UPPSALA
UNIVERSITET

F 18 025
Examensarbete 15 hp
Juni 2018

Construction of a Geiger counter

For cosmic radiation measurements in near space conditions.

Carl Tysk
Anton Olsson
Elvira Brenner

Populärvetenskaplig sammanfattning

CIGS solceller är en ny typ av solcell som utvecklas på Uppsala Universitet. De består av koppar, indium, gallium och selen som tillsammans bildar ett halvledarmaterial som har visat sig vara en väldigt effektiv ljusabsorbator. CIGS är så kallade tunnfilmssolceller, de kan tillverkas mycket tunnare än många andra halvledarmaterial eftersom att deras absorbtionsförmåga är så hög.

Det är idag vanligt att rymdfarkoster och sateliter använder sig av solceller för att utvinna elektricitet eftersom att behovet av batterier då minskar. Dock så innebär den kraftigare kosmiska strålningen utanför atmosfären att solcellerna slits i en högre takt. Jämförs CIGS med de konventionella kiselsolcellerna som finns på marknaden idag är de både lättare och billigare. Experiment har också visat att de är tåligare mot radioaktiv strålning. Dessa egenskaper gör dem till väldigt lämpliga kandidater för rymdteknik. Om de är tåligare mot strålning är det möjligt att minska hur mycket de behöver underhållas och repareras, vilket förstås är avsevärt svårare och dyrare för farkoster som redan är i rymden. Minskas massan på solcellerna som ska fästas på rymdfarkoster kan även bränslekostnaderna reduceras.

Det är svårt att veta hur stor potential CIGS solcellerna har inom rymdtekniken. Den kosmiska miljön är svår att simulera på jorden, och strålningspåverkan på CIGS solceller är ett aktivt forskningsämne. LODESTAR är ett projekt vars mål är att undersöka hur CIGS solceller påverkas av den kosmiska strålningen. Det är ett projekt inom det svensk-tyska programmet BEXUS som genomförs i samarbete med ESA. BEXUS ger universitetsstudenter möjligheten att sända upp egna experiment på en ballong från Esrange i Kiruna. LODESTAR kommer att göra mätningar på solcellerna före, under och direkt efter färden. "Construction of a Geiger counter" är en del av LODESTAR-projektet. Vårt mål är att tillverka en strålningsmätare som kan mäta strålningen som solcellerna utsätts för. Då mätaren skall tåla en ballongfärd upp till stratosfären måste den vara tålig. Inga kontakter i dess krets får lossna och den måste uthärda låga temperaturer och låga tryck. Vid låga tryck finns det en risk att elektriska urladdningar sker, vilket skulle kunna vara ytterst destruktivt för hela experimentet.

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Construction of a Geiger tube

Carl Tysk Anton Olsson Elvira Brenner

"Construction of a Geiger counter" is a project which aims to create a Geiger counter that is functional under near space conditions. It is a part of a bigger project called LODESTAR, which aims to measure how cosmic radiation affects CIGS solar cells. The counter will be used to measure how much radiation the solar cells get exposed to. The solar cells and the counter will be sent up on a balloon. The balloon will lift from Esrange in Kiruna and can reach an approximate altitude of 30 km. At this altitude the cosmic radiation can be approximated to consist of only high energy protons. The counter has to be able to detect these protons, at a rate of at least 40 counts per second. It also has to be small and lightweight. It is important that no cords come loose from its connections and that it can operate in the low temperatures it will encounter. The high voltages and low pressure might also cause electric discharges that can damage the circuit.

The Geiger-Müller counter consists of an Arduino Nano generating a square wave and a boost converter, stepping the voltage up to 400 V from the 12 V supply voltage. The 400V is needed for the Geiger-Müller tube to operate. Every detection the tube makes gets stored on the Arduino. When a discharge occurs in the circuit the recharge time is 500 microseconds, meaning the counter should be able to detect 2000 discharges per second. All components have been chosen so that they can operate for the high voltages and low temperatures.

Testing the Geiger-Müller counter showed that the circuit, which will be shielded inside a box during the balloon flight, could operate for temperatures down to -49^{circ}C, where -40^{circ}C is the expected temperature in the box in which it is placed. Furthermore, the tube could operate at temperatures down to -80^{circ}C, which is the lowest possible temperature that can be expected outside the box. A vacuum test showed that the circuit can operate for a pressure as low as 0.2 mBar. The expected pressure will be about 100 times greater. A radiation test showed that our tube could detect secondary radiation from a Am-241 radioactive source. It could not detect alpha radiation directly, but protons are much lighter than alpha particles so that will not be a problem. Simulations have shown that protons with an energy of 4 MeV or higher should be detected. The only protons that will pass through the plastic shielding covering the solar cells are those with an energy of 15 MeV or more. Therefore, the counter will be able to detect all the radiation that is potentially damaging the solar cells.

The Geiger-Müller counter constructed in this projects fulfills all the requirements stated in the beginning. This leads us to believe that the counter can measure the radiation in the LODESTAR project without breaking.

Handledare: Uwe Zimmerman
Ämnesgranskare: Michelle Åhlén
Examinator: Martin Sjödin
ISSN: 1401-5757, F 18 025

Contents

1 Introduction **4**
1.1 Background 4
1.2 Objective 4
1.3 Theory 4
1.3.1 Geiger-Müller tube 4
1.3.2 Electronic discharges 6
1.3.3 Boost converter 8

2 Method **9**
2.1 Work procedure 9
2.2 Arduino software 12
2.3 Tests 13
2.3.1 Temperature test 13
2.3.2 Pressure test 13
2.3.3 Radiation measurement 14

3 Result **15**
3.1 Temperature test 16
3.2 Pressure test 18
3.3 Radiation test 18

4 Discussion **19**

5 Conclusion **20**

1 Introduction

1.1 Background

The BEXUS program is realized under a bilateral Agency Agreement between the German Aerospace Center (DLR), Swedish National Space Board (SNSB) and the European Space Agency (ESA). The program allows students to carry out scientific and technological experiments on balloons. The balloon can reach a maximum altitude of 30 km and will fly for 2-5 hours. [1]

LODESTAR is one of the experiments scheduled to fly on the BEXUS balloon flight in October 2018. The objective of the experiment is to investigate the effects of cosmic radiation on CIGS solar cells. [2] To measure the radiation the solar cells gets exposed to a radiation measurement unit has to be installed. The experiment will launch from Esrange in Kiruna and reach an altitude of approximately 27 km. A research made over Murmansk showed that the cosmic radiation at the altitude mostly consists of high energy protons. [3] Since Murmansk is geographically close to Kiruna the cosmic ray spectrum can be approximated to only consist of protons.

1.2 Objective

The task is to design a Geiger counter that can be sent up on a balloon to measure the high energy protons. It should be small and lightweight to fit in the module that is being sent up. It is important that no wires lose contact during the flight. Also, it has to be able to endure the harsh external conditions, the low temperature and pressure. In high voltage systems spontaneous discharges can occur, especially in low pressure environments. The counter must be constructed in a way so that these discharges do not happen.

One request from the LODESTAR project is that the counter can send data to an Arduino Mega using I2C communication. The counter should also have a function to be turned off and on by command from the Arduino Mega. The radiation dosage should be displayed by diodes where a higher intensity results in a higher number of diodes being lit.

More specifically, the circuit should, according to simulations, at lowest have to endure -20°C but -40°C is the requirement that is set so that there is a safety margin. The pressure that is expected is about 20 mBar but the circuit will be tested down to 0.2mBar. The Geiger counter should be able to detect at least 40 detections per second.

1.3 Theory

1.3.1 Geiger-Müller tube

In a Geiger counter there is a Geiger-Müller tube detecting the radiation. Geiger-Müller tubes need a high voltage to operate. The tube consists of a chamber containing two electrodes and a potential difference of several hundred volts. The chamber is filled with a gas at a low pressure. A charged particle passing through the gas causes excited and ionized molecules along its path. A neutral molecule being ionized results in a positive ion and a free, negatively charged electron: an ion pair. Due to the high voltage there is a strong electric field accelerating the positive ions towards the cathode and the electrons towards the anode. Free electrons close to the anode gain sufficient energy to ionize more gas molecules due to a stronger electric field. This creates a

large number of electron avalanches, meaning the tube can produce a significant output pulse from a single original ionizing event. This phenomenon is called Townsend avalanche. [7]
Different tubes were considered for the experiment. However, the STS-5 Geiger-Müller tube has, for this experiment, overall the best properties and was therefore chosen. From an economic perspective it was cheap. According to its data sheet it can function down to -40°C and it has been used in a previous experiment measuring cosmic radiation in space. [8] The walls of the STS-5 consist of 50μm thick stainless steel. [9] The National Institute of Standards and Technology (NIST), provides tabulated values for the range of protons in Iron, given in g/cm^2. [10] Since stainless steel mainly consists of iron and has a similar density it is approximated that stainless steel have similar proton blocking abilities as iron. Figure 1 shows the range from NIST divided by the density of the walls in the Geiger tube, the wall thickness is marked by a red line in the figure. From this we can estimate that the minimum energy needed by a proton to ionize the tube is 4 MeV. SRIM is a free software used to calculate range and stopping power in matter. [11] Figure 2 shows the calculated predicted path of 2000 hydrogen ions with an energy of 4 MeV entering into stainless steel simulated with the SRIM software. There are only a few stray ions not managing to reach 50 μm into the stainless steel showing that the estimation used in figure 1 is valid. The energy needed for a proton to penetrate the plastic shielding used on the CIGS solar cells during the LODESTAR experiment is estimated to 15 MeV, this means the STS-5 is sensitive enough to measure the relevant radiation.

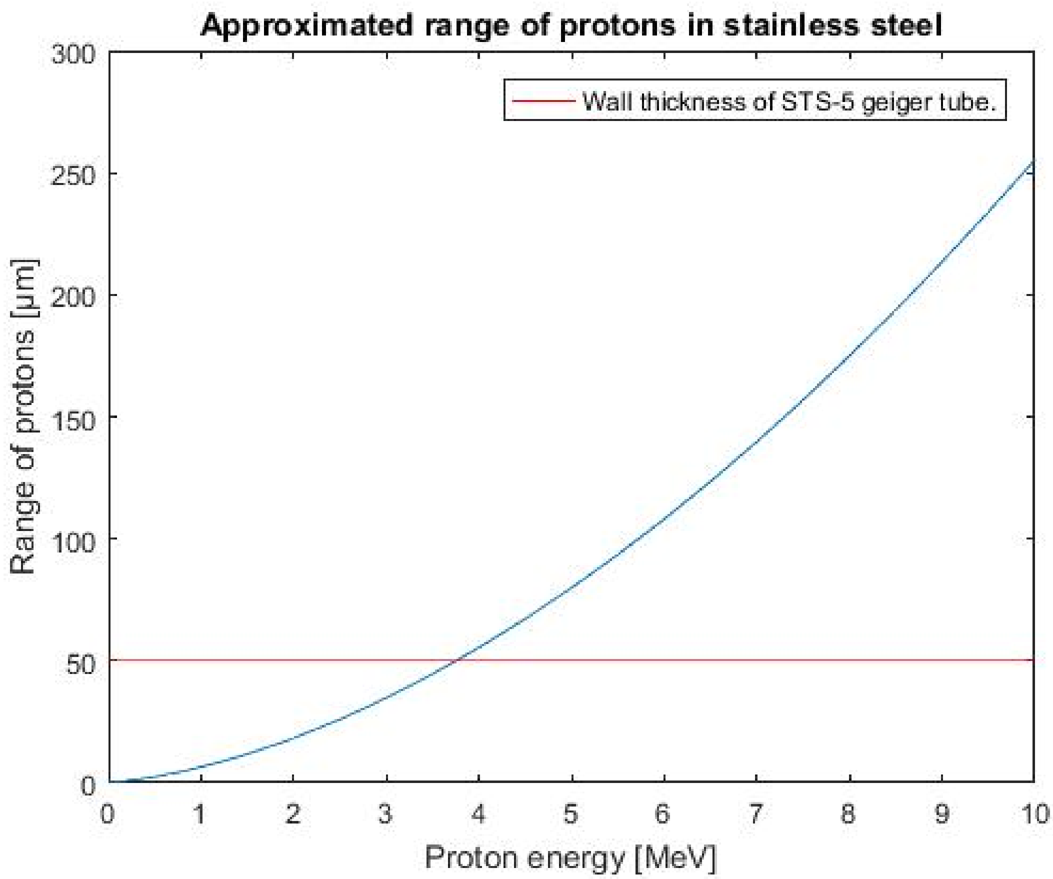

Figure 1: *Approximate range of protons with different kinetic energies in stainless steel.*

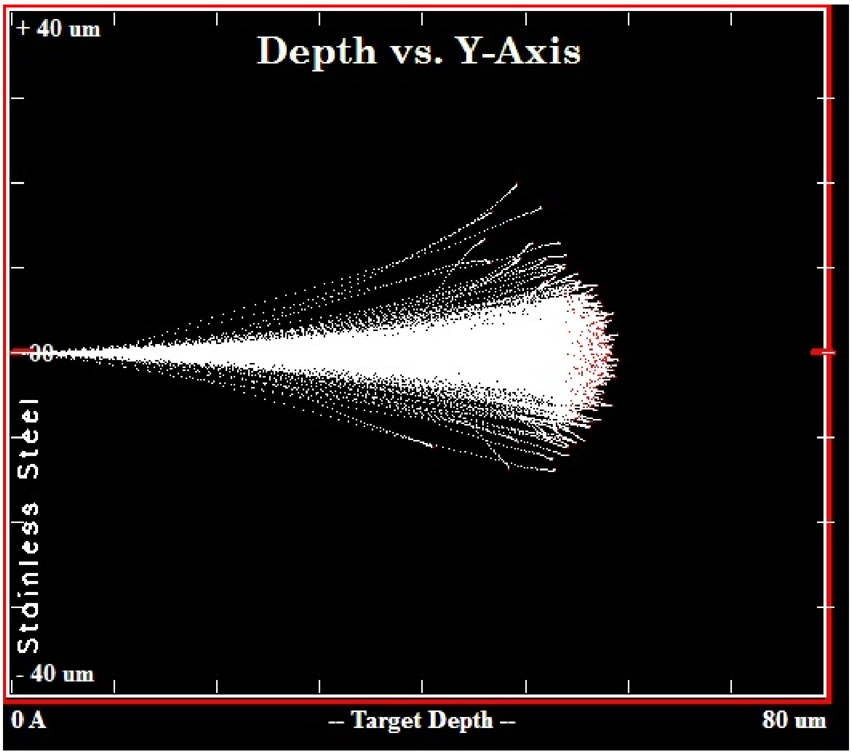

Figure 2: *Simulated path of 2000 H ions with 4 MeV energy projected in stainless steel.*

1.3.2 Electronic discharges

In high voltage systems spontaneous discharges can occur. This happens when the strength of the potential gradient from a conductors surface is stronger than the dielectric strength of the air surrounding it. If this happens then the air itself becomes ionized thus letting a current flow out of the conductor into the air, this phenomenom is called Corona discharge or the Corona effect. The dielectric strength of the air decreases when the pressure drops meaning this becomes even more important in low pressure conditions. Corona discharge may cause efficiency losses and it is undesired in most circuits. The maximum distance between two conductors with a certain potential that may cause Corona effect can be expressed with Peek's law,

$$d = re^{\frac{V_C}{m_v g_v r}}. \tag{1}$$

Where d is the maximum distance between the electrodes, V_C is the voltage, r is the radius of the conductors, m_v is a surface irregularity factor, g_v is the critical electric field that depends on the density of the gas. [4] A plot of Peek's law with the conditions we expect for our experiment can be seen in figure 3. From this graph we can see that for a 500 V potential the maximum distance between two electrodes to cause Corona discharge is approximately 2.8 mm. In the actual circuit the highest voltage will be about 400 V which will mean that the conductors can be closer without causing corona but we choose to use 500 V for calculations as a safety margin.

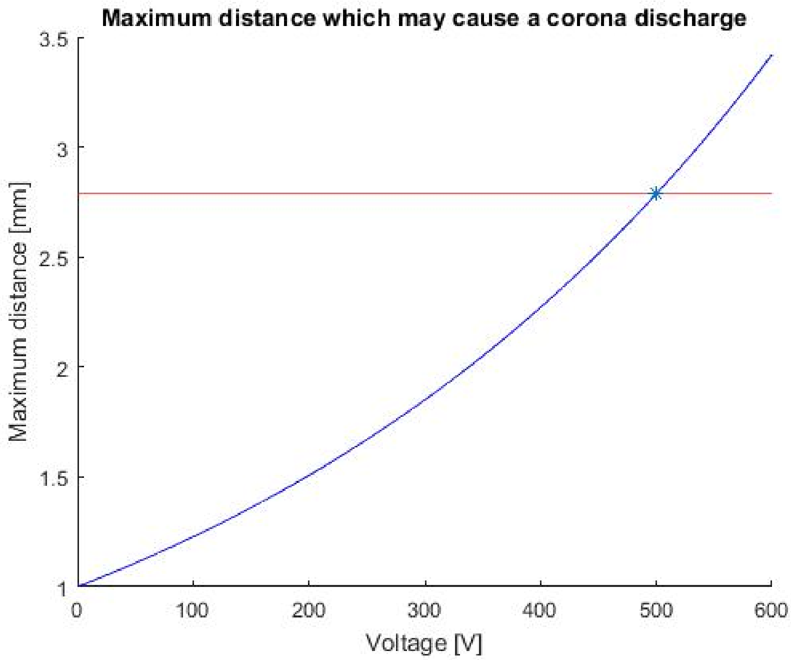

Figure 3: *Maximum distance between conductors which may cause a Corona discharge over voltage according to Peek's law.*

If the electric potential in the surrounding air becomes even greater a discharge between two electrodes can occur. This can damage other components in a circuit or cause physical harm. The voltage needed between two conductors can be calculated using Paschen's law,

$$V_B = \frac{Bpd}{ln(Apd) - ln(ln(1 + \frac{1}{\gamma_s e}))}. \tag{2}$$

Here, V_B is the breakdown voltage e.g. the voltage needed to cause a discharge, p is the pressure, d is the distance between the electrodes. [5] A, B and γ_{se} are related to the ionization energies in the gas and the number of emitted electrons from an ionization, these can be difficult to determine for different conditions. In figure 4 we can see the Paschen curve plotted for air at 20 degrees Celsius and with different values for the product of p and d. From the figure we can approximate for a 500 V potential, that the maximum product of pd needed to initiate a discharge is 35 mTorr m, this equals 4666 mm Pa. The pressure at 25 km altitude is approximately 1.76 kPa which means that the maximum distance between two conductors that can cause a discharge is approximately 2.6 mm.

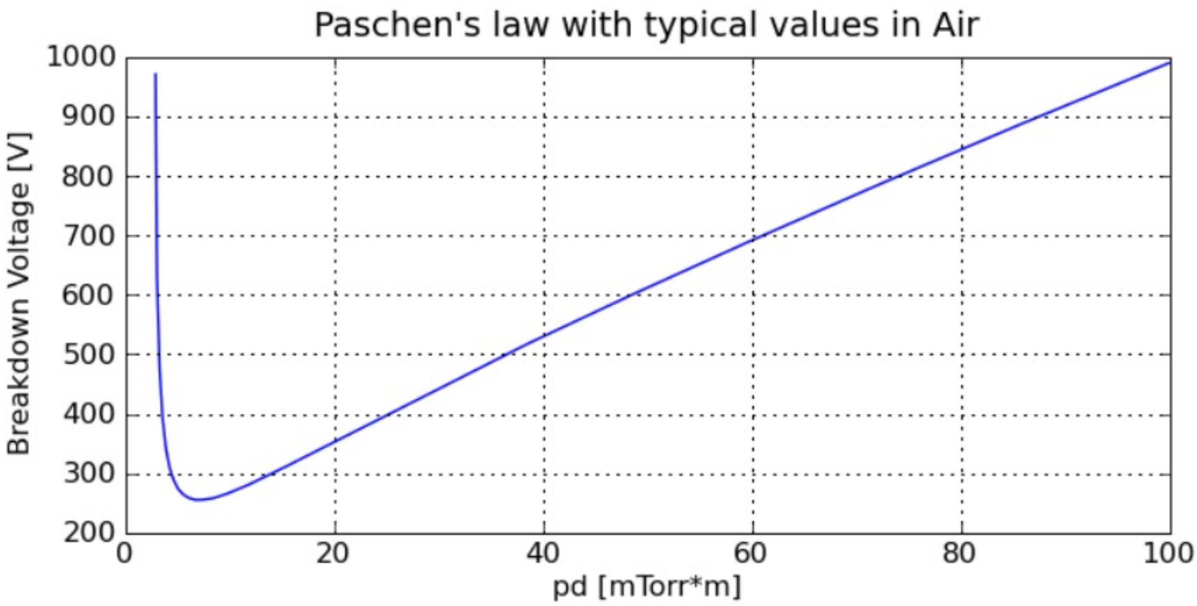

Figure 4: *Paschen curve for varying pd in air. [6]*

1.3.3 Boost converter

The LODESTAR experiment uses 28.8V batteries as a power source, a power regulator will be used to reduce the voltage to 12 V. The STS-5 tube has an operation voltage of 360-440V. One way of reaching this high voltage is by using a boost converter to step-up the voltage from 12V to about 400 V.

A boost converter is constructed using an inductor, a diode and a switch. In our case the switch will consist of a N-channel MOSFET transistor. If the gate connection of the transistor is supplied with a square wave it will act as a switch between an opened and closed state. The source connection of the transistor is connected to ground, meaning that when the switch is in an open state the transistor connects the inductor to ground, which creates a current flowing to the inductor (see fig.5). This stores energy in the form of an electric field. Once the switch closes the inductor is no longer directly connected to ground. This means that the current through the inductor is quickly reduced. Due to Faraday's law of induction,

$$v = L\frac{dI}{dt} \tag{3}$$

the change in current $\frac{dI}{dt}$, causes an electromotive force (e.m.t), forcing current through the inductor causing a voltage increase. [12] This pushes current through the diode and charges the capacitor. When the switch opens the diode prevents any current from returning from the capacitor through the transistor to ground. If the transistor is switching fast enough, the charge from the capacitor won't unload through the load resistance which can lead to much higher voltages at the output than what is given from the supply voltage.

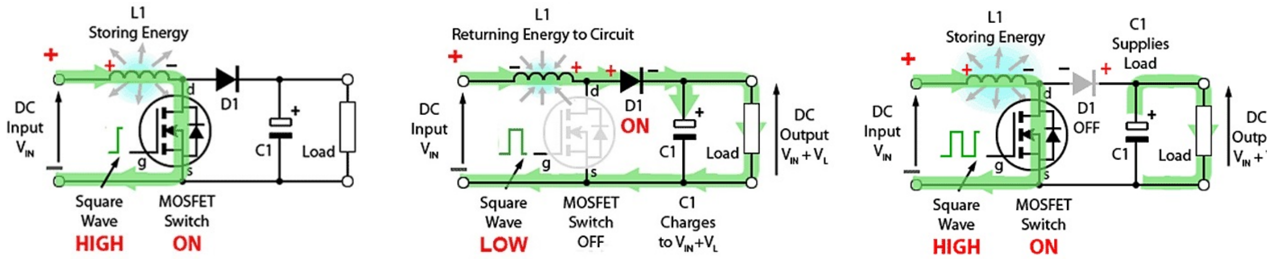

Figure 5: *Different states of a boost converter cycle* [13].

2 Method

2.1 Work procedure

To collect and store data some kind of microcontroller was needed. An Arduino Nano was chosen for this since it is cheap, not hard to find and relatively easy to program. The C based programming language is easy to use with a lot of preprogrammed functions. The switch in the boost converter, a MOSFET transistor, needs a square wave connected to its gate. At first a Wien bridge oscillator was used but it was later replaced with the Arduino using the function. *analogWrite*. Changing the frequency of the square wave was much easier with the Arduino, since it could be done through software changes. For the oscillator resistors and capacitors had to be removed and replaced which was far more time consuming. Also, the oscillator can only generate a square wave which is half on half off but the Arduino can vary the duty cycle, giving more control over the voltage. A circuit was simulated in LT-spice, connecting the square wave created by the Arduino to the gate of the a MOSFET transistor in the boost converter. The boost converter then steps the voltage up to 400 V. A capacitor was used replacing the Geiger-Müller tube and large resistors were used to keep the current low. The part to the right in the circuit, see figure 7, is a signal processing part connected to the Arduino. The capactitors keeps the output signal clean and the Arduino registers pulses from particles striking the tube. The output signal is 5V, when a discharge occurs in the Geiger-Müller tube it drops to 0 V. For every drop in the output voltage a variable counting them increases by one. The radiation can then be read from the Arduino in counts per minute.

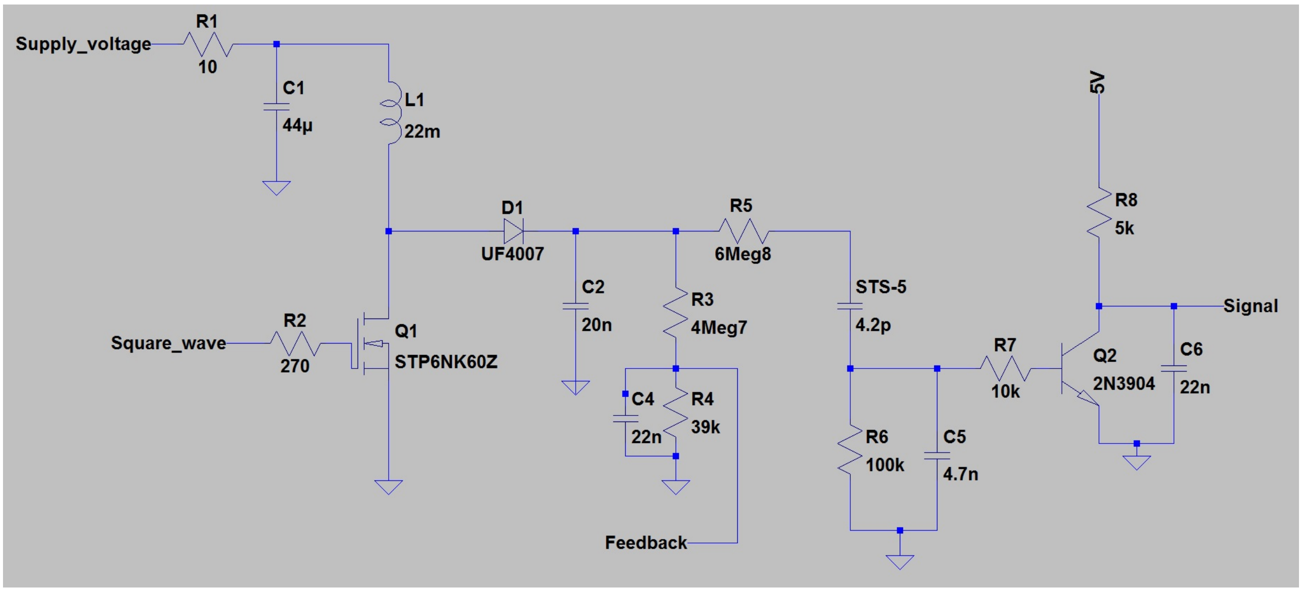

Figure 6: *Circuit diagram over the Geiger-Müller counter*

A feedback loop was added by making a voltage divider that lowered the voltage to a level readable by the Arduino. This ensured that the voltage over the Geiger-Müller tube would remain at 400 V even if some components were altered due to change in temperature. By varying the duty cycle of the square wave the circuit can be controlled easily by just making some minor changes in the code for the Arduino. The circuit was then built on a breadbord. Since parts of the circuit handle high voltages a few special components were needed. Table 1 shows a list of components and a motivation as to why the component was chosen. Since the Geiger counter need to be able to work in low pressures electrolytic capacitors need to be avoided since they contain a liquid that can boil if the pressure becomes to low. Only ceramic capacitors were used because of this.

Table 1. *List of components*

Component	Placement	Motivation
ST6NK60Z MOSFET transistor	Input of boost converter	Able to switch high voltages(600 V) while remaining a small size, fast switching speed.
UF4007 Diode	Diode in boost converter	High breakdown voltage(1000 V), fast switching speed.
High voltage capacitors	Capacitors in boost converter	Able of handling high voltages (2000 V), ceramic capacitors.
22 mH Inductor	Inductor in boost converter	Large inductor lowers the current needed to step up voltage.
2N3904 BJT	Pulse detection	High availability
Capacitors	Capacitors not in boost converter	Ceramic capacitors.
Resistors	Full circuit	Resistors with low temperature coefficient.

After the circuit worked properly on the breadboard the making of the printed circuit board (PCB) started. The circuit was drawn in KiCad and the design was printed on glossy transfer paper. The reason why the PCB was designed in KiCad was that it is user-friendly and it is relatively simple to transfer the schematic onto an actual copper board. The PCB board was passed through a laminator with the printed design and the paper residue was scrubbed off. To etch the PCB sodidum persulfate was mixed with warm water. A heating rod kept the water warm while a rack kept the PCB in a standing position in the solution. Sodium persulfate is highly oxidizing and dissolves copper to copper sulphate. To finish the PCB holes were drilled and components were soldered on. The legs of the N-channel MOSFET and the high voltage capacitors was insulated with heat-shrink tubing to avoid electronic discharges. The finished PCB was tested to see if it functioned properly.

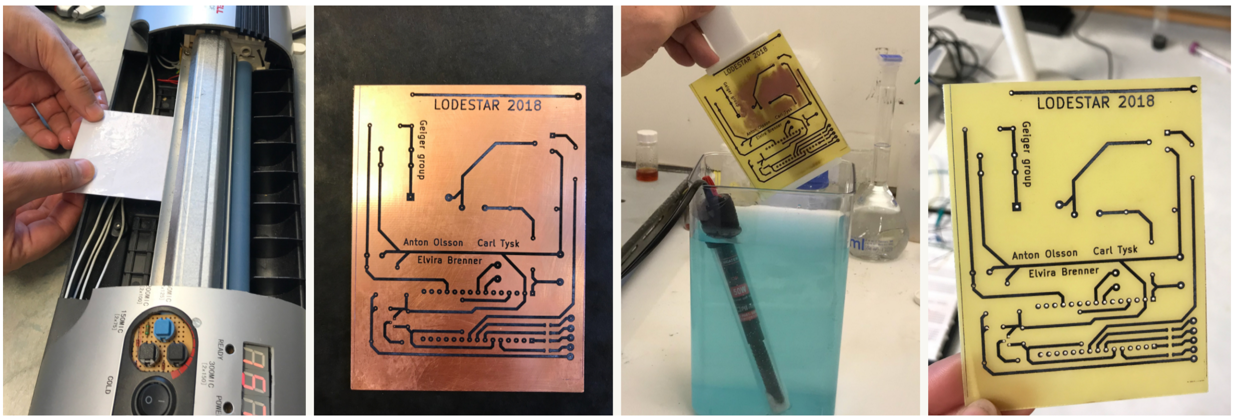

Figure 7: *The making of the PCB*

2.2 Arduino software

A few different functions are needed by the Arduino Nano. First, the Arduino needs to generate a square wave as input to the boost converter, this was accomplished by using the *analogWrite* function. This uses a method called pulse width modulation (PWM) and generates a square wave from an analog pin with a duty cycle between 0 and 255, where 0 gives a logical zero and 255 gives a logical one. The Arduino has a clock frequency of 62.5 kHz and by default the *analogWrite* function has a frequency of the clock frequency divided by 64 resulting in 980 Hz. The boost converter needs a higher frequency to step up the voltage so the PWM frequency was changed to the clocking frequency divided by 16, resulting in a frequency of 7.8 kHz. This has some unwanted side effects causing the time measurements of the Arduino system to run 8 times faster than normal. This has to be counteracted by multiplying all uses of the time function *millis* and the delay function *delay* by a factor eight. The duty cycle of the PWM have to be adapted to maintain a constant voltage since components characteristics may change due to changes in temperature and pressure. The Arduino reads the voltage at the Geiger tube with the function *analogRead* wich returns a value between 0 and 1023 where a 0 represents 0 volts and 1023 represents 5 volts. The Arduino library *PID* was the used to regulate the signal, this uses PID regulation to adapt the voltage in the feedback loop by altering the duty cycle. Different PID settings were tested and the settings found to give the best results was setting the derivative part, K_d, to zero and the integral part, K_i to 1 and the proportional part, K_p to 0.1. Since only small regulations are expected once the system reaches the target voltage it is reasonable that the integral part is the dominant since avoiding static errors are of higher priority than regulation speed.

Next the Arduino have to count the number of ionizations in the Geiger tube. This was done by implementing an Interrupt Service Protocol (ISP) with the Arduino function *attachInterrupt*. The ISP is prioritized above all other commands in the Arduino, and is activated when a change is detected on a digital pin. In our case the interrupt is activated when the signal on the designated pin switches from a logical one to a logical zero. When the interrupt is activated a Interrupt Service Routine (ISR) runs to perform actions wanted. The ISR *countPulse* registers an ionization by adding a one to two variables of int type, one of the variables are reset every minute and the other every second to enable the counts per minute and counts per second to be obtainable. When implementing the ISP some trouble with doubling were encountered, which means every pulse where counted twice. To avoid this a time check was added, only making one pulse each 500 microseconds possible. The Arduino software uses a continuous loop which repeatedly iterates its content. Inside the loop the Arduino checks if one minute has elapsed since its last reading, if that is the case then the number of counts are saved as an int named *countsPerMinute* and the count is reset. The same procedure is performed for the counts per second.

The Arduino Nano also needs to be able to communicate with an Arduino Mega over an I2C bus. This is accomplished by using the *Wire* library in Arduino. The Arduino Nano operates as a slave and it is assigned an address in the bus. The Arduino Mega operates as a master and can send a request for informaion or send information to the Nano. When the Mega requests information the int *countsPerMinute* is sent over the I2C bus. The bus can only handle byte size information and an int normally consists of two bytes. So the int has to be separated. This is managed easily with Arduino's functions *highByte* and *lowByte*. *highByte* returns the leftmost byte of the integer and *lowByte* returns the rightmost byte. The Mega may also send commands to the Nano which requires the Nano to start or stop the voltage supply for the Geiger-Müller tube. The mega sends one byte of information. If that byte is a one, the PID regulator is turned off and the duty cycle is set to zero, thus lowering the voltage to the supply voltage 12 volts. If the Mega sends a zero then the PID is turned back on and the voltage is resumed.

The int *countsPerSecond* is used to control four LED diodes indicating the amount of radiation measured by the unit. If the value exceeds a threshold value a certain amount of diodes are lit, this is controlled by the function *ledVar*. The full code can be seen in the appendix.

2.3 Tests

2.3.1 Temperature test

A requirement to send the constellation with the Bexus balloon is that our circuit has to be able to operate for temperatures as low as -40°C and our Geiger-Müller tube -80°C. Two temperature tests were done. In the first one the circuit and a radioactive Thorium source were placed together with a nitrogen gas supply in a freezer. The nitrogen is used to reduce the humidity and the formation of ice crystals. The freezer started at room temperature and as it was cooled down measurements were done continuously every time the temperature had decreased with about 10°C. Five measurements were done at each temperature of interest and an average was calculated along with the standard deviation. The circuit was cooled down to -80°C. When the temperature reached -37°C the voltage over the Geiger tube was increased from 400 V to 425 V to counteract a loss of data.

In the second test a new STS-5 tube was used and the electric circuit was placed outside of the freezer and only the Geiger-Müller tube and the Thorium source were placed inside. In this experiment no nitrogen gas was used. The freezer started at room temperature and the tube was cooled down to -80°C. Measurements were done continuously every time the temperature had dropped with about 10°C.

2.3.2 Pressure test

The circuit has to be able to operate low pressures, a requirement was 20 mBar. The circuit was placed in a vacuum chamber together with a radioactive Thorium source. To see if the circuit worked in there two LED-lights were connected. One light was on as long as the voltage over the tube stayed at 400V and the other was on as long as the counter measured over 5 counts per second. The pressure in the chamber decreased until it got to 0.2 mBar. Then a vault was opened to let air back in and the pressure increased to normal atmospheric pressure.

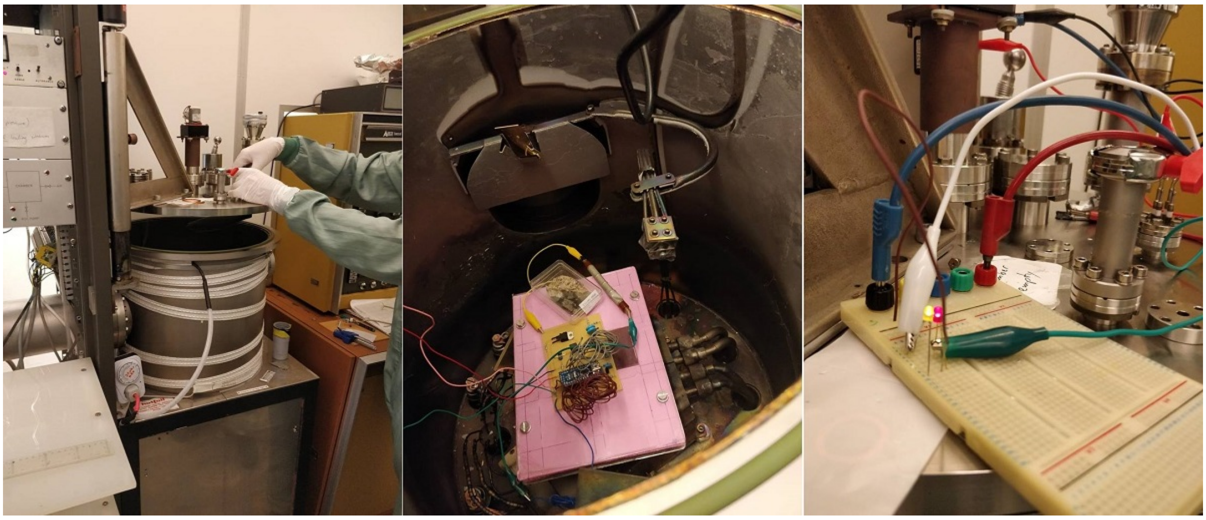

Figure 8: *The setup for the pressure test.*

2.3.3 Radiation measurement

The radiation from an americium-241 source with an intensity of 370 kBq was measured. The Am-241 emits mainly alpha-radiation but also secondary radiation like gamma radiation.[15] The Geiger-Müller tube was placed so that it touched the source and the radiation count was measured in counts per minute (CPM). Five measurements were done and an average was calculated. In the same way the source was measured at a distance of 2 cm, 4 cm and 8 cm. The same measurements were done using a RAM GENE-1 Meter and a Mini-Instruments Mini-Monitor G M Meter Type 510. The RAM GENE-1 Meter had a mica window covering its radiation detector instead of metal walls, enabling heavier particles to pass through. The background radiation was measured for all three measurement devices with the Am-241 source placed behind thick blocks of lead. The measured CPM was divided by the cross sectional area of the instrument used for the measurement.

Figure 9: *The setup for the radiation measurement test using our Geiger-Müller counter (left), RAM GENE-1 (middle) and Mini-Monitor G M Meter (right)*

3 Result

The finished product is a Geiger-Müller counter shown in figure 10. It can detect background radiation in a room, and also secondary radiation from a thorium source. Looking at a registered pulse in an oscilloscope the recharge time could be measured to 500 μs see figure 11, meaning the theoretical limit of counts the counter could detect is 2000 counts per second.

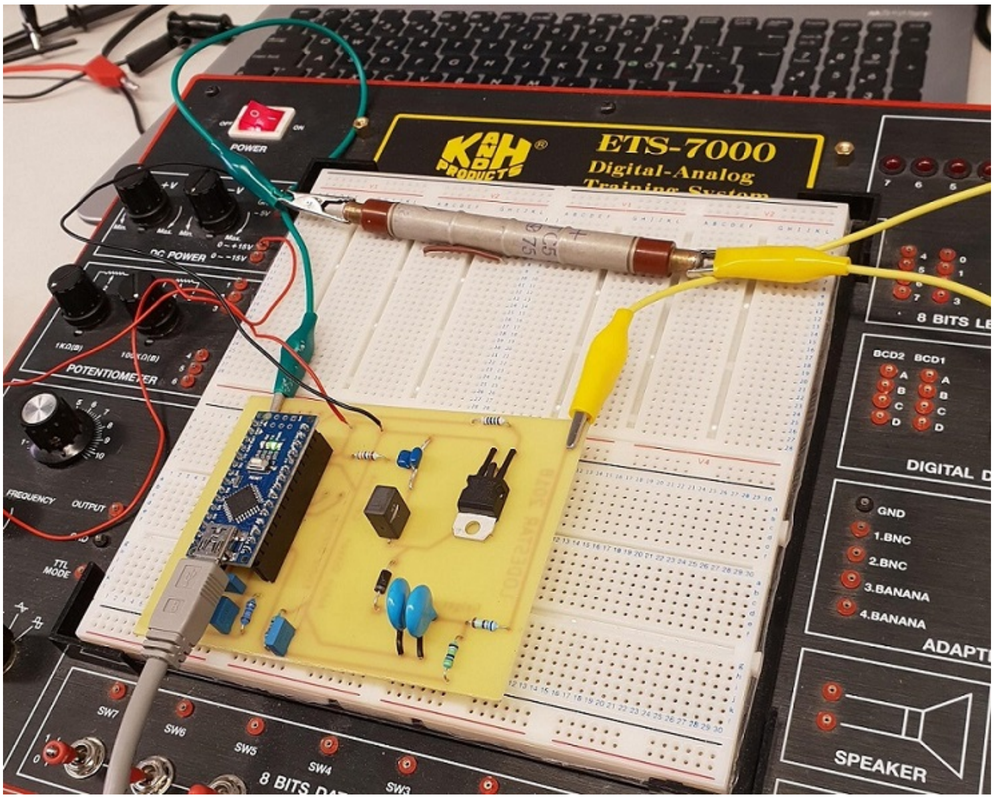

Figure 10: *The finished product*

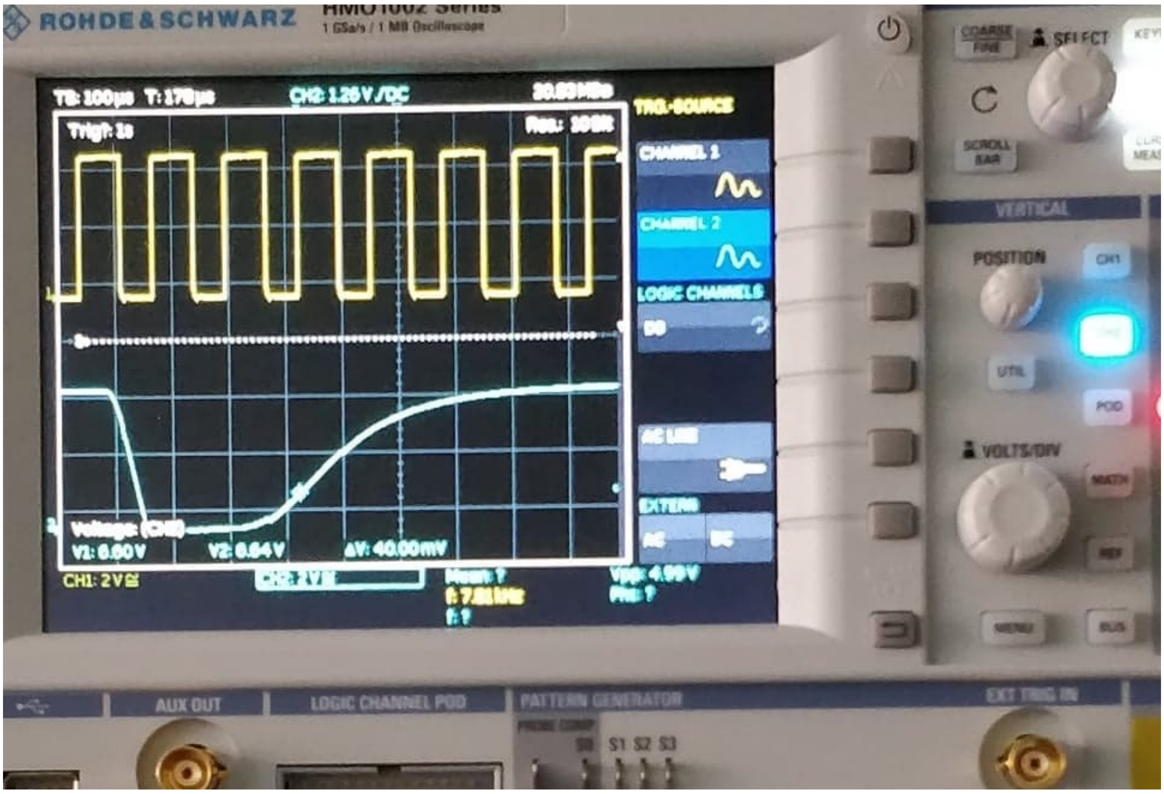

Figure 11: *Pulse from GM tube in the circuit(bottom channel), shown on oscilloscope. Top channel shows input signal from Arduino.*

3.1 Temperature test

The measurement data from the first temperature test can be seen in figure 12 and is represented as the CPM measured by the counter with the standard deviation. The counter worked down to -40°C, though the voltage had to be increased to 425 V at -37 ° in order to maintain correct measurements. When the temperature reached -49° the measured radiation dropped quickly down to 0 CPM resulting in the large standard deviation seen at -49 °. After the test was done, and the counter had been defrosted, it could be noted that the STS-5 tube had shrunk and had been deformed, see figure 14. However the tube was still able to measure background radiation. During the second cooling test the new STS-5 tube did not deform and it could function down to -80°C. However, the tube did not measure the same dosage at low temperatures as it did at high temperatures. A minor difference after -60°C could be noted, see figure 13.

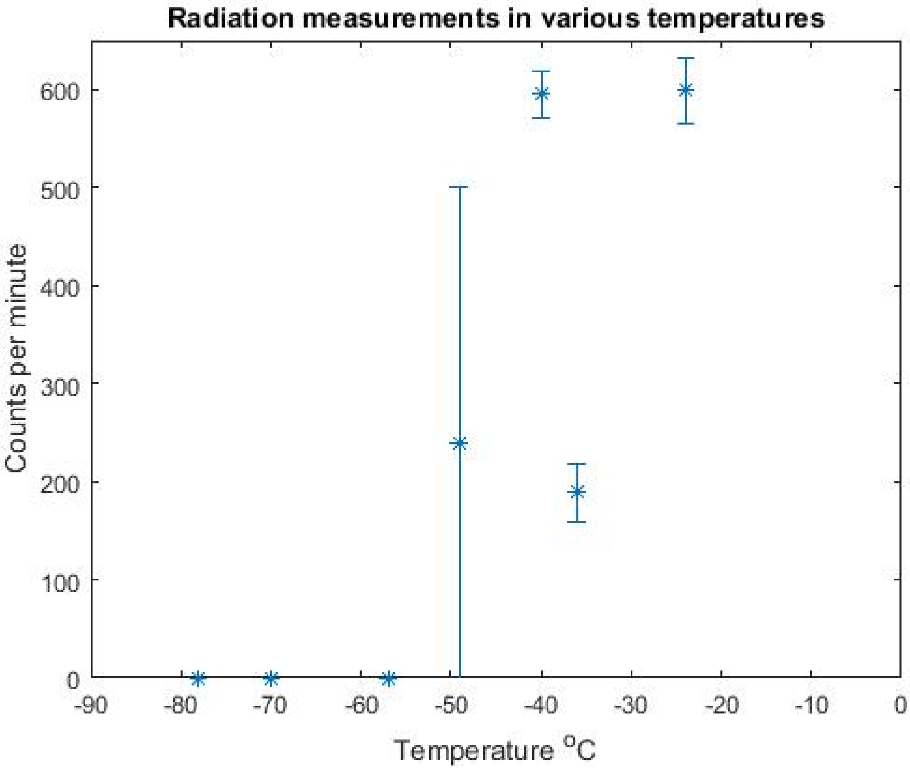

Figure 12: *Radiation measurements from the first temperature test.*

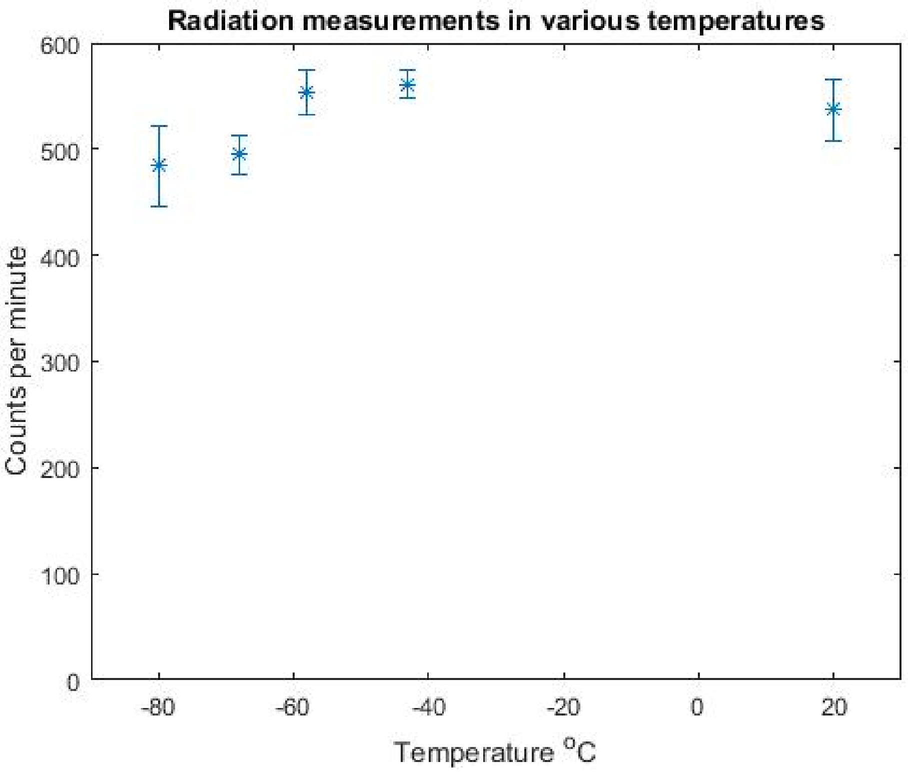

Figure 13: *Radiation measurements from the second temperature test.*

Figure 14: *The GM tube after the first cooling test.*

3.2 Pressure test

When the pressure dropped the circuit worked correctly, the voltage stayed above 400 V over the tube and it was doing measurements. When the test was done and a valve was opened to increase the pressure again the light that displayed the voltage over the tube blacked out, the light displaying the radiation measurements still worked.

3.3 Radiation test

The cross sectional areas of the instruments used during the radiation test was, 7.6 cm^2 for the STS-5, 39.6 cm^2 for the Mini Monitor and 15.6 cm^2 for the RAM GENE-1. Table 2 shows the CPM/cm^2 measured during the tests. Note that in the table neither the RAM GENE-1 nor the Mini-monitor shows any reading for 0 cm distance to the source. This is since the RAM GENE-1 reached its maximum level of radiation and the mini-monitor was enclosed by a metal shielding preventing the source to be placed closer than 2cm.

Table 2. *Radiation measurement results*

Distance to radiation source [cm]	STS-5 [min^{-1} cm^{-2}]	RAM GENE-1 [min^{-1} cm^{-2}]	Mini-monitor [min^{-1} cm^{-2}]
0	1349	overload	-
2	280	449	60
4	103	244	45
8	51	83	20
No source (background) radiation)	3.4	3.8	1.5-3.0

4 Discussion

The first temperature test showed that our construction could function down to -49°C. When the circuit instead was placed outside the freezer and the STS-5 tube inside freezer the tube did not deform in any way and was able to provide measurements down to the required -80 °C. This probably happened because the first tube had sustained some damage prior to the test. The tubes were constructed in 1975 and bought second hand on EBAY. There is no way to know how the tubes has been handled before being acquired for this experiment. Also the first tube was dropped on the floor by accident a few times which may have damaged the thin metal surrounding the ionization chamber. Furthermore, during the first test the radiation source was leaning on the tube which means that a force acted on the metal which could have led to the deformation of the tube. In the second temperature test the circuit did not measure as much radiation at the lower temperatures compared to the measurements done when the temperature was about room temperature. This may have been caused by the freezer having been opened after the third measurement to investigate whether the tube had deformed or not. This caused the freezer to shake and may have moved the radiation source slightly away from the tube. After the experiment when the equipment was taken out of the freezer the radiation source had moved from it's original position which supports this explanation. The circuit could function down to -49°C and the tube down to -80°C which means that both the circuit, that has to endure at worst -40°C, and the tube, which has to manage -80°C both are capable of withstanding the space environment.

The purpose of the pressure test was to analyze if the construction could operate at a low pressure without any electrical discharges. The pressure chamber went down to 0.2 mBar, whereas the lowest estimated pressure for the balloon flight is 20 mBar. As the pressure decreased the construction did measurements the entire time and there were no electrical discharges that could be seen from the observational window. Right before the vault opened to let air back in to increase the pressure the construction worked perfectly. According to Paschen's and Peek's laws a decrease in pressure increases the risk of discharges. However, the circuit stopped working when the pressure rose again. This may be because when the vault opened it shook our construction causing a connection to come loose. But most likely the Arduino was damaged from the first cooling test. Testing the Arduino showed that the feedback pin receiving the value of the voltage did not work and neither did several of the analog pins. But even with a damaged Arduino the counter could measure radiation down to a pressure 100 times lower than what is expected during the experiment which shows that the circuit is safe to use.

The Am-241 emits alpha radiation, which the RAM GENE-1 Meter could measure since it has a mica window that lets bigger particles pass through. The walls of our Geiger-Müller tube and the mini-monitors metal rod are too thick for the alpha particles to penetrate it. The alpha radiation emitted from the source will ionize particles in the air surrounding it as well as in the walls of the Geiger-Müller tube. This will result in secondary radiation which can be registered by our counter and the mini monitor. Since the RAM GENE-1can measure both types of radiation it is expected that it would measure a higher dose than the two other devices which it did. The mini-monitor also measured more counts than our Geiger counter, but when the radiation over the cross sectional area per cm^2 was calculated our tube registered more counts/cm^2 than the mini monitor. The reason this comparison was needed is because the larger the area of the measuring device is the more counts can strike it. The mini monitor had a measuring device almost twice the size of the STS-5 tube. This comparison is based on an approximation that all of the tubes cross sections receives the same amount of radiation, it is very likely that since the mini-monitor was much longer than our STS-5 the ends of the mini-monitor tube had very little activity. The mini-monitor was also covered partly by a metal casing that may have shielded it from some radiation. For a better comparison two tubes with similar shape should be used.

The test shows that our construction competes well with other windowless Geiger counters but that a window counter is far better for measuring alpha particles. But we are only interested in measuring the high energy protons that can enter our tube. Since charged particles have almost 100% efficiency in gaseous counters once they penetrated the walls, all relevant radiation should be detected by the counter. [16] The test also shows that the counter can register much higher doses than the 40 CPS required.

There are several ways the construction and the tests could be improved. Firstly, another radiation detection could be used, for example an ionization chamber or a semiconductor detector which can also measure the energy levels of the protons. If we had used a Geiger-Müller tube which could measure alpha radiation as well our counter could be used in more types of projects. We could also try to improve the recharge time when a pulse is detected so that it can measure more pulses than 2000 per second by using smaller capacitors in the signal processing part of the circuit. As far as testing goes one could measure exactly how thick the STS-5 shell is to do better estimations of what energy a particle needs to be detected. Since we are only interested in measuring protons during the experiment it would be more relevant to perform a radiation test using a proton source. This is however hard to accomplish since proton beams can be expensive to operate.

5 Conclusion

The radiation measurement unit constructed in this project is able to measure radiation, work at low temperatures and at a low pressure. When exposing it to extremely low temperatures and pressures, much less than it would experience for the BEXUS balloon flight, the circuit did not function properly. The circuit managed down to -49° C. When exposing the circuit to low pressures the measurement worked as it should but when the pressure was increased, the circuit stopped measuring. However, this was probably due to damage caused to the Arduino during the cooling test. The objective was fulfilled, the circuit can endure -40° C, a pressure as low as 0.2mBar and the tube can measure more than 40 detections per second and can endure temperatures as low as -80° C. Conclusively, our radiation unit will be able to measure cosmic radiation in near space conditions.

Appendix

Arduino code

```
//Include relevant libraries
#include <Wire.h>
#include <PID_v1.h>

//Definitions
#define sigPin 6        //Pin Supplying PWM signal (DIGITAL)
#define feedback 0      //Pin reading feedback loop (ANALOG)
#define gm_input 3      //Pin recieving signal from GM tube (DIGITAL)

//Variables
double tresh = (3.3/5.0)*1023;    //Setpoint value in feedback, 3.3 gives
    400V.
int ledArray [] = {2,1,4,5};      //Pins used for diodes.

//Variables for ISR.
long interruptTime = 0;          //Time for a new Interrupt.
long lastInterrupt = 0;          //Time when last interrupt occured.
long bouncePreventTime = 4;      //Minimum allowed time between interrupts.

//Variables for PID
double Input;                    //Input value.
double Output;                   //Output value.
double Setpoint;                 //Setpoint value.

//Variables for radiation count.
unsigned int countPerSecond = 0;       //Counts per second (CPS).
unsigned int countPerMinute = 0;       //Counts per minute (CPM).
unsigned int count = 0;                //Counts used in ISR
unsigned int countDiode = 0;           //Counts used in ISR for diode
    control.
long timePreviousMeassure = 0;         //Time for previous CPM reading.
long timePreviousFlash = 0;            //Time for previous CPS reading.

//Threshhold values for diodes.
int diode1 = 2;
int diode2 = 100;
int diode3 = 1;
int diode4 = 2;

//Set PID tuning parameters. Kp = 0.1, Ki = 1, Kd = 0.
//Direct sets direction. Higher output gives higher input.
PID feedbackPID(&Input, &Output, &Setpoint, 0.1, 1, 0, DIRECT);

void setup() {
  //Define pins as input or output.
  pinMode(sigPin,OUTPUT);
  for (int i=0;i<4;i++){
    pinMode(ledArray[i],OUTPUT);
  }
```

```
  pinMode(gm_input,INPUT);
  digitalWrite(gm_input,HIGH); //Set high as "normal" mode.
  //PID startup
  Input = analogRead(feedback);      //Reads input
  Setpoint = tresh;                  //Defines setpoint.
  feedbackPID.SetMode(AUTOMATIC);          //Turn regulator ON.
  feedbackPID.SetOutputLimits(24, 200);    //Sets limits for dutyCycle.
  feedbackPID.SetSampleTime(10);           //Sets regulator speed.

  //Alter PWM frequency to 62500/8 (Instead of 62500/64).
  TCCR0B = (TCCR0B & 0b11111000) | 0x02;

  //Attaches ISP to gm_inpt pin.
  // When pulse is detected ISR countPulse is initiated.
  attachInterrupt(digitalPinToInterrupt(gm_input), countPulse, FALLING);

  //i2c bus start up.
  Wire.begin(3);                  // Join i2c bus with address #3.
  Wire.onRequest(requestEvent); // Run requestEvent when requested by MEGA.
}

void loop() {
  //Run PID regulation
  Input = analogRead(feedback);      //Read input.
  feedbackPID.Compute();             //Compute regulation needed.
  analogWrite(sigPin, Output);       //Apply regulation

  //Check if one minute has passed since last measurement.
  if ((millis())-timePreviousMeassure > 480000){          //8*60 s = 480000
      ms
    countPerMinute = count;                               //Set CPM.
    timePreviousMeassure = millis();                      //Update
        measurement time.
    count = 0;                                            //Reset count.
  }
  //Check if one second has passed since last measurement.
  if (millis()-timePreviousFlash > 8000){                //8*1 s = 8000 ms
    countPerSecond = countDiode;                         //Set CPS.
    timePreviousFlash = millis();                        //Update
        measurement time.
    countDiode = 0;                                      //Reset countDiode.

    //Check CPS vs thresholds. Run ledVar function based on results.
    if(countPerSecond <= diode1) ledVar(0);
    if((countPerSecond <= diode2)&&(countPerMinute>diode1)) ledVar(1);
    if((countPerSecond <= diode3)&&(countPerMinute>diode2)) ledVar(2);
    if((countPerSecond <= diode4)&&(countPerMinute>diode3)) ledVar(3);
    if(countPerSecond>diode4) ledVar(4);
}
```

```
//Function that executes whenever data is requested by master.
//If master sends data turn on/off signal for boost converter.
void requestEvent() {
  if (Wire.available() > 0){                    //Check if master sends data.
    byte c = Wire.read();                       //Read data sent by master.
    if (c == 1) {
     feedbackPID.SetMode(MANUAL);               //Turn regulator OFF if 1 is sent
         .
     Output = 0;
    }
    if (c==0) {
     feedbackPID.SetMode(AUTOMATIC);               //Turn regulator ON if 0 is
        sent.
    }
  }
  else{                                            //If no data is sent send CPM
   Wire.write(highByte(countPerMinute));           //byte by byte.
   Wire.write(lowByte(countPerMinute));
  }
}


//ISR Interrupt Service Routine countPulse.
void countPulse(){
  interruptTime = millis();               //Set time for interrupt, check for
     bouncing.
  if (interruptTime - lastInterrupt > bouncePreventTime){
  //Detach Interrupt from input pin.
  detachInterrupt(digitalPinToInterrupt(gm_input));
  count++;                                                  //Add counts.
  countDiode++;
  //Reattach interrupt.
   attachInterrupt(digitalPinToInterrupt(gm_input),countPulse,FALLING);
  lastInterrupt = interruptTime;                          //Update last
     interrupt time.
}
}

//Function controlling diodes.
void ledVar(int value){
  if (value > 0){                    //If value > 0 light that number of diodes
     .
    for(int i=0;i<value;i++){
      digitalWrite(ledArray[i],HIGH);
    }
    for(int i=3;i>value;i--){    //Turn off remaining diodes.
      digitalWrite(ledArray[i],LOW);
    }
  }
  else {                           //If value = 0 turn of all diodes.
    for(int i=3;i>=0;i--){
      digitalWrite(ledArray[i],LOW);
    }
  }
}
```

References

[1] http://rexusbexus.net

[2] http://lodestarbexus.se

[3] Y.I. Stozhkov et al. *Cosmic ray measurements in the atmosphere*, Lebedev Physical Institute, Russian Academy of Sciences, Moscow, Russia

[4] https://en.wikipedia.org/wiki/Peek%27s_law

[5] https://en.wikipedia.org/wiki/Paschen%27s_law

[6] https://w3.pppl.gov/scied/docs/Paschens_law.pdf

[7] Glenn F Knoll. *Detection and Measurement,* third edition 2000. John Wiley and sons

[8] S.N Vernov, I.A Savenko, M.V Teltsov, P.I Shavrin, *Investigation of the outer proton belt*, Proceedings of the 9th International Cosmic Ray Conference, Vol. 1

[9] http://www.gstube.com/data/4540/

[10] https://physics.nist.gov/cgi-bin/Star/ap_table.pl

[11] http://www.srim.org/SRIM/SRIMINTRO.htm

[12] https://en.wikipedia.org/wiki/Inductor

[13] http://www.learnabout-electronics.org/PSU/psu32.php

[14] https://www.pocketmagic.net/tube-sts-5-%d1%81tc-5-geiger-tube/

[15] http://www.nucleide.org/DDEP_WG/Nuclides/Am-241_tables.pdf

[16] https://en.wikipedia.org/wiki/Counting_efficiency