = CreateObject("Scripting.FilesystemObject")
        Set Datei = FSO.OpentextFile(fNameAndPath)
        Str_String = Datei.readall
        Datei.Close

        Arr = Split(Str_String, vbCrLf) 'Nach Datensätzen splitten

        KeyStrike = "$BD"     'Schlüsselwort für Blitzdaten
        KeyPulse = "$BS"      'Schlüsselwort für PPS
        sPosStrike = 1
        L = 0

        'Array arr_Data nur beim ersten Durchlauf (erste Textdatei) redimensionieren
        If I = 1 Then
            Do While sPosStrike > 0             'Anzahl der Datensätze ermitteln
                sPosStrike = InStr(sPosStrike + 1, Str_String, KeyStrike)
                L = L + 1
            Loop
    '        ReDim arr_Data(L + 30, 2)
            ReDim arr_Data((fnumber * L) + 5, 4) 'Länge des arrays ist die Anzahl der Datensätze mal die Anzahl der Dateien
        End If

        L = row

        For pointer = 1 To UBound(Arr)
            If InStr(Arr(pointer), KeyStrike) And InStr(Arr(pointer - 1), KeyPulse) Then 'IF $BD folgt $BS

                counter = 1 'Zähler für $BS-Zeile
                diff_counter = CDec("&H" & Mid(Arr(pointer), 5, 6)) - CDec("&H" & Mid(Arr(pointer - counter), 5, 6)) 'Berechnung der Differenz der Counterwerte für PPS und Strike

                arr_Data(L, 0) = Mid(Arr(pointer - counter), 21, 2) & "." & Mid(Arr(pointer - counter), 23, 2) & "." & Mid(Arr(pointer - counter), 25, 2) + 2000    'Datum ausschneiden

                'Addieren von einer Sekunde, wenn der Zeitstempel von zwei aufeinanderfolgenden PPS-Signalen gleich ist
                If Mid(Arr(pointer - counter), 14, 6) = Mid(Arr(pointer - counter - 1), 14, 6) Then
                    arr_Data(L, 1) = Mid(Arr(pointer - counter), 14, 2) & ":" & Mid(Arr(pointer - counter), 16, 2) & ":" & Mid(Arr(pointer - counter), 18, 2) + 1 'Zeit ausschneiden
                Else
```

```
            arr_Data(L, 1) = Mid(Arr(pointer - counter), 14, 2) & ":" & Mid(Arr(pointer - counter), 16, 2) & ":" & Mid(Arr(pointer - counter), 18, 2)     'Zeit ausschneiden
        End If

        If diff_counter >= 0 Then
            arr_Data(L, 2) = diff_counter / 2500089 * 10 ^ 9    'Berechnung der Nanosekunden ohne Überlauf
        Else
            arr_Data(L, 2) = (diff_counter + CDec("&HFFFFFF")) / 2500089 * 10 ^ 9   'Berechnung der Nanosekunden mit Überlauf
        End If

        str_Hex = Mid(Arr(pointer), 12, 256)
        Do While str_Hex <> ""
            'arr_Data(L, 4) = ""
            If Len(str_Hex) >= 4 Then
                arr_Data(L, 3) = arr_Data(L, 3) & Left(str_Hex, 2)      'Daten Kanal 1 auschneiden
                arr_Data(L, 4) = arr_Data(L, 4) & Mid(str_Hex, 3, 2)    'Daten Kanal 2 auschneiden
                str_Hex = Right(str_Hex, Len(str_Hex) - 4)              'HEX-Str kürzen
            Else
                str_Hex = ""    'String leeren, wenn zu wenige Zeichen = Fehler
            End If
        Loop

        counter = counter + 1   'Zähler für $BS-Zeile hochsetzen
        L = L + 1
    ElseIf InStr(Arr(pointer), KeyStrike) And InStr(Arr(pointer - 1), KeyStrike) Then   'IF $BD folgt $BD

        diff_counter = CDec("&H" & Mid(Arr(pointer), 5, 6)) - CDec("&H" & Mid(Arr(pointer - counter), 5, 6))
        'Berechnung der Differenz der Counterwerte für PPS und Strike

        arr_Data(L, 0) = Mid(Arr(pointer - counter), 21, 2) & "." & Mid(Arr(pointer - counter), 23, 2) & "." & Mid(Arr(pointer -
        counter), 25, 2) + 2000    'Datum ausschneiden

        'Addieren von einer Sekunde, wenn der Zeitstempel von zwei aufeinanderfolgenden PPS-Signal gleich ist
        If Mid(Arr(pointer - counter), 14, 6) = Mid(Arr(pointer - counter - 1), 14, 6) Then
            arr_Data(L, 1) = Mid(Arr(pointer - counter), 14, 2) & ":" & Mid(Arr(pointer - counter), 16, 2) & ":" & Mid(Arr(pointer -
            counter), 18, 2) + 1        'Zeit ausschneiden
        Else
            arr_Data(L, 1) = Mid(Arr(pointer - counter), 14, 2) & ":" & Mid(Arr(pointer - counter), 16, 2) & ":" & Mid(Arr(pointer -
            counter), 18, 2)            'Zeit ausschneiden
        End If

        If diff_counter >= 0 Then
            arr_Data(L, 2) = diff_counter / 2500089 * 10 ^ 9    'Berechnung der Nanosekunden ohne Überlauf
        Else
```

```
                    arr_Data(L, 2) = (diff_counter + CDec("&HFFFFFF")) / 2500089 * 10 ^ 9   'Berechnung der Nanosekunden mit Überlauf
                End If


                str_Hex = Mid(Arr(pointer), 12, 256)
                Do While str_Hex <> ""
                    If Len(str_Hex) >= 4 Then
                        arr_Data(L, 3) = arr_Data(L, 3) & Left(str_Hex, 2)       'Daten Kanal 1 auschneiden
                        arr_Data(L, 4) = arr_Data(L, 4) & Mid(str_Hex, 3, 2)     'Daten Kanal 2 auschneiden
                        str_Hex = Right(str_Hex, Len(str_Hex) - 4)               'HEX-Str kürzen
                    Else
                        str_Hex = ""    'String leeren, wenn zu wenige Zeichen = Fehler
                    End If
                Loop
                counter = counter + 1   'Zähler für $BS-Zeile hochsetzen
                L = L + 1
            End If
        Next
    End If
    J = I + 1
    fNameAndPath = Replace(fNameAndPath, I & ".txt", J & ".txt") 'nächste Datei zum Datenimport festlegen
    I = I + 1
    row = L
Loop

'Ausgeben. Anpassen.
Sheets("Blitzortung_Sensor").Range(begin, Selection.End(xlDown)).Resize(UBound(arr_Data) + 1, 5) = arr_Data

'Einfügen der Formel zur Berechnung der Zeit mit Nano- und Schaltsekunden
Range("F6").Select
    ActiveCell.FormulaR1C1 = _
        "=RC[-5]+RC[-4]+RC[-3]/10^9/24/3600+R1C10/24/3600"
    ActiveCell.Offset(0, -2).Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(0, 2).Select
    ActiveCell.FormulaR1C1 = "1/1/1900  12:00:00 AM"
    Selection.End(xlUp).Select
    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    ActiveSheet.Paste
    Application.CutCopyMode = False

End Sub
```

A.2.2. Data Import Sensor Red

```

'Daten Einlesen von Sensor Green

Option Explicit
Public deletedata As Integer
Public fnumber As Integer
```

```
Public fNameAndPath As Variant
Public UFclose As Boolean



Public Sub DatenLesenSensor()

'Dim fNameAndPath As Variant
Dim Arr
Dim Datei
Dim FSO
Dim L As Long
Dim arr_HexAll, arr_ASCIIall As Variant
Dim I, J As Integer
Dim Str_String, Str_Data, Str_Time As String
Dim sPos, sPosData3, sPosData4 As Long
Dim KeyData3, KeyData4, KeyTime As String
Dim arr_Data
Dim FoundIt As Boolean
Dim begin As String
Dim row As Long


UFclose = 1 'globale Variable zum Schließen der Userform mit dem Schließenkreuz auf 1 setzen

import_data.Show vbModal 'userform zum datenimport öffnen
If UFclose = 0 Then Exit Sub 'exit wenn die Userform geschlossen wird, kein Dateipfad oder keine Dateianzahl eingegeben wird

'Daten markieren und löschen

    If deletedata = -1 And ActiveWorkbook.Worksheets("Blitzortung_Sensor").Range("A6").Value <> "" Then
        Range(Cells(6, 1), Cells(1048576, 16384).End(xlDown)).Select
        Selection.Delete Shift:=xlUp
        begin = "A6"
    ElseIf ActiveWorkbook.Worksheets("Blitzortung_Sensor").Range("A6").Value = "" Then
        begin = "A6"
    Else
        Range("A6").Select
        Selection.End(xlDown).Select
        ActiveCell.Offset(1, 0).Select
        begin = ActiveCell.Address
    End If

    row = 0


'####################
'Textdatei auslesen

I = 1

'Einlesen der Textdatei in ein Array
'(Das zeilenweise Einlesen der Datei und das anschließende Bearbeiten benötigt zu viel Zeit oder führt zum Absturz von Excel)
Do While I <= fnumber
    If Dir(fNameAndPath) <> "" Then
        Set FSO = CreateObject("Scripting.FilesystemObject")
        Set Datei = FSO.OpentextFile(fNameAndPath)
        Str_String = Datei.readall
```

```
        Datei.Close


        Str_String = Replace(Str_String, vbCrLf + vbCrLf, vbCrLf) ' doppelte Zeilenumbrüche ersetzen
        Str_String = Replace(Str_String, "  ", " ") 'doppelte Leerzeichen ersetzen
        Arr = Split(Str_String, vbCrLf) 'Nach Datensätzen splitten
        ReDim arr_HexAll(UBound(Arr))   'Array redimensionieren
        ReDim arr_ASCIIall(UBound(Arr)) 'Array redimensionieren

        'HEX-Werte und ASCII-Zeichen ausschneiden
        For L = 0 To UBound(Arr)
            arr_HexAll(L) = Mid(Arr(L), 6, 47)
            arr_ASCIIall(L) = Right(Arr(L), 17)
        Next

        Str_Data = Join(arr_HexAll, " ") 'Zusammenfügen der HEX-Datenzeilen
        Str_Time = Join(arr_ASCIIall, " ") 'Zusammenfügen der ASCII-Datenzeilen
        KeyData3 = Txt2Hex("data 3")    'Schlüsselwort für Kanal 1 - data 0 oder data 3
        KeyData4 = Txt2Hex("data 4")    'Schlüsselwort für Kanal 2 - data 1 oder data 4
        sPos = 1
        L = 1

        'Array arr_Data nur beim ersten Durchlauf redimensionieren
        If I = 1 Then
            Do While sPos > 0           'Anzahl der Datensätze ermitteln
                sPos = InStr(sPos + 1, Str_Time, ".time")
                L = L + 1
            Loop
'            ReDim arr_Data(L + 30, 2)
            ReDim arr_Data((fnumber * L) + 5, 3) 'Länge des arrays ist die Anzahl der Datensätze mal die Anzahl der Dateien
        End If

        sPos = 1
        L = row
        FoundIt = True
        Do While FoundIt    'Nur so lange das Schlüsselwort enthalten
            sPos = InStr(sPos + 1, Str_Time, ".time") 'Startposition des Zeitstempels suchen
            sPosData3 = InStr(sPosData3 + 1, Str_Data, KeyData3) 'Startposition des ersten Datensatzes suchen
            sPosData4 = InStr(sPosData4 + 1, Str_Data, KeyData4)    'Startposition des zweiten Datensatzes suchen

            If sPos = 0 Or sPosData3 = 0 Or sPosData4 = 0 Then
                FoundIt = False
                Exit Do
            ElseIf sPosData3 > sPosData4 Then
                sPos = InStr(sPos + 1, Str_Time, ".time") 'Startposition des Zeitstempels suchen
                sPosData4 = InStr(sPosData4 + 1, Str_Data, KeyData4)    'Startposition des zweiten Datensatzes suchen
            ElseIf (sPosData4 - sPosData3) > 2000 Then
                sPos = InStr(sPos + 1, Str_Time, ".time") 'Startposition des Zeitstempels suchen
                sPosData3 = InStr(sPosData3 + 1, Str_Data, KeyData3) 'Startposition des ersten Datensatzes suchen
            End If
            arr_Data(L, 0) = Replace(Mid(Str_Time, sPos + 5, 24), " ", "") 'Zeitstempel ausschneiden und alle Leerzeichen innerhalb des ↩
```

```
                Zeitstempels löschen
            arr_Data(L, 1) = Replace(Mid(Str_Time, sPos + 30, 9), " ", "") '↩
                Nanosekunden ausschneiden und alle Leerzeichen innerhalb des ↩
                Zeitstempels löschen
            arr_Data(L, 0) = Left(arr_Data(L, 0), 10) + " " + Right(arr_Data(L, 0),↩
                 8) 'Datum und Zeit mit Leerzeichen trennen
            arr_Data(L, 2) = Mid(Str_Data, sPosData3 + 36, 1536) 'Datensatz 1 in ↩
                Array arr_Data abspeichern
            arr_Data(L, 3) = Mid(Str_Data, sPosData4 + 36, 1536) 'Datensatz 2 in ↩
                Array arr_Data abspeichern
            L = L + 1
        Loop

        J = I + 1
        fNameAndPath = Replace(fNameAndPath, I & ".txt", J & ".txt") 'nächste Datei↩
             zum Datenimport festlegen
        I = I + 1
        row = L
        Else
        Exit Do
    End If
Loop

'Ausgeben. Anpassen.
Sheets("Blitzortung_Sensor").Range(begin, Selection.End(xlDown)).Resize(UBound(↩
    arr_Data) + 1, 4) = arr_Data

'Einfügen der Formel zur Berechnung der Zeit mit Nano- und Schaltsekunden
Range("E6").Select
    ActiveCell.FormulaR1C1 = _
        "=RC[-4]+RC[-3]/10^9/24/3600+R1C9/24/3600"
    ActiveCell.Offset(0, -2).Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(0, 2).Select
    ActiveCell.FormulaR1C1 = "1/1/1900  12:00:00 AM"
    Selection.End(xlUp).Select
    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    ActiveSheet.Paste
    Application.CutCopyMode = False

End Sub

Public Function Txt2Hex(TextString As String) As String
' Geschrieben von Hannes Pichler (ALDIS)
' Wandle einen ASCII-Text in Hexzahlen mit Leerzeichen dazwischen um
    Dim I As Long

    For I = 1 To Len(TextString)
        Txt2Hex = Txt2Hex & Hex(Asc(Mid(TextString, I, 1))) & " "
    Next I
    Txt2Hex = LCase(Txt2Hex)
End Function
```

A.2.3. Waveform

```
Sub Show_Wave()
' Zeige Kurve

```

```
Dim int_last_line As Long
Dim db_time As Double
Dim I As Long
Dim str_data1, str_data2 As String


    Sheets("waveform").Select
    db_time = Cells(1, 5)                'Datum und Zeit einlesen

    Sheets("Blitzortung_Sensor").Select
    Range("E6").Select
    Selection.End(xlDown).Select
    int_last_line = ActiveCell.row       'letzte Datenzeile bestimmen

    For I = 6 To int_last_line           'Blitzsignal zum entsprechenden Datum und ↩
        Zeit suchen
        If Cells(I, 5) = db_time Then
            str_data1 = Cells(I, 3)      'Daten Kanal 1 abspeichern
            str_data2 = Cells(I, 4)      'Daten Kanal 2 abspeichern
            Exit For
        End If
    Next

    Sheets("waveform").Select
    Cells(1, 8) = str_data1              'Daten Kanal 1 ausgeben
    Cells(1, 10) = str_data2             'Daten Kanal 2 ausgeben

End Sub

Sub prev()
' Zeige vorherige Kurve
Dim int_last_line As Long
Dim db_time As Double
Dim I As Long
Dim str_data1, str_data2 As String
Dim int_Zeile As Integer

Application.ScreenUpdating = False

    Sheets("Waveform").Select
    int_Zeile = Cells(1, 12)
    db_time = Cells(1, 5)

    Sheets("Aldis_Sensor").Select
    Range("A7").Select
    Selection.End(xlDown).Select
    int_last_line = ActiveCell.row       'letzte Datenzeile bestimmen
    Selection.End(xlUp).Select
    int_first_line = ActiveCell.row
    If Sheets("aldis_Sensor").Cells(int_Zeile, 1) = db_time Then
        If int_Zeile = 0 Then
            For I = int_last_line To 7 Step -1           'Blitzsignal zum ↩
                entsprechenden Datum und Zeit suchen
                If Cells(I, 1) = db_time And Cells(I, 1).Interior.Color = 10092543 ↩
                    Then
                    int_Zeile = Cells(I, 1).row
                    Exit For
                End If
            Next
        End If

        If int_Zeile = int_first_line Or int_Zeile = int_first_line + 1 Then
            int_Zeile = int_last_line + 1
```

```
        End If

        int_Zeile = int_Zeile - 1

        For I = int_Zeile To 7 Step -1
            If Cells(I, 3) = 1 And Cells(I, 1).Interior.Color = 10092543 Then
                Sheets("waveform").Cells(1, 12) = I
                db_time = Cells(I, 1)
                Exit For
            End If
        Next


        Sheets("Blitzortung_Sensor").Select
        Range("A6").Select
        Selection.End(xlDown).Select
        int_last_line = ActiveCell.row      'letzte Datenzeile bestimmen

        For I = 6 To int_last_line
            If Cells(I, 5) = db_time Then
                str_data1 = Cells(I, 3)      'Daten Kanal 1 abspeichern
                str_data2 = Cells(I, 4)      'Daten Kanal 2 abspeichern
                Exit For
            End If
        Next

        Sheets("Waveform").Select

        Cells(1, 8) = str_data1
        Cells(1, 10) = str_data2
        Cells(1, 5) = db_time
    Else
        Call Show_Wave

        Sheets("Aldis_Sensor").Select
        For I = 7 To int_last_line
            If Cells(I, 3) = 1 And Cells(I, 1) = db_time And Cells(I, 1).Interior.↩
                Color = 10092543 Then
                Sheets("Waveform").Cells(1, 12) = I
                db_time = Cells(I, 1)
                Exit For
            End If
        Next
        Sheets("Waveform").Select
    End If
Application.ScreenUpdating = True
End Sub

Sub next_wave()
' Zeige nächste Kurve
Dim int_last_line As Long
Dim db_time As Double
Dim I As Long
Dim str_data1, str_data2 As String
Dim int_Zeile As Integer

Application.ScreenUpdating = False

    Sheets("Waveform").Select
    int_Zeile = Cells(1, 12)
    db_time = Cells(1, 5)

    Sheets("Aldis_Sensor").Select
```

```
    Range("A7").Select
    Selection.End(xlDown).Select
    int_last_line = ActiveCell.row      'letzte Datenzeile bestimmen

    If Sheets("aldis_Sensor").Cells(int_Zeile, 1) = db_time Then
        If int_Zeile = 0 Then
            For I = 7 To int_last_line          'Blitzsignal zum entsprechenden ↩
                Datum und Zeit suchen
                If Cells(I, 1) = db_time And Cells(I, 1).Interior.Color = 10092543 ↩
                    Then
                    int_Zeile = Cells(I, 1).row
                    Exit For
                End If
            Next
        End If

        If int_Zeile = int_last_line Or int_Zeile = int_last_line - 1 Then
            int_Zeile = 6
        End If

        int_Zeile = int_Zeile + 1

        For I = int_Zeile To int_last_line
            If Cells(I, 3) = 1 And Cells(I, 1).Interior.Color = 10092543 Then
                Sheets("waveform").Cells(1, 12) = I
                db_time = Cells(I, 1)
                Exit For
            End If
        Next


        Sheets("Blitzortung_Sensor").Select
        Range("A6").Select
        Selection.End(xlDown).Select
        int_last_line = ActiveCell.row      'letzte Datenzeile bestimmen

        For I = 6 To int_last_line
            If Cells(I, 5) = db_time Then
                str_data1 = Cells(I, 3)     'Daten Kanal 1 abspeichern
                str_data2 = Cells(I, 4)     'Daten Kanal 2 abspeichern
                Exit For
            End If
        Next

        Sheets("Waveform").Select

        Cells(1, 8) = str_data1
        Cells(1, 10) = str_data2
        Cells(1, 5) = db_time
    Else
        Call Show_Wave

        Sheets("Aldis_Sensor").Select
        'int_Zeile = int_Zeile + 1
        For I = 7 To int_last_line
            If Cells(I, 3) = 1 And Cells(I, 1) = db_time And Cells(I, 1).Interior.↩
                Color = 10092543 Then
                Sheets("waveform").Cells(1, 12) = I
                db_time = Cells(I, 1)
                Exit For
            End If
        Next
        Sheets("Waveform").Select
```

```
    End If
Application.ScreenUpdating = True
End Sub
```

A.2.4. Data Import Blitzortung.org

```
Sub DatenLesenServer()
'
' DatenLesenServer Makro
'
Dim fNameAndPath As Variant              ' enthält den Dateiname der Datei die geöffnet werden soll
Dim FileNum As Long                      ' freie file nummer zum Öffnen herausfinden
Dim DataLine As String                   ' Daten einer Zeile der Datei
Dim Line As Long                         ' Zeile in der in Excel geschrieben wird

    ' öffnet einen Dialog der nach Dateinamen fragt, Abbruch, wenn keine Datei gewählt wird
    fNameAndPath = Application.GetOpenFilename(FileFilter:="TEXT (*.txt), *.txt, CSV (*.csv), *.csv", Title:="Select File To Be Opened")
    If fNameAndPath = False Then Exit Sub

    'Daten markieren und löschen

    Range(Cells(6, 1), Cells(1048576, 16384).End(xlDown)).Select
    Selection.Delete Shift:=xlUp

    Close                                'alle ev. geöffneten Dateien schließen, sollte nicht erforderlich sein, aber gerade beim debuggen bleiben immer geöffnete Dateien (Abbruch vor "close") zurück, dann geht excel irgenwann die freien Kanäle aus
    FileNum = FreeFile()                 ' freien Kanal finden (normalerweise 1)
    Line = 6

     'Textdatei importieren'
     With ActiveSheet.QueryTables.Add(Connection:= _
        "TEXT;" & fNameAndPath _
        , Destination:=Range("$A$6"))
        .Name = "test_1"
        .FieldNames = False
        .RowNumbers = False
        .FillAdjacentFormulas = True
        .PreserveFormatting = True
        .RefreshOnFileOpen = False
        .RefreshStyle = xlOverwriteCells
        .SavePassword = False
        .SaveData = True
        .AdjustColumnWidth = False
        .RefreshPeriod = 0
        .TextFilePromptOnRefresh = False
        .TextFilePlatform = 850
        .TextFileStartRow = 1
        .TextFileParseType = xlDelimited
        .TextFileTextQualifier = xlTextQualifierDoubleQuote
        .TextFileConsecutiveDelimiter = False
        .TextFileTabDelimiter = True
        .TextFileSemicolonDelimiter = True
        .TextFileCommaDelimiter = False
        .TextFileSpaceDelimiter = False
```

```
        .TextFileColumnDataTypes = Array(1, 1, 1, 1, 1, 1, 1, 1, 1, xlSkipColumn, ↩
            xlSkipColumn)
        .TextFileTrailingMinusNumbers = True
        .TextFileStartRow = 2
        .Refresh BackgroundQuery:=False

    End With

    'Einfügen der Formel zur Berechnung der Distance
    Range("I6").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(0, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "1"
    ActiveCell.Offset(0, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "1/1/1900  12:00:00 AM"
    Range("J6").Select
    ActiveCell.FormulaR1C1 = _
        "=orthodrome(Blitzortung_Sensor!R2C5,Blitzortung_Sensor!R2C6,RC[-6],RC[-5])↩
            "
    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    ActiveSheet.Paste
    Application.CutCopyMode = False

    'Einfügen der Formel zur Berechnung der Zeit mit Nano und Laufzeit
    Range("K6").Select
    ActiveCell.FormulaR1C1 = _
        "=RC[-10]+RC[-9]+RC[-8]/10^9/24/3600+RC[-1]/(3*10^5)/24/3600"
    Range("K6").Select
    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    ActiveSheet.Paste
    Application.CutCopyMode = False


End Sub
```

A.2.5. Data Import ALDIS

```
Sub DatenLesenAldis()
'
' DatenLesenServer Makro
'
Dim fNameAndPath As Variant           ' enthält den Dateiname der Datei die ↩
    geöffnet werden soll
Dim FileNum As Long                   ' freie file nummer zum Öffnen herausfinden
Dim DataLine As String                ' Daten einer Zeile der Datei
Dim Line As Long                      ' Zeile in der in Excel geschrieben wird

    ' öffne einen Dialog der nach Dateinamen frägt, Abbruch, wenn keine Datei ↩
        gewählt wird
    fNameAndPath = Application.GetOpenFilename(FileFilter:="TEXT (*.txt), *.txt, ↩
        CSV (*.csv), *.csv", Title:="Select File To Be Opened")
    If fNameAndPath = False Then Exit Sub

    'Daten markieren und löschen

    Range(Cells(6, 1), Cells(1048576, 16384).End(xlDown)).Select
    Selection.Delete Shift:=xlUp
```

```

    Close                               'alle ev. geöffneten Dateien schließen, sollte nicht erforderlich sein, aber gerade beim debuggen bleiben immer geöffnete Dateien (Abbruch vor "close") zurück, dann geht excel irgenwann die freien Kanäle aus
    FileNum = FreeFile()                ' freien Kanal finden (normalerweise 1)
    Line = 6

     'Textdatei importieren'
     With ActiveSheet.QueryTables.Add(Connection:= _
        "TEXT;" & fNameAndPath _
        , Destination:=Range("$A$6"))
        .Name = "test_1"
        .FieldNames = False
        .RowNumbers = False
        .FillAdjacentFormulas = True
        .PreserveFormatting = True
        .RefreshOnFileOpen = False
        .RefreshStyle = xlOverwriteCells
        .SavePassword = False
        .SaveData = True
        .AdjustColumnWidth = False
        .RefreshPeriod = 0
        .TextFilePromptOnRefresh = False
        .TextFilePlatform = 850
        .TextFileStartRow = 4
        .TextFileParseType = xlDelimited
        .TextFileTextQualifier = xlTextQualifierDoubleQuote
        .TextFileConsecutiveDelimiter = False
        .TextFileTabDelimiter = True
        .TextFileSemicolonDelimiter = True
        .TextFileCommaDelimiter = False
        .TextFileSpaceDelimiter = False
        .TextFileColumnDataTypes = Array(xlSkipColumn, 1, 1, 1, 1, 1, 1, 1, 1, 1)
        .TextFileTrailingMinusNumbers = True
        .TextFileStartRow = 2
        .Refresh BackgroundQuery:=False
    End With

    ActiveSheet.QueryTables(1).Delete 'Anfrage löschen

    'Einfügen der Formel zur Berechnung der Zeit mit Nano und Laufzeit

    Range("I7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(0, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "1/1/1900  12:00:00 AM"
    Range("J7").Select
    ActiveCell.FormulaR1C1 = _
        "=RC[-9]+RC[-8]+RC[-1]/10^9/24/3600+RC[-3]/(3*10^5)/24/3600"
    Range("J7").Select

    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    ActiveSheet.Paste
    Application.CutCopyMode = False

    'Löschen von unnötigen Informationen
    Rows("6:6").Select
    Selection.Delete Shift:=xlUp

End Sub
```

A.2.6. Data Comparison ALDIS versus Sensor

```
Sub Aldis_Sensor()
'
' Datenvergleich zwischen Aldis und Sensor

Dim F_end As String 'Gibt das Ende der Daten an
Dim I As Integer


    'Filter löschen

    Sheets("Aldis_Sensor").Select
    Range("A6:F6").Select
    Selection.AutoFilter

    'Daten markieren und löschen

    Range(Cells(6, 1), Cells(1048576, 16384).End(xlDown)).Select
    Selection.Delete Shift:=xlUp

    'Format löschen
    With Selection.Interior
        .Pattern = xlNone
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With

    'Einfügen der relevanten Blitzdaten von Aldis
    Sheets("Aldis").Select
    Range("J6").Select
    Range(Selection, Selection.End(xlDown)).Select
    Selection.Copy
    Sheets("Aldis_Sensor").Select
    ActiveSheet.Range("A7").Select
    'ActiveSheet.Paste
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Sheets("Aldis").Select
    Range("E6:G6").Select
    Range(Selection, Selection.End(xlDown)).Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Aldis_Sensor").Select
    Range("D7").Select
    ActiveSheet.Paste

    'Einfügen der Sensordaten
    Sheets("Blitzortung_Sensor").Select
    Range("E6").Select
    Range(Selection, Selection.End(xlDown)).Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Aldis_Sensor").Select
    Range("A7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(1, 0).Range("A1").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False
    'Daten gelb markieren
    With Selection.Interior
```

```
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 10092543
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With

    'Grenzen des Datenbereichs bestimmen
    Range("A7").Select
    Selection.End(xlDown).Select
    A_End = ActiveCell.Address
    ActiveCell.Offset(0, 5).Select
    F_end = ActiveCell.Address

    'Einfügen der Formel zur Berechnung des Timeshifts
    Range("B7").Select
    ActiveCell.FormulaR1C1 = "=(RC[-1]-R[-1]C[-1])*24*3600*10^6"
    Range("A7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(0, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "1"
    'Selection.End(xlUp).Select
    Range("B7").Select
    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    ActiveSheet.Paste
    Application.CutCopyMode = False

    'Einfügen der Formel für den Flag zum Filtern der Daten
    Range("C7").Select
    'alle Blitzdaten, die bezügl. der Zeit auf 30 mycrosekunden genau übereinstimmen werden mit 1 markiert
    ActiveCell.FormulaR1C1 = _
        "=IF(AND(RC2<30,RC2<>""""",farbe<>farbe_oben),1,IF(AND(R[1]C2<30,R[1]C2<>"""",farbe<>farbe_unten),1,0))"
    Range("B7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(0, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "1"
    Range("C7").Select
    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    ActiveSheet.Paste
    Application.CutCopyMode = False

    'restliche Sensordaten gelb markieren
    Range("D7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(1, -2).Range("A1").Select
    Range(Selection, ActiveCell.Offset(0, 4)).Select
    Range(Selection, Selection.End(xlDown)).Select

    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 10092543
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With

    'Sortieren der Daten nach dem Zeitstempel
    Range("A7").Select
    Range("A7:F7").Select
```

```
    Range(Selection, Selection.End(xlDown)).Select
    ActiveWorkbook.Worksheets("Aldis_Sensor").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("Aldis_Sensor").Sort.SortFields.Add Key:=Range( _
        "A7", A_End), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
        xlSortNormal
    With ActiveWorkbook.Worksheets("Aldis_Sensor").Sort
        .SetRange Range("A7", F_end)
        .Header = xlGuess
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With

    'Filter setzen
    ActiveSheet.Range("A6", "F6").AutoFilter Field:=3, Criteria1:="=1", _
        Operator:=xlAnd

    Range("A7").Select
End Sub
```

A.2.7. Data Comparison ALDIS versus Blitzortung.org

```
Sub Aldis_Server()
'
' Datenvergleich zwischen ALDIS und Blitzortung

Dim A_End, H_end As String

    'Filter löschen
    Sheets("Aldis_Server").Select
    Range("A6:H6").Select
    Selection.AutoFilter

    'Daten markieren und löschen

    Range(Cells(6, 1), Cells(1048576, 16384).End(xlDown)).Select
    Selection.Delete Shift:=xlUp

    'Format löschen
    With Selection.Interior
        .Pattern = xlNone
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With

    'Einfügen der relevanten Blitzdaten von Aldis
    Sheets("Aldis").Select
    Range("J6").Select
    Range(Selection, Selection.End(xlDown)).Select
    Selection.Copy
    Sheets("Aldis_Server").Select
    ActiveSheet.Range("A7").Select
    'ActiveSheet.Paste
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Sheets("Aldis").Select
    Range("C6:G6").Select
    Range(Selection, Selection.End(xlDown)).Select
```

```
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Aldis_Server").Select
    Range("D7").Select
    ActiveSheet.Paste

    'Einfügen der relevanten Blitzdaten von Blitzortung
    Sheets("Blitzortung_Server").Select
    Range("K6").Select
    Range(Selection, Selection.End(xlDown)).Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Aldis_Server").Select
    Range("A7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(1, 0).Range("A1").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False
    'Daten gelb markieren
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 10092543
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With

    Sheets("Blitzortung_Server").Select
    Range("D6:E6").Select
    Range(Selection, Selection.End(xlDown)).Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Aldis_Server").Select
    Range("D7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(1, 0).Range("A1").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False

    Sheets("Blitzortung_Server").Select
    Range("J6").Select
    Range(Selection, Selection.End(xlDown)).Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Aldis_Server").Select
    Range("H7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(1, 0).Range("A1").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False

    'Grenzen des Datenbereichs bestimmen
    Range("A7").Select
    Selection.End(xlDown).Select
    A_End = ActiveCell.Address
    ActiveCell.Offset(0, 7).Select
    H_end = ActiveCell.Address

    'Einfügen der Formel zur Berechnung des Timeshifts
    Range("B7").Select
```

```
    ActiveCell.FormulaR1C1 = "=(RC[-1]-R[-1]C[-1])*24*3600*10^6"
    Range("A7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(0, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "1"
    Range("B7").Select
    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    ActiveSheet.Paste
    Application.CutCopyMode = False

    'Einfügen der Formel für den Flag zum Filtern der Daten
    Range("C7").Select
    'alle Blitzdaten, die bezügl. der Zeit auf 30 mycrosekunden genau übereinstimmen werden mit 1 markiert
    ActiveCell.FormulaR1C1 = _
        "=IF(AND(RC2<30,RC2<>"""",farbe<>farbe_oben),1,IF(AND(R[1]C2<30,R[1]C2<>"""",farbe<>farbe_unten),1,0))"
    Range("B7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(0, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "1"
    Range("C7").Select
    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    ActiveSheet.Paste
    Application.CutCopyMode = False

    'restliche Serverdaten gelb markieren
    Range("F7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(1, -4).Range("A1").Select
    Range(Selection, ActiveCell.Offset(0, 6)).Select
    Range(Selection, Selection.End(xlDown)).Select

    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 10092543
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With

    'Sortieren der Daten nach dem Zeitstempel
    Range("A7").Select
    Range("A7:H7").Select
    Range(Selection, Selection.End(xlDown)).Select
    ActiveWorkbook.Worksheets("Aldis_Server").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("Aldis_Server").Sort.SortFields.Add Key:=Range( _
        "A7", A_End), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
        xlSortNormal
    With ActiveWorkbook.Worksheets("Aldis_Server").Sort
        .SetRange Range("A7", H_end)
        .Header = xlGuess
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With

    'Filter setzen
    ActiveSheet.Range("A6", "H6").AutoFilter Field:=3, Criteria1:="=1", _
        Operator:=xlAnd
```

```
    Range("A7").Select

End Sub
```

A.2.8. Data Comparison Sensor versus Blitzortung.org

```
Sub Sensor_Server()
'
' Datenvergleich zwischen Sensor und Server

Dim F_end As String

    'Filter löschen
    Sheets("Sensor_Server").Select
    Range("A6:F6").Select
    Selection.AutoFilter

    'Daten markieren und löschen

    Range(Cells(6, 1), Cells(1048576, 16384).End(xlDown)).Select
    Selection.Delete Shift:=xlUp

    'Format löschen
    With Selection.Interior
        .Pattern = xlNone
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With

    'Einfügen der relevanten Blitzdaten von Blitzortung
    Sheets("Blitzortung_Server").Select
    Range("K6").Select
    Range(Selection, Selection.End(xlDown)).Select
    Selection.Copy
    Sheets("Sensor_Server").Select
    ActiveSheet.Range("A7").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Sheets("Blitzortung_Server").Select
    Range("D6:E6").Select
    Range(Selection, Selection.End(xlDown)).Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Sensor_Server").Select
    Range("D7").Select
    ActiveSheet.Paste

    Sheets("Blitzortung_Server").Select
    Range("J6").Select
    Range(Selection, Selection.End(xlDown)).Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Sensor_Server").Select
    Range("F7").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False

    'Einfügen der Sensordaten
```

```
    Sheets("Blitzortung_Sensor").Select
    Range("E6").Select
    Range(Selection, Selection.End(xlDown)).Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Sensor_Server").Select
    Range("A7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(1, 0).Range("A1").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False
    'Daten gelb markieren
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 10092543
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With

    'Grenzen des Datenbereichs bestimmen
    Range("A7").Select
    Selection.End(xlDown).Select
    A_End = ActiveCell.Address
    ActiveCell.Offset(0, 5).Select
    F_end = ActiveCell.Address

    'Einfügen der Formel zur Berechnung des Timeshifts
    Range("B7").Select
    ActiveCell.FormulaR1C1 = "=(RC[-1]-R[-1]C[-1])*24*3600*10^6"
    Range("A7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(0, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "1"
    Range("B7").Select
    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    ActiveSheet.Paste
    Application.CutCopyMode = False

    'Einfügen der Formel für den Flag zum Filtern der Daten
    Range("C7").Select
    'alle Blitzdaten, die bezügl. der Zeit auf 30 mycrosekunden genau übereinstimmen werden mit 1 markiert
    ActiveCell.FormulaR1C1 = _
        "=IF(AND(RC2<30,RC2<>"""",farbe<>farbe_oben),1,IF(AND(R[1]C2<30,R[1]C2<>"""",farbe<>farbe_unten),1,0))"
    Range("B7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(0, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "1"
    Range("C7").Select
    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    ActiveSheet.Paste
    Application.CutCopyMode = False

    'restliche Sensordaten gelb markieren
    Range("D7").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(1, -2).Range("A1").Select
    Range(Selection, ActiveCell.Offset(0, 4)).Select
```

```
    Range(Selection, Selection.End(xlDown)).Select

    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 10092543
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With

    'Sortieren der Daten nach dem Zeitstempel
    Range("A7").Select
    Range("A7:F7").Select
    Range(Selection, Selection.End(xlDown)).Select
    ActiveWorkbook.Worksheets("Sensor_Server").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("Sensor_Server").Sort.SortFields.Add Key:=Range( _
        "A7", A_End), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
        xlSortNormal
    With ActiveWorkbook.Worksheets("Sensor_Server").Sort
        .SetRange Range("A7", F_end)
        .Header = xlGuess
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With

    'Filter setzen
    ActiveSheet.Range("$A$6", "F6").AutoFilter Field:=3, Criteria1:="=1", _
        Operator:=xlAnd

    Range("A7").Select

End Sub
```

A.2.9. Orthodrome

```

Public Function Orthodrome(Latitude_A As Double, Longitude_A As Double, Latitude_B ↩
    As Double, Longitude_B As Double) As Double
' Written by Hannes Pichler (ALDIS)
' Distance in km (accuracy ±50m all over the earth)
' http://de.wikipedia.org/wiki/Orthodrome
'
Dim F, G, L, S, C, W, D, R, H1, H2 As Double

Const FE = 1 / 298.257223563
Const AE = 6378137 / 1000
Const rad = 3.1415927 / 180

    F = (Latitude_A + Latitude_B) / 2 * rad
    G = (Latitude_A - Latitude_B) / 2 * rad
    L = (Longitude_A - Longitude_B) / 2 * rad

    S = Sin(G) ^ 2 * Cos(L) ^ 2 + Cos(F) ^ 2 * Sin(L) ^ 2
    C = Cos(G) ^ 2 * Cos(L) ^ 2 + Sin(F) ^ 2 * Sin(L) ^ 2
    W = Atn(Sqr(S / C))
    D = 2 * W * AE

```

```
    R = Sqr(S * C) / W
    H1 = (3 * R - 1) / 2 / C
    H2 = (3 * R + 1) / 2 / S

    Orthodrome = D * (1 + FE * H1 * Sin(F) ^ 2 * Cos(G) ^ 2 - FE * H2 * Cos(F) ^ 2 ↩
        * Sin(G) ^ 2)

End Function
```

A.2.10. Negative CG-strokes

```
Sub negWolkeErde()
'
' Auswertung der negativen Wolke-Erde Blitze nach Distanz und Stärke

Dim dlength As Long  'Länge des Datensatzes
Dim begin_row, row As Integer 'erste Zeile
Dim strength_low, strength_high As Integer 'Intervallgrenzen Stromstärke
Dim distance_low, distance_high As Integer 'Intervallgrenzen Distanz
Dim I As Integer
Dim source As String    'Arbeitsblatt auf das die Matrixformel angewandt wird


    'Analyse der Blitzdaten von Aldis (Referenz) nach Stromstärke und Distanz

    'Filter im Arbeitsblatt Aldis löschen
    Sheets("Aldis").Select
    Range("A5:J5").Select
    Selection.AutoFilter
    Selection.AutoFilter
    Range("A6").Select
    Selection.End(xlDown).Select
    dlength = ActiveCell.row
    source = ActiveSheet.Name

    'Löschen der alten Daten
    Sheets("neg_CG-Strokes").Select
    Range("B6").Select
    row = ActiveCell.row
    Range(Selection, ActiveCell.Offset(7, 0)).Select
    Range(Selection, ActiveCell.Offset(0, 11)).Select
    Selection.ClearContents

    'Anfangswerte festlegen
    begin_row = row
    strength_low = 0
    strength_high = -5 'negative CG-Blitze
    distance_low = 0
    distance_high = 50

    'Einfügen der Matrixformeln
    I = 2
    For I = 2 To 13
        If distance_high <= 150 Then
           Do While Cells(13, I).Value = ""
                Select Case strength_low
                    Case Is >= -10
                        'Matrixformel: summe(intra-cloud=0, distance_low<distance<=↩
                            distance_high, strength_high<=strength<strength_low)
```

```
                    Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=0)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(" & source & "!R6C5:R" & dlength & "C5< " & strength_low & ")*(" & source & "!R6C5:R" & dlength & "C5>= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high - 5 'Stromstärkeintervall um 5A erhöhen
                    row = row + 1
                Case Is >= -40
                    'Matrixformel: summe(intra-cloud=0, distance_low<distance<=distance_high, strength_high<=strength<strength_low)
                    Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=0)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(" & source & "!R6C5:R" & dlength & "C5< " & strength_low & ")*(" & source & "!R6C5:R" & dlength & "C5>= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high - 10 'Stromstärkeintervall um 10A erhöhen
                    row = row + 1
                Case Else
                    'Matrixformel: summe(intra-cloud=0, distance_low<distance<=distance_high, strength_high<=strength<strength_low)
                    Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=0)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(" & source & "!R6C5:R" & dlength & "C5< " & strength_low & "))"
            End Select
        Loop
       distance_low = distance_high
       distance_high = distance_high + 50 'Distanzintervall um 50 km erhöhen
    ElseIf distance_low <= 1000 Then
        Do While Cells(13, I).Value = ""
            Select Case strength_low
                Case Is >= -10
                    'Matrixformel: summe(intra-cloud=0, distance_low<distance<=distance_high, strength_high<=strength<strength_low)
                    Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=0)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(" & source & "!R6C5:R" & dlength & "C5< " & strength_low & ")*(" & source & "!R6C5:R" & dlength & "C5>= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high - 5  'Stromstärkeintervall um 5A erhöhen
                    row = row + 1
                Case Is >= -40
                    'Matrixformel: summe(intra-cloud=0, distance_low<distance<=distance_high, strength_high<=strength<strength_low)
                    Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
```

```
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=0)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(" & source & "!R6C5:R" & dlength & "C5< " & strength_low & ")*(" & source & "!R6C5:R" & dlength & "C5>= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high - 10  'Stromstärkeintervall um 10A erhöhen
                    row = row + 1
                Case Else
                    'Matrixformel: summe(intra-cloud=0, distance_low<distance<=distance_high, strength_high<=strength<strength_low)
                    Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=0)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(" & source & "!R6C5:R" & dlength & "C5< " & strength_low & "))"
            End Select
        Loop
        distance_low = distance_high
        distance_high = distance_high + 100 'Distanzintervall um 100 km erhöhen
    End If
    'Zurücksetzen auf die Anfangswerte
    row = begin_row
    strength_low = 0
    strength_high = -5
Next

'Analyse der zeitkorrelierten Blitzdaten von Blitzortung nach Stromstärke und Distanz

'Filter im Arbeitsblatt Aldis_Server löschen
Sheets("Aldis_Server").Select
Range("A6:H6").Select
Selection.AutoFilter
Selection.AutoFilter
Range("A7").Select
Selection.End(xlDown).Select
dlength = ActiveCell.row
source = ActiveSheet.Name

'Löschen der alten Daten
Sheets("neg_CG-Strokes").Select
Range("B18").Select
row = ActiveCell.row
Range(Selection, ActiveCell.Offset(7, 0)).Select
Range(Selection, ActiveCell.Offset(0, 11)).Select
Selection.ClearContents

'Anfangswerte festlegen
begin_row = row
strength_low = 0
strength_high = -5
distance_low = 0
distance_high = 50

'Einfügen der Matrixformeln
I = 2
For I = 2 To 13
```

```
        If distance_high <= 150 Then
           Do While Cells(25, I).Value = ""
                Select Case strength_low
                    Case Is >= -10
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<↩
                            distance<=distance_high, strength_high<=strength<↩
                            strength_low)
                        Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=0)*(↩
                                Aldis_Server!R7C3:R" & dlength & "C3=1)*(↩
                                Aldis_Server!R7C8:R" & dlength & "C8> " & ↩
                                distance_low & ")*(Aldis_Server!R7C8:R" & dlength & ↩
                                "C8<= " & distance_high & ")" & _
                            " *(Aldis_Server!R7C6:R" & dlength & "C6< " & ↩
                                strength_low & ")*(Aldis_Server!R7C6:R" & dlength & ↩
                                "C6>= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high - 5   'Stromstärkeintervall ↩
                            um 5A erhöhen
                        row = row + 1
                    Case Is >= -40
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<↩
                            distance<=distance_high, strength_high<=strength<↩
                            strength_low)
                        Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=0)*(↩
                                Aldis_Server!R7C3:R" & dlength & "C3=1)*(↩
                                Aldis_Server!R7C8:R" & dlength & "C8> " & ↩
                                distance_low & ")*(Aldis_Server!R7C8:R" & dlength & ↩
                                "C8<= " & distance_high & ")" & _
                            " *(Aldis_Server!R7C6:R" & dlength & "C6< " & ↩
                                strength_low & ")*(Aldis_Server!R7C6:R" & dlength & ↩
                                "C6>= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high - 10  'Stromstärkeintervall ↩
                            um 10A erhöhen
                        row = row + 1
                    Case Else
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<↩
                            distance<=distance_high, strength_high<=strength<↩
                            strength_low)
                        Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=0)*(↩
                                Aldis_Server!R7C3:R" & dlength & "C3=1)*(↩
                                Aldis_Server!R7C8:R" & dlength & "C8> " & ↩
                                distance_low & ")*(Aldis_Server!R7C8:R" & dlength & ↩
                                "C8<= " & distance_high & ")" & _
                            " *(Aldis_Server!R7C6:R" & dlength & "C6< " & ↩
                                strength_low & "))"
                End Select
            Loop
           distance_low = distance_high
           distance_high = distance_high + 50   'Distanzintervall um 50 km erhöhen
        ElseIf distance_low <= 1000 Then
            Do While Cells(25, I).Value = ""
                Select Case strength_low
                    Case Is >= -10
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<↩
                            distance<=distance_high, strength_high<=strength<↩
                            strength_low)
                        Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=0)*(↩
                                Aldis_Server!R7C3:R" & dlength & "C3=1)*(↩
```

```
                                Aldis_Server!R7C8:R" & dlength & "C8> " & ↩
                                distance_low & ")*(Aldis_Server!R7C8:R" & dlength & ↩
                                "C8<= " & distance_high & ")" & _
                            " *(Aldis_Server!R7C6:R" & dlength & "C6< " & ↩
                                strength_low & ")*(Aldis_Server!R7C6:R" & dlength & ↩
                                "C6>= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high - 5   'Stromstärkeintervall ↩
                            um 5A erhöhen
                        row = row + 1
                    Case Is >= -40
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<↩
                            distance<=distance_high, strength_high<=strength<↩
                            strength_low)
                        Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=0)*(↩
                                Aldis_Server!R7C3:R" & dlength & "C3=1)*(↩
                                Aldis_Server!R7C8:R" & dlength & "C8> " & ↩
                                distance_low & ")*(Aldis_Server!R7C8:R" & dlength & ↩
                                "C8<= " & distance_high & ")" & _
                            " *(Aldis_Server!R7C6:R" & dlength & "C6< " & ↩
                                strength_low & ")*(Aldis_Server!R7C6:R" & dlength & ↩
                                "C6>= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high - 10  'Stromstärkeintervall ↩
                            um 10A erhöhen
                        row = row + 1
                    Case Else
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<↩
                            distance<=distance_high, strength_high<=strength<↩
                            strength_low)
                        Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=0)*(↩
                                Aldis_Server!R7C3:R" & dlength & "C3=1)*(↩
                                Aldis_Server!R7C8:R" & dlength & "C8> " & ↩
                                distance_low & ")*(Aldis_Server!R7C8:R" & dlength & ↩
                                "C8<= " & distance_high & ")" & _
                            " *(Aldis_Server!R7C6:R" & dlength & "C6< " & ↩
                                strength_low & "))"
                End Select
            Loop
            distance_low = distance_high
            distance_high = distance_high + 100 'Distanzintervall um 100 km erhöhen
        End If
        'Zurücksetzen auf die Anfangswerte
        row = begin_row
        strength_low = 0
        strength_high = -5
    Next

    'Analyse der zeitkorrelierten Blitzdaten vom Sensor nach Stromstärke und ↩
        Distanz

    'Filter im Arbeitsblatt Aldis_Server löschen
    Sheets("Aldis_Sensor").Select
    Range("A6:F6").Select
    Selection.AutoFilter
    Selection.AutoFilter
    Range("A7").Select
    Selection.End(xlDown).Select
    dlength = ActiveCell.row
    source = ActiveSheet.Name

```

```
    'Löschen der alten Daten
    Sheets("neg_CG-Strokes").Select
    Range("B30").Select
    row = ActiveCell.row
    Range(Selection, ActiveCell.Offset(7, 0)).Select
    Range(Selection, ActiveCell.Offset(0, 11)).Select
    Selection.ClearContents


    'Anfangswerte festlegen
    begin_row = row
    strength_low = 0
    strength_high = -5
    distance_low = 0
    distance_high = 50

    'Einfügen der Matrixformeln
    I = 2
    For I = 2 To 13
        If distance_high <= 150 Then
           Do While Cells(37, I).Value = ""
                Select Case strength_low
                    Case Is >= -10
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<↩
                            distance<=distance_high, strength_high<=strength<↩
                            strength_low)
                        Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=0)*(" & ↩
                                source & "!R7C3:R" & dlength & "C3=1)*(" & source & ↩
                                "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" ↩
                                & source & "!R7C6:R" & dlength & "C6<= " & ↩
                                distance_high & ")" & _
                            " *(" & source & "!R7C4:R" & dlength & "C4< " & ↩
                                strength_low & ")*(" & source & "!R7C4:R" & dlength ↩
                                & "C4>= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high - 5   'Stromstärkeintervall ↩
                            um 5A erhöhen
                        row = row + 1
                    Case Is >= -40
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<↩
                            distance<=distance_high, strength_high<=strength<↩
                            strength_low)
                        Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=0)*(" & ↩
                                source & "!R7C3:R" & dlength & "C3=1)*(" & source & ↩
                                "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" ↩
                                & source & "!R7C6:R" & dlength & "C6<= " & ↩
                                distance_high & ")" & _
                            " *(" & source & "!R7C4:R" & dlength & "C4< " & ↩
                                strength_low & ")*(" & source & "!R7C4:R" & dlength ↩
                                & "C4>= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high - 10   'Stromstärkeintervall ↩
                            um 10A erhöhen
                        row = row + 1
                    Case Else
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<↩
                            distance<=distance_high, strength_high<=strength<↩
                            strength_low)
                        Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=0)*(" & ↩
                                source & "!R7C3:R" & dlength & "C3=1)*(" & source & ↩
```

```
                                "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" ↩
                                & source & "!R7C6:R" & dlength & "C6<= " & ↩
                                distance_high & ")" & _
                            " *(" & source & "!R7C4:R" & dlength & "C4< " & ↩
                                strength_low & "))"
                End Select
            Loop
           distance_low = distance_high
           distance_high = distance_high + 50   'Distanzintervall um 50 km erhöhen
        ElseIf distance_low <= 1000 Then
            Do While Cells(37, I).Value = ""
                Select Case strength_low
                    Case Is >= -10
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<↩
                            distance<=distance_high, strength_high<=strength<↩
                            strength_low)
                        Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=0)*(" & ↩
                                source & "!R7C3:R" & dlength & "C3=1)*(" & source & ↩
                                "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" ↩
                                & source & "!R7C6:R" & dlength & "C6<= " & ↩
                                distance_high & ")" & _
                            " *(" & source & "!R7C4:R" & dlength & "C4< " & ↩
                                strength_low & ")*(" & source & "!R7C4:R" & dlength ↩
                                & "C4>= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high - 5   'Stromstärkeintervall ↩
                            um 5A erhöhen
                        row = row + 1
                    Case Is >= -40
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<↩
                            distance<=distance_high, strength_high<=strength<↩
                            strength_low)
                        Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=0)*(" & ↩
                                source & "!R7C3:R" & dlength & "C3=1)*(" & source & ↩
                                "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" ↩
                                & source & "!R7C6:R" & dlength & "C6<= " & ↩
                                distance_high & ")" & _
                            " *(" & source & "!R7C4:R" & dlength & "C4< " & ↩
                                strength_low & ")*(" & source & "!R7C4:R" & dlength ↩
                                & "C4>= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high - 10   'Stromstärkeintervall ↩
                            um 10A erhöhen
                        row = row + 1
                    Case Else
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<↩
                            distance<=distance_high, strength_high<=strength<↩
                            strength_low)
                        Worksheets("neg_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=0)*(" & ↩
                                source & "!R7C3:R" & dlength & "C3=1)*(" & source & ↩
                                "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" ↩
                                & source & "!R7C6:R" & dlength & "C6<= " & ↩
                                distance_high & ")" & _
                            " *(" & source & "!R7C4:R" & dlength & "C4< " & ↩
                                strength_low & "))"
                End Select
            Loop
            distance_low = distance_high
            distance_high = distance_high + 100 'Distanzintervall um 100 km erhöhen
        End If
```

```
        'Zurücksetzen auf die Anfangswerte
        row = begin_row
        strength_low = 0
        strength_high = -5
    Next

End Sub
```

A.2.11. Positive CG-strokes

```

Sub posWolkeErde()
'
' Auswertung der positiven Wolke-Erde Blitze nach Distanz und Stärke

Dim dlength As Long  'Länge des Datensatzes
Dim begin_row, row As Integer 'erste Zeile
Dim strength_low, strength_high As Integer 'Intervallgrenzen Stromstärke
Dim distance_low, distance_high As Integer 'Intervallgrenzen Distanz
Dim I As Integer
Dim source As String    'Arbeitsblatt auf das die Matrixformel angewandt wird

    'Analyse der Blitzdaten von Aldis (Referenz) nach Stromstärke und Distanz

    'Filter im Arbeitsblatt Aldis löschen
    Sheets("Aldis").Select
    Range("A5:J5").Select
    Selection.AutoFilter
    Selection.AutoFilter
    Range("A6").Select
    Selection.End(xlDown).Select
    dlength = ActiveCell.row
    source = ActiveSheet.Name

    'Löschen der alten Daten
    Sheets("pos_CG-Strokes").Select
    Range("B6").Select
    row = ActiveCell.row
    Range(Selection, ActiveCell.Offset(7, 0)).Select
    Range(Selection, ActiveCell.Offset(0, 11)).Select
    Selection.ClearContents

    'Anfangswerte festlegen
    begin_row = row
    strength_low = 0
    strength_high = 5   'positive CG-Blitze
    distance_low = 0
    distance_high = 50

    'Einfügen der Matrixformeln
    I = 2
    For I = 2 To 13
        If distance_high <= 150 Then
           Do While Cells(13, I).Value = ""
                Select Case strength_low
                    Case Is <= 10
                        'Matrixformel: summe(intra-cloud=0, distance_low<distance<=↩
                            distance_high, strength_low<strength<=strength_high)
                        Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
```

```
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=0)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(" & source & "!R6C5:R" & dlength & "C5> " & strength_low & ")*(" & source & "!R6C5:R" & dlength & "C5<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 5
                    row = row + 1
                Case Is <= 40
                    'Matrixformel: summe(intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                    Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=0)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(" & source & "!R6C5:R" & dlength & "C5> " & strength_low & ")*(" & source & "!R6C5:R" & dlength & "C5<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 10  'Stromstärkeintervall um 10A erhöhen
                    row = row + 1
                Case Else
                    'Matrixformel: summe(intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                    Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=0)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(" & source & "!R6C5:R" & dlength & "C5> " & strength_low & "))"
            End Select
        Loop
        distance_low = distance_high
        distance_high = distance_high + 50   'Distanzintervall um 50 km erhöhen
    ElseIf distance_low <= 1000 Then
        Do While Cells(13, I).Value = ""
            Select Case strength_low
                Case Is <= 10
                    'Matrixformel: summe(intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                    Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=0)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(" & source & "!R6C5:R" & dlength & "C5> " & strength_low & ")*(" & source & "!R6C5:R" & dlength & "C5<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 5   'Stromstärkeintervall um 5A erhöhen
                    row = row + 1
                Case Is <= 40
                    'Matrixformel: summe(intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                    Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=0)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low
```

```
                                & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                            " *(" & source & "!R6C5:R" & dlength & "C5> " & strength_low & ")*(" & source & "!R6C5:R" & dlength & "C5<= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high + 10  'Stromstärkeintervall um 10A erhöhen
                        row = row + 1
                    Case Else
                        'Matrixformel: summe(intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                        Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R6C6:R" & dlength & "C6=0)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                            " *(" & source & "!R6C5:R" & dlength & "C5> " & strength_low & "))"
                End Select
            Loop
            distance_low = distance_high
            distance_high = distance_high + 100 'Distanzintervall um 100 km erhöhen
        End If
        'Zurücksetzen auf die Anfangswerte
        row = begin_row
        strength_low = 0
        strength_high = 5
    Next

    'Analyse der zeitkorrelierten Blitzdaten von Blitzortung nach Stromstärke und Distanz

    'Filter im Arbeitsblatt Aldis_Server löschen
    Sheets("Aldis_Server").Select
    Range("A6:H6").Select
    Selection.AutoFilter
    Selection.AutoFilter
    Range("A7").Select
    Selection.End(xlDown).Select
    dlength = ActiveCell.row
    source = ActiveSheet.Name

    'Löschen der alten Daten
    Sheets("pos_CG-Strokes").Select
    Range("B18").Select
    row = ActiveCell.row
    Range(Selection, ActiveCell.Offset(7, 0)).Select
    Range(Selection, ActiveCell.Offset(0, 11)).Select
    Selection.ClearContents

    'Anfangswerte festlegen
    begin_row = row
    strength_low = 0
    strength_high = 5
    distance_low = 0
    distance_high = 50

    'Einfügen der Matrixformeln
    I = 2
    For I = 2 To 13
        If distance_high <= 150 Then
           Do While Cells(25, I).Value = ""
```

```
            Select Case strength_low
                Case Is <= 10
                    'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                    Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=0)*(Aldis_Server!R7C3:R" & dlength & "C3=1)*(Aldis_Server!R7C8:R" & dlength & "C8> " & distance_low & ")*(Aldis_Server!R7C8:R" & dlength & "C8<= " & distance_high & ")" & _
                        " *(Aldis_Server!R7C6:R" & dlength & "C6> " & strength_low & ")*(Aldis_Server!R7C6:R" & dlength & "C6<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 5   'Stromstärkeintervall um 5A erhöhen
                    row = row + 1
                Case Is <= 40
                    'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                    Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=0)*(Aldis_Server!R7C3:R" & dlength & "C3=1)*(Aldis_Server!R7C8:R" & dlength & "C8> " & distance_low & ")*(Aldis_Server!R7C8:R" & dlength & "C8<= " & distance_high & ")" & _
                        " *(Aldis_Server!R7C6:R" & dlength & "C6> " & strength_low & ")*(Aldis_Server!R7C6:R" & dlength & "C6<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 10  'Stromstärkeintervall um 10A erhöhen
                    row = row + 1
                Case Else
                    'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                    Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=0)*(Aldis_Server!R7C3:R" & dlength & "C3=1)*(Aldis_Server!R7C8:R" & dlength & "C8> " & distance_low & ")*(Aldis_Server!R7C8:R" & dlength & "C8<= " & distance_high & ")" & _
                        " *(Aldis_Server!R7C6:R" & dlength & "C6> " & strength_low & "))"
            End Select
        Loop
       distance_low = distance_high
       distance_high = distance_high + 50   'Distanzintervall um 50 km erhöhen
    ElseIf distance_low <= 1000 Then
        Do While Cells(25, I).Value = ""
            Select Case strength_low
                Case Is <= 10
                    'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                    Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=0)*(Aldis_Server!R7C3:R" & dlength & "C3=1)*(Aldis_Server!R7C8:R" & dlength & "C8> " & distance_low & ")*(Aldis_Server!R7C8:R" & dlength &
```

```
                            "C8<= " & distance_high & ")" & _
                        " *(Aldis_Server!R7C6:R" & dlength & "C6> " & strength_low & ")*(Aldis_Server!R7C6:R" & dlength & "C6<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 5   'Stromstärkeintervall um 5A erhöhen
                    row = row + 1
                Case Is <= 40
                    'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                    Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=0)*(Aldis_Server!R7C3:R" & dlength & "C3=1)*(Aldis_Server!R7C8:R" & dlength & "C8> " & distance_low & ")*(Aldis_Server!R7C8:R" & dlength & "C8<= " & distance_high & ")" & _
                        " *(Aldis_Server!R7C6:R" & dlength & "C6> " & strength_low & ")*(Aldis_Server!R7C6:R" & dlength & "C6<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 10  'Stromstärkeintervall um 10A erhöhen
                    row = row + 1
                Case Else
                    'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                    Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=0)*(Aldis_Server!R7C3:R" & dlength & "C3=1)*(Aldis_Server!R7C8:R" & dlength & "C8> " & distance_low & ")*(Aldis_Server!R7C8:R" & dlength & "C8<= " & distance_high & ")" & _
                        " *(Aldis_Server!R7C6:R" & dlength & "C6> " & strength_low & "))"
            End Select
        Loop
        distance_low = distance_high
        distance_high = distance_high + 100 'Distanzintervall um 100 km erhöhen
    End If
    'Zurücksetzen auf die Anfangswerte
    row = begin_row
    strength_low = 0
    strength_high = 5
Next

'Analyse der zeitkorrelierten Blitzdaten vom Sensor nach Stromstärke und Distanz

'Filter im Arbeitsblatt Aldis_Server löschen
Sheets("Aldis_Sensor").Select
Range("A6:F6").Select
Selection.AutoFilter
Selection.AutoFilter
Range("A7").Select
Selection.End(xlDown).Select
dlength = ActiveCell.row
source = ActiveSheet.Name

'Löschen der alten Daten
Sheets("pos_CG-Strokes").Select
```

```
    Range("B30").Select
    row = ActiveCell.row
    Range(Selection, ActiveCell.Offset(7, 0)).Select
    Range(Selection, ActiveCell.Offset(0, 11)).Select
    Selection.ClearContents


    'Anfangswerte festlegen
    begin_row = row
    strength_low = 0
    strength_high = 5
    distance_low = 0
    distance_high = 50

    'Einfügen der Matrixformeln
    I = 2
    For I = 2 To 13
        If distance_high <= 150 Then
           Do While Cells(37, I).Value = ""
                Select Case strength_low
                    Case Is <= 10
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                        Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=0)*(" & source & "!R7C3:R" & dlength & "C3=1)*(" & source & "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" & source & "!R7C6:R" & dlength & "C6<= " & distance_high & ")" & _
                            " *(" & source & "!R7C4:R" & dlength & "C4> " & strength_low & ")*(" & source & "!R7C4:R" & dlength & "C4<= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high + 5   'Stromstärkeintervall um 5A erhöhen
                        row = row + 1
                    Case Is <= 40
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                        Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=0)*(" & source & "!R7C3:R" & dlength & "C3=1)*(" & source & "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" & source & "!R7C6:R" & dlength & "C6<= " & distance_high & ")" & _
                            " *(" & source & "!R7C4:R" & dlength & "C4> " & strength_low & ")*(" & source & "!R7C4:R" & dlength & "C4<= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high + 10  'Stromstärkeintervall um 10A erhöhen
                        row = row + 1
                    Case Else
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                        Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=0)*(" & source & "!R7C3:R" & dlength & "C3=1)*(" & source & "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" & source & "!R7C6:R" & dlength & "C6<= " &
```

```
                                distance_high & ")" & _
                            " *(" & source & "!R7C4:R" & dlength & "C4> " & strength_low & "))"
                End Select
            Loop
           distance_low = distance_high
           distance_high = distance_high + 50   'Distanzintervall um 50 km erhöhen
        ElseIf distance_low <= 1000 Then
            Do While Cells(37, I).Value = ""
                Select Case strength_low
                    Case Is <= 10
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                        Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=0)*(" & source & "!R7C3:R" & dlength & "C3=1)*(" & source & "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" & source & "!R7C6:R" & dlength & "C6<= " & distance_high & ")" & _
                            " *(" & source & "!R7C4:R" & dlength & "C4> " & strength_low & ")*(" & source & "!R7C4:R" & dlength & "C4<= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high + 5   'Stromstärkeintervall um 5A erhöhen
                        row = row + 1
                    Case Is <= 40
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                        Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=0)*(" & source & "!R7C3:R" & dlength & "C3=1)*(" & source & "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" & source & "!R7C6:R" & dlength & "C6<= " & distance_high & ")" & _
                            " *(" & source & "!R7C4:R" & dlength & "C4> " & strength_low & ")*(" & source & "!R7C4:R" & dlength & "C4<= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high + 10  'Stromstärkeintervall um 10A erhöhen
                        row = row + 1
                    Case Else
                        'Matrixformel: summe(flag=1,intra-cloud=0, distance_low<distance<=distance_high, strength_low<strength<=strength_high)
                        Worksheets("pos_CG-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=0)*(" & source & "!R7C3:R" & dlength & "C3=1)*(" & source & "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" & source & "!R7C6:R" & dlength & "C6<= " & distance_high & ")" & _
                            " *(" & source & "!R7C4:R" & dlength & "C4> " & strength_low & "))"
                End Select
            Loop
            distance_low = distance_high
            distance_high = distance_high + 100 'Distanzintervall um 100 km erhöhen
        End If
        'Zurücksetzen auf die Anfangswerte
        row = begin_row
```

```
            strength_low = 0
            strength_high = 5
        Next


End Sub
```

A.2.12. CC-strokes

```

Sub WolkeWolke()
'
' Auswertung der Wolke-Wolke Blitze nach Distanz und Stärke

Dim dlength As Long  'Länge des Datensatzes
Dim begin_row, row As Integer 'erste Zeile
Dim strength_low, strength_high As Integer 'Intervallgrenzen Stromstärke
Dim distance_low, distance_high As Integer 'Intervallgrenzen Distanz
Dim I As Integer
Dim source As String    'Arbeitsblatt auf das die Matrixformel angewandt wird

    'Analyse der Blitzdaten von Aldis (Referenz) nach Stromstärke und Distanz

    'Filter im Arbeitsblatt Aldis löschen
    Sheets("Aldis").Select
    Range("A5:J5").Select
    Selection.AutoFilter
    Selection.AutoFilter
    Range("A6").Select
    Selection.End(xlDown).Select
    dlength = ActiveCell.row
    source = ActiveSheet.Name

    'Löschen der alten Daten
    Sheets("CC-Strokes").Select
    Range("B6").Select
    row = ActiveCell.row
    Range(Selection, ActiveCell.Offset(7, 0)).Select
    Range(Selection, ActiveCell.Offset(0, 11)).Select
    Selection.ClearContents

    'Anfangswerte festlegen
    begin_row = row
    strength_low = 0
    strength_high = 5
    distance_low = 0
    distance_high = 50


    'Einfügen der Matrixformeln
    I = 2
    For I = 2 To 13
        If distance_high <= 150 Then
           Do While Cells(13, I).Value = ""
                Select Case strength_low
                    Case Is <= 10
                        'Matrixformel: summe(intra-cloud=1, distance_low<distance<=↩
                            distance_high, strength_low<abs(strength)<=strength_high↩
```

```
                        )
                    Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=1)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(ABS(" & source & "!R6C5:R" & dlength & "C5)> " & strength_low & ")*(ABS(" & source & "!R6C5:R" & dlength & "C5)<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 5   'Stromstärkeintervall um 5A erhöhen
                    row = row + 1
                Case Is <= 40
                    'Matrixformel: summe(intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                    Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=1)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(ABS(" & source & "!R6C5:R" & dlength & "C5)> " & strength_low & ")*(ABS(" & source & "!R6C5:R" & dlength & "C5)<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 10  'Stromstärkeintervall um 10A erhöhen
                    row = row + 1
                Case Else
                    'Matrixformel: summe(intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                    Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=1)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(ABS(" & source & "!R6C5:R" & dlength & "C5)> " & strength_low & "))"
            End Select
        Loop
       distance_low = distance_high
       distance_high = distance_high + 50   'Distanzintervall um 50 km erhöhen
    ElseIf distance_low <= 1000 Then
        Do While Cells(13, I).Value = ""
            Select Case strength_low
                Case Is <= 10
                    'Matrixformel: summe(intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                    Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R6C6:R" & dlength & "C6=1)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                        " *(ABS(" & source & "!R6C5:R" & dlength & "C5)> " & strength_low & ")*(ABS(" & source & "!R6C5:R" & dlength & "C5)<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 5   'Stromstärkeintervall um 5A erhöhen
                    row = row + 1
```

```
                    Case Is <= 40
                        'Matrixformel: summe(intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                        Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R6C6:R" & dlength & "C6=1)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                            " *(ABS(" & source & "!R6C5:R" & dlength & "C5)> " & strength_low & ")*(ABS(" & source & "!R6C5:R" & dlength & "C5)<= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high + 10  'Stromstärkeintervall um 10A erhöhen
                        row = row + 1
                    Case Else
                        'Matrixformel: summe(intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                        Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R6C6:R" & dlength & "C6=1)*(" & source & "!R6C7:R" & dlength & "C7> " & distance_low & ")*(" & source & "!R6C7:R" & dlength & "C7<= " & distance_high & ")" & _
                            " *(ABS(" & source & "!R6C5:R" & dlength & "C5)> " & strength_low & "))"
                End Select
            Loop
            distance_low = distance_high
            distance_high = distance_high + 100 'Distanzintervall um 100 km erhöhen
        End If
        'Zurücksetzen auf die Anfangswerte
        row = begin_row
        strength_low = 0
        strength_high = 5
    Next

    'Analyse der zeitkorrelierten Blitzdaten von Blitzortung nach Stromstärke und Distanz

    'Filter im Arbeitsblatt Aldis_Server löschen
    Sheets("Aldis_Server").Select
    Range("A6:H6").Select
    Selection.AutoFilter
    Selection.AutoFilter
    Range("A7").Select
    Selection.End(xlDown).Select
    dlength = ActiveCell.row
    source = ActiveSheet.Name

    'Löschen der alten Daten
    Sheets("CC-Strokes").Select
    Range("B18").Select
    row = ActiveCell.row
    Range(Selection, ActiveCell.Offset(7, 0)).Select
    Range(Selection, ActiveCell.Offset(0, 11)).Select
    Selection.ClearContents


    'Anfangswerte festlegen
    begin_row = row
    strength_low = 0
```

```
strength_high = 5
distance_low = 0
distance_high = 50

'Einfügen der Matrixformeln
I = 2
For I = 2 To 13
    If distance_high <= 150 Then
       Do While Cells(25, I).Value = ""
            Select Case strength_low
                Case Is <= 10
                    'Matrixformel: summe(flag=1,intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                    Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=1)*(Aldis_Server!R7C3:R" & dlength & "C3=1)*(Aldis_Server!R7C8:R" & dlength & "C8> " & distance_low & ")*(Aldis_Server!R7C8:R" & dlength & "C8<= " & distance_high & ")" & _
                        " *(ABS(Aldis_Server!R7C6:R" & dlength & "C6)> " & strength_low & ")*(ABS(Aldis_Server!R7C6:R" & dlength & "C6)<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 5   'Stromstärkeintervall um 5A erhöhen
                    row = row + 1
                Case Is <= 40
                    'Matrixformel: summe(flag=1,intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                    Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=1)*(Aldis_Server!R7C3:R" & dlength & "C3=1)*(Aldis_Server!R7C8:R" & dlength & "C8> " & distance_low & ")*(Aldis_Server!R7C8:R" & dlength & "C8<= " & distance_high & ")" & _
                        " *(ABS(Aldis_Server!R7C6:R" & dlength & "C6)> " & strength_low & ")*(ABS(Aldis_Server!R7C6:R" & dlength & "C6)<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 10  'Stromstärkeintervall um 10A erhöhen
                    row = row + 1
                Case Else
                    'Matrixformel: summe(flag=1,intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                    Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=1)*(Aldis_Server!R7C3:R" & dlength & "C3=1)*(Aldis_Server!R7C8:R" & dlength & "C8> " & distance_low & ")*(Aldis_Server!R7C8:R" & dlength & "C8<= " & distance_high & ")" & _
                        " *(ABS(Aldis_Server!R7C6:R" & dlength & "C6)> " & strength_low & "))"
            End Select
        Loop
       distance_low = distance_high
       distance_high = distance_high + 50   'Distanzintervall um 50 km erhöhen
    ElseIf distance_low <= 1000 Then
        Do While Cells(25, I).Value = ""
            Select Case strength_low
```

```
                Case Is <= 10
                    'Matrixformel: summe(flag=1,intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                    Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=1)*(Aldis_Server!R7C3:R" & dlength & "C3=1)*(Aldis_Server!R7C8:R" & dlength & "C8> " & distance_low & ")*(Aldis_Server!R7C8:R" & dlength & "C8<= " & distance_high & ")" & _
                        " *(ABS(Aldis_Server!R7C6:R" & dlength & "C6)> " & strength_low & ")*(ABS(Aldis_Server!R7C6:R" & dlength & "C6)<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 5   'Stromstärkeintervall um 5A erhöhen
                    row = row + 1
                Case Is <= 40
                    'Matrixformel: summe(flag=1,intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                    Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=1)*(Aldis_Server!R7C3:R" & dlength & "C3=1)*(Aldis_Server!R7C8:R" & dlength & "C8> " & distance_low & ")*(Aldis_Server!R7C8:R" & dlength & "C8<= " & distance_high & ")" & _
                        " *(ABS(Aldis_Server!R7C6:R" & dlength & "C6)> " & strength_low & ")*(ABS(Aldis_Server!R7C6:R" & dlength & "C6)<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 10  'Stromstärkeintervall um 10A erhöhen
                    row = row + 1
                Case Else
                    'Matrixformel: summe(flag=1,intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                    Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((Aldis_Server!R7C7:R" & dlength & "C7=1)*(Aldis_Server!R7C3:R" & dlength & "C3=1)*(Aldis_Server!R7C8:R" & dlength & "C8> " & distance_low & ")*(Aldis_Server!R7C8:R" & dlength & "C8<= " & distance_high & ")" & _
                        " *(ABS(Aldis_Server!R7C6:R" & dlength & "C6)> " & strength_low & "))"
            End Select
        Loop
        distance_low = distance_high
        distance_high = distance_high + 100 'Distanzintervall um 100 km erhöhen
    End If
    'Zurücksetzen auf die Anfangswerte
    row = begin_row
    strength_low = 0
    strength_high = 5
Next

'Analyse der zeitkorrelierten Blitzdaten vom Sensor nach Stromstärke und Distanz

'Filter im Arbeitsblatt Aldis_Server löschen
Sheets("Aldis_Sensor").Select
Range("A6:F6").Select
```

```
Selection.AutoFilter
Selection.AutoFilter
Range("A7").Select
Selection.End(xlDown).Select
dlength = ActiveCell.row
source = ActiveSheet.Name

'Löschen der alten Daten
Sheets("CC-Strokes").Select
Range("B30").Select
row = ActiveCell.row
Range(Selection, ActiveCell.Offset(7, 0)).Select
Range(Selection, ActiveCell.Offset(0, 11)).Select
Selection.ClearContents


'Anfangswerte festlegen
begin_row = row
strength_low = 0
strength_high = 5
distance_low = 0
distance_high = 50

'Einfügen der Matrixformeln
I = 2
For I = 2 To 13
    If distance_high <= 150 Then
       Do While Cells(37, I).Value = ""
            Select Case strength_low
                Case Is <= 10
                    'Matrixformel: summe(flag=1,intra-cloud=1, distance_low<↩
                        distance<=distance_high, strength_low<abs(strength)<=↩
                        strength_high)
                    Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R7C5:R" & dlength & "C5=1)*(" & ↩
                            source & "!R7C3:R" & dlength & "C3=1)*(" & source & ↩
                            "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" ↩
                            & source & "!R7C6:R" & dlength & "C6<= " & ↩
                            distance_high & ")" & _
                        " *(ABS(" & source & "!R7C4:R" & dlength & "C4)> " & ↩
                            strength_low & ")*(ABS(" & source & "!R7C4:R" & ↩
                            dlength & "C4)<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 5   'Stromstärkeintervall ↩
                        um 5A erhöhen
                    row = row + 1
                Case Is <= 40
                    'Matrixformel: summe(flag=1,intra-cloud=1, distance_low<↩
                        distance<=distance_high, strength_low<abs(strength)<=↩
                        strength_high)
                    Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                        "=SUM((" & source & "!R7C5:R" & dlength & "C5=1)*(" & ↩
                            source & "!R7C3:R" & dlength & "C3=1)*(" & source & ↩
                            "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" ↩
                            & source & "!R7C6:R" & dlength & "C6<= " & ↩
                            distance_high & ")" & _
                        " *(ABS(" & source & "!R7C4:R" & dlength & "C4)> " & ↩
                            strength_low & ")*(ABS(" & source & "!R7C4:R" & ↩
                            dlength & "C4)<= " & strength_high & "))"
                    strength_low = strength_high
                    strength_high = strength_high + 10  'Stromstärkeintervall ↩
                        um 10A erhöhen
                    row = row + 1
```

```
                    Case Else
                        'Matrixformel: summe(flag=1,intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                        Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=1)*(" & source & "!R7C3:R" & dlength & "C3=1)*(" & source & "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" & source & "!R7C6:R" & dlength & "C6<= " & distance_high & ")" & _
                            " *(ABS(" & source & "!R7C4:R" & dlength & "C4)> " & strength_low & "))"
                End Select
            Loop
            distance_low = distance_high
            distance_high = distance_high + 50   'Distanzintervall um 50 km erhöhen
        ElseIf distance_low <= 1000 Then
            Do While Cells(37, I).Value = ""
                Select Case strength_low
                    Case Is <= 10
                        'Matrixformel: summe(flag=1,intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                        Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=1)*(" & source & "!R7C3:R" & dlength & "C3=1)*(" & source & "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" & source & "!R7C6:R" & dlength & "C6<= " & distance_high & ")" & _
                            " *(ABS(" & source & "!R7C4:R" & dlength & "C4)> " & strength_low & ")*(ABS(" & source & "!R7C4:R" & dlength & "C4)<= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high + 5   'Stromstärkeintervall um 5A erhöhen
                        row = row + 1
                    Case Is <= 40
                        'Matrixformel: summe(flag=1,intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                        Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=1)*(" & source & "!R7C3:R" & dlength & "C3=1)*(" & source & "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" & source & "!R7C6:R" & dlength & "C6<= " & distance_high & ")" & _
                            " *(ABS(" & source & "!R7C4:R" & dlength & "C4)> " & strength_low & ")*(ABS(" & source & "!R7C4:R" & dlength & "C4)<= " & strength_high & "))"
                        strength_low = strength_high
                        strength_high = strength_high + 10  'Stromstärkeintervall um 10A erhöhen
                        row = row + 1
                    Case Else
                        'Matrixformel: summe(flag=1,intra-cloud=1, distance_low<distance<=distance_high, strength_low<abs(strength)<=strength_high)
                        Worksheets("CC-Strokes").Cells(row, I).FormulaArray = _
                            "=SUM((" & source & "!R7C5:R" & dlength & "C5=1)*(" & source & "!R7C3:R" & dlength & "C3=1)*(" & source & "!R7C6:R" & dlength & "C6> " & distance_low & ")*(" & source & "!R7C6:R" & dlength & "C6<= " & distance_high & ")" & _
```

```
                              " *(ABS(" & source & "!R7C4:R" & dlength & "C4)> " & ↩
                                  strength_low & "))"
                    End Select
                Loop
                distance_low = distance_high
                distance_high = distance_high + 100 'Distanzintervall um 100 km erhöhen
            End If
            'Zurücksetzen auf die Anfangswerte
            row = begin_row
            strength_low = 0
            strength_high = 5
        Next



End Sub
```

A.2.13. Userform Import Data

```
Private Sub cmd_data_Click()
    fNameAndPath = Application.GetOpenFilename(FileFilter:="Text (*.txt), *.txt", ↩
        Title:="Select File To Be Opened")
    If fNameAndPath <> Falsch Then
        txt_file = fNameAndPath
    End If
End Sub

Private Sub cmd_OK_Click()
    If txt_number <> "" And txt_file <> "" Then
        fnumber = txt_number
    Else
    UFclose = False
    End If
    deletedata = box_löschen
    import_data.Hide
    fNameAndPath = txt_file
End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, closemode As Integer)
    UFclose = False
End Sub
```

Bibliography

[1] Cummins, Kenneth L.; Murphy, Martin J.; Tuel Jeffrey V.: *Lightning detection methods and meteorological applications.* In *IV International Symposium on Military Meteorology Hydro-Meteorological Support of Allied Forces and PfP Members Tasks Realization*, Malbork, Poland, 2000. `http://www.atmo.arizona.edu/students/courselinks/spring07/atmo589/articles/Cummins_Poland2000.pdf`, visited on 07.10.2014.

[2] Dowden, Richard L.; Brundell, James B.; Rodger Craig J.: *Vlf lightning location by time of group arrival (toga) at multiple sites.* Journal of Atmospheric and Solar-Terrestrial Physics, 64, 2002. `http://wwlln.net/publications/dowden.toga.article.pdf`, visited on 07.10.2014.

[3] Euclid. `http://www.euclid.org`, visited on 02.10.2014.

[4] Friese, Wolfgang. `http://www.friese-electronic.de/sfericsempfang/komponenten/ferritantennen/`, visited on 04.11.2014.

[5] Hadrian, Wolfgang and Gerhard Diendorfer: *Blitzschutz (Skriptum zur Vorlesung).* TU Wien, 2013.

[6] Smith, D. A. et al.: *A method for determining intracloud lightning and ionospheric heights from vlf/lf electric field records.* Radio Science, 39, 2004. `http://www.forte.lanl.gov/science/publications/2004/Smith_2004_RS1010_Method.pdf`, visited on 04.11.2014.

[7] Wanke, Egon: *A low-cost time-of-arrival lightning location network.* Blitzortung.org, 2010.

[8] Wanke, Egon; Anderson, Riko; Volgnandt Tobias: *A world wide low-cost community-based time-of-arrival lightning detection and lightning location network.* Blitzortung.org, 2014.