

## BACHELOR PROJECT

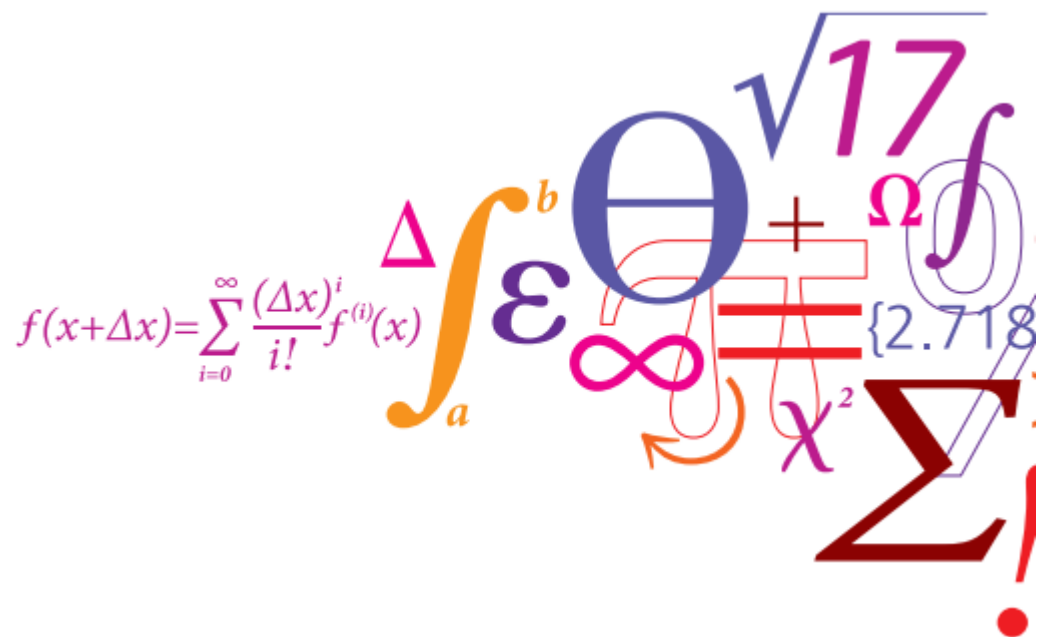
---

# GAMMA RAY MAPPING WITH UAV'S

---

Pernille Boldrup Østfeldt (s144144)  
Philip Lind (s135387)

SUPERVISORS:  
Jakob Jakobsen  
Irfan Kuvvetli



## **Abstract**

Nuclear power has since its original conception become a more widely used way to generate large amounts of electricity for civilian purposes. The close location to large populations as well as the storage of the tons of very hazardous waste material that is produced at these power plants means that it is crucial to closely monitor the radiation levels surrounding the power plants and waste storage locations. This project was made to design and build a gamma ray mapping system which could be mounted on an UAV-platform. A MATLAB program with a GUI was written to match the system. The program processes the data from the system and generate interpolated maps of the intensity and energy levels of the gamma radiation. The final system and program functions as intended and can locate and identify even relatively weak sources.

**Keywords:** Gamma radiation, Spectra, GPS, MATLAB, UAV, Interpolation, Mapping, Surveying.



# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Proof of concept . . . . .	4
<b>2</b>	<b>Theory</b>	<b>5</b>
2.1	Radioactivity . . . . .	5
2.1.1	Cobalt 60 . . . . .	5
2.2	Semiconductor detectors . . . . .	6
2.3	GNSS technology . . . . .	6
2.4	Coordinate Reference Systems . . . . .	7
2.4.1	Geographic coordinate systems . . . . .	7
2.4.2	Projection reference systems . . . . .	8
2.4.3	Universal Transverse Mercator . . . . .	8
<b>3</b>	<b>Hardware</b>	<b>10</b>
3.1	UAV-platform . . . . .	10
3.2	Fitlet . . . . .	11
3.3	Kromek GR1-A+ gamma spectrometer . . . . .	11
3.4	GPS receiver board and antenna . . . . .	11
3.5	Payload box . . . . .	11
<b>4</b>	<b>Software</b>	<b>13</b>
4.1	Wireshark . . . . .	13
4.2	u-center . . . . .	13
4.3	MATLAB . . . . .	13
4.4	Windows Remote Desktop Protocol (RDP) . . . . .	13
4.5	KMSTrans2 . . . . .	13
<b>5</b>	<b>Platform operation</b>	<b>14</b>
5.1	Taking control of the fitlet . . . . .	14
5.2	Synchronise UTC time . . . . .	14
5.3	Start data collection . . . . .	14
5.4	Flight and data saving . . . . .	15
<b>6</b>	<b>Experiments</b>	<b>16</b>
6.1	Test 1 & 2 - Spiral & parallel track area mapping . . . . .	16
6.2	Test 3 - Distance measurement of the gamma source . . . . .	17
<b>7</b>	<b>Data</b>	<b>18</b>
7.1	Gamma radiation data . . . . .	18
7.2	GPS data . . . . .	19
<b>8</b>	<b>Data processing</b>	<b>21</b>
8.1	Gamma data formatting . . . . .	21
8.2	Coordinate formatting . . . . .	22
8.3	Data pairing . . . . .	22
8.4	Energy . . . . .	23
8.5	Interpolation . . . . .	23
8.6	Present data . . . . .	24
8.7	Overview of the program . . . . .	25
<b>9</b>	<b>Results</b>	<b>26</b>
<b>10</b>	<b>Discussion</b>	<b>31</b>
<b>11</b>	<b>Further Improvements</b>	<b>33</b>
11.1	Program further development . . . . .	33
11.2	Raspberry Pi with Windows 10 IoT Core. . . . .	33
11.3	Intel NUC . . . . .	33

<b>12 Conclusion</b>	<b>35</b>
<b>Appendices</b>	<b>36</b>
<b>A Payload Setup</b>	<b>36</b>
A.1 Setting up the fitlet . . . . .	36
A.1.1 Ad-hoc WiFi network . . . . .	36
A.1.2 Remote control over network . . . . .	38
A.1.3 Software installation . . . . .	38
A.2 Payload assembly . . . . .	40
<b>B Unforeseen difficulties</b>	<b>46</b>
<b>C User guide</b>	<b>47</b>
<b>D Three week preliminary project</b>	<b>53</b>
<b>E Datasheet for the Cobalt-60 source</b>	<b>70</b>
<b>F Datasheet for the Kromek GR1-A+</b>	<b>72</b>
<b>G Kromek USB Spectrometer Interface – Technical Manual</b>	<b>77</b>
<b>H MATLAB program code</b>	<b>82</b>
<b>References</b>	<b>83</b>
<b>Figures</b>	<b>85</b>

# 1 Introduction

The goal of this project is to develop a proof of concept for a mobile gamma ray mapping solution which can be deployed in areas where the contamination is too high for human presence. The mapping solution is intended to be very low weight so it can be carried by an UAV-platform. The project consists of two main parts. The first is to find hardware which can function together with the Kromek GR1-A+ gamma detector and capture both the gamma radiation data from the detector and GPS location data. The second part is to create a program which will allow a person without any programming knowledge to still be able to create visual presentation of the data.

Uses for such an application could for example be on Greenland. Every year more ice thaws on Greenland which is threatening to release up to 24 million litres of radioactive coolant water from Camp Century's nuclear power plant at the abandoned cold war U.S. Army base just 800 miles from the North Pole[1]. The base was abandoned decades ago despite efforts to maintain the underground tunnels. The continuously shifting ice sheets eventually forced the Army to abandon the base in 1966[2]. The atomic waste still remains and if it's allowed to seep into the oceans it could lead to further pollution and have geopolitical implications.

Another place where such a solution would be applicable is nuclear power plants. As of early June 2017 the IAEA reports that there are 449 nuclear power reactors in operation spread over 31 countries with another 60 under construction[3]. A small and airborne mapping solution would be able to identify and locate radioactive leaks at a low cost while still maintaining high precision. The big risk with nuclear power is the large scale contamination in case of a meltdown or other accidents. The Fukushima and Chernobyl nuclear accidents are the two most well known. In case of a reactor meltdown a small unmanned platform would be able to quickly gather data on the severity of the contamination as well as provide constant updates on the situation. The small mapping system would also be ideal because of the ease of use, which requires only an experienced drone-pilot to fly the UAV and another person to process the data.

In this report there will first be an overview on radioactivity and GNSS theory followed by short descriptions on the hardware and software which has been used for the project. After that a guide to how the system is used in the field is given, which is then followed by the description of the 3 main experiments done for this project. It is then explained what data is obtained from these experiments and how the newly developed program processes this data into formats which can be analysed. The report ends with suggestions of further improvements that can be made to the gamma mapping systems current state.

## 1.1 Proof of concept

This bachelor project follows a three week course used to gain familiarity with the equipment and to create a proof of concept of the system. A table test was used to collect the amount of radiation in counts per second from both a smoke detector and from the background radiation. Then simple data processing, such as manual data pairing and MATLAB interpolation, was used to create a contour plot of the counts which was exported to Google Earth to locate the gamma source.

During the three week course it was found that the Kromek analysis software only was able to save 120 measurements. Since velocity of the drone will be approximately  $1 \frac{\text{m}}{\text{s}}$ , only 2 minutes of gamma data will be available. This means that a program which could process the data input and convert it to useful data needed to be designed.

The report can be seen in appendix D.

## 2 Theory

### 2.1 Radioactivity

Radioactivity is spontaneous decay of unstable nuclei [4]. An unstable isotope will emit particle radiation or electromagnetic radiation until it reaches a stable isotope [5]. The most common types of radiation are alpha, beta and gamma. In alpha radiation a helium nucleus consisting of two protons and neutrons is emitted from the unstable isotope [6]. Due to the large mass and charge, alpha radiation has a very short range and very little penetration. In beta radiation the weak interactions changes a down quark to an up quark, changing a neutron to a proton [7]. Due to conservation of charge an electron and an electron antineutrino is emitted. Beta radiation refers to the electron. Because of the smaller mass and larger energy than the alpha particles, beta radiation has larger penetration. The last type of radiation is gamma radiation. In gamma decay, a nucleus changes from a higher energy state to a lower energy state through the emission of electromagnetic radiation [8]. There are no change in the number of protons and neutrons in the nucleus during the process, so the parent and daughter atoms are the same isotope. Gamma radiation have very high energy, and are therefore very penetrating.

The decay of a radioactive isotope is described by its half-life, which is the time it takes half the nucleus to decay. The half-life can vary from a few microseconds to billions of years [9]. Half-life is based on probability. Another way to describe a radioactive material is its activity, which is the number of disintegrations per second [12]. Radioactive materials with high half-lives and high activity can be problematic and harmful to the environment surrounding the source.

When measuring energy levels from a radioactive source, other energy levels will be present in the measured radiation spectrum, which come from the background radiation. Background radiation is radiation from sources such as space, soil and the air. The level of background radiation is described in units of mSv/year, where 1 Sv equals 1 joule per kilogram [10]. The world average is 2.40 mSv/year. Figure 1 shows a gamma background spectrum measured inside a concrete building. Since gamma rays can penetrate concrete, a gamma background spectrum measured outside will be very similar.

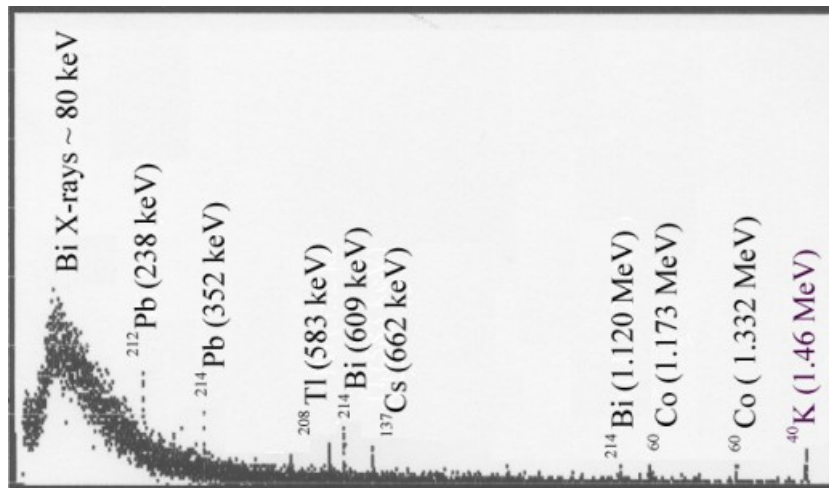


Figure 1: Gamma background radiation inside a concrete building.

#### 2.1.1 Cobalt 60

For the tests a Cobalt-60 source is used. Cobalt-60 is a radioactive isotope that decays to Nickel-60 and an electron and an electron antineutrino by beta minus decay of energy 0.31 MeV. The beta decay can be seen in equation 1. The Nickel-60 is a nuclear excited state of Nickel-60 which emits two gamma ray photons to reach the stable state of the Nickel isotope. The energy of the gamma radiation is 1,17 MeV and 1,33 MeV. 0.12% of the time the beta minus decay has an energy of 1.48 MeV, and the excited Nickel-60 emits only one gamma ray photon with energy 1.33 MeV. The gamma decay is shown in figure 2. The half life of Cobalt-60 is 5.27 years.



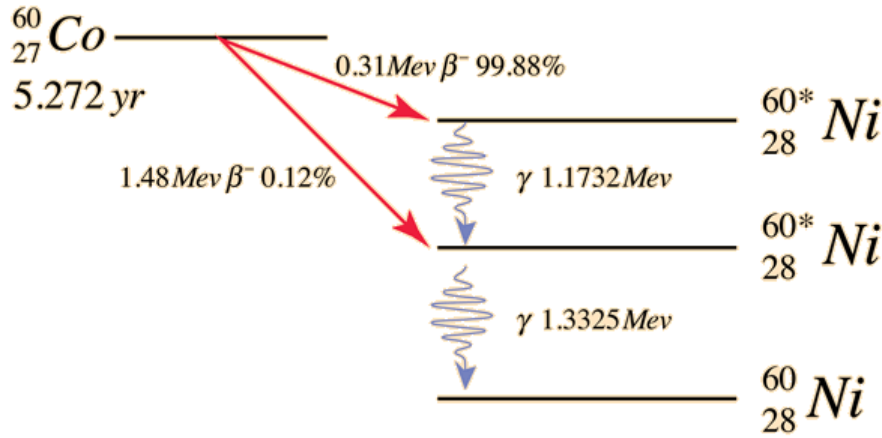


Figure 2: Decay of Cobalt-60.

The Cobalt-60 source used for the tests has activity 20,1 mCi measured in 1978, see appendix E. Equation (2) shows the calculated activity of the Cobalt-60 source today [11].

$$A = A_0 \cdot e^{-\lambda \cdot t} = 20.1 \cdot 10^{-3} \cdot e^{-\frac{\ln(2)}{5.27} 39} = 0.1 \text{ mCi} = 3.7 \cdot 10^6 \text{ Bq} \quad (2)$$

Where 1 Ci =  $3.7 \cdot 10^{10}$  Bq, which is the number of disintegrations per second. [12].

## 2.2 Semiconductor detectors

One of the ways to measure gamma radiation is by using a semiconductor detector. A semiconductor detector is a type of ionisation detector, which rely on a material to produce ion pairs in response to ionisation radiation [13]. The ionisation happens when charged particles or photons pass through the atoms of the material at a high velocity, which produces a large enough electromagnetic field to separate the electrons, creating ions. In semiconductor materials a relatively large number of carriers occur for each incident radiation event, giving it the best energy resolution in modern radiation detectors. The carriers are electron-hole pairs created along the path of the radiation through the detector material, and they form a measurable electric current when an external electric field is applied [13].

One of the materials used in semiconductor detectors are Cadmium Zinc Telluride (CZT) which can measure gamma rays with energies greater than 20 keV due to the high density of these materials. It also has a wide band gap, which gives it a high resistivity, meaning that it can operate at room temperature. Other common semiconductor detector materials are Silicon and Germanium, but both are not suitable for measuring gamma radiation using UAV platforms. Silicon has a lower density and are therefore not able to measure energy above 20 keV, and germanium operates at 77 K, so it requires liquid cooling. [14]

## 2.3 GNSS technology

The Global Navigation Satellite System consists of multiple satellites. There are 24 operational satellites at all times which are orbiting the Earth twice a day at an altitude of 20.000 km [16]. The GNSS include GPS, Galileo, GLONASS and BeiDou, where GPS is the most commonly used term to describe the system, and so this term will also be used in some parts this report. The satellites transmit radio signals which is detected by a GPS receiver. The signal consists of three parts. The first is a pseudo random code carrying the identity of the specific satellite transmitting the signal. The second is almanac data which contains information about the course of all the GNSS satellites. The receiver uses this information to determine which satellites are above the horizon so the receiver only focuses on these satellites. The last is ephemeris data which contains information about the specific satellites position at any time during the day [15].

Each satellite carries atomic clocks with an nominal accuracy of 1 nanosecond. to achieve the level of accuracy used in the GNSS system, a nominal accuracy of 20-30 nanoseconds is needed. Because the satellites move at a constant speed relative to the Earth, the atomic clocks carried on the satellites will experience time dilation, so the clocks run slow as seem from the Earth's inertial system. Due to the

effect of special relativity the atomic clocks will fall behind the clocks on the Earth by 7 microseconds a day. Because the satellites are orbiting at 20.000 km, the curvature of the spacetime is smaller than at the surface of the Earth, so due to general relativity the clocks will run fast by 45 microseconds. Due to these effects the nominal accuracy will exceed 30 microsecond after only two seconds. In order to synchronise the satellites these effect must be taken into account [16].

When the satellites are synchronised a satellite signal is transmitted from each satellite at the same instant as seen from the Earth's inertial system [15]. When the receiver receives a signal it calculates the time it takes the signal to reach the receiver, and then calculates the distance to the satellite. The position is determined through trilateration where the distance to four different satellites are used to calculate the 3-dimensional position. As seen in figure 3 a satellite transmits the signal in all directions, creating a sphere with the distance to the receiver as radius. Intersection between two spheres determines a circle on which the receiver lies. The intersection with a third sphere creates two points where the receiver lies on one of the points. The intersection with a fourth sphere determines the 3-dimensional position [17].

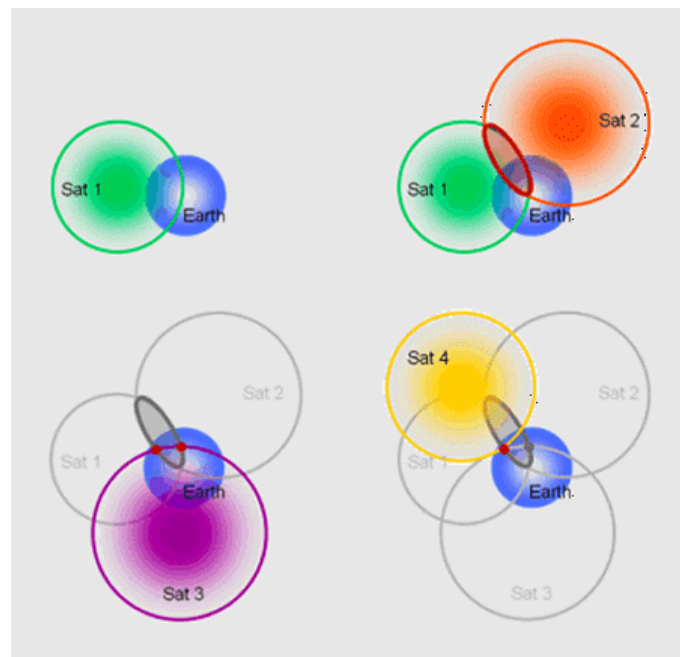


Figure 3: The principle of trilateration.

## 2.4 Coordinate Reference Systems

Coordinate reference systems specify positions on Earth using coordinates, which are sets of three numbers. Coordinate reference systems can be divided into two groups. The first is geographic coordinate systems which specify positions using longitude and latitude, and the second is projected coordinate systems, which uses Cartesian coordinates.

### 2.4.1 Geographic coordinate systems

Geographic coordinates systems (GCS) uses a three-dimensional spherical surface to define locations on the earth [19]. The reference systems used by GPS uses the World Geodetic System 1984 datum, which is a spheroid based on constants and model parameters that describe the Earth's size, shape, gravity and geomagnetic fields [20]. The locations are described by degrees of latitude and longitude [18]. Latitude are lines that run parallel to the equator and divide the earth into 180 equally spaced sections from South to North. The reference line at which the latitude is zero degrees is the Equator. The latitude goes from 0 to -90 degrees in the southern hemisphere and from 0 to +90 degrees in the northern hemisphere. Longitude are lines that run perpendicular to the Equator and therefore converges at the poles, with the reference longitude, also called prime meridian, going through Greenwich, England. The longitude goes from 0 to +180 degrees towards East and from 0 to -180 towards West of the prime meridian.

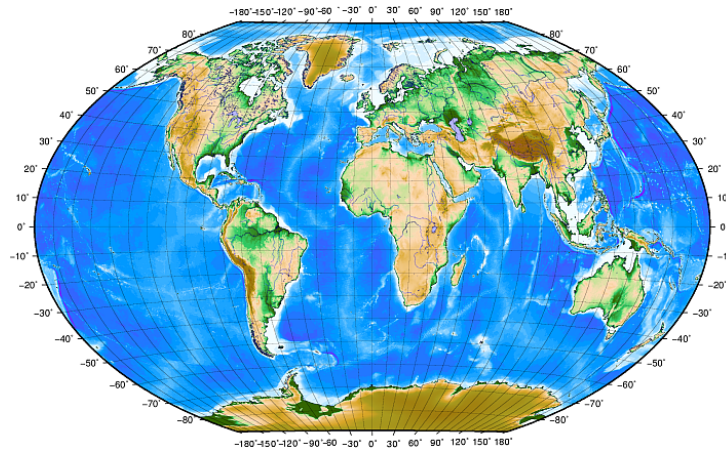


Figure 4: Geographic coordinate system with lines of latitude parallel to the equator and lines of longitude with the prime meridian going through Greenwich.

Because the lines of longitude converges at the poles, a small quadrilateral grid will be distorted when plotting the latitude and longitude. This will make it more difficult to locate a radioactive source when using a UAV platform. The coordinates need to be projected onto a 2-dimensional map.

#### 2.4.2 Projection reference systems

Projection coordinate systems are defined in a 2-dimensional surface [21]. Projection systems are commonly described by two axes, where the horizontal axis is called the x-axis, and the vertical axis is the y-axis, and at a right angle they form a xy-plane. In a 3-dimensional system, a z-axis is added at a right angle to the xy-plane. Unlike a geographical reference system where the lines of longitude converges at the poles, a projected coordinate system has constant angles, length and area [21]. In the northern hemisphere the x values increase to the East relative to the specific longitude and the y values increase northwards from the equator. In the southern hemisphere the x value increases to the West and the y value increases southwards.

#### 2.4.3 Universal Transverse Mercator

The Universal Transverse Mercator (UTM) is a global map projection. The system has origin at the Equator at a specific longitude, where the world is divided into 60 zones of 6 degrees each, starting at 180 degrees West going eastwards to 180 degrees East [21]. The latitude bands are not a part of the UTM but of the military grid reference system (MGRS) [22].

The coordinate position is given by the UTM zone number and the easting and northing value in meters. The easting value is the distance from the centre meridian of the UTM zone in meters and the northing value is the distance from the equator in meters [21]. The easting value increases towards East and the northing value increases northwards. The equator is positioned at 10.000.000 m North and the central meridian is positioned to coincide with 50.000 m East. This is done so all easting and northing values are positive. The UTM zone used in this project is UTM zone 33, which covers East Zealand.



Figure 5: UTM zones in Europe.



### 3 Hardware

For this project 5 main hardware components are required. 4 electric and 1 non-electric.

- UAV-platform.
- Onboard computer for data collection.
- Gamma detector.
- GPS receiver board and antenna.
- Mounting solution for the payload on the UAV-platform.

#### 3.1 UAV-platform

The UAV-platform used for this project is the DJI Spreading Wings S900. The S900 is a hexacopter. The takeoff weight of the S900 is from 4.2 kg to 8.2 kg. It is rated to handle mean winds up to 8 m/s and can safely handle gust up to 13 m/s. The S900 has an official working environment temperature range from -10 to + 40 degrees Celsius.

The reason that a multirotor UAV-platform is used rather than a fixed-winged platform is that while the fixed-wing might have a longer flight time and can cover a larger area per flight, its minimum speed is too fast for the gamma detector to gather any meaningful data. The drone has a measured actual flight time of around 18 minutes of with the Tattus 16000 mAh batteries.

The S900 with the 16000 mAh has room for up to a little over 2 kg of payload. This range fit into the anticipated payload weight of around 1.3 kg. This would give a total takeoff weight of 7.62 kg. The S900 has a maximum draw power of 3000 W from the battery while flying. When the takeoff weight is at 6.8 kg the drone uses 1000 W in order to maintain hover in a windless environment[24]. This ratio is very favourable since a rule of thumb is that the desired maximum thrust should be 1.5-2 times the weight of the multirotor in order to have a stable and safe platform. Thus even with the slightly heavier payload from this project the S900 still fulfils this criteria.

Having a stable platform helps with the data gathering since it makes it easier to fly in a grid so as to gather uniform data. When flying in a grid be it manually or autonomously it is imperative that the drone has the ability to hold itself in a relatively low deviation from the grid which is being flown. The A2 system on the S900 has a flying mode where it uses GPS data to maintain its position in a 3-dimensional space. This GPS, although not as precise in its ability to ascertain the drones position as a RTK-GPS system, is still accurate enough to have the S900 stay very close to its position, and it takes a while before it will have drifted any significant distance such as 1 meter from the grid.

The DTU drone service has a standardised modular payload system which has been designed with the S900 in mind. This modular setup has been tested with payloads of up to 2 kg maximum weight and can fly up to 5 m/s completely stable with experimental equipment mounted in the payload box. It can also handle an uneven weighted 2 kg payload as long as the bulk of the weight is located inside the payload box.

The power to the payload was drawn from the S900 through a power supply. The power supply was the TRACO power TMDC 40-2412 which can take a 9-36 VDC input and transform it to a 12 VDC / 3330 mA output.



Figure 6: The DJI Spreading Wings S900 drone with the Finalised payload mounted just before a mapping test flight.

### 3.2 Fitlet

The fitlet used for this project is the XA10 LAN Barebone, specifically FITLET-GX-C67-FLAN-W. It is a fanless mini pc which can run a full version of Windows. The fitlet is a low power system with power draw ranging from 4.5W - 10.5W depending on system load. It has a quad core processor - the AMD A10 Micro-6700T with a core clock of 1.2 Ghz and boost speed of 2.2 GHz. It has no internal Wi-Fi but comes bundled with an USB Wi-Fi adapter.

It has 2 USB3.0 ports and 3 USB2.0 ports. The USB3.0 ports run on their own USB3.0 controller and the USB2.0 ports run on two USB EHCI Host controllers.

The fitlet used in this project was configured with a 32-bit version of Windows 10 Professional, 4 GB of RAM and a 1 TB mSATA SSD.

Over the course of the project it was attempted to substitute the fitlet with a BeagleBone Black. Another fitlet was also temporarily used but this was scrapped but both due to unforeseen difficulties, see appendix B.

### 3.3 Kromek GR1-A+ gamma spectrometer

The component which the entire project was based around was the Kromek GR1-A+ CZT semiconductor detector. All the other hardware was chosen based on whether or not they functioned together with the Kromek GR1-A+. The Kromek GR1-A+ is a USB-powered gamma spectrometer using a 1 cm<sup>3</sup> Cadmium Zink Telluride detector. Cadmium-Zink-Telluride, or CZT, is a semiconductor that directly converts gamma photons into electrons [23]. The data sheet for the Kromek GR1-A+ gamma detector can be found in appendix F.

The Kromek GR1-A+ is the optimal choice for the project for the exact reason that it is a CZT semiconductor. CZT semiconductors operate at room temperature 0 to 40°C and the Kromek GR1-A+ is also very small ( 25 x 25 x 63 mm) and light (60 g) making it ideal for UAV-operations. The energy range of the spectrometer is 30 keV to 3.0 MeV, with an energy resolution of <2.0% FWHM at 662 keV. Electronic noise on the detector is <10 keV. It has 4096 (12 bit) channels and a maximum throughput of 30,000 counts per second, see appendix F.

The GR1-A+ has low power draw as well at 250mW making it possible to drive it over most USB ports on different small form-factor computers and USB-controllers.

### 3.4 GPS receiver board and antenna

For this project the UBLOX NEO-M8T-0-10 GPS receiver board with SMA connector is used. The NEO-M8T GNSS modules are concurrent GNSS receivers that can receive and track multiple GNSS systems: GPS, Galileo, GLONASS and BeiDou. Due to the dual-frequency RF front-end architecture either GLONASS or BeiDou can be processed concurrently with GPS and Galileo signals providing reception of three GNSS systems at the same time. By default the M8T receivers are configured for concurrent GPS and GLONASS, including QZSS reception. While this isn't a very precise GPS solution it was still within acceptable range for testing the proof of concept solution and was also a lightweight solution, adding only the board, an USB cable and a GPS antenna to the payload.

### 3.5 Payload box

The experimental equipment is mounted inside a payload box which in turn is mounted on the S900. The payload box is a standardised modular system designed for the S900 platform by Jakob Jakobsen. The modular system has been tested through its use in other projects.

The payload box is made from 3mm plywood which has been lasercut into the separate pieces which makes up the box. This gives a space of approximately 6 litres for experimental equipment.

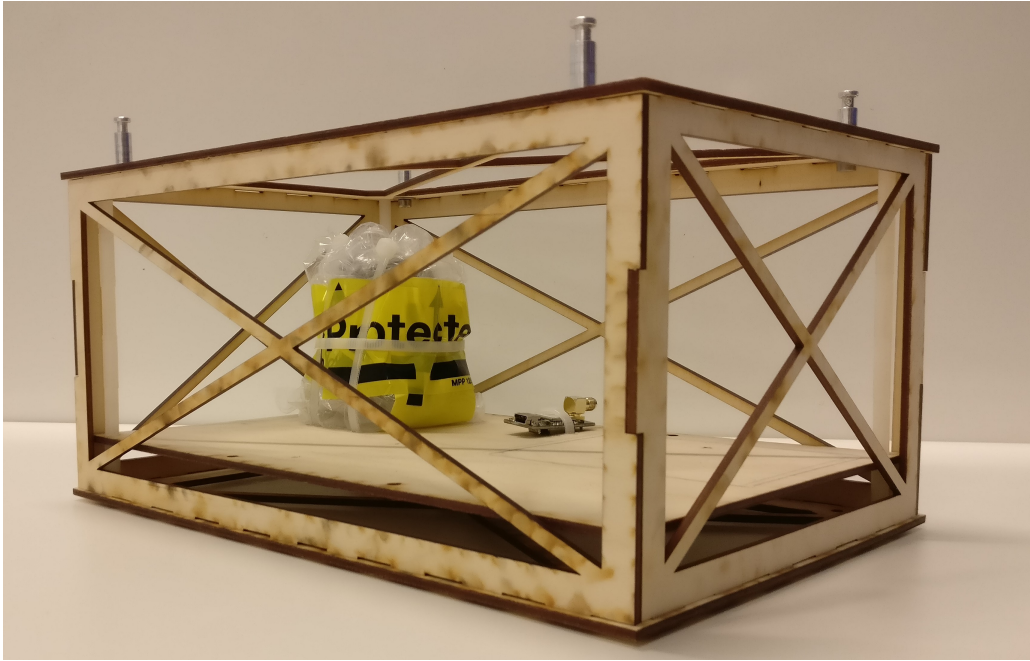


Figure 7: Picture of the payload box with the foam and bubblewrap protection for the gamma detector as well as the mounted GPS receiver board.

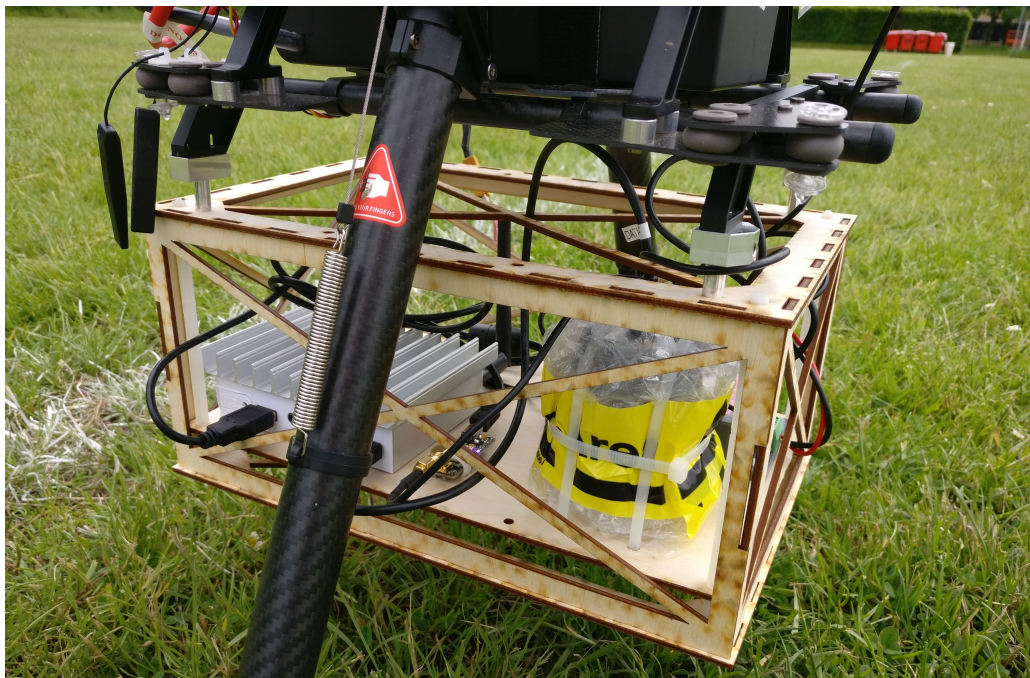


Figure 8: Picture of the finalised payload which was used for the experiments. Most data from this report was gathered with this exact version of the setup.

## 4 Software

The software needed to both create and utilise the gamma ray mapping system is comprised of 5 programs/applications. The OS which is used is Windows 10 with the 64-bit version for development and the 32-bit version for the fitlet.

### 4.1 Wireshark

For the collection of gamma radiation data from the Kromek gamma detector, the program Wireshark is used. The reason that this software is used instead of the Kromek software Multi-Spect, is because of the limited possibilities of saving the measurements when using Multi-Spect. When measuring gamma radiation at a frequency of 1 Hz in Multi-Spect, only the last 120 measurement are available, which gives two minutes of data, whereas Wireshark captures all packets sent from the Kromek GR1-A+.

### 4.2 u-center

u-center is a GNSS evaluation software which supports all u-blox GNSS receivers. It is a free of charge program which besides evaluating the GNSS data also functions as a GNSS configuration tool for the receiver boards. u-center also provides real time performance visualisation of u-blox GNSS receiver products such as docking views with real-time cockpit instruments and satellite status charts which allow easy observation of the static and dynamic behaviour of the GNSS receiver. This is used so that the mapping data gathering isn't started until a valid GPS position is ascertained.

### 4.3 MATLAB

For this project the data processing will be performed in the multi-paradigm numerical computing environment: MATLAB. MATLAB is a mathematics-computer program which focuses on the application of vectors and matrices which the name is derived of: MATrix LABoratory. It is a proprietary programming language developed by MathWorks. It is a multi-paradigm numerical computing environment and fourth-generation programming language. MATLAB is one of the most used mathematics-computer programs among engineers around the world. In Denmark it is taught at Aalborg University, Syddansk University, Technical University of Denmark - DTU, etc. As of 2017, MATLAB has over 2 million users across the world.

MATLAB can handle more than just vectors and matrices with features which include: plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java and Python.

### 4.4 Windows Remote Desktop Protocol (RDP)

For controlling the fitlet remotely Windows 10's own remote control solution has been selected. The choice to use Microsoft's own software was because of it's simplicity but also because the fitlet was having its CPU usage at 80-100 % just by having u-center and Wireshark running. The newest version (10.0) of RDP included utilising H.264/AVC and other improvements with the graphics compression allowing for less stress on the CPU.

RDP or Remote Desktop Protocol is the Windows protocol taking care of remote desktop connections. Remote Desktop Connection RDC (also called Remote Desktop) is the client application for Windows Remote Desktop Services using RDP to allow a user to remotely log into a networked computer running the terminal services server. RDC presents the desktop interface (or application GUI) of the remote system as if it were accessed locally.

### 4.5 KMSTrans2

KMSTrans2 (v2.1) from "styrelsen for dataforsyning og effektivisering" is a program where coordinates are effectively transformed from one reference system to another. The program can handle data sets of coordinates even if some coordinates are missing from the data set.



## 5 Platform operation

This section of the report focuses on how the process of capturing and saving the data for the experiments/tests described in the next section are done.

### 5.1 Taking control of the fitlet

For taking control of the fitlet start the **Remote Desktop Connection APP**. Then write the static IP which has been assigned to the fitlet (for this project 192.168.2.2 was used) and enter the user name (SPC-DRONE1-PC-fitlet). After this step one will be prompted for the password to the user if the user has a password (???123Qwe). The user will now be logged on and the remote controlled session is started.

### 5.2 Synchronise UTC time

The first thing to do before anything else after the fitlet has become remotely controlled, is to connect the fitlet to an internet connection. This can either be over Wi-Fi (which won't interrupt the remote controlled session, or over 1 of the 4 Ethernet ports on the fitlet. After that, navigate to the bottom right on the screen and right click on the time and date on the taskbar. Select the option called **Adjust Date/Time**. After that enable the function which says **Set time automatically**. If this is already enabled, disable it and then enable it again to trigger the service which syncs the time over the internet. This step is important because the fitlet's internal clock easily can drift more than 5 seconds a day when de-powered and not connected to the internet.

### 5.3 Start data collection

When the time has been synchronised it is time to start up the two programs u-center and Wireshark on the fitlet. First start up u-center and check whether GPS position is available. If it is pres F11 to open table view. In this table add in this order: UTC, Longitudinal position, Latitudinal position, height MSL(Mean Sea Level). When all four columns are continuously filled with data input on every second, press the red circle symbol named **record** and save the file in the wanted directory. Press **no** to the prompt that comes after. The GPS data is now being gathered.

Now open Wireshark with Admin privileges in order to successfully detect all devices and controllers. Select the USBPcap matching the controller which the Kromek GR1-A+ has been connected to and navigate through the menu down to **HID-compliant vendor-defined device** and press **Start**. This will start the capturing of the data from the gamma detector.

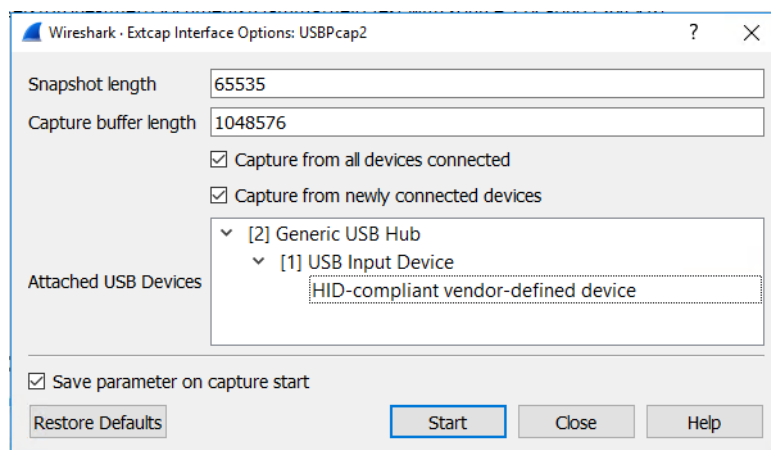


Figure 9: Wireshark menu for choosing which device it should capture packets from. USBPCap2 was usually used since it was connected to the front right USB2.0 port which was the only port on the USB-controller.

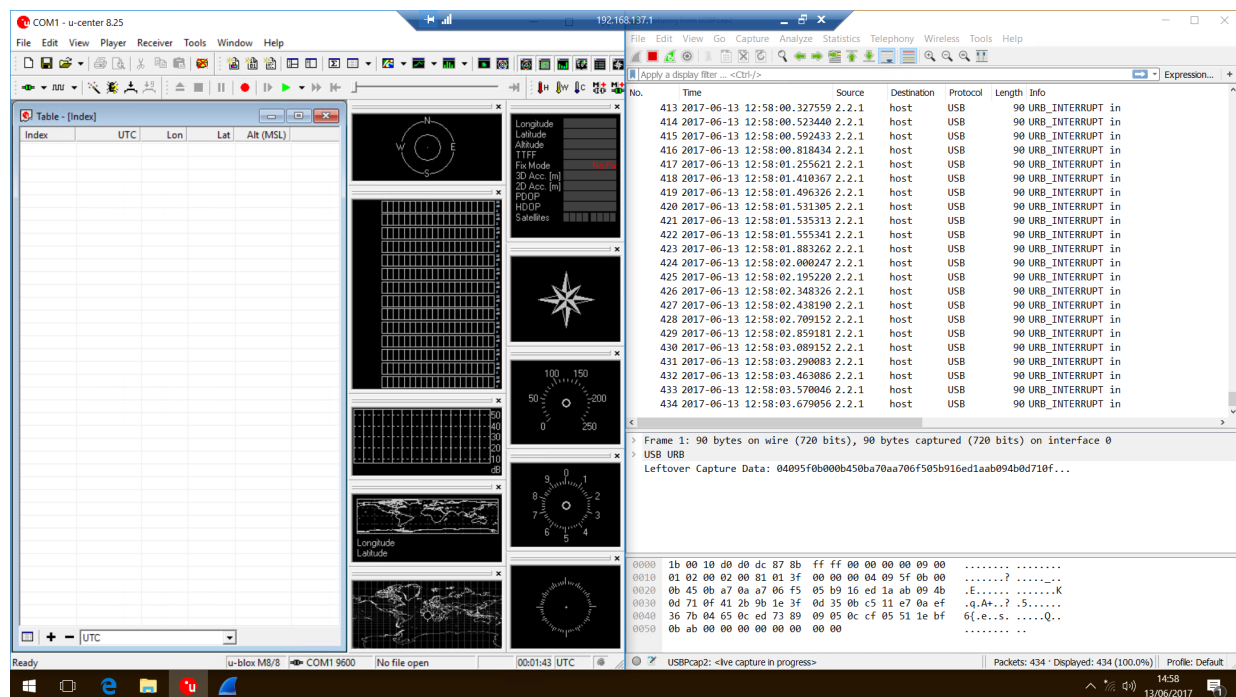


Figure 10: Screenshot of u-center and Wireshark running on the remote controlled fitlet. \*u-center not connected to GPS receiver board. Wireshark is capturing data.

## 5.4 Flight and data saving

When both programs are capturing the data go down to the Windows **Start** menu and select **Power > Disconnect**. This will end the session and log of the remote user but the data gathering will still be running in the background.

The pilot can now arm the UAV-platform and fly over the area designated for gamma mapping.

After the flight is over and the drone has landed, the data needs to be saved. This has to be done manually so it is necessary to start another remote controlled session. When the connection is back up simply stop the Capture in Wireshark by pressing the red square titled **Stop capturing packets**. Then go to **File** and select **save** and save the file in the preferred directory. The data should then be saved as a **.txt** file for further processing in MATLAB. This is done by going to **File** and selecting **Export Packet Dissections** and then choosing **As Plain Text...** In the pop-up window under the category **Packet Format** only tick off **Packet summary line** and **Packet Bytes**.

For u-center stop the capture and in the top menus select **Edit > Select all** then right click on the selected data in the table view and select **Save as CSV**, then choose the preferred directory for the saved file.

The files can now be transferred to the computer which is designated to handle the post-processing of the data. The files are by default now saved in the correct way to make it possible for the written MATLAB program to process it, but the **.pcapng** format can be opened by any Wireshark installation to then save the data as **.txt** with the correct formatting for the program.

## 6 Experiments

Three tests were made on the football field at DTU on the 2/6-2017. A quadrilateral area of approximately 15 times 20 meters was chosen, and a Cobalt-60 source was placed in the centre of the area. The background radiation was measured to be  $0.15 \mu\text{Sv/h}$ .

### 6.1 Test 1 & 2 - Spiral & parallel track area mapping

The first two field tests are designed to simulate a real world use case. A gamma radiation source is placed in the center of the designated area and the drone with attached payload is flown in a pattern to gain an uniform collection of data. For the experiments 4 things are needed:

- S900 drone.
- Experimental payload.
- Laptop with wireless network capabilities.
- Gamma radiation source (Cobalt-60).

The difference between the two experiments lies in the patterns they were flown with. Field test 1 was flown with a spiral pattern and field test 2 was flown in a series of parallel tracks as seen in figure 11. Both patterns were flown with a distance of 1-1.5 meters between each track and with a flying height of 1.5 meters.

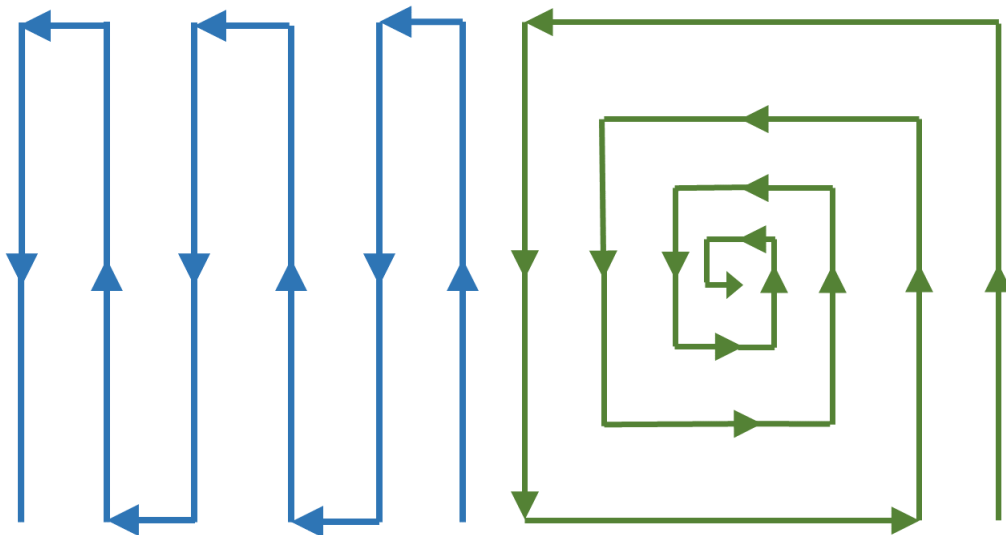


Figure 11: Visual presentation of the two types of patterns which were flown. Amount of tracks not representative of actual amount during the test.

The flying height was set to keep the location of the gamma detector around 1.5 meters above the ground. Because of small gust of wind the S900's onboard barometer had problems with maintaining the altitude correctly and the drone sometimes drifted down to 50 cm height above ground. Due to low battery the second test was stopped before the pattern was completed but there where flown on 5 meters on both sides of the gamma source.

## 6.2 Test 3 - Distance measurement of the gamma source

The third test is not a mapping test but a test of what the gamma detector measures at different distances from the gamma source. To do the experiment only 4 things are needed.

- Gamma detector.
- Gamma radiation source (Cobalt-60).
- PC connected to gamma detector.
- Distance measurement device.

The source was placed on top of the bucket which the sources was transported in, which had a height of ca 30 cm. The gamma detector was still attached to the payload after the first two tests so the solution was to tilt the S900 on its side so as to point the front of the gamma detector towards the Cobalt-60 source. There was then done measurements of 15 seconds before the gamma detector was moved further away from the Cobalt 60 source. The measurements were done at 3 different distances: 50 cm, 100 cm and 150 cm.

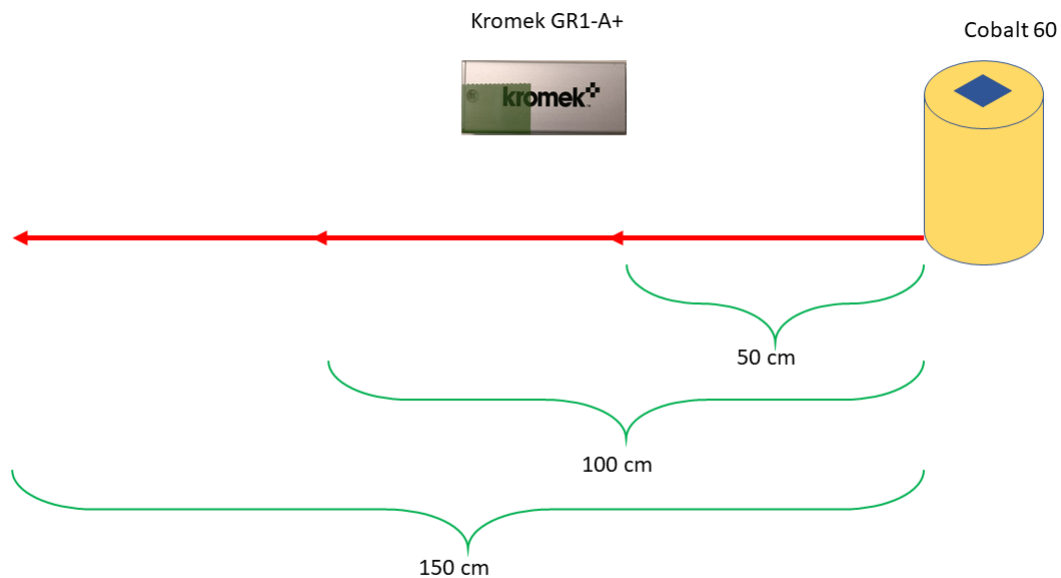


Figure 12: Visual presentation of the distance measurement test.



## 7 Data

### 7.1 Gamma radiation data

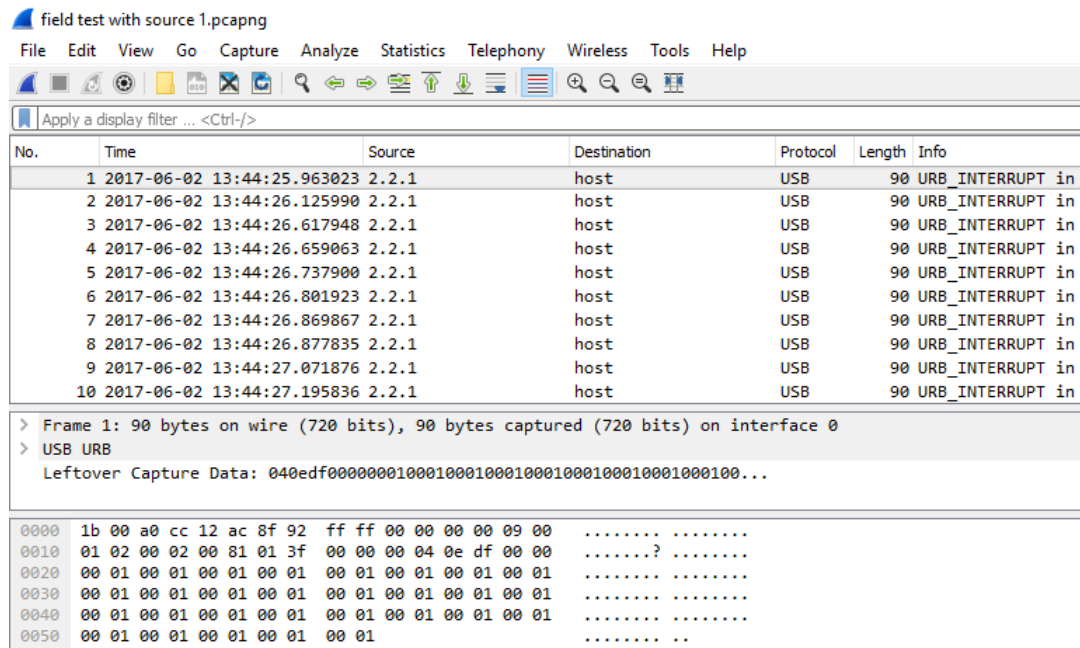


Figure 13: Presentation of the measured data in Wireshark.

All the data measured from the gamma spectrometer is saved as a PCAP file. The PCAP file is then imported to Wireshark, where the data is saved to the right format for processing. To export the data from Wireshark, one must go to **File > Export packet Dissections > As Plain Text...** Under **Packet Format**, only the boxes **Packet Summary Line** and **Packet Bytes** should be chosen, to secure that the MATLAB script can handle the format. The data should be saved as a **.txt** file.

```

99 2017-03-13 09:34:09.283983 1.4.1 host
USB 90 Yes Yes USB_INTERRUPT in

0000  1b 00 d0 b4 d2 4e 07 bc ff ff 00 00 00 00 09 00  .....N.....
0010  01 01 00 04 00 81 01 3f 00 00 00 04 02 b9 02 b3  .....?.....
0020  02 c9 08 d3 05 81 04 00 04 ab 04 55 02 c5 02 75  .....U...u
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
  
```

Figure 14: A packet shown as a **.txt** file. The second line starting with "USB" is usually in extension of the first line but this solution fit the image better.

The gamma radiation data now contains the index, time stamp and packet bytes. Because the Fitlet gets access to the internet, the time stamp is the UTC. The packet bytes which contains the channel number which the events has occurred in is send via report 4 which is used to transfer spectrum data from the detector to the USB Host device. Report 4 is always 63 bytes long, where the first byte contains the report identifier, and the remaining 62 bytes contain 31 channels, with two bytes per channel. The first 12 bits is the channel number, determining the radiation energy, and the last bit of the second byte determines whether the channel is valid or not. If the least significant bit of the second channel byte is 0, the report will be invalid and all following channels in the report will also be invalid. If the least significant bit is 1, the channel is valid.

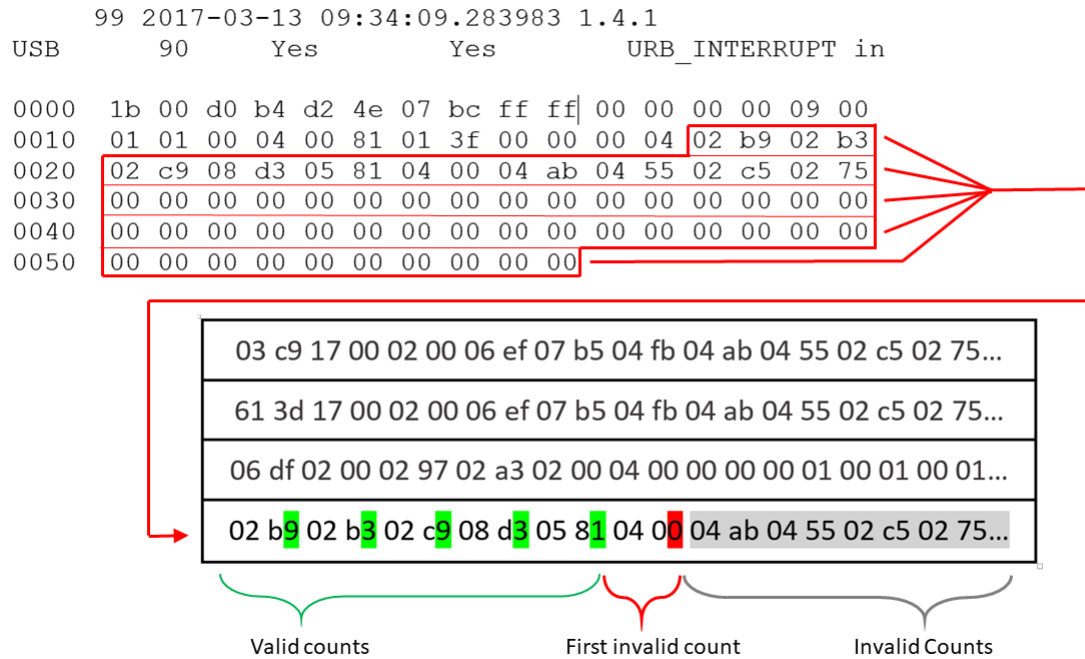


Figure 15: The data is collected by analysing each packet that the detector sends.

- Field test 1 contains gamma measurements from 13:44:25.963023 to 13:55:03.003530.
- Field test 2 contains gamma measurements from 14:00:55.720841 to 14:10:47.375603.
- The distance test contains gamma measurements from 14:13:49.812453 to 14:16:30.893177.

## 7.2 GPS data

The GPS data contains a measurement index, UTC time stamp measured at a rate of 2 Hz, longitude and latitude. To view the measured data in a table, the menu **View > Table View** was selected, and then **Column add** was selected to add the desired parameters to the table. Figure ?? shows data in u-center.

Table - [Index]			
Index	UTC	Lon	Lat
430	13:47:11.000 06/02/2017	12.52252833	55.79016383
431	13:47:11.500 06/02/2017	12.52252417	55.79016417
432	13:47:12.000 06/02/2017	12.52252017	55.79016433
433	13:47:12.500 06/02/2017	12.52251583	55.79016483
434	13:47:13.000 06/02/2017	12.52251133	55.79016550
435	13:47:13.500 06/02/2017	12.52250667	55.79016633
436	13:47:14.000 06/02/2017	12.52250250	55.79016717
437			
438	13:47:20.000 06/02/2017	12.52245417	55.79017367
439	13:47:20.500 06/02/2017	12.52244967	55.79017450
440	13:47:21.000 06/02/2017	12.52244483	55.79017517
441	13:47:21.500 06/02/2017	12.52243983	55.79017567
442	13:47:22.000 06/02/2017	12.52243433	55.79017633
443	13:47:22.500 06/02/2017	12.52242850	55.79017700
444	13:47:23.000 06/02/2017	12.52242217	55.79017800
445	13:47:23.500 06/02/2017	12.52241600	55.79017883
446	13:47:24.000 06/02/2017	12.52240933	55.79017983
447	13:47:27.000 06/02/2017	12.52236867	55.79018633
448	13:47:27.500 06/02/2017	12.52236200	55.79018750
449	13:47:28.000 06/02/2017	12.52235567	55.79018850
450	13:47:28.500 06/02/2017	12.52234983	55.79018950

Figure 16: GPS data in u-center.

As seen in figure ??, the GPS was unable to record the UTC time stamp for some time period, usually lasting only a few seconds. This creates a single empty row in the table, only containing the index number.

- Field test 1 contains GPS measurements from 13:44:39.500 to 13:55:47.000.  
The longitude range is  $[12.52213^{\circ} ; 12.52262^{\circ}]$  and the latitude range is  $[55.79000^{\circ} ; 55.79019^{\circ}]$ .
- Field test 2 contains GPS measurements from 13:56:34.000 to 14:17:30.500.  
The longitude range is  $[12.52223^{\circ} ; 12.52260^{\circ}]$  and the latitude range is  $[55.79003^{\circ} ; 55.79017^{\circ}]$ .
- The distance test does not contain any GPS measurements.

## 8 Data processing

The majority of the processing is done in MATLAB using a program build around the script `Mainscript_GAMMA.m` created specifically for this project. `KMSTrans2` is needed to efficiently transform the GPS data from longitude and latitude to easting and northing coordinates. The MATLAB script calls functions based on input from the user.

### 8.1 Gamma data formatting

The first step in processing the data is to extract the desired data from the gamma data captured through Wireshark. This is done using the function `Packet.Format`, which finds the index, UTC time stamp and the bytes containing information about the gamma radiation for each packet.

In the `.txt` file, each packet takes up seven lines, and the index, time stamp and bytes are all placed at the same positions for each packet. `Packet.Format` locates and saves the positions for the index and time stamps as arrays. The bytes containing the gamma data are found by locating the report identifier, which has the byte value 04. Then all bytes after the report identifier are saved to a string. A for-loop is run to find the values for each packet.

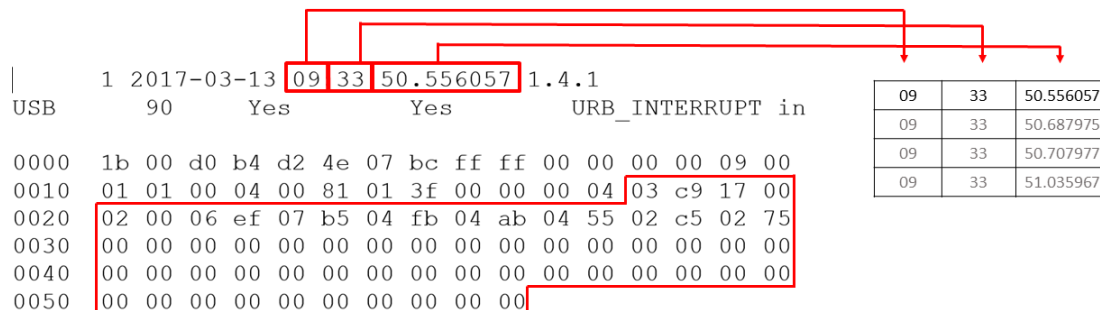


Figure 17: UTC time formatting done by `Packet.Format`.

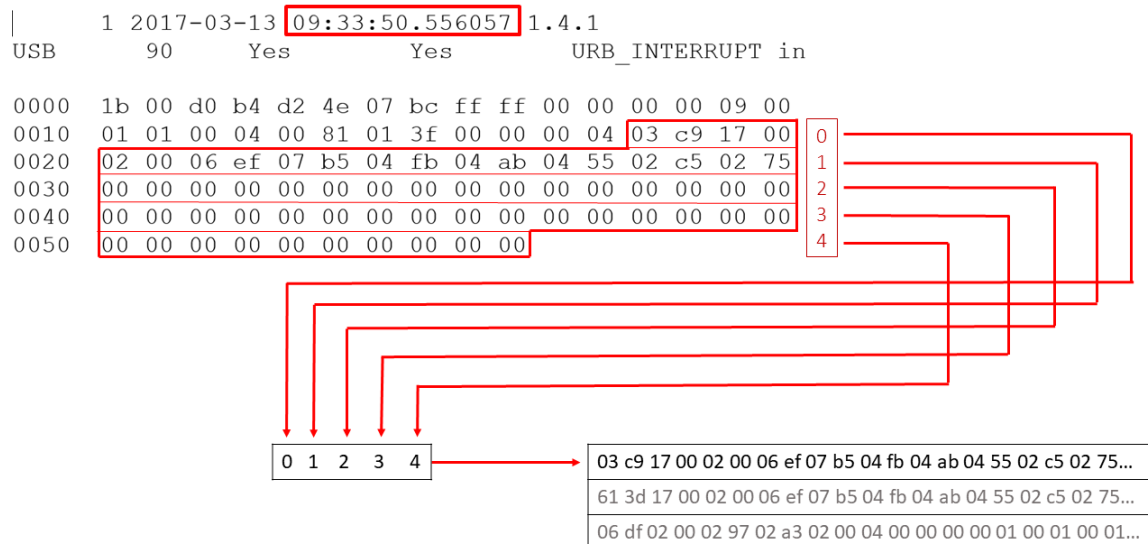


Figure 18: `Packet.Format` making the different lines of data into a single cell.

The function `Packet.Analyzer` takes the string containing the channel bytes and finds the valid channels. First the string is split into hexadecimal numbers of 2 digits using the MATLAB function `strsplit`. Then a for-loop is run for half the length of the string, corresponding to the number of channels. First the two first bytes are located, and they are grouped into 2 (3 digits - 12 bits describing the channel number, and 1 digit - 4 bits determining whether the channel is valid or not). `Packet.Analyzer` then tests if the channel is valid by checking if the last hexadecimal number is even or uneven. If it is uneven, the channel

is valid and the loop continues to check the next channel, but if it is even, the loop breaks and saves all valid channel numbers.

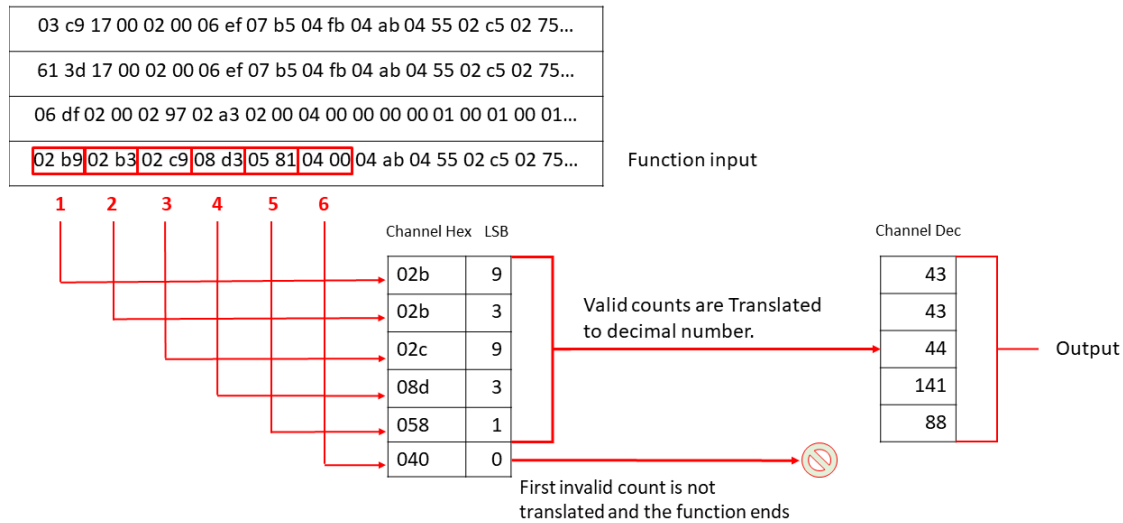


Figure 19: Packet\_Analyzer

Lastly the counts per second is calculated using the function `Counts_Per_Second`. The function finds the counts per packet, and then floors the seconds for each packet to match the frequency of the GPS measurements. Floor is chosen instead of ceil, because there is a small delay between the detection and Wireshark logging the data capture time. Now the amount of counts per second is calculated. First the function finds the hour-values containing data, and a for-loop is run from the first hour to the last hour. For each hour-value another for-loop is run to find all minute-values for each hour-value. At last a for-loop is run to find all second-values for each minute-value.

Now for each second an if statement checks if there are any counts measured that second. If there are any counts measured, the function adds the counts together. If there are no counts measured, the function writes zero and continues to the next second. The function adds zeros for 1.000.000.000 seconds before stopping. Then it finds the last second-value that has measured counts, and deletes all seconds after that.

Because the hours are found using the MATLAB function `unique` with the stable order, it is possible to perform measurements over UTC midnight and still process the data correctly. If the first measurement was made between UTC 22:00:00 and 22:59:59, the MATLAB script can handle data until UTC 21:59:59. The function can handle a maximum of 23:59:59 hours.

## 8.2 Coordinate formatting

GPS coordinates measured in longitude and latitude are saved as a `.txt` file using the function `LatLon_save.m`. The function opens a dialog box for saving files and lets the user select the file name and directory. The saved file is then uploaded to KMSTrans2, where the coordinates are transformed to coordinates of easting and northing. This is done under the "batch" menu point by selecting "TEXT" under Driver/Format and then uploading the saved latitude and longitude and selecting an output destination and file name.

The input system is set to `geo_etr89` which is a geographical reference system, and the output system is set to `utm33_etr89` which is the Universal Transverse Mercator zone 33. Both use the European Terrestrial Reference System. The coordinates in easting and northing are now loaded into MATLAB using the function `KMS_Load.m`.

## 8.3 Data pairing

Because the data might not be of the same length, the data needs to be paired so that only the overlapping data is used for further analysis. This is done by the MATLAB function `Data_Pairing`, which uses multiple for-loops and if-statements to delete all gamma data that does not correspond to data in the

GPS measurements, and the same for GPS data that does not correspond to gamma data.

The first loop locates all GPS time stamps not equal to a whole second, and deletes these measurements from the GPS data set, which include the GPS time stamp, easting, northing, longitude and latitude. This is done to ensure that the GPS data matches the computed counts per second.

The next loop uses `ceil` on the gamma time stamp and an if-statement finds all seconds equal to 60 and makes them equal to zero. Because of this, the minute time stamp gets a minute added, and an if-statements check if the minute reach 60. If so the minute value is transformed to zero and the hour value gets added one hour. Lastly the hour value is checked to see if it reaches 24, and if so the value is likewise transformed to zero.

The next loop uses `unique` to find all the gamma time stamps containing counts. Now the locations of the counts per second containing no counts is found and the time stamp corresponding to that count is inserted into the vector of gamma time stamps containing no counts. This is done to ensure that the gamma time stamp matches the counts per second since a zero in the counts per second corresponds to a missing time stamp in the gamma data set, and therefore the vector containing counts per second will be longer than the gamma time stamp. As with the previous loop, if-statements are run to ensure that if the seconds, minutes or hours exceeds their limits, the numbers are transformed to their appropriate values.

Now the gamma data set and the GPS data set are ready to be compared. First a loop is run for all gamma time stamps, and each time stamp is compared to the GPS time stamp. If a time stamp does not exist in the GPS time stamp, the gamma time stamp and the corresponding counts per second is deleted. Then a loop is run to find the GPS time stamp that does not exist in the gamma time stamp. If the GPS time stamp does not correspond, it is deleted along with the corresponding easting, northing, longitude and latitude. Now the two data sets have the same length and it is then possible to interpolate the counts per second based on easting and northing coordinates.

## 8.4 Energy

It can be of interest to know the energy of the measured counts or to process data within a specific energy range. All functions related to energy can be run without GPS data. If the gamma data and GPS data has been paired some of the captured gamma data might be lost if they do not match the GPS data.

The energy of each count can be calculated. This is done with the function `Energy_Per_Second.m`. The function takes the gamma packets, which is an array containing the channel number of each measurement and calculates the energy of each packet. There is a linear relation between the channel number and the energy given by the calibration equation:

$$\text{Energy} = -478.06 \text{ eV} + \text{Channel} \cdot 754.77 \text{ eV} \quad (3)$$

Since the channel numbers run from 0 to 4095, the energy range is  $-478.06 \text{ eV}$  to  $3.09 \cdot 10^6 \text{ eV}$ . Then a for loop is run for all values of the counts per second array. Then the number of energy measurements corresponding to the amount of counts are grouped into a cell. The output is an array of the same length as the counts per second, with each row containing a cell of all measured energies during that second.

The next function `Gamma_Spectrum.m` creates a plot showing the counts as a function of energy. The function `Energy_Interval.m` lets the user choose an energy interval for further data processing of the energy data. All data not within the chosen interval is deleted, and number of energies in each cell is calculated, creating an array of counts per second within the energy interval. This function is optional and is only used for creating an energy interpolation. If an energy interval is selected, the plot created using the `Gamma_Spectrum.m` function will show the selected interval.

## 8.5 Interpolation

Now the data is ready to be interpolated. The function `Interpolation_Counts.m` interpolates all counts based on easting and northing coordinates, and the function `Interpolation_Energy.m` interpolate the

counts in a chosen energy interval based on easting and northing coordinates. The MATLAB function `meshgrid` is used to create a 2-dimensional data set over the area where the counts should be interpolated. The spatial resolution of the grid can be chosen by the user via the function `Interp_res.m`, and a maximum resolution of  $0.01 \text{ m}^2$  is recommended. Then the counts are interpolated using `griddata`, which handles scattered data.

## 8.6 Present data

The data can be presented in two ways. The first is as a contour plot of the interpolated counts based on the easting and northing coordinates either with or without a selected energy interval. This is done using `Interp_Plot_Counts.m` for all counts, or `Interp_Plot_Energy.m` for counts in a selected energy interval. Both functions uses the MATLAB function `contourf`, which creates a filled contour plot of the interpolated data.

The second way of presenting the data is by using `ge_plot3` from the Google Earth Toolbox created by Scott Lee Davis. The function creates a 2-dimensional route based on longitude and latitude, and the height of the route is the counts per second. The data is saved as a `.klm` file using the function `GoogleEarth_Counts.m` for all counts, or `GoogleEarth_Energy.m` for counts in a selected energy interval. The saved data can then be presented in Google Earth.

## 8.7 Overview of the program

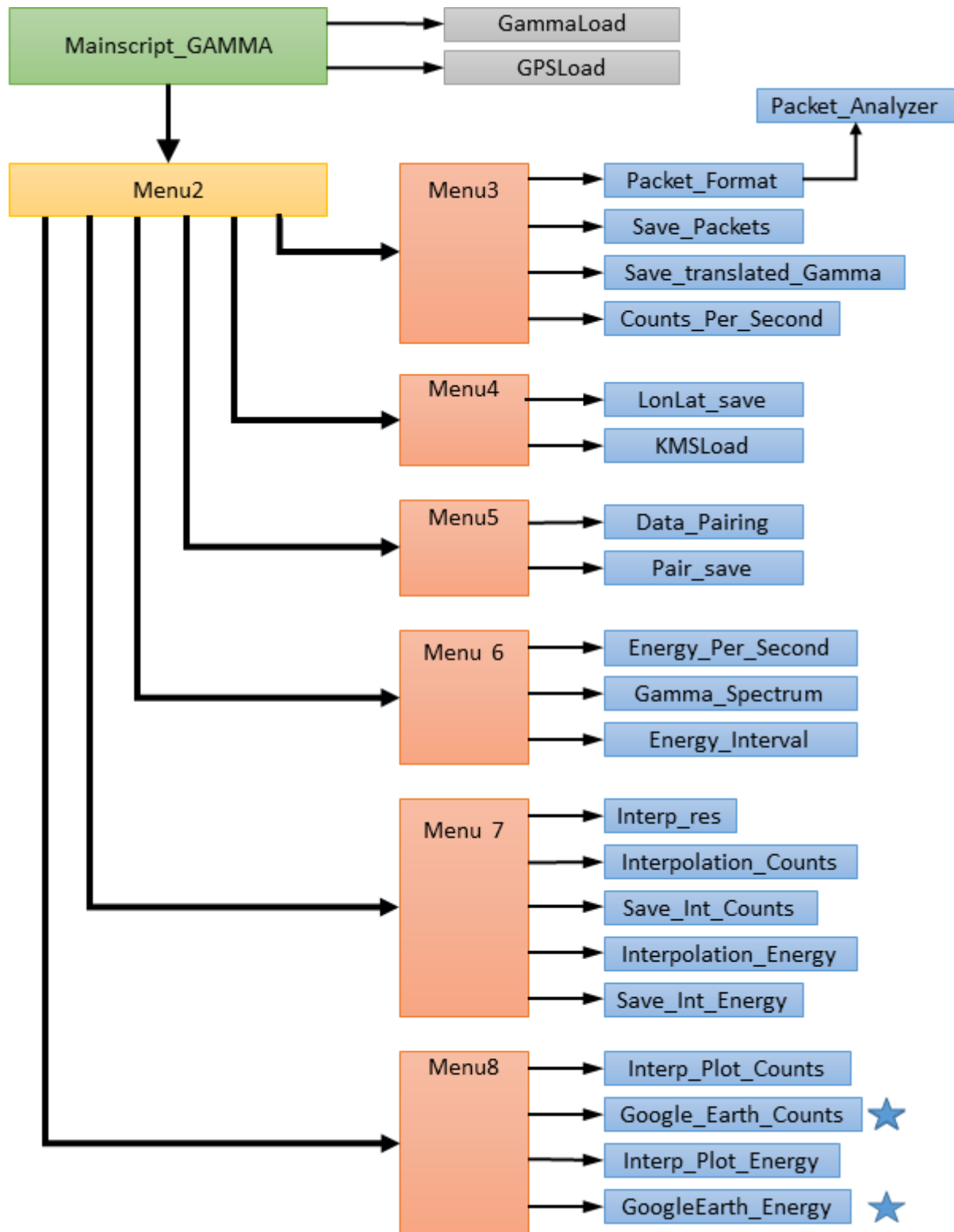


Figure 20: Overview of the program. This picture shows how all of the different functions are connected. Arrows indicate the call of the function which is pointed at. Every item is a Function except Mainscript.GAMMA.

The stars indicate the call of a function called "ge-plot3" from the googleeearth toolbox directory.



## 9 Results

First the data from the two field tests were processed. Figure 21 shows the full gamma ray spectrum for field test 1 and figure 22 shows the full gamma ray spectrum for field test 2. The figures are created using `Gamma.Spectrum.m` without selecting an energy interval.

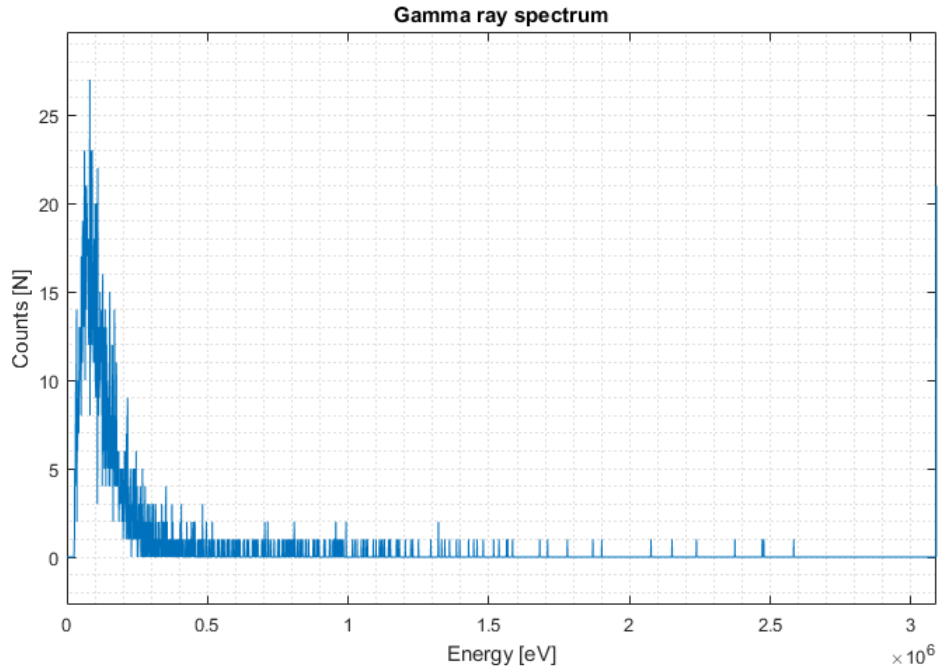


Figure 21: Gamma ray spectrum from field test 1.

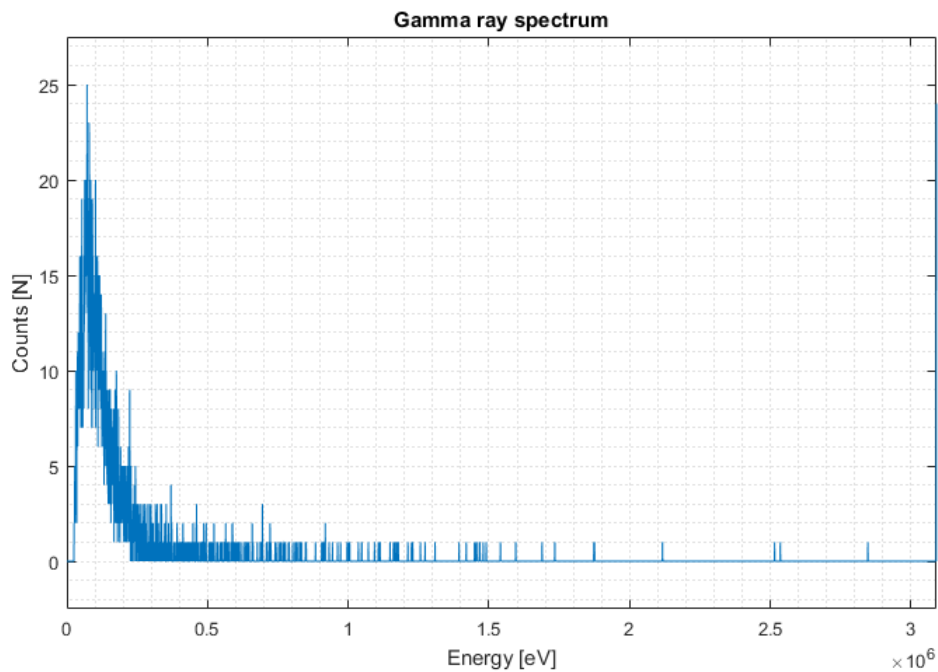


Figure 22: Gamma ray spectrum from field test 2.

Neither of the gamma spectra show significant peaks at 1.17 MeV and/or 1.33 MeV corresponding to photons emitted from a Cobalt-60 source.

Now gamma data and GPS data are paired, and the interpolations of all counts based on easting and northing coordinates are created. Figure 23 shows the interpolation for field test 1 and figure 24 shows the interpolation for field test 2. The interpolation grid size is  $0.25\text{m}^2$ .

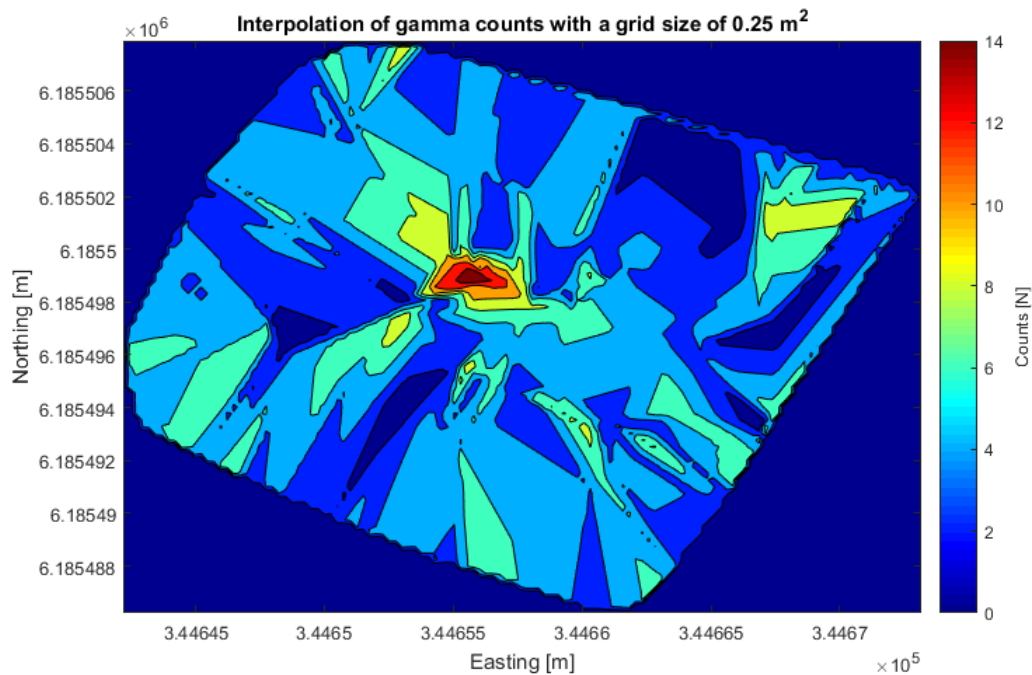


Figure 23: Counts per second for field test 1.

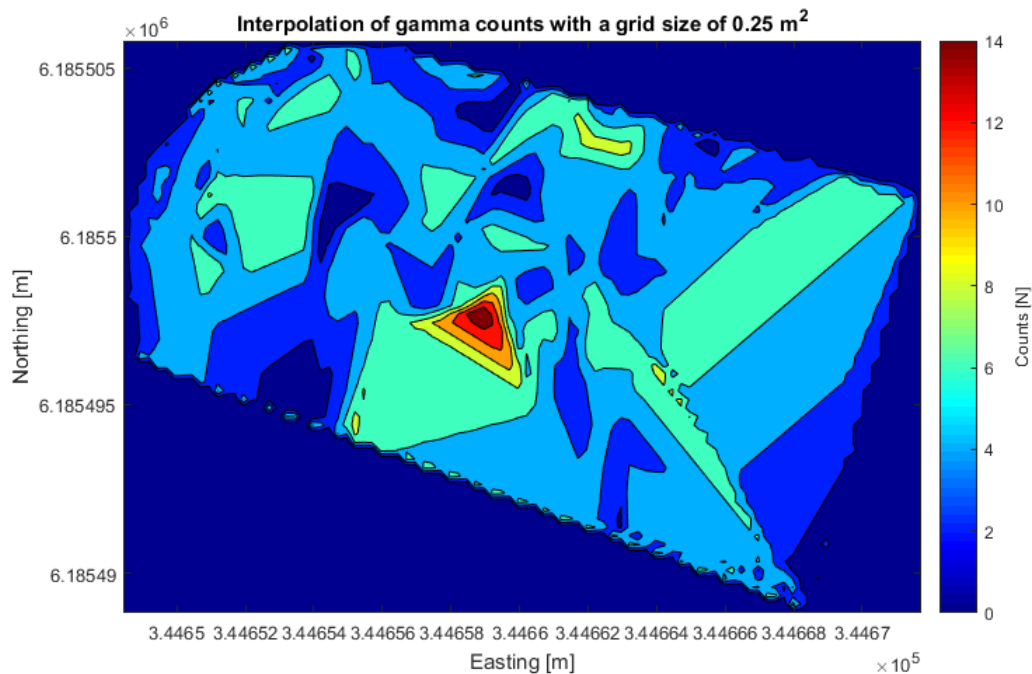


Figure 24: Counts per second for field test 2.

Both figures show a larger amount of counts at the position of the Cobalt-60 source. It peaks at around 14 counts per second at the source, and the counts elsewhere does not reach more than nine counts per second and generally varying between zero and six counts per second. The position in coordinates of easting and

northing of the source is found to be (334656m ; 6185499m) for field test 1 and (334659m ; 6185498m) for field test 2.

Now an energy interval is chosen and the counts per second within the interval is plotted. The energy interval is [500,000 eV ; 3,000,000 eV], which contain the majority of the background radiation. This is done to ensure that the peaks in the centre of the area is a gamma source and not just a large amount of background radiation. Figure 25 show the counts per second in the energy interval for field test 1, and figure 26 for field test 2.

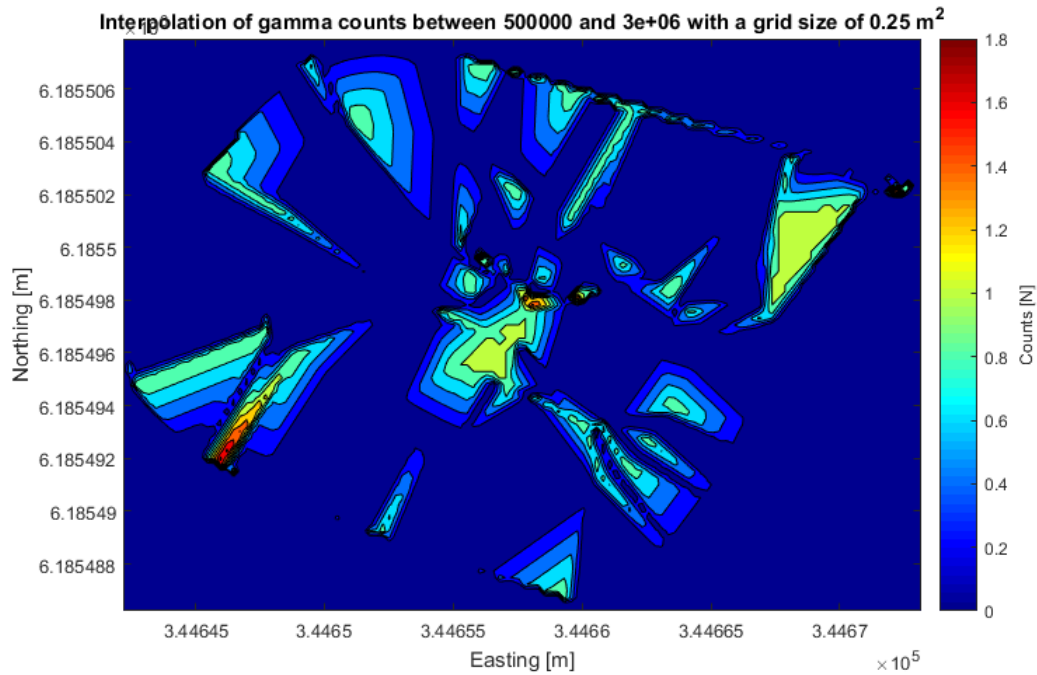


Figure 25: Counts per second for field test 1 in the energy interval [500,000 eV ; 3,000,000 eV].  
Note: Northing unit is  $\times 10^6$ .

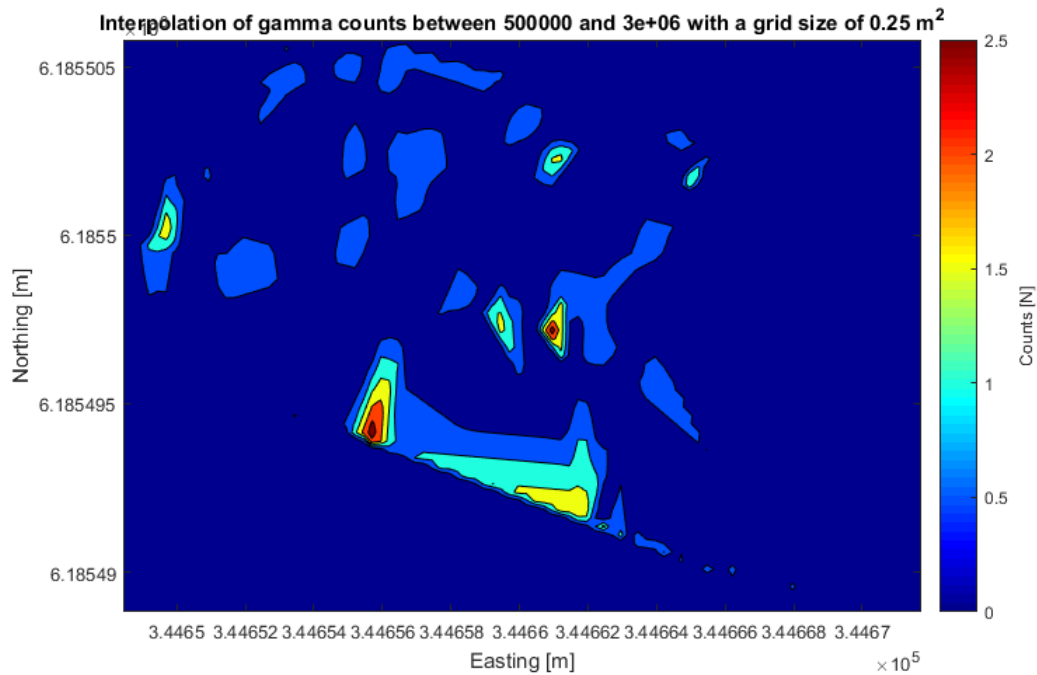


Figure 26: Counts per second for field test 2 in the energy interval [500,000 eV ; 3,000,000 eV].  
Note: Northing unit is  $\times 10^6$ .

There still appear to be counts at the position of the Cobalt-60 source, but also at other positions further away from the source. This happens because of the limited amount of measured counts. As seen in figure 21 and 22 the energy levels caused by background radiation are as prominent as the high-energy background radiation. Even if all the counts at the energy levels of a gamma source is measured during the same second, another second might contain the same amount of counts just from high-energy background radiation. The two measurement will have the same peak height in the interpolation.

To confirm that the source is Cobalt-60, enough measurements are needed to see one prominent peak from the gamma source, even when removing the majority of the background radiation. Figure 27 shows the gamma spectrum for the distance test, where data was collected close to the Cobalt-60 source.

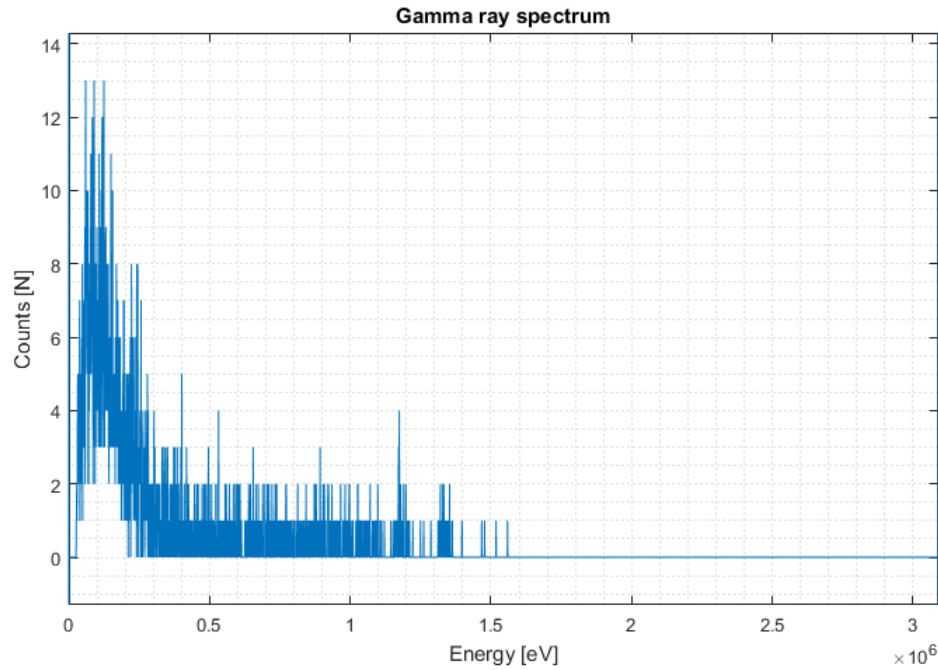


Figure 27: Gamma ray spectrum from the distance test.

The gamma ray spectrum appears to have a peak at 1.17 MeV and a relatively large amount of counts at 1.33 MeV. The two energies correspond to the energies emitted in the radioactive decay of Cobalt-60. Figure 28 shows the gamma spectrum zoomed in at the these two energy intervals. The intervals boundaries are  $\pm 2.0\%$  of the emitted energies, which correspond to the energy resolution of the Kromek GR1-A+.

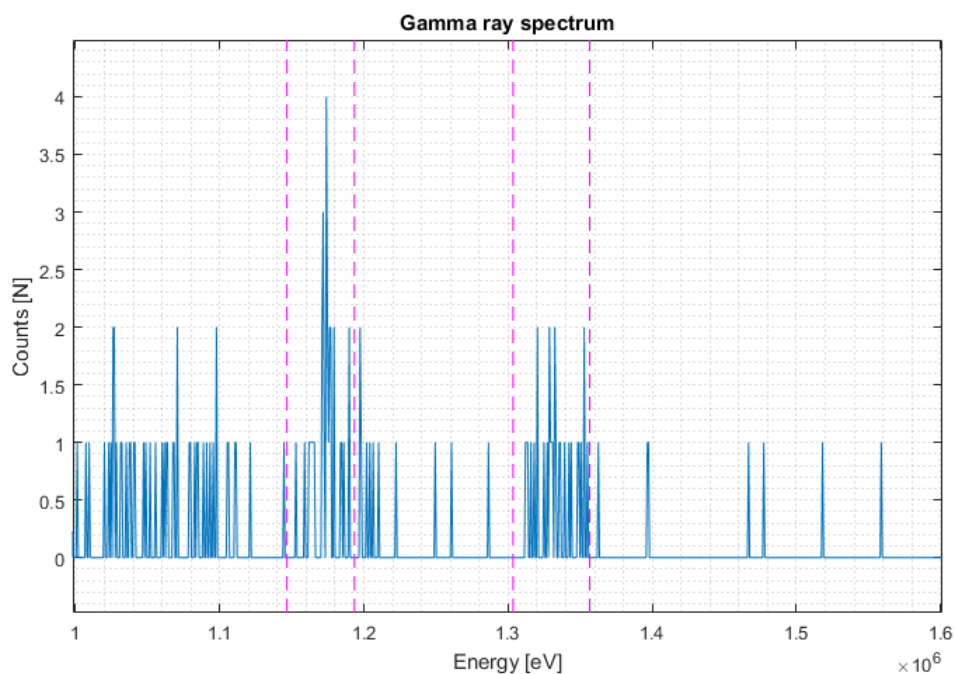


Figure 28: Gamma ray spectrum from the distance test zoomed in at the energies emitted from Cobalt-60.

By using the interpolation from the field tests and the gamma ray spectrum from the distance test, the gamma source has been located and identified as Cobalt-60.

## 10 Discussion

As a result of the post processing of the data measured during the three tests, the position of the source was located and identified as Cobalt-60. The results were based on very few tests, which did not contain many measurements, and so the results are not especially convincing despite being successful. This is however related to a few different factors. This include the activity of the gamma source, the speed of the UAV platform, the flight grid size and interpolation grid size as well as the hardware on the payload.

Using equation (2) the activity of the Cobalt-60 source was found to have an activity of 0.1 mCi today. Since a radioactive source sends out radiation uniformly in all directions, the theoretical amount of activity measured by a detector can be calculated. Assuming that the average distance from the detector to the source during the second of where the gamma detector is closest to the source (1 meter), the surface area of a sphere of radius = 1m is found.

$$A_{\text{surf}} = 4 \cdot \pi \cdot r^2 \text{ m}^2 = 4 \cdot \pi \text{ m}^2 \quad (4)$$

Now the number of disintegrations reaching the detector per second can be found.

$$A_{\text{detector}} = \frac{A_{\text{detector}}}{A_{\text{surf}}} \cdot A = \frac{0.01\text{m} \cdot 0.01\text{m}}{4 \cdot \pi \text{ m}^2} \cdot 3.7 \cdot 10^6 \text{ Bq} = 29 \text{ Bq} \quad (5)$$

The number of disintegrations reaching the detector per second would be  $\approx 5980 \text{ Bq}$  if the source had the same activity as it did in May 1978.

Because the source has a very low activity it is more difficult to distinguish it from background radiation compared to a source with higher activity. If a high-activity source was used during the tests, the peak at the position of the source would be more prominent, both when interpolating all measured counts and when interpolating only selected energies. When interpolating all counts, the large amount of background radiation will create a relatively uniform interpolation with only one peak from the source. When filtering away the majority of the background radiation, the small amount of background radiation will create a less uniform interpolation, but the peak from the source will appear even more prominent.

Another way to improve the amount of measured radiation is by flying at a lower speed. When flying at a lower speed the amount of time spent close to the source is increased, which will increase the number measured counts corresponding to the energies emitted by Cobalt-60. The more time spent at or near the position of the source, the better the source-to-background-ratio will become.

A way to improve the source-to-background-ratio while still mapping a large area is by first doing a relatively fast flight test (1 m/s) which will show a small peak in the interpolation of the counts per second. Then another test can be made by hovering at the position of the source, which will give a better source-to-background-ratio than flying in a grid. By processing this data, a gamma spectrum showing significant peaks at the energies emitted by the source can be produced.

The results are presented in a  $0.25 \text{ m}^2$  interpolation grid size. If the grid size is very large the position will be less accurate and the source-to-background-ratio will be smaller. If the grid size is very small, the radiation from the source might be divided into multiple less prominent peaks. The best results happen when the grid size is small but still show one prominent peak from the source.

Figure 29, 30 and 31 show interpolations of all counts for field test 1 when using interpolation grid size  $0.05 \text{ m}^2$ ,  $0.25 \text{ m}^2$  and  $1.00 \text{ m}^2$ . The  $0.05 \text{ m}^2$  grid size show a distorted peak, which can make the positioning of the source difficult, whereas the  $1.00 \text{ m}^2$  grid size shows a relatively large area of the peak, making positioning the source difficult as well. A  $0.25 \text{ m}^2$  grid size was therefore chosen as the optimal grid size for the data processing of the two field tests.

When comparing the interpolation of all counts per second from the two tests, they both show one prominent peak and have similar background radiation, meaning that both flight patterns works equally well. The flight patterns can still be improved by choosing a smaller spaced grid. This is however not always possible due to battery life and the size of the area of interest. When using the UAV platform used in the tests, the battery life is 18 min. which gives a flight distance of 216 m at a velocity of 0.2 m/s, which would be a reasonable flight speed for measuring the cobalt-60 source used in the tests, due to its low activity.

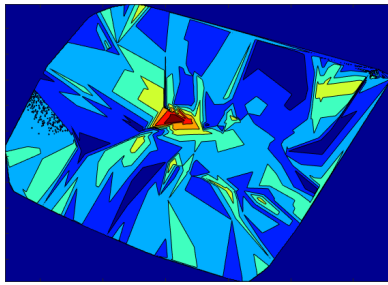


Figure 29: Interpolation grid size: 0.05 m<sup>2</sup>.

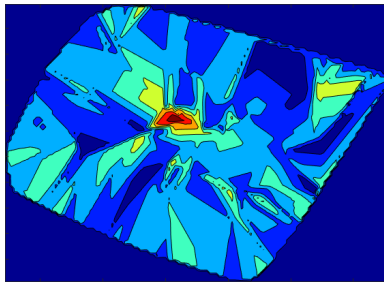


Figure 30: Interpolation grid size: 0.25 m<sup>2</sup>.

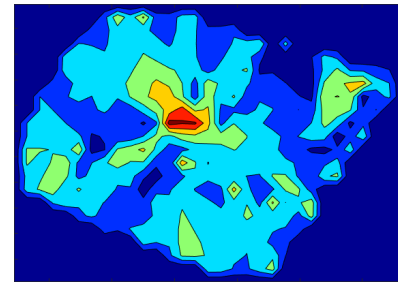


Figure 31: Interpolation grid size: 1.00 m<sup>2</sup>.

During the project two different fitlets were used. They were the same model except the second one was black compared to the aluminium finish on the first. The fitlet which was used in the final version of the payload was the aluminium finish one, since the other fitlet couldn't interface properly with the Kromek GR1-A+ detector. This fitlet has problems with slow responsiveness and something as simple as clicking on the start menu can cause it to hang for a second or two. This sluggish performance is not just in the form of input lag. Launching programs, simple use of software and booting is very slow compared to the black fitlet with the exact same specs. The startup process for instance takes 2-3 times as long as it did on the black one. Originally this was thought to be just nuisance and since the software worked the project went through with the aluminium fitlet.

However after the experiments had been made and the data processing commenced it was found that the GPS data now had empty rows as seen in figure ???. The data before and after the empty row can be up to several seconds apart creating an area of the mapping without any data. This behaviour had not been observed with the black fitlet. Originally it was assumed that connection was lost to the satellites. It was then observed while monitoring a 10 minute stream of GPS data collection that the data written in the table would slowly lag behind. The time and location was correct, but the data was not being written as fast as it was coming in. The empty rows are suspected to be a way for u-center to catch back up after what we suspect to be a filled buffer. The empty row was only there for the time a normal data input would be there and not the time frame difference between the two counts before and after. It looks like beyond a certain point the buffered data is discarded so it can keep capturing the data. A supporting factor that shows that empty rows has to do with the computational power of the fitlet, is that during GPS data capture test it could be seen that the GPS receiver board always had strong signal to at least 9 GPS satellites.

During the project there was experimented with using the BeagleBone Black instead of the fitlet due to its lower weight and it would be easy to integrate to more precise GPS solutions which were already in use at DTU. The BeagleBone has very low weight compared to the Fitlet, but the Kromek GR1-A+ and the BeagleBone was unable to handshake due to issues with matching the report descriptors. Because of this, the BeagleBone was unable to receive data from Kromek GR1-A+. The Linux driver from Canberra might solve this, but if the Kromek GR1-A+ driver makes it wait for input instead of sending a constant data stream, then it would be unusable with the MATLAB program. Another perk of the fitlet is its ability to synchronise its time over the internet via hot-spot leading to accurate pairing of the data from the gamma detector and GPS system.

## 11 Further Improvements

### 11.1 Program further development

The interpolation maps which the program currently can create are only 2D. Adding the option choose a third dimension to the plot as either intensity or actual flight height would help give more information on the gamma radiation in the surveyed area.

At the time of this report the program only present the interpolated maps and does not give location of the areas with high intensity. A feature which could be added would be the ability to search for certain combinations of peaks and locate their sources or make the program able to interact with a database, which contains the different spectra for different sources. Another feature in the same ballpark would be to have the program point out areas where different peaks appear on the interpolation and give an estimate of the locations/centre of the radioactive material.

The program currently only has the ability to plot a 3-dimensional route in Google Earth where the height is given by the counts per second, where 1 count equals 1 meter. A Further development for this feature would be to add the ability to plot the interpolated data in Google Earth or to create functionality with the Open Source QGIS.

### 11.2 Raspberry Pi with Windows 10 IoT Core.

Since the BeagleBone Black was not able to be used due to the problems with the interfacing between the gamma ray detector and the OS then another way to go about it might be the Raspberry Pi running Windows.

The Raspberry Pi is a single-board computer, with a much smaller form factor than the fitlet and comparable to the BeagleBone Black in size, is capable of running "Windows 10 IoT Core". The Iot is the designation that refers to an OS which is designed specifically for use in small footprint, low-cost devices and IoT(internet of things) scenarios. It is a re-branded version of the earlier embedded OSs. Windows 10 IoT Core runs on ARM devices like the Raspberry Pi and on both x86/x64 devices.

However, at the moment the Raspberry Pi is not an usable platform for one main reason. The reason is that the Wireshark buildbots does not currently include the capability to build a version of the Wireshark software for ARM platforms.

However even before this one would need USBPcap library for ARM as well. In the field of computer network administration, packet capture (pcap) consists of an API (application programming interface) for capturing network traffic. Unix-like systems implement pcap in the libpcap library whereas Windows uses a port of libpcap known as WinPcap.

Since there isn't even a WinPcap for ARM yet then it is safe to assume that it will be at least some years before Wireshark is able to be run on windows ARM platforms.

### 11.3 Intel NUC

The Intel NUC (Next Unit of Computing) is another small form factor computer which is designed by Intel. Being designed by Intel the platform uses Intel processors ranging from 15 W i3-XXXX-u series all the way up to the 45 W i7-XXXXHQ. Sold as bare bone kits much like the fitlets Intel does offer just the NUC motherboards, which have a built-in CPU for sale.

Using the *NUC6i3SYK* (i3-6100u) as the basis the NUC weighs in at 520 g including SSD and memory and is larger than a fitlet measuring 11.5 cm by 11.1 cm by 3.2 cm. It has 4 x USB 3.0 ( 2 front, 2 rear ) 2 x USB 2.0 ( 2 internal ) so it matches the fitlet in USB connectivity solutions and is more than enough to handle the required 3 for the payload setup. As for wireless connectivity even the least expensive NUC has 802.11ac with Bluetooth 4.0



Its strength lies in its processor range. While the cheapest Sixth generation NUC - the NUC6i3SYK only has the dual core i3-6100u processor it is still more powerful than the AMD A10 Micro-6700T APU found in the fitlet. The 6700t scoring 1787 in CPU Mark with a Single Thread Rating of 738 vs the i3-6100u with CPU Mark score of 3880 and a single Thread Rating of 1334.

Table 1: Comparison of Intel Core i3-6100U and AMD A10 Micro-6700T [27]

Model	L2 Cache + L3 Cache	TDP (Watt)	MHz - Turbo	Cores/ Threads	Process (nm)	Cinebench R11.5 CPU Single 64Bit	Cinebench R11.5 64Bit
Intel Core i3-6100U	512KB + 3MB	15	2300	2/4	14	1.11	2.75
AMD A10 Micro-6700T	2MB	5	1200 - 2200	4/4	28	0.55	1.49

One of the problems with the current payload is that the fitlet is very slow to work with and an increase in processing power would help with both the troubleshooting and further development but also with the use of the application itself, and the increase in performance per WATT would favour the NUC as well despite the higher processor TDP of 15 W vs the 6700T's 5 W since the architecture and manufacturing process gives the Intel processors an advantage in power per Watt.

## 12 Conclusion

The development of the gamma mapping system has been a success as the payload and program is working as originally envisioned. While there are some problems with the flight which makes for some gaps in the data the measurements are still usable and the problem can be solved with a new flight or Intel NUC pc.

The written MATLAB program works as intended. It can translate the captured data-packets from the Kromek GR1-A+ CZT gamma detector. The program can pair the translated data with GPS positions and interpolate the data in order to generate maps of the radiation over the surveyed areas. These maps can be restricted to multiple energy intervals. Gamma spectra of the captured data can be used to help identify the source(s) present in the area of interest. Because the program has a simple GUI it is possible for people without any knowledge of how the program functions to still use it to process data.

By using the constructed payload and the MATLAB data processing program a relatively weak Cobalt-60 source was successfully located and identified. The program can locate sources with meter-precision. The system was tested to its limits with the weak Cobalt-60 source having only 2 and 3 measured counts in the corresponding energy levels. The way to use the system when dealing with low intensity sources is to first fly a mission at 1 m/s and then after the first batch of data has been processed perform another flight with a speed of 0.2 m/s around the area of the measured peak. The system works with higher activity sources and is ideal to use in either Greenland or at nuclear accident sites.

## A Payload Setup

The Payload setup has two main parts. The first is the software setup of the fitlet. The second is assembling the hardware and mounting everything in the payload box.

### A.1 Setting up the fitlet

For setting up the fitlet there are three main steps.

1. Automatic ad hoc network/access point.
2. Remote control over network
3. Software installation

#### A.1.1 Ad-hoc WiFi network

The first step in setting up the ad hoc network is to check whether or not the currently enabled drivers support the hosted network function. This is done by entering the command: "netsh wlan show drivers"

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\drone-fitlet>netsh wlan show drivers

Interface name: WiFi

Driver               : 802.11n USB Wireless LAN Card
Vendor               : Ralink Technology, Corp.
Provider             : MediaTek, Inc.
Date                 : 21/04/2015
Version              : 5.1.22.0
INF file             : ???
Type                 : Native Wi-Fi Driver
Radio types supported : 802.11b 802.11g 802.11n
FIPS 140-2 mode supported : Yes
802.11w Management Frame Protection supported : Yes
Hosted network supported : Yes
Authentication and cipher supported in infrastructure mode:
    Open                None
    Open                WEP-40bit
    Open                WEP-104bit
    Open                WEP
    WPA-Enterprise      TKIP
    WPA-Enterprise      CCMP
    WPA-Personal        TKIP
    WPA-Personal        CCMP
    WPA2-Enterprise     TKIP
    WPA2-Enterprise     CCMP
    WPA2-Personal       TKIP
    WPA2-Personal       CCMP
    Vendor defined      TKIP
    Vendor defined      CCMP
    WPA2-Enterprise     Vendor defined
    WPA2-Enterprise     Vendor defined
    Vendor defined      Vendor defined
    Vendor defined      Vendor defined
    Vendor defined      Vendor defined
Authentication and cipher supported in ad-hoc mode:
    Open                None
    Open                WEP-40bit
    Open                WEP-104bit
    Open                WEP
    WPA2-Personal       CCMP
    Vendor defined      Vendor defined
Wireless Display Supported: Yes (Graphics Driver: Yes, Wi-Fi Driver: Yes)

C:\Users\drone-fitlet>
```

Figure 32: The result of the "netsh wlan show drivers" on the fitlet.

The next step is to give windows permission to set a hosted network. This network also needs an SSID (service set identifier) as well as a key (password) of no less than 8 characters. This can be done using the netsh wlan command again:

```
"netsh wlan set hostednetwork mode=allow ssid=** key=**"
```

where the \*\* represent the SSID and the password of the hosted network. we have used the SSID: "Drone-fitlet" and the key: "zkp9m96b".

The next step is to initiate the hosted network. This can be done by the command: "netsh wlan start hosted network". This command needs to be run at every time the fitlet boots or the hosted network won't be initiated. In order to do this a batch script is made which contains the command.

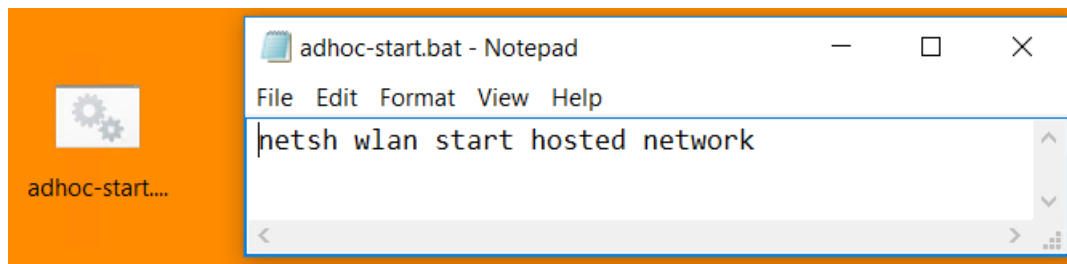


Figure 33: The batch script on the desktop and its content.

There are two ways to proceed after creating the batch file. The first is to have it run as a startup program. This is done by first creating a shortcut for the batch file. Once the shortcut has been created, right-click the file and select Cut. Then press "Windows logo key" + "r" to launch the run command window. Type "shell:startup" to open the start folder. Once the Startup folder has been opened click the "Home" tab at the top of the folder and select "Paste" to paste the shortcut into the folder. This will make it so that every time the user logs in then the batch script will be executed. This method therefore requires the fitlet to be set up to automatic log on to an admin account on boot. This was enabled during the initial installation of windows 10.

However on some windows 10 systems there is a bug with the automatic log on functionality and the remote desktop application which results in the local user being logged on automatically every time a remote desktop user logs on to the PC. If this is the case then the previous method cannot be used since it requires the automatic log on after boot.

A solution to this problem is to use Windows build in "Task scheduler" application.

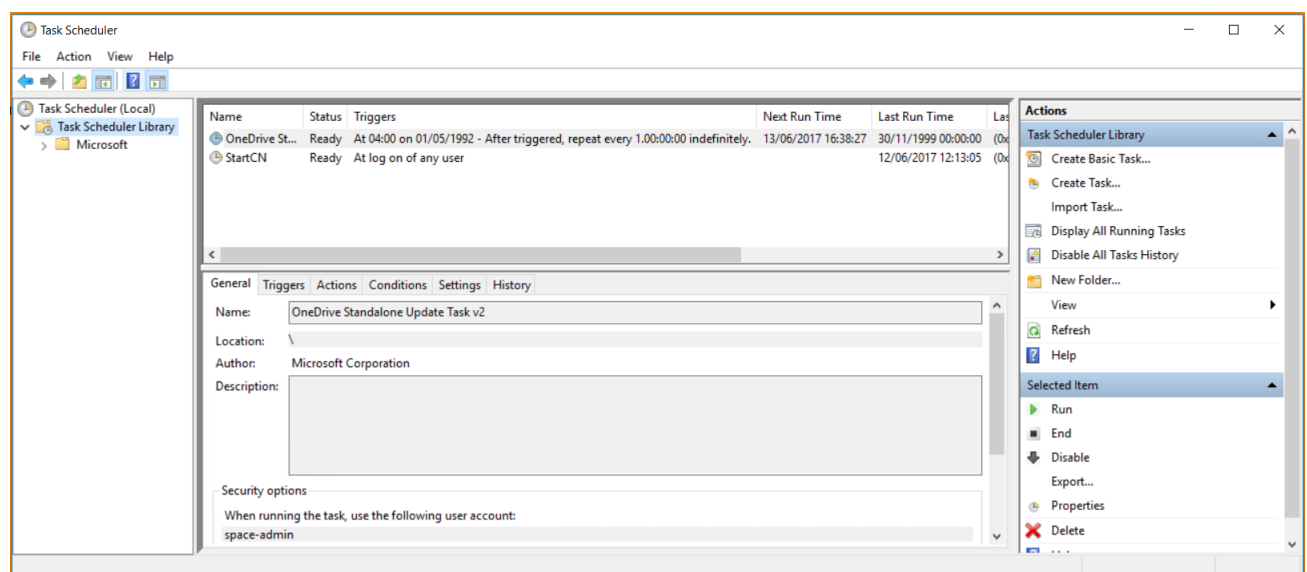


Figure 34: The Task scheduler application.

Using this application a new task must be created. It is important to check the two boxes for: "Run whether user is logged on or not" and "Run with highest privileges" It also needs to be configured for windows 10.

Next the a trigger must be added. Go to the "Triggers" tap and select "New..." then on the drop down menu "Begin the task" select "At startup". Next step is to navigate to the "Actions" tab and select "New...". The default new action should be "Start a program" then under click "Browse..." and locate the batch script. The task now has all the correct parameters and can be created. On every startup the batch script will now be executed before any log on and create the hosted network.

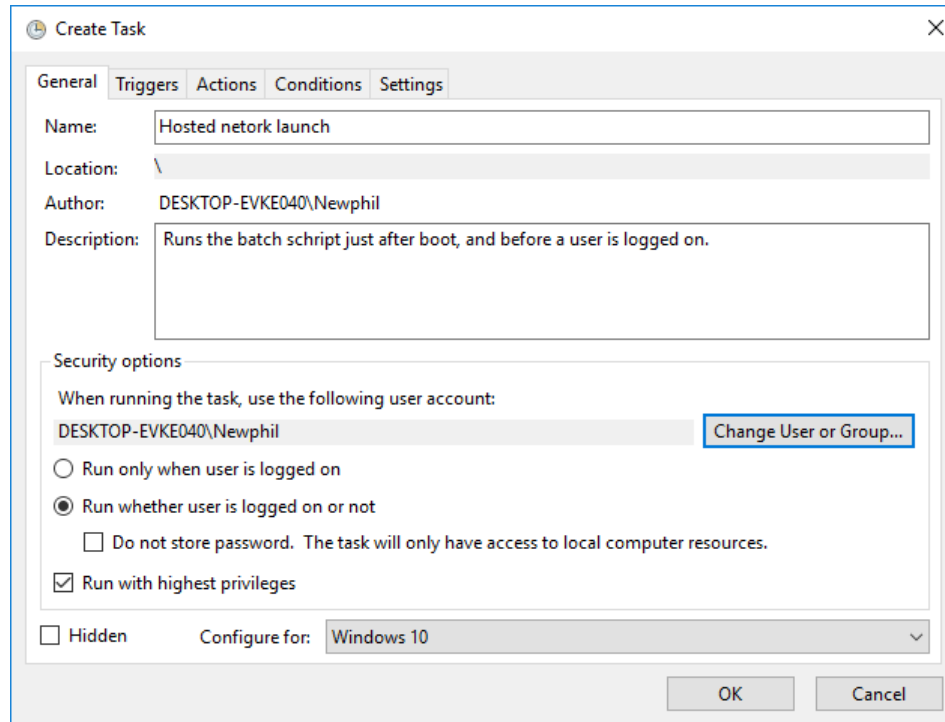


Figure 35: The Task scheduler application \*ignore "Author" and the user account chosen for running the task.

### A.1.2 Remote control over network

Before enabling RDC hosting on a Windows 10 pc its important to check wether or not the OS supports it, as the Home version of windows only has remote control capabilities and not hosting.

To enable the remote control functionality on a Windows 10 pc the following steps must be taken:

1. On the remote PC, open **Settings** and select **System > About**. Then, under **Related settings**, select **System info**.
2. In the left pane of the **System** window, select **Advanced system settings**.
3. On the **Remote** tab of the **System Properties** dialog box, under **Remote Desktop** select **Allow remote connections** to this computer and then choose **OK**.

It is important to make sure the settings for sleep and hibernation on the remote computer are set to **Never** since you can't connect to a computer that's asleep or hibernating.

### A.1.3 Software installation

To install u-center go to: <https://www.u-blox.com/en/product/u-center-windows> and download the installation file. Once this file is downloaded simple run it and u-center will be installed. There is no further internet connection required so it can be downloaded on another computer and then installed via

USB on the fitlet.

The Wireshark software can be downloaded from <https://www.wireshark.org/download.html>. It is important to chose the 32-bit for version since the fitlet is running a 32-bit OS. Wireshark can also be installed over USB. During installation it is importan to check the box which says "Install USBPcap". If the USBPcap library is not installed then Wireshark won't be able to gather data from the USB controllers, rendering the gamma data noncollectable.

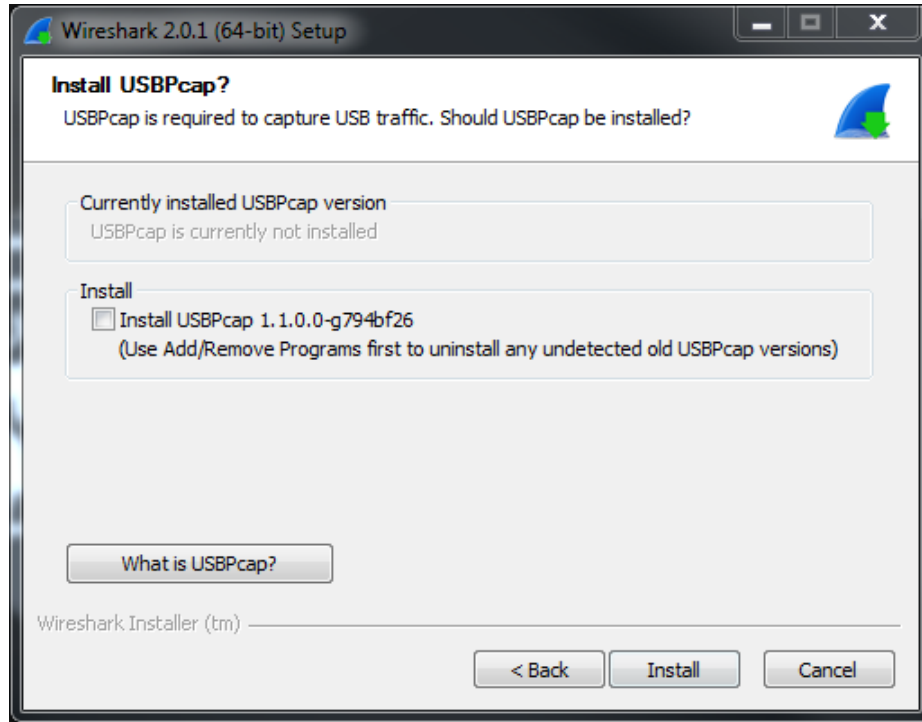


Figure 36: The dialog box which shows up during the installation of the Wireshark software \*The image is from a 64-bit installation on another pc.

## A.2 Payload assembly

There are 6 pieces of electronics that make up the payload they are:

- Fitlet
- Traco power supply
- Kromek GR1-A+
- USB Wi-Fi adapter
- UBLOX NEO-M8T-0-10 GPS
- GPS antenna.

The Wi-Fi adapter, Kromek GR1-A+ and the UBLOX GPS receiver board are all connected to the fitlet with USB, and the Power is via the fitlets DC input. the GPS antenna is attached to the UBLOX NEO-M8T-0-10 via the SMA connector.

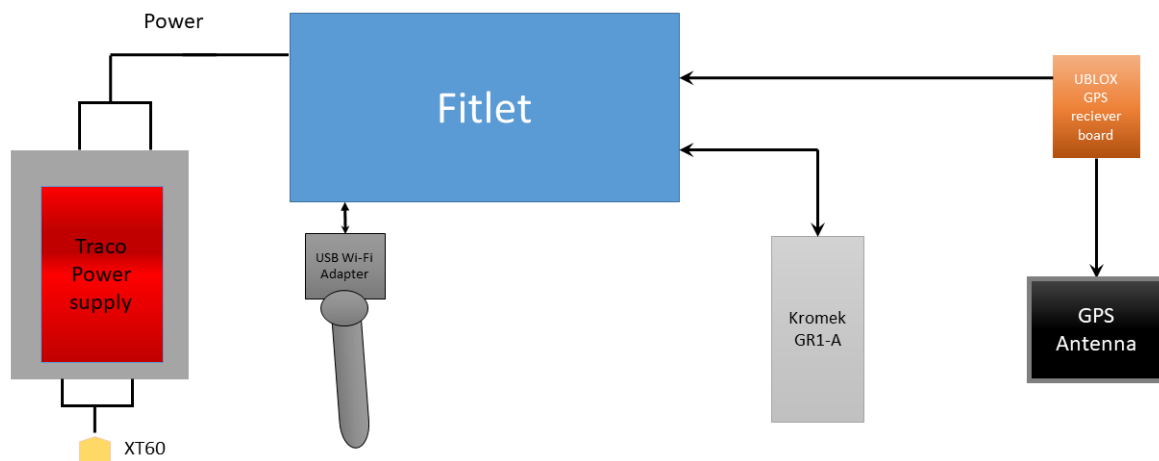


Figure 37: Schematic of the connections between the hardware which where mounted in the payload box.  
 \*Arrows only indicate connections not actual locations of connection to the fitlet.

They payload box was laser cut, but the holes for the screws and strips used to hold the different devices in place was drilled with a hand drill. The next 2 pictures shows the holes and strips on the Payload box:



Figure 38: The payload seen from the top with most of the hardware unmounted in order to show the mounting holes.





Figure 39: The payload seen from underneath. The two large strips hold in the gamma detectors foam and bubblewrap protection, whereas the singular strip holds the UBLOX NEO-M8T-0-10 GPS receiver board.

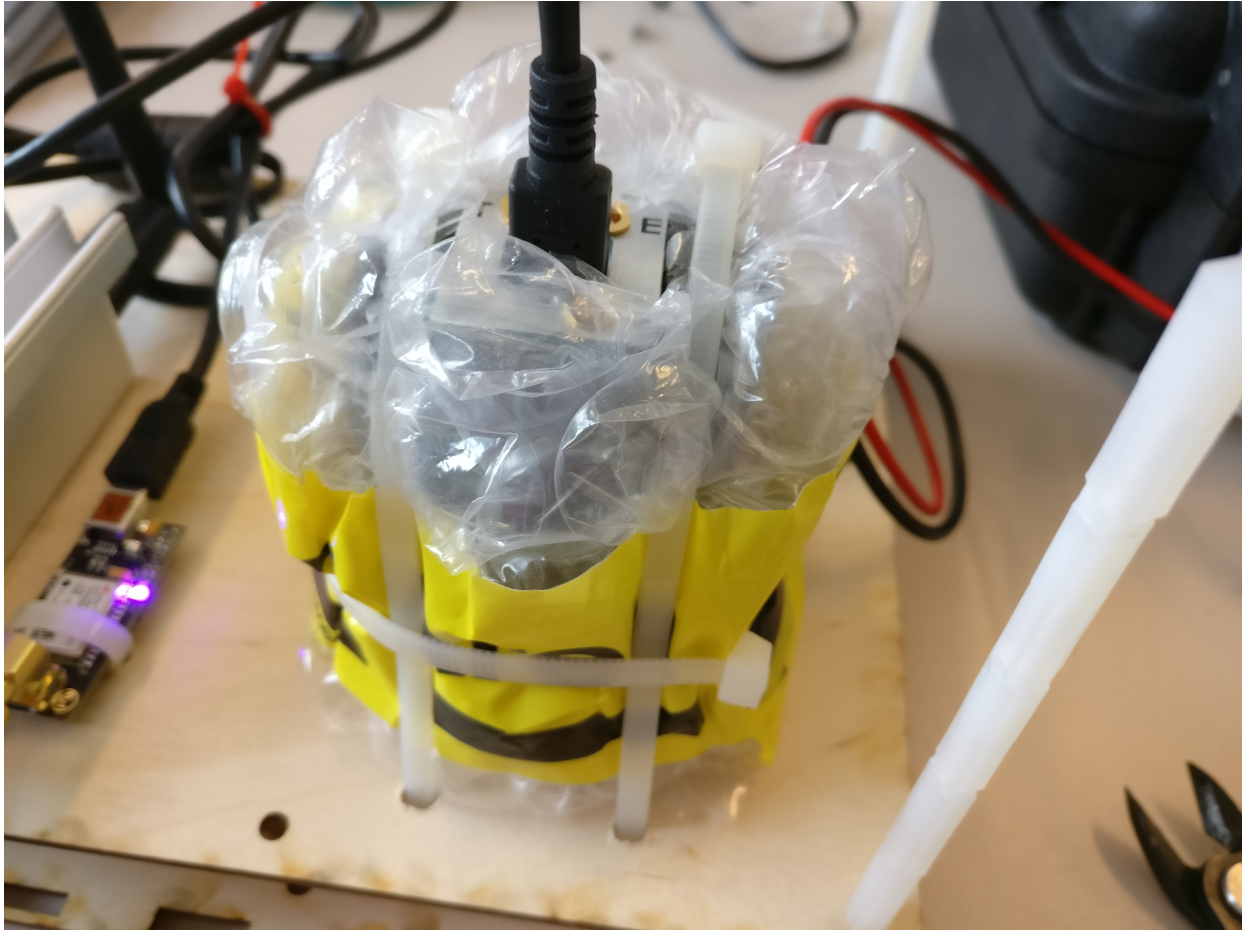


Figure 40: Picture of the mounting solution for the Kromek GR1-A+ gamma detector. The gamma detector needed to be protected from most vibrations and to soften the blow of potential rough landings so we build a double layer of soft transportation foam and bubble wrap to protect it. The foam is held in place with the two large strips. It is possible to remove the gamma detector without having to remove anything else but requires some finesse since we needed to be sure the gamma detector was safely positioned.

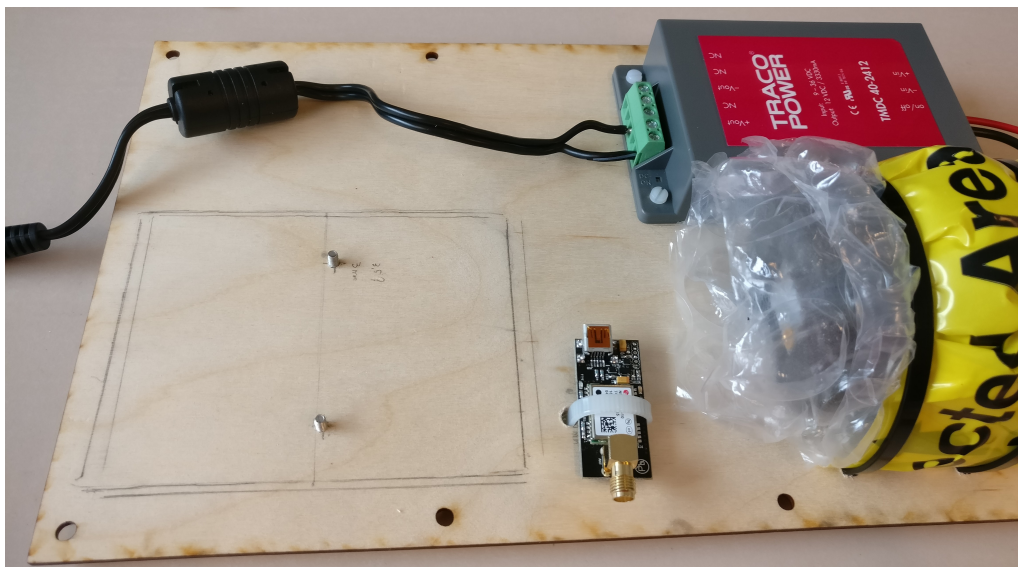


Figure 41: Early start of the assembly process of the first iteration of the payload. Gamma detector pointing forward rather than the final downwards solution.





Figure 42: This was the first iteration version of the payload with the black (faulty) fitlet and the Gamma detector pointing forward rather than the final downwards solution.

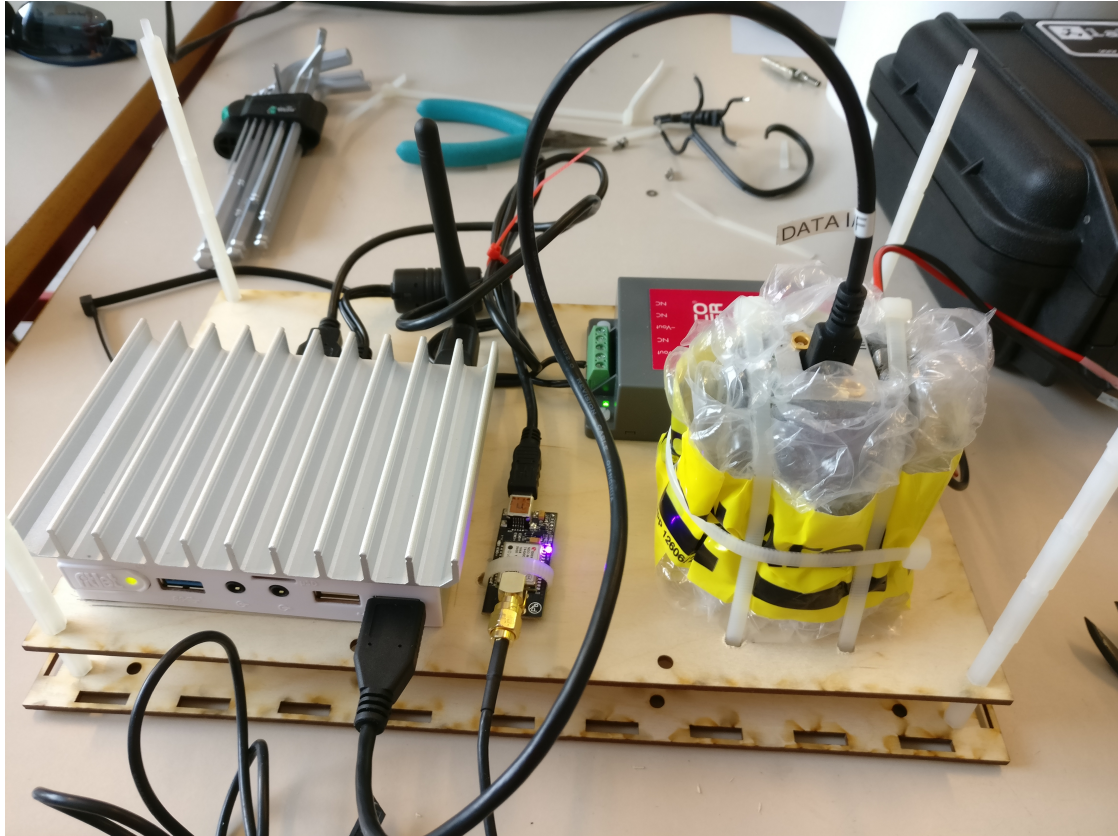


Figure 43: This was the final iteration version of the payload with the non painted aluminium finlet and the Gamma detector pointing downwards. It was the front right USB of the finlet to the gamma detector, since it was detected as it's own USB controller and therefor there were less chances of miss-clicking and collecting the data form either the USB Wi-Fi adapter or the GPS receiver board.

## B Unforeseen difficulties

During the last two months of the project a series of unforeseen circumstances set the project back about 1 months worth of work. Two months before the deadline the project was going good. Everything worked when tested with normal Windows 10 computers so it was decided that in order to reduce the weight of the payload to transfer the data gathering from the fitlet Windows platform, to the Linux based BeagleBone Black board. This solution would also help to implement a more precise GPS solution from Novatel. So time was spent on figuring out how Wireshark worked on the fitlet and writing a script which would capture the GGA input from the USB connection to whatever GPS solution would be used. This process took about 3 weeks and then it was discovered that the Kromek GR1-A+ gamma detectors data wasn't captured by Wireshark on the BeagleBone Black board. The origin of this problem was found when running the `LSUSB` command on the gamma detector. It could be seen that Linux could identify everything about the Kromek GR1-A+ down to it's company and fundamental configurations but the `InterfaceProtocol` was "unavailable". This was despite the Kromek GR1-A+ being a HID-device.

This discovery meant that we had to move back the fitlet solution. However in the meantime the fitlet had been assigned to a concurrent project which required the fitlet to be send the Spain the next week. This meant that a new fitlet had to be ordered but the SSD was on back order so it took 11 days before the new fitlet arrived. This new fitlet of course had to have windows installed from the ground and we now had to figure out a way to start the GPS and Gamma collection since the automatic setup for the BeagleBone Black did not work in Windows. Thus the `Remote desktop` over the ad hoc network was the solution which was decided upon.

Immediately a new problem arose because of the new fitlet having been setup with an automatic log on to a local admin account. Every time a `Remote Desktop` session was started the local admin account would immediately log onto the fitlet and kick the remote session user without any prompt. This was a recognised problem to a lot of people on the internet and it had no official solution with many people opting for other remote controlled solutions until Microsoft would patch it. This problem was fixed after 4 days of enabling and disabling over 25 different registries on the fitlet and finally completely replacing and deleting the `superuser` from the fitlet with one that did not log on automatically.

After this it was discovered that the Kromek GR1-A+ would suddenly stop transmitting data at random times and not resuming the data transfer until it had been unplugged and plugged into the fitlet again. This problem only occurred on the fitlet. It was then discovered that it wasn't always the same packages/errors which was captured by Wireshark. The two most frequent were that the port would suddenly send out a new `Get_Status request` which would initiate a multiple packet exchange between the Kromek GR1-A+ and the fitlet. However at either the 4th or the 5th exchange the fitlet would send and error message and tell the gamma detector to either idle or disable to port it plugged into. The second error was even harder to troubleshoot since the Wireshark data only showed that the packets from the gamma detector suddenly stopped being received with no further packet exchange from either device.

After not being able to solve these problems for 4 days the original fitlet returned from it's 1 month trip to Spain. it was given to us for testing. We immediately no longer had any of the previous problems with both the Kromek GR1-A+ working flawlessly with the fitlet and the automatic log on admin user did not log on when a `Remote Desktop` session was in progress. The old fitlet was however very slow but it was decided that since it worked then we would not re install the OS on the fitlet.

After having flown the field tests described in section 6 we realised that the GPS data was suddenly missing some UTC time stamps for some locations. U-center's solution to this was to put in an empty rows of data without a time stamp and the next row would be the next measured UTC time stamp which could be several seconds later. This required a complete re-write of the entire data pairing section of the data processing program. This re-write took 20 man hours. We suspect that the fitlet(the original which was sent to Spain) simply is too slow to be able to keep up with the data-flow. Two supporting facts to this theory is that we could actively see the packets slowly lagging behind the UTC time on the Wireshark capture, and when an empty row occurred it the rows still kept coming in at a rate of one every half second instead of having a delay matching the time skip.

## C User guide

This is a user guide to the MATLAB data processing program used to process gamma and GPS data...

Figure 20 contains an overview of the program `Mainscript_GAMMA.m`. The Google Earth Toolbox was created by Scott Lee Davis (last updated 2/2-2012).

Run `Mainscript_Gamma.m`

- Select the option **Load gamma data** to select and read a `.txt` file containing gamma data saved from Wireshark as described in section 5: Platform operation.
- Select **Load GPS data** to choose a `.CVS` file containing GPS data corresponding to the gamma data saved from u-center as described in section 5: Platform operation.  
*This step can be skipped if only gamma data needs processing.*
- If **Delete all data** is selected, all data currently uploaded and processed will be deleted from the work-space.

Press **Continue** to enter the main menu "Data processing". Each point on the main menu opens another menu. The menus accessible from the main menu is:

- Gamma data processing
- GPS data processing
- Data pairing
- Energy
- Interpolation
- Present data

The menus should be chosen in the order above.

- Select **Terminate program** to terminate the program. Data in the MATLAB work-space is not deleted when the program is terminated.

### Gamma data processing

- Select **Gamma data translation** to translate the gamma packet from hexadecimal to channel numbers.
- Select **Compute counts per second** to upload a `.CVS` file containing GPS data saved from u-center as described in section 5: Platform operation.

### GPS data processing

*This menu can be skipped if only gamma data needs processing.*

- Select **Save lon/lat coordinates** to save the longitude and latitude coordinates. The saved file should then be uploaded to KMSTrans2 and processed.
  1. Open KMSTrans2
  2. Select "Batch"
  3. Do this:

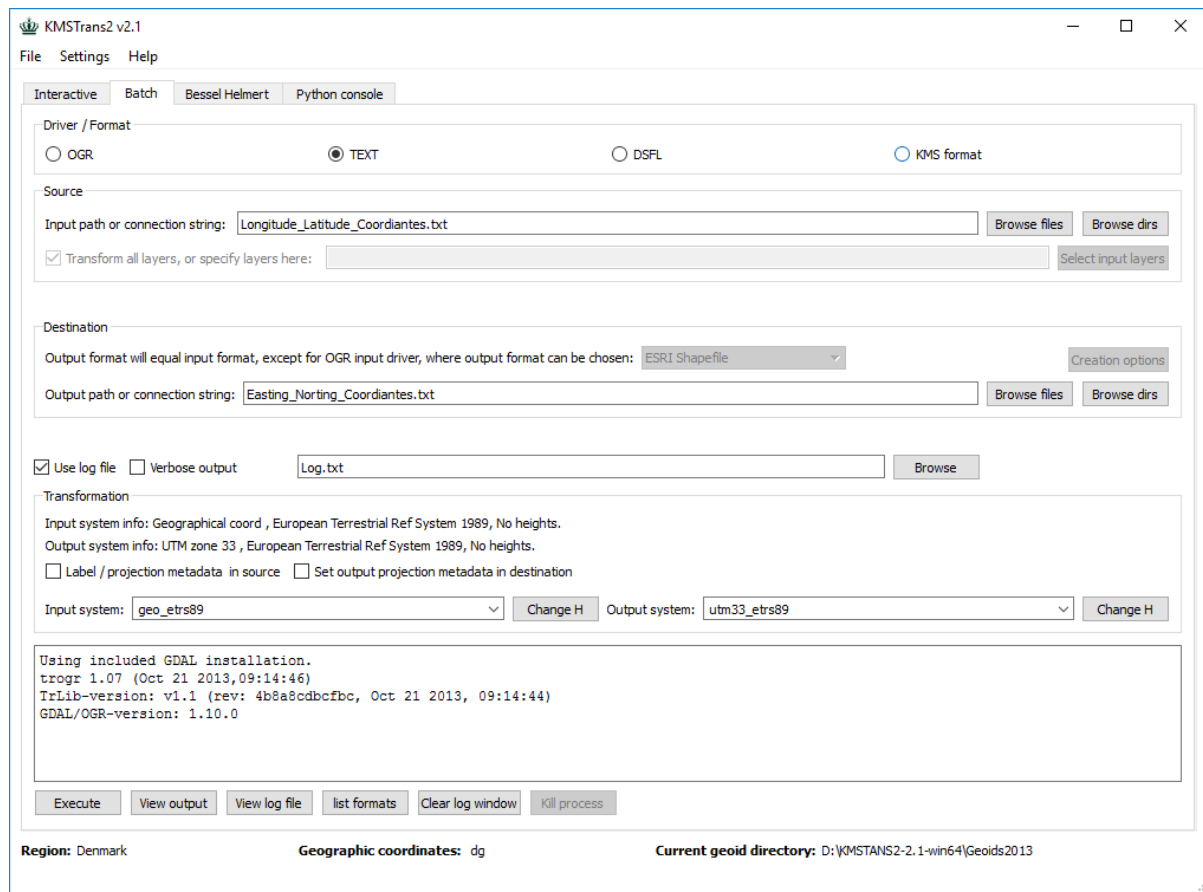


Figure 44

4. Go to "Settings" and select "TEXT/KMS format settings". The settings should match the settings in figure 45

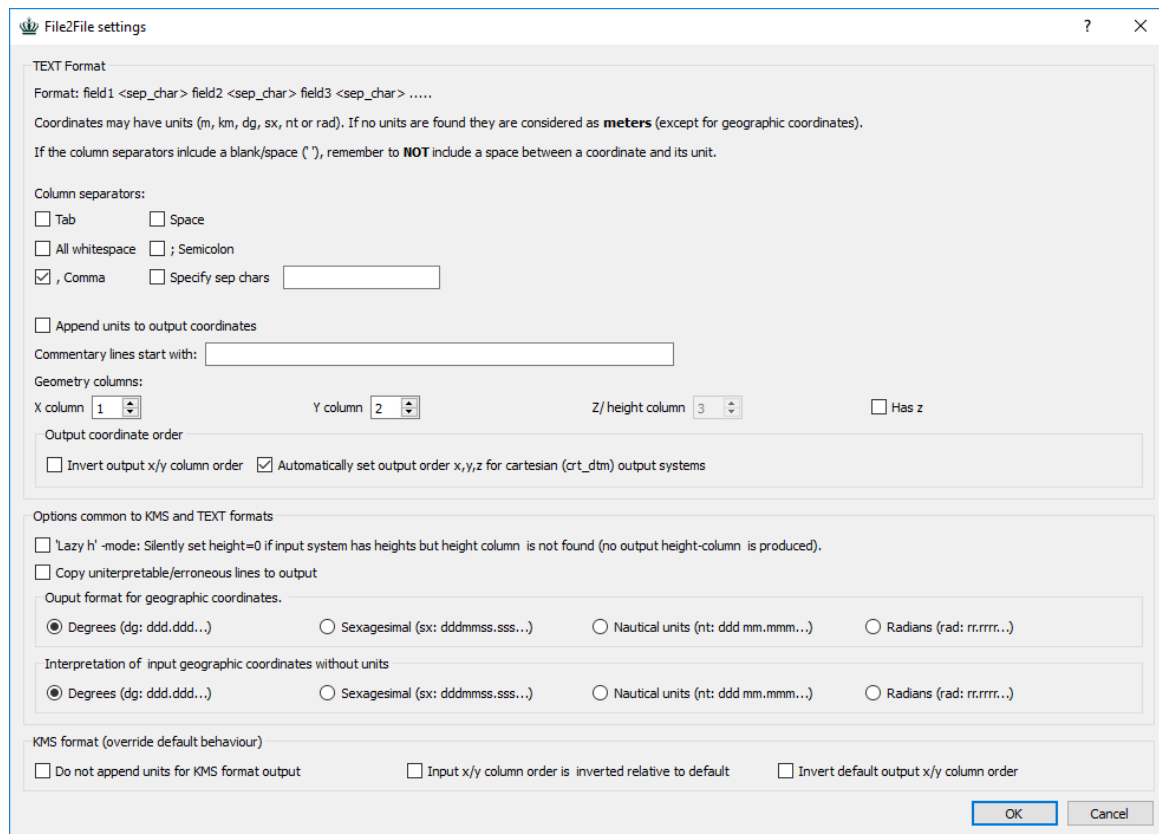


Figure 45: Caption

- Select **Load KMSTrans2 data** to upload a .txt file containing the easting and northing coordinates.

### Data pairing

*This menu can be skipped if only gamma data needs processing.*

- Select **Pair data** to pair data. This will delete all GPS data .... It will then delete GPS data where the timestamp does not match the timestamp of the gamma data. It will then delete gamma data where the timestamp does not match the timestamp of the GPS data.
- Select **Save paired data** to save a .txt file containing easting, northing and counts per second.

*This step can be skipped with no consequences for further data processing.*

### Energy

- Select **Calculate energy per second** to calculate the energy of all counts.
- Select **Show gamma spectrum** to show a gamma spectrum with energy on the x-axis and number of counts on the y-axis. If **Pair data** has been selected, only the counts of the paired data will be showed. In **Choose energy interval** has been selected, the last selected energy interval will be shown in the figure.

*This step can be skipped with no consequences for further data processing.*

- Select **Choose energy interval** to select an energy interval for further processing. A pop-up window will appear. Write the desired lower energy boundary and upper energy boundary in units of eV. Multiple energy intervals can be selected. Select **Choose energy interval** to add another energy interval. The counts per second within the chosen intervals will be added together.

### Interpolation



- Select **Choose interpolation grid size** to select the size of the grid used for the interpolation. A pop-up window will appear. Write the desired grid size in units of  $\text{m}^2$ . Choosing a very small grid size can cause "out of memory" issues when interpolating the data. The recommended minimum grid size is 0.01 m.
- Select **Interpolate counts** to interpolate all counts per second based on the easting and northing coordinates.
- Select **Interpolate energy** to interpolate all counts per second within the energy interval selected in **Choose energy interval**.

#### Present data

- Select **Plot interpolated counts** to show the interpolated counts per second. The x-axis is easting in units of meters and the y-axis is northing in units of meters.
- Select **Save counts for Google Earth** to save a .kml file containing the 2-dimensional route in longitude and latitude coordinates and the counts per second. When the file is opened in Google Earth the height of the route is the amount of counts per second.
- Select **Plot interpolated energy** to show the interpolated counts per second within the energy interval selected in **Choose energy interval**. The x-axis is easting in units of meters and the y-axis is northing in units of meters.
- Select **Save counts for Google Earth** to save a .kml file containing the 2-dimensional route in longitude and latitude coordinates and the counts per second within the energy interval selected in **Choose energy interval**. When the file is opened in Google Earth the height of the route is the amount of counts per second within the energy interval.

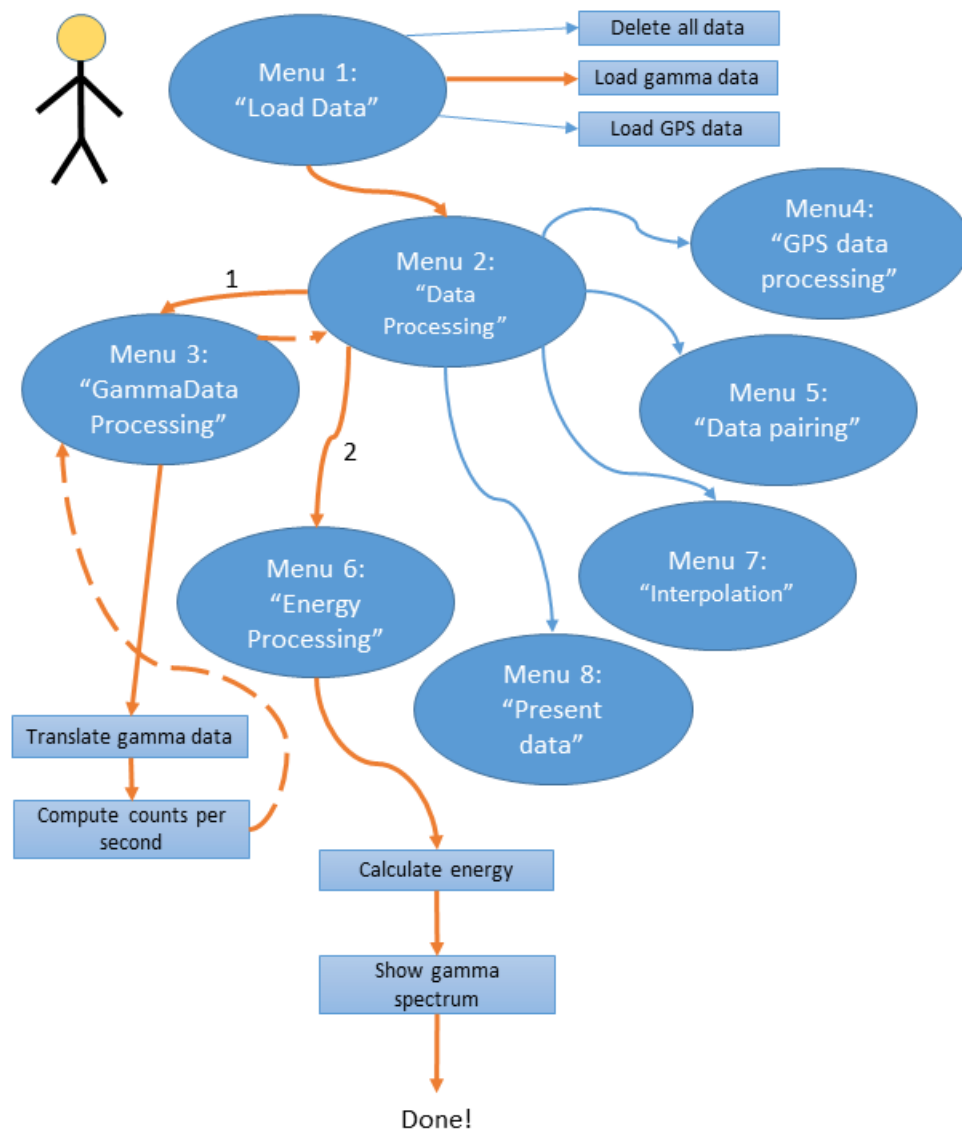


Figure 46: Visual representation of the steps that needs to be taken in order to generate a gamma spectrum based with the MATLAB program.

The circles indicate different menus in the program and the squares options in the menu, which the arrow is coming from. The orange arrows indicate the next step in the process and the striped lines indicate that this is the last action be taken in the menu.

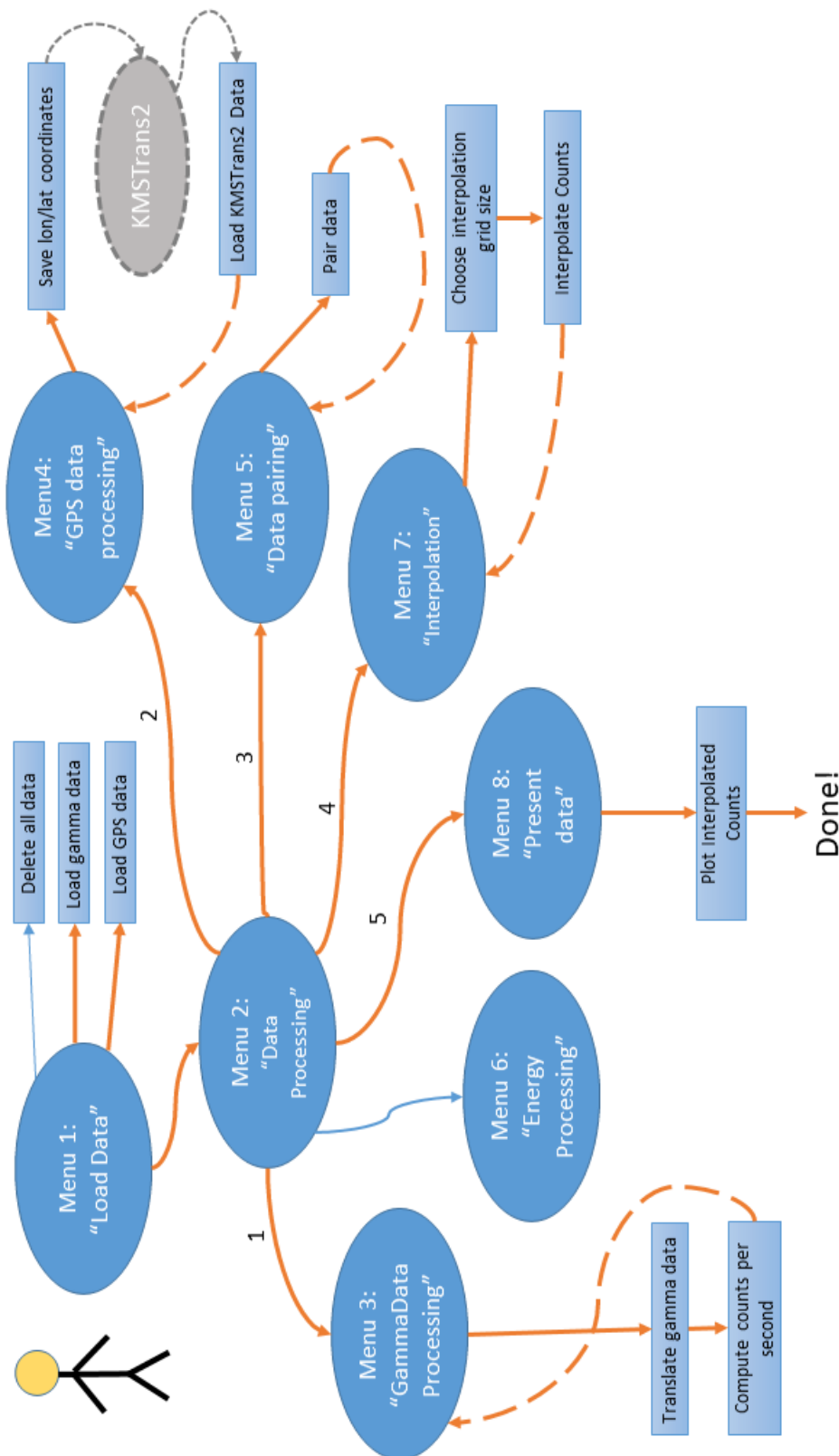


Figure 47: Visual representation of the steps that needs to be taken in order to generate a interpolated map of the counts per second with the MATLAB program.

The circles indicate different menus in the program and the squares options in the menu, which the arrow is coming from. The orange arrows indicate the next step in the process and the striped lines indicate that this is the last action be taken in the menu the grey colour indicates a step which is made outside of the MATLAB program.

## D Three week preliminary project

---

# GAMMA-RAY DATA LOGGING SYSTEM FOR UAV'S

---

Pernille Boldrup Østfeldt (s144144)  
Philip Lind (s135387)

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Equipment</b>	<b>3</b>
2.1	Kromek gamma detector . . . . .	3
2.2	Fitlet . . . . .	3
2.3	GPS . . . . .	3
<b>3</b>	<b>UAV platform</b>	<b>4</b>
3.1	DJI S900 . . . . .	4
<b>4</b>	<b>Test setup</b>	<b>5</b>
4.1	Basic test setup . . . . .	5
<b>5</b>	<b>Data &amp; data processing</b>	<b>5</b>
5.1	GPS data . . . . .	5
5.2	Gamma detector data . . . . .	5
5.3	Data processing . . . . .	6
<b>6</b>	<b>Results &amp; status</b>	<b>6</b>
6.1	Results . . . . .	6
<b>7</b>	<b>Discussion &amp; conclusion</b>	<b>10</b>
7.1	Discussion . . . . .	10
7.2	Conclusion . . . . .	11
<b>8</b>	<b>Further improvements</b>	<b>12</b>
8.1	Main points . . . . .	12
8.2	Fitlet setup . . . . .	12
8.3	Matlab . . . . .	12
8.4	Detector data acquisition . . . . .	12
8.5	Drone and Ground control station . . . . .	12
	<b>Appendices</b>	<b>13</b>
<b>A</b>	<b>Log</b>	<b>13</b>

# 1 Introduction

This report is for a preliminary 3-week project which leads up to a bachelor project that will resolve around using a UAV platform to fly around with a gamma ray detector in order to create maps showing the intensity levels in radiation over a larger area.

The purpose of this 3-week course is to test the equipment needed during the bachelor project, as well as to create a proof of concept with the use of pre-developed software and hardware. The project is therefore meant to help gain familiarity with the equipment needed for the bachelor project, and to asses which areas needs the most work. This report presents the equipment used, and the data collected from it. The data processing is described and the results are presentet, and further improvements are discussed. A log is added as an appendix, showing what has been done each day of the course.

## 2 Equipment

### 2.1 Kromek gamma detector

To measure the gamma counts from a radioactive source, a Kromek gamma detector is used. The model is *Kromek Advanced GR1*, also called GR1-A, which is a gamma spectrometer with a 1 cm<sup>3</sup> cadmium zink telluride (CZT) detector. It has a temperature range from 0°C to 40°C. [2]

To collect the data the software Kromek MultiSpect is used. MultiSpect collects measurements for the number of counts pr. second and the energy of each count.

### 2.2 Fitlet

The portable computer which will be used is the *Fitlet XA10 LAN Barebone*. It is a low-power pc which runs the full version of windows 10 32-bit. It is equipped with a 1 terra-byte SSD so which will make it more resilient to vibrations than a normal HDD. The fitlet has plenty of IO which will enable us to use it efficiently without unnecessary addons.

The specific model was chosen based on this projects requirements and the system requirements for the hyperspectral camera and it's software. The software for the hyperspectral camera is also the reason why the fitlet runs a 32-bit version of windows.

### 2.3 GPS

For the GPS we use a *u-blox ANN-MS-0-005*, which is a high performance active GPS antenna. The temperature range is -40°C to 85°C. This is not the final GPS to be used, but this will be able to collect data to create the proof of concept. The GPS used in the final setup will be a RTK GPS.

The software used is u-center, which is a GNSS evaluation and visualization tool. u-center can collect data for many parameters, where we are interested in longitude, latitude and height, and the time of each measurement.

Longitude and latitude are displayed according to the datum selected in the GNSS device, with WGS-84 as default. Time is displayed with reference to UTC. The elevation is displayed with reference to either MSL (Height above Mean Sea Level) or HAE (Height Above WGS-84-Ellipsoid). [1]



### 3 UAV platform

#### 3.1 DJI S900

The expected UAV-platform which will be used in the bachelor project is the S900 hexa-copter from DJI. The multicopter has a takeoffweight from 4.7 to 8.2 kg. It can handle mean wind of up to 8 m/s with gust up to 13 m/s before it starts to become unstable.

Despite being a multi-rotor UAV the S900 is very ideal for missions that require measuring to be done over a large area. The drone should have around 18 minutes of flight time with the 16000 mAh batteries while carrying the finalised payload. It also has a battery bay mounting system that makes swapping batteries a process which only takes 1 minute. The S900 has an official working environment temperature range from -10 to + 40 degrees Celsius.

The S900 can be flown either manually but the A2 flight controller on the UAV allows the user to upload singular way points or even a complex route to the drone which it can then fly autonomously. For the autonomous flight it is possible to either place the points down individually or to put down specific types of grids and figures for the drone to fly in.

When flying in a grid be it manually or autonomously it is imperative that the drone has the ability to hold itself in a relatively low deviation to the grid which is being flown. The A2 system on the S900 has a flying mode where it uses GPS data to maintain its position in a 3 dimensional space. This GPS although not nearly as precise in its ability to ascertain the drones position as a RTK-GPS system is still able to have the S900 stay very close to its position and it can takes a while before it will have drifted any significant distance such as 1 meter from the grid.

With regard to the altitude there is a bit higher uncertainty than with the longitudinal and latitudinal position of the UAV. However this is mostly predominant when the drone is flying in very windy weather as well as when the drone is hovering over a fixed position.

The S900's maximum flight speed is 16m/s when using the A2 flight controller and without any payload as well as a standard battery. DJI however do not recommend to fly at this kind of speed nor is it necessary for this project. We intend to in the final version base ourselves on flying in grids with a velocity of 1 m/s and only when that works test out higher velocities.

Since we are going to fly it is also imperative to have a stable system with power to spare so that it can correct itself as safely as possible. The DJI S900 has a max power consumption of 3000W, however when it weighs 6.8 kg it uses only 1000W in order to maintain a stable hover. This leaves a lot of power to ensure a stable platform given that the drone is calibrated correctly.

Another important reason to use this drone is that DTU drone service has a standardised modular payload system designed for the S900 platform. This payload box can be seen in figure 1. This modular setup has been tested with payloads of up to 2 kg maximum weight and can fly up to 5 m/s completely stable with experimental equipment mounted in the payload box.



Figure 1: S900 Drone tipped over to show payload box type, which will be used in the bachelor-project

## 4 Test setup

### 4.1 Basic test setup

For the basic test setup the GPS antenna is connected to a computer running u-center, and the gamma detector is connected to another computer running MultiSpect. The equipment was carried around in a grid pattern at Richard Petersens plads at DTU Lyngby, close to a constant velocity. A constant velocity is preferred when making a contour plot where all points count equally. At some point along the grid, a smoke detector was put close to the detector to simulate a drone flying close to a gamma source.

For the GPS antenna u-center saved coordinates for every second, and during the whole test there were a few seconds where u-center did not save any coordinates. For the gamma detector MultiSpect were measuring gamma counts every second, but was only able to save the last 120 measurements.

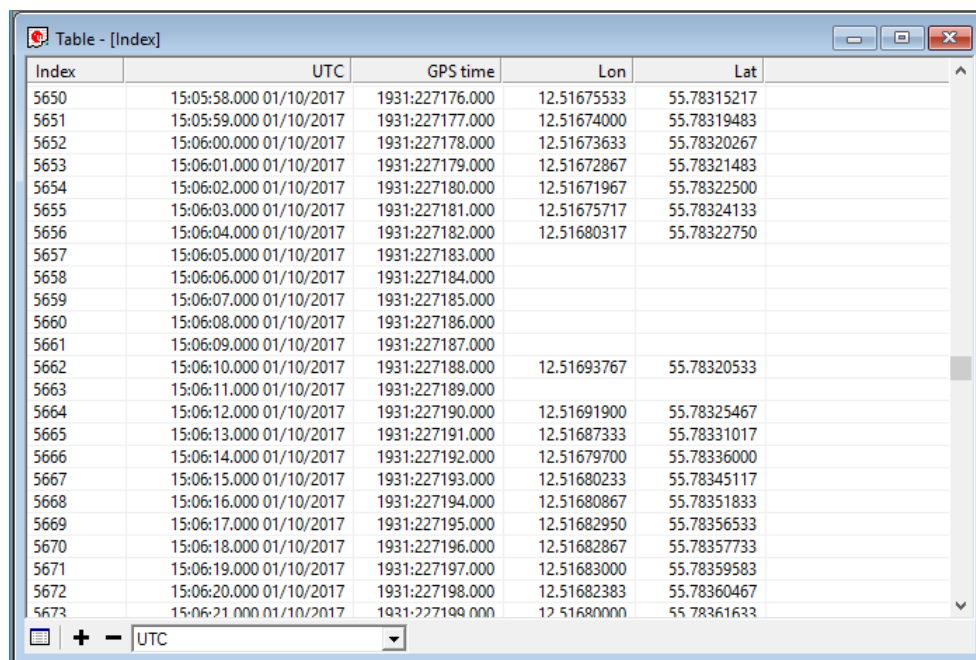
## 5 Data & data processing

### 5.1 GPS data

The GPS measures the following parameters:

- Index: Number of measurement with reference to the first measurement.
- UTC: UTC time
- GPS time: Time [s] after
- Lon: Position LTP Longitude (according to WGS-84)
- Lat: Position LTP Latitude (according to WGS-84)

These are the parameters for the basic test setup. The final setup will also include the height.



Index	UTC	GPS time	Lon	Lat
5650	15:05:58.000 01/10/2017	1931:227176.000	12.51675533	55.78315217
5651	15:05:59.000 01/10/2017	1931:227177.000	12.51674000	55.78319483
5652	15:06:00.000 01/10/2017	1931:227178.000	12.51673633	55.78320267
5653	15:06:01.000 01/10/2017	1931:227179.000	12.51672867	55.78321483
5654	15:06:02.000 01/10/2017	1931:227180.000	12.51671967	55.78322500
5655	15:06:03.000 01/10/2017	1931:227181.000	12.51675717	55.78324133
5656	15:06:04.000 01/10/2017	1931:227182.000	12.51680317	55.78322750
5657	15:06:05.000 01/10/2017	1931:227183.000		
5658	15:06:06.000 01/10/2017	1931:227184.000		
5659	15:06:07.000 01/10/2017	1931:227185.000		
5660	15:06:08.000 01/10/2017	1931:227186.000		
5661	15:06:09.000 01/10/2017	1931:227187.000		
5662	15:06:10.000 01/10/2017	1931:227188.000	12.51693767	55.78320533
5663	15:06:11.000 01/10/2017	1931:227189.000		
5664	15:06:12.000 01/10/2017	1931:227190.000	12.51691900	55.78325467
5665	15:06:13.000 01/10/2017	1931:227191.000	12.51687333	55.78331017
5666	15:06:14.000 01/10/2017	1931:227192.000	12.51679700	55.78336000
5667	15:06:15.000 01/10/2017	1931:227193.000	12.51680233	55.78345117
5668	15:06:16.000 01/10/2017	1931:227194.000	12.51680867	55.78351833
5669	15:06:17.000 01/10/2017	1931:227195.000	12.51682950	55.78356533
5670	15:06:18.000 01/10/2017	1931:227196.000	12.51682867	55.78357733
5671	15:06:19.000 01/10/2017	1931:227197.000	12.51683000	55.78359583
5672	15:06:20.000 01/10/2017	1931:227198.000	12.51682383	55.78360467
5673	15:06:21.000 01/10/2017	1931:227199.000	12.51680000	55.78361633

Figure 2: A table showing the data from the GPS antenna. Some of the coordinates have not been recorded.

### 5.2 Gamma detector data

The gamma detector measures the following parameters:

- UTC: UTC time
- TotalCounts: Counts pr. seconds

### 5.3 Data processing

The GPS data from u-center is exported as a CSV file, and then saved as a text file without the first row containing the names of the columns. The text file is imported into KMSTrans2, where the coordinates are transformed from the input system geo\_etr89 (latitude and longitude in degrees) to the output system utm33\_etr89 (easting and northing in meters). During the transformation the measurements that have no coordinates are removed, which makes UTC the best parameter for comparing GPS data with gamma data. Easting and northing are then imported into Matlab. Here the gamma counts are plotted as a function of easting and northing. In order to locate the gamma source a contour plot is created. The function `griddata` is used to make an interpolation of the measured gamma counts on a meshgrid of the part of the GPS route where there are gamma measurements. Finally the toolbox *googleearth*, created by Scott Lee Davis, is downloaded and added to the current Matlab folder. Then a plot showing longitude and latitude for the route and gamma data is plotted on top of a Google Earth image.

## 6 Results & status

### 6.1 Results

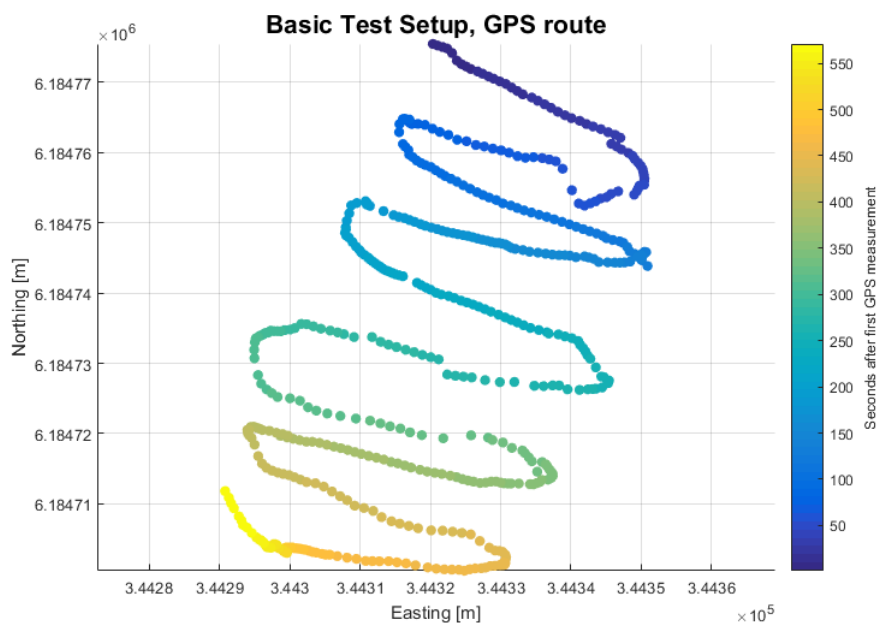


Figure 3: GPS route for basic test setup

Figure 3 shows the GPS route at Richard Petersens plads at DTU Lyngby. The route starts at the blue points and ends at the yellow points. The equipment were moving with a constant velocity most of the time, but due to cars and ice on the road, the velocity increased or decreased a little, which can be seen between 325 and 350 seconds after the beginning of the route.

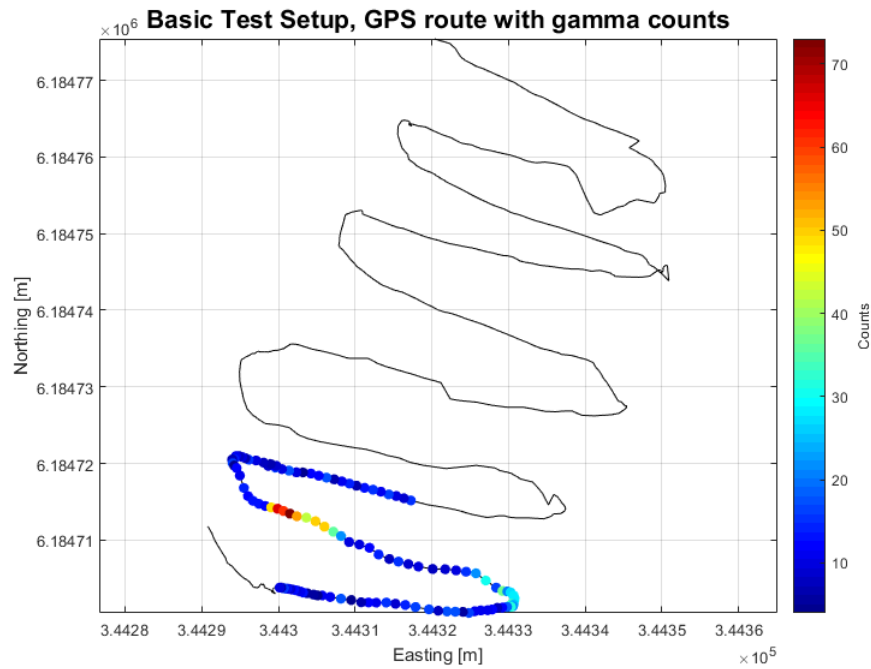


Figure 4: Gamma counts along the GPS route

Figure 4 shows the gamma counts measurements in the last 120 seconds of the route. It shows an area with 40-70 counts pr. second, and another area with 20-35 counts pr. second.

In order to locate the gamma source a contour plot is created. The function `griddata` is used to make an interpolation of the measured gamma counts on a meshgrid of the part of the GPS route where there are gamma measurements.

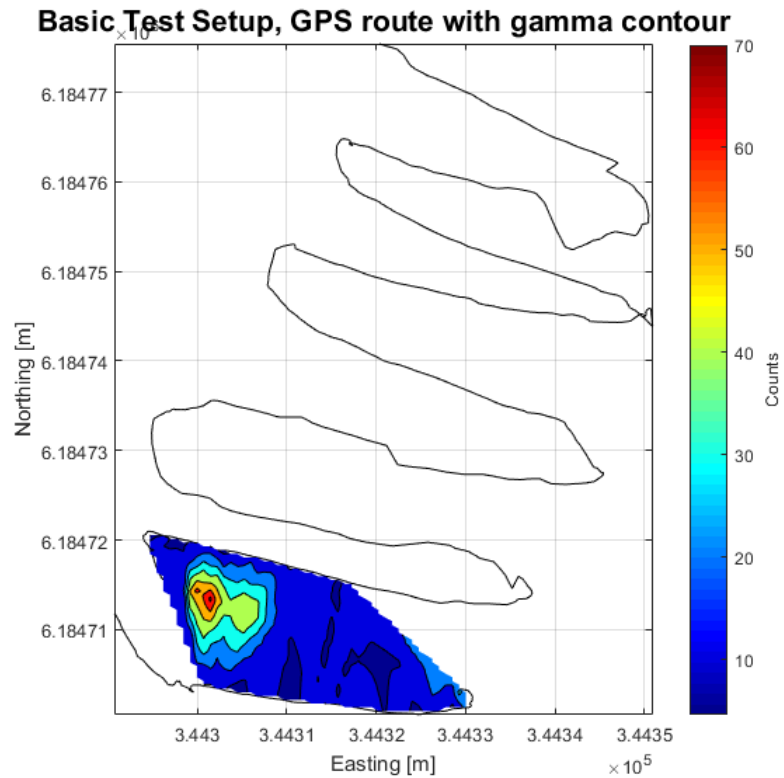


Figure 5: Contour plot over the counts in the area with gamma measurements

On figure 5 the contour plot is shown on top of the route. It shows the area with the most gamma counts based on the measured gamma counts on the route. During the final setup in the bachelor project, the contour plot is expected to be similar to the contour plot in figure 5. Since this setup was able to locate where on the route the gamma source was close to the detector, it will be used with the fitlet in the bachelor project.

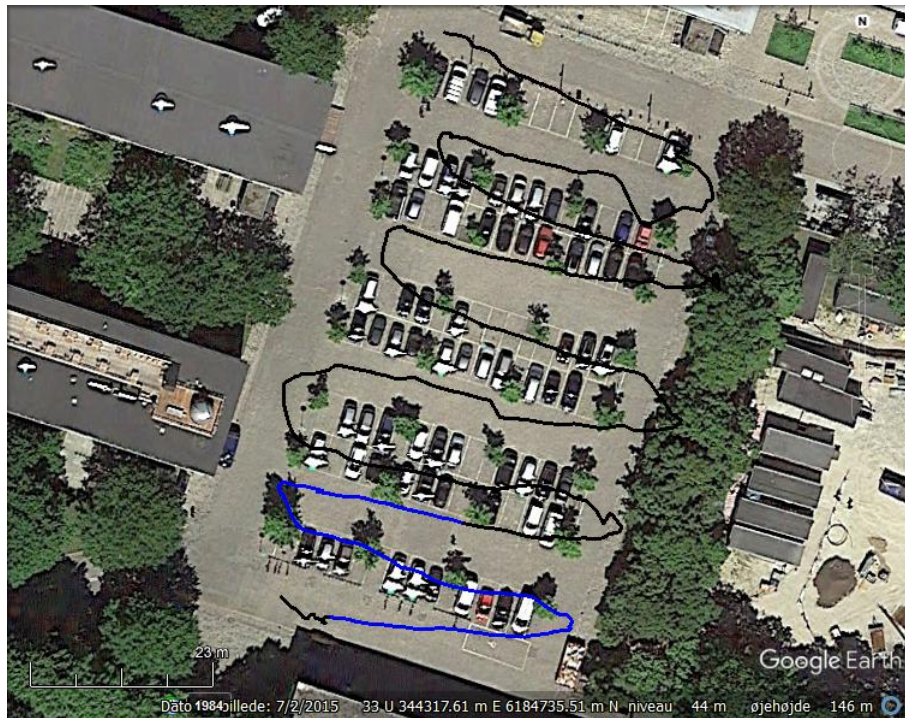


Figure 6: Basic test setup route on Google Earth

Figure 6 shows the route plotted on Google Earth. The black route is the full route, and the blue part is where the gamma measurements are. When comparing the measured route to the actual route, there are deviations up to ca. 8 meters in some parts, but most of the time, the deviation is less than 3 meters. Most of the deviations appears to be in a southwards direction, so there might be a systematic deviation, but more GPS data and routes are needed to verify this.



## 7 Discussion & conclusion

### 7.1 Discussion

The basic test setup was able to both record the coordinates of the mobile experiment, and the gamma counts from the background radiation and the gamma source. Since both u-center and MultiSpect were saving data every second with reference to UTC, it was easy to pair gamma counts with the coordinates in Matlab. This setup will therefore be used in the bachelor project.

When plotting the data, all the gamma counts from coordinate 1 to coordinate 2 will be plotted at coordinate 2. Therefore the measured locations of the gamma source will slightly off. It will be necessary to display the gamma counts midways between the two coordinates for the best estimate of the location of the gamma source.

As can be seen in figure 5 the peak area is not entirely uniform. The cause of this might be that we during our test did not move the smoke alarm uniformly to and from the sensor as we wanted, but it can also be due to having an irregular grid and therefor non-uniform distribution of the data with respect to position. If it is the latter it will be somewhat alleviated by flying in a 1m by 1m block-grid with a constant speed.

In the bachelor project we will also use sources much stronger than the smoke detector with which the tests was conducted with in this pre-project. This combined with a 1m by 1m grid would make for more clearly defined areas of radiation with a more easily defined positions of the Radioactive sources.

Concerning the actual flying in a grid formation it is important to be able to create the set of way-points the drone needs in order to fly these autonomously since it can be quite difficult to manually fly in a perfect 1 by 1 meter grid. The software that have been used so far is DJI's *Ipad Ground Station* app. While reliable it is also extremely limited in it's capabilities to program detailed grids for the drone and has only 4 different grids which can be used. Each of these are predefined with respect to the amount of way-points and increasing the area to fly in would therefor also increase the spacing between the way-points ruining the 1 by 1 meter grid. Instead using another ground control station with better support for such grids would be the ideal solution.

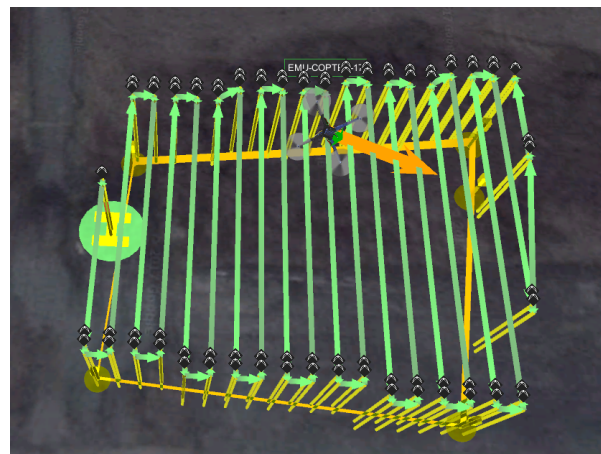


Figure 7: Simulated drone flying in a user defined grid in UgCS. It can be seen how the drone is adjusting its heading while flying along the grid.

*UgCS* is the most likely candidate at the moment since it supports the A2 system on the S900. It also has an "Area scan" function in which one manually places the borders down for the area which one wishes to fly in and the software then generates the flying grid for you based on information such as the side distance and so on.

## 7.2 Conclusion

With regards of designing a mobile experimental setup which can be flown with a UAV we have successfully tested the basic points as well as the functionality of the different hardware. We have also determined the next steps which needs to be taken in order to proceed with the bachelor project.

We have a functioning Matlab script which can plot the data that we want into different types of plots and graphs depending on what we want to show.

The GR1-A operates and functions as it should with the currently available software being our only problem as the data is gathered in the *Monitor* window/menu can only be exported as 120 values with specific intervals. This makes it so we only have either averaged data in pieces of minimum 15 seconds or only 2 minutes of data.



## 8 Further improvements

### 8.1 Main points

There are 4 main points which need to be improved upon before the application can be considered complete. These are:

- Fitlet setup
- Matlab
- Detector data acquisition
- GPS data implementation

### 8.2 Fitlet setup

The fitlet in itself works as it should but for the final application it is necessary for the fitlet to be used without a screen and without a lot of input. The ideal solution to this problem is to write an executable script which automatically runs on the fitlet on boot. The script would then launch both the GPS sampling software and the data solution software for the GR1-A. This solution would save a lot of weight since there is no need to add any sort of input device in order to make the fitlet run the programs. It would also help with making the application more user friendly and effective since powering up the board would be all that is needed for the data acquisition.

### 8.3 Matlab

To be able to create some of the plots, some of the data were selected and imported manually. During the bachelor the scripts needs to be optimized, and handle data based on UTC, since both the GPS antenna and the gamma detector records UTC. The gamma data also need to be plotted in Google Earth, to be able to show where the gamma source is. At last the Matlab script will be transformed into a .exe file.

### 8.4 Detector data acquisition

The gamma detector logs data every second with the Multipect software, but MultiSpect only saves the last 120 measurements. Kromek has a program called Genie 2000 Analysis, which is able to collect data in a simpler way than Multispect but is also incapable of running any form of analysis by itself.

The GR1 family of sensors is also supported by a software Kromek development kit (SDK). These software tools can be purchased from CANBERRA. The software would allow us to develop a customised software application for the GR1-A that would be capable of gathering the data which we want.

An e-mail has been sent to CANBERRA to get the price of the software.

### 8.5 Drone and Ground control station

While the UgCS software has the functions and abilities to navigate the drone in a grid of our desired type there still needs to be done a lot of tests on the consistency of the drones grid flying.

This will include test on how the turns affect the drones ability to maintain it's position the grid. This would determine how big an area the grid will have to encompass in order account for the distance the drone will need in order to get back on track to measure the desired points. There will also have to be done several test with regards to stability. Finally an important test would be to fly several times in the same 1 by 1 meter grid with the RTK-GPS system and then compare how consistently the drone maintains the same deviations of the wanted path.

# Appendices

## A Log

### 03/01-17

- Created time schedule.
- Created questions for the first meeting regarding gamma detector, fitlet and report.

### 04/01-17

- Meeting with Jakob.

### 05/01-17

- Found out which fitlet we needed, and ordered it.
- Downloaded the GPS software u-center.
- First collection of GPS data points.
- Saved GPS data points as CSV file and imported to Matlab.
- Read about interpolation in Matlab.
- Created a vector with random numbers and interpolated the random data over the GPS data.

### 06/01-17

- Not much to do until we get the gamma detector.

### 09/01-17

- Meeting with Irfan.
- Looking at which parameters we need from the GPS software.
- Measured data with the gamma detector.

### 10/01-17

- First test with gamma detector at Richard Petersens plads.
- Realized that the gamma detector software MultiSpect only saves the last 120 measurements.
- Data processing in Matlab with the measured data.

### 11/01-17

- Unable to do outside tests because of the weather.
- Working with MultiSpect, trying to figure out how we can save more measurements.

### 12/01-17

- Second test with gamma detector at Richard Petersens plads, this time using a 30 minute window. This saved the averaged data for every 15 second, so we did not use the data from this test.

### 13/01-17

- Working with and reading about KMSTrans2.
- Written the report.

### 16/01-17

- Introduction to KMSTrans2.
- Working with KMSTrans2.
- Trying to make plots in Matlab work with Google Earth.

**17/01-17**

- Working in Matlab.
- Creating simple plots in Google Earth (single color scatter plots).
- Written the report.

**18/01-17**

- Working in Matlab.
- Written the report.

**19/01-17**

- Written the report.

**20/01-17**

- Meeting with Irfan.
- Contacted Canberra and Kromek, asking for prices for software for the Kromek gamma detector.
- Finished the report.

## References

- [1] [https://www.u-blox.com/sites/default/files/u-center\\_UserGuide\\_\(UBX-13005250\).pdf](https://www.u-blox.com/sites/default/files/u-center_UserGuide_(UBX-13005250).pdf)
- [2] [https://www.u-blox.com/sites/default/files/ANN\\_ProductSummary\\_](https://www.u-blox.com/sites/default/files/ANN_ProductSummary_)

## E Datasheet for the Cobalt-60 source

14-5 85

ok 1/3 88

Dato:

11/5-78

Isotop:  $^{60}\text{Co}$ 

Kemisk form: \_\_\_\_\_

Anskaffet d. 5/5-78

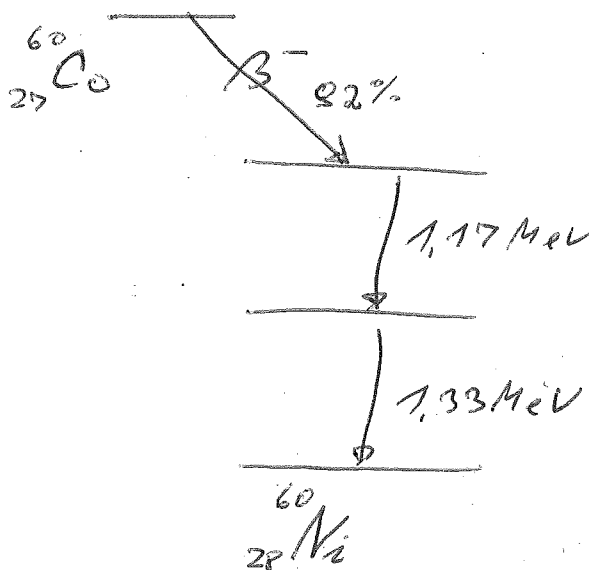
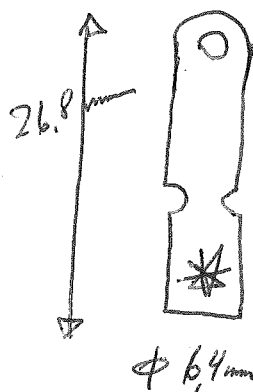
fra: Amersham

Styrke ved anskaffelse: \_\_\_\_\_

Styrke d. 2/5-78: 20,1 mCi styrke d. \_\_\_\_\_: \_\_\_\_\_ =

Udseende:

Henfaldsskema:

Amersham  
kapseltype X44

Anvendes til: Bestrølingsøvelse

Befinder sig normalt i: Brandskab, holder lokale i blybeholder

Strålingsniveau i: \_\_\_\_\_ cm afstand: \_\_\_\_\_, d. \_\_\_\_\_

Målt/beregnet: \_\_\_\_\_

Kommentarer: \_\_\_\_\_

X

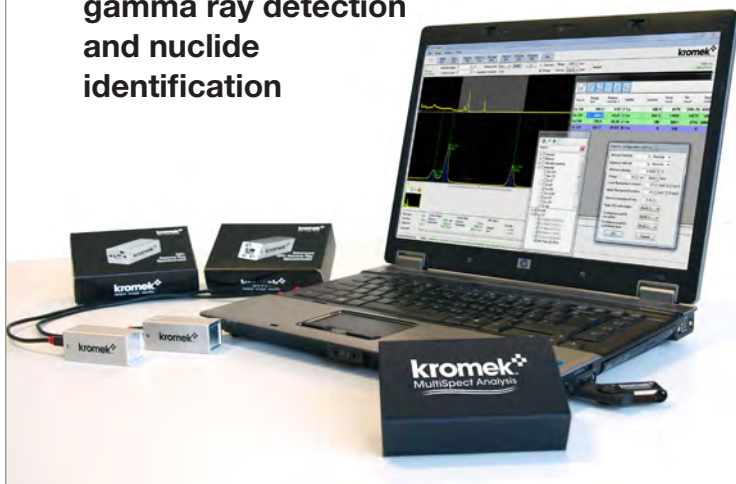
## **F Datasheet for the Kromek GR1-A+**

# GR1-A® and GR1®

Now with flexible  
functionality to meet  
specialist requirements



**Compact high-resolution  
gamma ray detection  
and nuclide  
identification**



## Main features:

### Common to GR1 - GR1-A

- High spectral resolution
- High efficiency
- Compact
- Simple to use
- CZT solid state detector
- USB Powered
- Low power consumption

### GR1-A specific

- Operational flexibility
- MCX connectors
- Analogue energy output
- Timing output
- MCA gate input

**kromek**®  
detect image identify

## GR1-A® and GR1® the World's Smallest and Highest Resolution Room Temperature Gamma-Ray Spectrometers

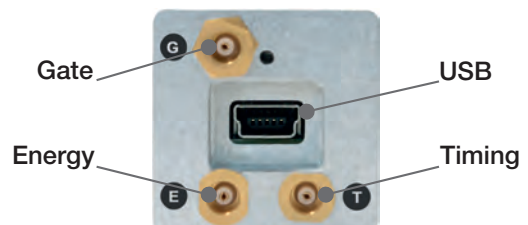
The **GR1®** and **GR1-A®** high-performance gamma-ray spectrometers utilize a 1 cm<sup>3</sup> CZT solid state detector and offer world-leading specification in a compact form.

Both are completely self-contained, with a built-in preamplifier, shaping amplifier, baseline restorer, pulse height digitizer, and HV supply. The digitized pulse heights of detected gamma-ray signals are sent to a PC via the mini-USB which also powers the unit, so no external power supply is required.

The **GR1®** comes with K-Spect®, Kromek's entry-level Windows®-based (7, 8, and 10) software, built for detailed sample and spectral analysis. K-Spect, which is available to download, free of charge, from the Kromek website, provides the spectrum acquisition, display, analysis, and storage functions.

The **GR1-A®** is the advanced model of the GR1®, containing auxiliary input/outputs that greatly increase operational flexibility. These allow the GR1-A® to be tailored for specific applications in nuclear research and academia. Three MCX connectors provide energy and timing outputs and gate inputs.

The GR1-A® is supplied with MultiSpect Analysis® software, a unique application that allows connection of multiple GR1-As to a PC, and has the ability to display multiple spectra; both live and saved, from previous measurements. It provides the spectrum acquisition, display and storage functions, and the export of data for further analysis as well as match spectra to a pre-loaded library of over 400 radionuclides.



**Energy Output:** shaped and buffered detector output pulses with amplitude proportional to energy suitable as input to an external multichannel analyser (MCA)

**Timing Output:** A logic pulse triggered by each detected event and coincident with each output pulse

**Gate Input:** used to suppress pulse height output via the USB interface to K-Spect for anticoincidence. Energy and timing outputs are unaffected.

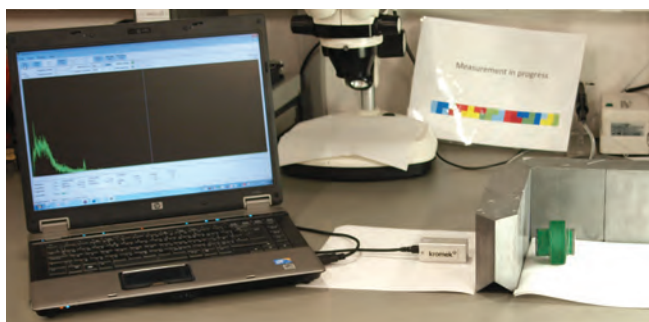




Health Physics



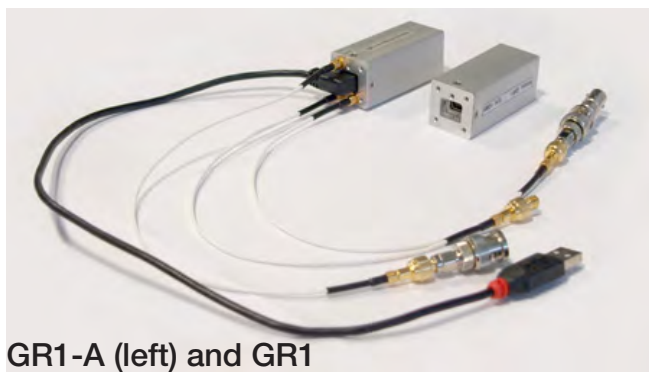
InSitu Monitoring



Nuclear Spectroscopy



Nuclear Industry



GR1-A (left) and GR1

# Specifications

Common to GR1 / GR1-A:	
Detector	10x10x10mm CZT detector
Energy range	30 keV to 3.0 MeV
Energy resolution	<2.5% FWHM @ 662 keV
Electronic noise	<10 keV
Maximum throughput (USB)	30,000 cps
Number of channels	4096 (12 bit)
Differential non-linearity	< ± 1%
Power consumption	250 mW
Dimensions	25mm x 25mm x 63mm
Weight	60 gram
Temperature range	0 to +40°C

Specific to GR1-A:		
Energy Output	Rise time	3 µs
	Decay time	10 µs
	Output impedance	< 150 Ω
	Maximum throughput (analogue)	50,000 cps
Timing Output	Shape	TTL compatible rectangular pulse
	Amplitude	5.0 V
	Duration	8 µs
	Output impedance	< 150 Ω
	Timing Resolution	< 100 ns
Gate Input	Threshold	3 V
	Maximum input voltage	5 V
	Input impedance	10 kΩ
	Timing	Input must be above threshold from at least 0.5µs before the energy signal maximum to at least 2µs after it.

*Note: In the absence of any connection the gate input is held low and all pulses are processed normally.*

Tested by the National Physical Laboratory in accordance with the conditions in;

**ANSI N42.31 (2003)** “Measurement procedures for resolution and efficiency of wide-bandgap semiconductor detectors of ionizing radiation”

**ANSI N42.34 (2006) Section 7.1** “Performance criteria for hand-held instruments for the detection and identification of radionuclides”

**BS EN 62327:2011 Section 9.6** “Hand-held instruments for the detection and identification of radionuclides and for the indication of ambient dose equivalent rate from photon radiation”

**NPL Good Practice Guide No. 14** “The examination, testing and calibration of portable radiation protection instruments”

**Environmental:** meets or exceeds: EN55011:1998 +A1:1999 +A2:2002 (Radiated Emissions), EN61000-4-2:1995 +A1:1998 + A2:2001 (Immunity to ESD), EN61000-4-3:2002 (Radiated Immunity)

# Gamma Spectrometry Software

**MultiSpect Analysis®** software is written specifically for Kromek's GR1® and GR1-A® for use with Windows® (7, 8, 10) -based PCs or tablets.

The GR1® comes with **K-Spect®**, Kromek's entry-level software which is available to download, free of charge from: [www.kromek.com](http://www.kromek.com) (click on the downloads tab, and follow the on-screen instructions).

**K-Spect®** receives the data and performs the spectrum acquisition, display, analysis and storage functions. Signals from the CZT solid-state detector are processed and digitized, and the pulse height data is transferred to the computer via the mini-USB.

K-Spect® users can upgrade to MultiSpect Analysis® or MultiSpect Analysis Premium®. To upgrade, simply contact our commercial team via the Kromek website.

*Kromek's feature-rich MultiSpect Analysis Software*

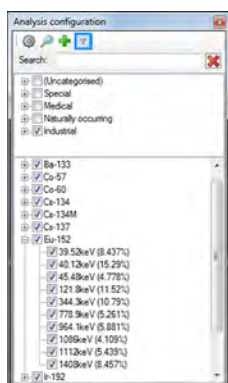


Source	Energy (keV)	Relative intensity (%)	Half-life	Composed (keV)	Gross count	Net count	Upper confidence	Lower confidence	P(100%) (keV)	P(95%) (keV)	P(50%) (keV)
Co-134	569.3	6.87	2.1 yr	569.7	1692	580.5	745.7723	415.2277	11.09	21.81	0.5083
Co-134	604.7	43.61	2.1 yr	606.6	3261	2121	Peak detected above critical limit (95.2%)	25.88	0.4815		
Co-134	795.8	38.20	2.1 yr	798.2	1741	1354	1475.1957	1333.8443	12.39	28.32	0.4374
Co-137	661.7	91.63	30.1 yr	0	125	0	0	0	0	0	0

*Filter by critical limit, relative intensity of the emission lines, energy windows and half life.*

*Colour coded results showing which lines are above the critical limit of the information.*

**MultiSpect Analysis®** is Kromek's feature-rich software and is included on a flash disk with each GR1-A®. In addition to all the K-Spect® functionality, MultiSpect Analysis® allows users to acquire and display live spectra from multiple devices simultaneously alongside saved spectra from previous measurements. Data is transferred to the computer via USB. It also enables grouping and summing of individual spectra plus the ability to match spectra to an on-board library of over 400 nuclides.



*User configurable library of radionuclide emissions*

**MultiSpect Analysis Premium®** provides the added benefit of efficiency calibration to enable quantitative analysis ideally suited to data collected in any fixed geometry such as marinelli beakers.

## Manual Efficiency Calibration

Where the detector and radioactive sources are used in a fixed geometry, an efficiency calibration of the system can be used, together with the measured count rates in spectrum peaks, to calculate source activity. Tools are provided allowing the user to determine the efficiency of their system using a calibration source of known activity.

Feature	K-Spect	MultiSpect Analysis	MultiSpect Premium
Spectral acquisition from single GR1, GR1-A, SIGMA or TN15	✓	✓	✓
Spectral acquisition from multiple detectors of the same type simultaneously	✗	✓	✓
Energy calibration facility	✓	✓	✓
Efficiency calibration	✗	✗	✓
Display multiple detector information and more than 5 saved Spectra	✗	✓	✓
Display calibrated spectra at the same energy scales to allow comparison	✗	✓	✓
Compatible with K102 Multichannel Analyser	✓	✓	✓
Thumbnail indication of loaded spectra	✓	✓	✓
Ability to save spectra in SPE, CSV or N42.42 formats	✓	✓	✓
Ability to export data	✓	✓	✓
Ability to save detector calibration information	✓	✓	✓
Association of calibration data with particular detectors by serial number	✓	✓	✓
Aggregation of multiple spectra into one spectrum	✗	✓	✓
Built in library of 416 isotopes	✗	✓	✓
Industry standard categorisation of isotopes	✗	✓	✓
Import from RayMon10	✗	✓	✓
Multiple regions of interest with Spectra	✓	✓	✓
Automated peak analysis of Spectra	✗	✓	✓
Quantitative analysis	✗	✗	✓

*Drivers are available for both Windows® (7, 8, & 10) and Linux® operating systems*

GR1 Family variants	Detector size mm	Resolution at 662 keV	USB	Gate input	Timing output	Energy output	Software included
GR1	10x10x10	<2.5%	✓	-	-	-	
GR1+	10x10x10	<2.0%	✓	-	-	-	
GR1-A	10x10x10	<2.5%	✓	✓	✓	✓	MultiSpect Analysis
GR1-A+	10x10x10	<2.0%	✓	✓	✓	✓	MultiSpect Analysis
GR05	5x5x5	<2.5%	✓	-	-	-	
GR05+	5x5x5	<2.0%	✓	-	-	-	

*K-Spect is compatible with all products and is available to download free of charge from the Kromek website*

*The GR1 is available with Linux® driver and can be interfaced with other software packages; driver information can be provided as part of the project deliverables for selected partners.*

*3x MCX and 3x BNC adaptors included with GR1-A products*

## Example applications of the GR1 / GR1-A

- **Radionuclide identification:** The GR1 and MultiSpect Analysis provide a powerful platform to perform radionuclide identification in a variety of industrial and health physics applications such as environmental monitoring and sample analysis.
- **Classroom teaching:** The GR1 module is an ideal and portable tool that can be utilised for educational purposes in teaching concepts of radiation as well as for training in the use of radiation sensors.



## Example research applications of the GR1-A

- **Particle-gamma coincidence measurements:** The high resolution of the GR1-A allows it to be paired with charged-particle or neutron detectors in order to observe coincidence particle decays with gamma-ray emission.
- **Gamma-gamma coincidence measurements:** Multiple GR1-A's can be used in unison in order to identify multiple gamma rays that are emitted in coincidence; because of this, the GR1-A can be used as a tool in studies of nuclear structure.
- **Pulse-shape analysis:** The analogue energy output can be used for studies of charge collection pulse-shapes, which can yield information on interaction points, scattering and charge mobility within the detector crystal. The timing output can also be employed as an input for time-to-analogue converters.



© 2016 Kromek Group. All rights reserved.

### Kromek Group plc

**Kromek Ltd** NETPark Thomas Wright Way Sedgfield County Durham TS21 3FD UK T: +44 (0) 1740 626060

**Kromek USA** 373 Saxonburg Blvd Saxonburg PA 16056 USA T: +1(0) 724 352 5288

E: sales@kromek.com W: www.kromek.com

## **G Kromek USB Spectrometer Interface – Technical Manual**

# Kromek USB Spectrometer Interface – Technical Manual

---

*Current Version 1.1*

Revision	Date	Author	Notes
1.1	7 January 2013	S Foreman	Updated Report 1 / LLD information
1.0	3 October 2012	S Foreman	

# Contents

---

Introduction .....	3
Overview .....	3
GR1 USB Identifiers .....	3
Report 1 .....	3
Report 4 .....	3
References / External Links.....	4
Recommended Reading .....	4



## Introduction

This document describes how the USB interface to the GR1 works. It assumes familiarity with the USB HID specification.<sup>1</sup>

The document is intended to be used by someone who needs to communicate with the device directly rather than through a supplied API. (For example if an API is not available for the platform you wish to develop on).

Finally the document describes how to process the data returned from the spectrometers.

## Overview

The GR1 is a USB HID device<sup>2</sup>. It uses USB HID report 4 to send data from the GR1 to the USB Host at a maximum rate of one report every millisecond. Report 4 is always 63 bytes long and can contain up to 31 counts.

## GR1 USB Identifiers

The GR1 has the following USB descriptors:

Vendor Id = 0x04D8

Product Id = 0x0000

The GR1 will identify itself as a 'PGS1', this is our internal name for the GR1.

## Report 1

Report 1 is used to set the LLD for the GR1. This report is always 3 bytes long.

The first byte is the report number 0x01.

The next two bytes form a 16 bit value (word) which is the value of the LLD. Note that the LLD value only requires 12 bits which are the most significant 12 bits of the word. For example to set an LLD of 0x20 the two bytes should be (0x02, 0x00).

## Report 4

Report 4 is used to transfer spectrum data from the GR1 to the USB Host device.

The GR1 sends back channel numbers that events have occurred in. So if an event occurs in channel 45 the value 45 is sent back. Up to 31 channels can be transferred in one report. The report frequency can vary depending on how many events are occurring, but a maximum of 1 report every millisecond can be sent.

The GR1 has 4096 channels, which are transferred back as 0 – 4095 (0x000 – 0xffff), ie 12 bits. Therefore every event requires 2 bytes (a word) to represent the channel number. One byte is required at the beginning of the report to identify the report number. Following that are 31 words representing up to 31 events.

31 channels (2 bytes per channel) + 1 bytes for the report identifier = 63 bytes per report.

This report is always 63 bytes regardless of how many events have occurred. So if only 1 event occurs a report will be sent but only 1 channel will be valid.

To determine if a channel is valid check the least significant bit of the word to see if it is 1. If the least significant bit of the word is 0 then that channel is not valid and neither are any following channels in that report.

The channel number is the first (most significant) 12 bits of the word.

Example:

The following example shows how two events are transferred in a report. Note how the least significant 4 bits of the word may contain data, even though only the least significant bit is used.

It is important to stop processing further events once an invalid event is received, even though there may be what appears to be valid data following in the report. The GR1 does not always fill the report with 0x00 once it has transferred all channels required. This is demonstrated in the example.

Byte	Value	Notes
1	0x04	Report number 4.
2	0xef	Most significant 2 bytes of the first event.
3	0x15	Least significant bit is 1, so channel is valid. Channel is 0xef1 (3825).
4	0x00	Most significant 2 bytes of the second event.
5	0x13	Least significant bit is 1, so channel is valid. Channel is 0x001 (1).
6	0x34	Most significant 2 bytes of the third event.
7	0xf2	Least significant bit is 0, so channel is not valid and is ignored. Stop processing this report.
8	0xa4	Ignored as the report is no longer being processed.
9	0x31	Ignored as the report is no longer being processed.
...		

Any events occurring outside the range of the GR1 will be placed into channel 4095.

## References / External Links

- 1 – USB HID Specification - [http://www.usb.org/developers/devclass\\_docs/HID1\\_11.pdf](http://www.usb.org/developers/devclass_docs/HID1_11.pdf)
- 2 – USB HID - [http://en.wikipedia.org/wiki/USB\\_HID](http://en.wikipedia.org/wiki/USB_HID)

## Recommended Reading

USB Complete – Jan Axelson, Third Edition, Chapters 11, 12, 13



## H MATLAB program code

This Appendix was handed in as a secondary .zip file.

## References

- [1] Schultz, Tobias Hjortkjær (2016), "*Tøende amerikansk atomaffald truer Grønland*", [www.illvid.dk](http://www.illvid.dk) Internet: <http://illvid.dk/naturen/klimaforandringer/toeende-amerikansk-atomaffald-truer-groenland>
- [2] Panke, Ben (2016) "*A Radioactive Cold War Military Base Will Soon Emerge From Greenland's Melting Ice*" [www.smithsonian.com](http://www.smithsonian.com) Internet: <http://www.smithsonianmag.com/science-nature/radioactive-cold-war-military-base-will-soon-emerge-greenlands-melting-ice-180960036/>
- [3] Power Reactor Information System (PRIS) as of 15/6-2017, Internet :<https://www.iaea.org/pris/>
- [4] Steinberg, Ellis P. and Rasmussen, John O. (2017), "*Radioactivity*", [www.Britannica.com](http://www.Britannica.com) Internet: <https://global.britannica.com/science/radioactivity#toc48275>
- [5] Matis, Howard S. (2000), "*Radioactivity*", <http://www2.lbl.gov> Internet: <http://www2.lbl.gov/abc/wallchart/chapters/03/0.html>
- [6] Nave, Carl R. (2017), "*Radioactivity*", [www.hyperphysics.phy-astr.gsu.edu](http://www.hyperphysics.phy-astr.gsu.edu) Internet: <http://hyperphysics.phy-astr.gsu.edu/hbase/Nuclear/radact.html#c5>
- [7] Nave, Carl R. (2017), "*Beta Radioactivity*", [www.hyperphysics.phy-astr.gsu.edu](http://www.hyperphysics.phy-astr.gsu.edu) <http://hyperphysics.phy-astr.gsu.edu/hbase/Nuclear/beta.htmlc2>
- [8] Matis, Howard S. (2000), "*Gamma decay*", <http://www2.lbl.gov> Internet: <http://www2.lbl.gov/abc/wallchart/chapters/03/3.html>
- [9] NDT Resource Center, "*Radioactive half-life*", [www.nde-ed.org](http://www.nde-ed.org) Internet: <https://www.nde-ed.org/EducationResources/HighSchool/Radiography/half-life2.htm>
- [10] Wikipedia (2017), "*Sievert*", [www.wikipedia.org](http://www.wikipedia.org) Internet: <https://en.wikipedia.org/wiki/Sievert>
- [11] Wagenaar, Douglas J. (1995), "*Radioactive decay*", [www.med.harvard.edu](http://www.med.harvard.edu) Internet: [http://www.med.harvard.edu/jpnm/physics/nmltd/radprin/sect2/2.2/2\\_2.1.html](http://www.med.harvard.edu/jpnm/physics/nmltd/radprin/sect2/2.2/2_2.1.html)
- [12] <https://www.britannica.com/science/activity-radioactivity>
- [13] Fraden, J. (2010): "*Handbook of Modern Sensors*", fourth edition, Springer, New York
- [14] Wikipedia (2017), "*Semiconductor detectors*", [www.wikipedia.org](http://www.wikipedia.org) Internet: [https://en.wikipedia.org/wiki/Semiconductor\\_detector](https://en.wikipedia.org/wiki/Semiconductor_detector)
- [15] Griffin, Darren (2011), "*How does the Global Positioning System work ?*", [www.pocketgpsworld.com](http://www.pocketgpsworld.com) Internet: <http://www.pocketgpsworld.com/howgpsworks.php>
- [16] Pogge, Richard W. (2017), "*Real-World Relativity: The GPS Navigation System*", [www.astronomy.ohio-state.edu](http://www.astronomy.ohio-state.edu) Internet: <http://www.astronomy.ohio-state.edu/~pogge/Ast162/Unit5/gps.html>
- [17] GISGeography, "*Trilateration vs Triangulation – How GPS Receivers Work*", [www.gisgeography.com](http://www.gisgeography.com) Internet: <http://gisgeography.com/trilateration-triangulation-gps/>
- [18] Documentation QGIS 2.0, "*Coordinate Reference Systems*", [www.http://docs.qgis.org](http://www.http://docs.qgis.org) Internet: [http://docs.qgis.org/2.0/en/docs/gentle\\_gis\\_introduction/coordinate\\_reference\\_systems.html](http://docs.qgis.org/2.0/en/docs/gentle_gis_introduction/coordinate_reference_systems.html)
- [19] Esri (2013), "*What are geographic coordinate systems?*", [www.resources.arcgis.com](http://www.resources.arcgis.com) Internet: <http://resources.arcgis.com/en/help/main/10.1/index.html#//003r00000006000000>
- [20] Internet: <https://confluence.qps.nl/pages/viewpage.action?pageId=29855173>

- [21] Esri (2013), "*What are projected coordinate systems?*", [www.resources.arcgis.com](http://resources.arcgis.com/en/help/main/10.1/index.html#/What_are_projected_coordinate_systems/003r0000000p000000/)  
Internet: [http://resources.arcgis.com/en/help/main/10.1/index.html#/What\\_are\\_projected\\_coordinate\\_systems/003r0000000p000000/](http://resources.arcgis.com/en/help/main/10.1/index.html#/What_are_projected_coordinate_systems/003r0000000p000000/)
- [22] Wikipedia (2017), "*Universal Transverse Mercator coordinate systems*", [www.wikipedia.org](http://www.wikipedia.org)  
Internet: [https://en.wikipedia.org/wiki/Universal\\_Transverse\\_Mercator\\_coordinate\\_system](https://en.wikipedia.org/wiki/Universal_Transverse_Mercator_coordinate_system)
- [23] Internet: <http://www.gammapdata.se/no/produkter/radioaktivitetssmaling/detectors/cadmium-zinc-telluride-czt/>
- [24] DJI (2017), "*SPREADING WINGS S900SPECS*"  
Internet: <http://www.dji.com/spreading-wings-s900/info>
- [25] u-blox (2016), "u-blox M8 concurrent GNSS timing modules" Data Sheet,  
Internet: [https://www.u-blox.com/sites/default/files/NEO-LEA-M8T-FW3\\_DataSheet\\_%28UBX-15025193%29.pdf](https://www.u-blox.com/sites/default/files/NEO-LEA-M8T-FW3_DataSheet_%28UBX-15025193%29.pdf)
- [26] Foreman, S. (2013), "*Kromek USB spectrometer Interface - Technical Manual*"
- [27] Internet <https://www.notebookcheck.net/Mobile-Graphics-Cards-Benchmark-List.844.0.html>

All references has been updated 16/06-2017

## Figures

Figure 1 Internet: <https://owlcation.com/stem/Gamma-Ray-Spectroscopy>

Figure 2 Internet: <http://hyperphysics.phy-astr.gsu.edu/hbase/Nuclear/betaex.html>

Figure 3 Internet: <https://gis.stackexchange.com/questions/12866/why-does-gps-positioning-require-four-satellites>

Figure 4 Internet: [http://docs.qgis.org/2.0/en/docs/gentle\\_gis\\_introduction/coordinate\\_reference\\_systems.html](http://docs.qgis.org/2.0/en/docs/gentle_gis_introduction/coordinate_reference_systems.html)

Figure 5 Internet: <https://www.ibm.com/developerworks/library/j-coordconvert/>

Figure 6 Photo taken by Philip Lind

Figure 7 Photo taken by Philip Lind

Figure 8 Photo taken by Philip Lind

Figure 9 Screenshot taken by Philip Lind

Figure 10 Screenshot taken by Philip Lind

Figure 11 Schematic made by Philip Lind

Figure 12 Schematic made by Philip Lind

Figure 13 Screenshot taken by Pernille Boldrup Østfeldt

Figure 14 Screenshot taken by Philip Lind

Figure 15 Screenshot taken by Pernille Boldrup Østfeldt

Figure 17 Schematic made by Philip Lind

Figure 18 Schematic made by Philip Lind

Figure 19 Schematic made by Philip Lind

Figure 20 Schematic made by Philip Lind

Figure 21 Plot created with the MATLAB program.

Figure 22 Plot created with the MATLAB program.

Figure 23 Plot created with the MATLAB program.

Figure 24 Plot created with the MATLAB program.

Figure 25 Plot created with the MATLAB program.

Figure 26 Plot created with the MATLAB program.

Figure 27 Plot created with the MATLAB program.

Figure 28 Plot created with the MATLAB program.

Figure 29 Plot created with the MATLAB program.

Figure 30 Plot created with the MATLAB program.

Figure 31 Plot created with the MATLAB program.