UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Dual band TPR radiometer for Snow Water Content determination

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Nicholas J.R.M. Albarracin Roberts

In partial fulfilment

of the requirements for the degree in

ELECTRONIC SYSTEMS ENGINEERING

Advisor: Dr. Albert Aguasca

Barcelona, February 2020

Abstract

This degree thesis is about being able to collect Brightness Temperature using a drone Radiometer. The project is designed to fix an ongoing radiometer project and calibrate it to be able to read small differences of temperature on the surface. The goal of the project is to make a flying Microwave Radiometer (MWR) and be able to calculate the Snow Water Equivalent (SWE) of the snow. The SWE is done by calculating the Brightness temperature of the snow by the difference of temperature in the two antennas. This thesis will explain the problem that the MWR originally had, the solution, the outcome and the struggles to get the Radiometer ready for flight. The project has had its last-minute hardware problems due to easily breakable parts and unscheduled inconveniences but the outcome is the following: For now, the MWR can read temperatures and distinguish differences between sky temperature/ Blackbody temperature and terrains.

Resum

El projecte que ens ocupa té com a principal objectiu l'ús d'una tecnologia mitjançant la qual es pugui mesurar la temperatura de la brillantor de les superfícies, col·locant-se aquest, alhora, en un dron. Es tracta d'una tesis que parteix d'un projecte de radiòmetre anterior i es pretén, de fet, poder calibrar-lo correctament i preparar-lo perquè pugui mesurar petites diferències de temperatura en la superfície. Mes concretament, el propòsit és configurar un MWR volador capaç de llegir l'equivalent d'aigua de neu (SWE) present en una superfície de neu; el SWE, cal afegir, s'identifica calculant la temperatura de la brillantor de la neu per la diferència de temperatura a les dues antenes. La tesis parteix de la identificació dels problemes en el projecte anterior amb l'objectiu, precisament, de resoldre'ls, fent-ho alhora funcional i preparant-lo pel vol. Tot i que el projecte ha patit problemes de hardware d'última hora, l'estat actual en què es troba és el següent: el radiòmetre és capaç de llegir temperatures de la brillantor, identificant les diferències entre la temperatura del cel, la temperatura d'un cos negre i la dels terrenys.

Resumen

El proyecto que nos ocupa tiene como principal objetivo el uso de una tecnología mediante la cual pueda medirse la temperatura de brillo de las superficies, colocándose éste, a su vez, en un dron. Se trata de una tesis que parte de un proyecto de radiómetro anterior y se persigue, de hecho, el calibrarlo correctamente y prepararlo para que pueda medir pequeñas diferencias de temperatura en la superficie. Más concretamente, el propósito es configurar un MWR volador capaz de leer el equivalente de agua de nieve (SWE) presente en una superficie de nieve; el SWE, cabe añadir, se identifica calculando la temperatura de brillo de la nieve por la diferencia de temperatura en las dos antenas. La tesis parte de la identificación de los problemas en el proyecto anterior con el objetivo, precisamente, de resolverlos, haciéndolo a la vez funcional y preparándolo para el vuelo. Si bien el proyecto ha topado con problemas de hardware en el último momento, el estado actual en el que se encuentra es el siguiente: el radiómetro es capaz de leer temperaturas de brillo, identificando las diferencias entre la temperatura del cielo, la temperatura de un cuerpo negro y de los terrenos.

I would like to dedicate this TFG to my family because their help on the last day made this thesis be successful.

Acknowledgements

I would like to express my deepest appreciation to my advisor, Dr. Albert Aguasca, who has been of great support throughout this project, offering advice and help when needed. Without his guidance and help this thesis would not have been possible.

I would also like to thank the staff at D3, from the IT department to the laboratory clerks. They were able to assist me in critical moments, such as recovering data from a dead PC to showing me practical skills like printing my own PCB or using the tools at hand.

Revision history and approval record

Revision	Date	Purpose
0	16/01/2020	Document creation
1	dd/mm/yyyy	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Nicholas J.A. Roberts	
Dr. Albert Aguasca	

Written by:		Reviewed and approved by:	
Date	16/01/2020	Date	dd/mm/yyyy
Name	Nicholas J.A. Roberts	Name	Dr. Albert Aguasca
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables:

1. Introduction

This project was an ongoing project which was brought to my intention by my tutor. He proposed the radiometer on a drone. The radiometer itself was already built but he mentioned that he could not stabilize the interior heat to be able to get a correct reading from the radiometer. The project's objective is to be able to study the snow water estimate (SWE) in the snow sheets on the surface. This radiometer would be the first fly born that would study the snow by drone. The radiometer works with two microwave receivers, one works at 18 GHz and the other at 37 GHz. The 37 GHz is used as a measurement frequency sensitive to snow grain volume scattering and the 18 GHz is a frequency that is insensitive to snow. The greater the difference between temperature brightness measurements at these two frequencies, the higher the estimate of SWE until a certain degree. [1]

The figure below shows the parts of the radiometer system.

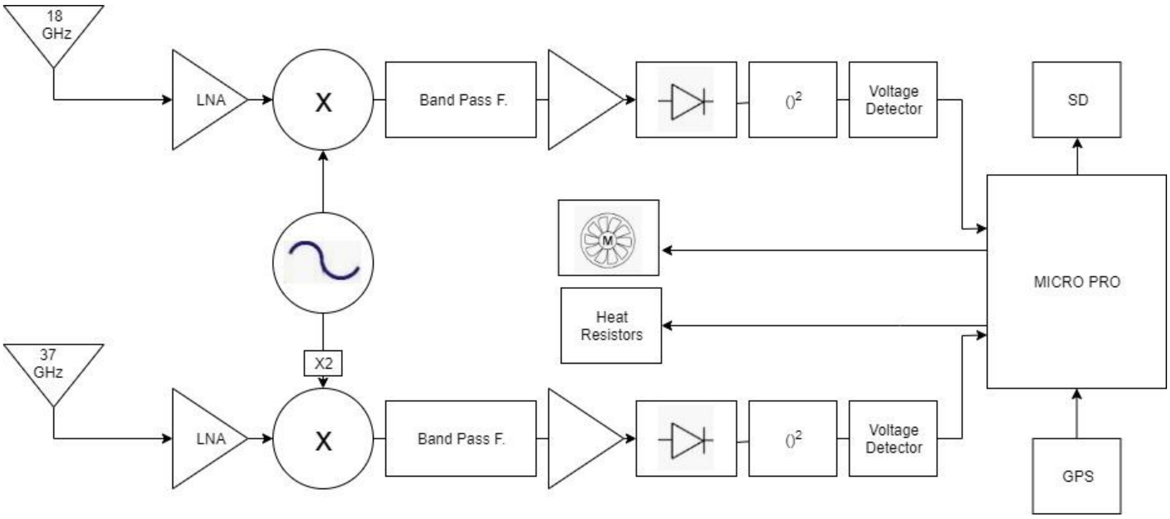


Figure 1. Block diagram of the Dual band Radiometer

The radiometer system needs to be supplied with two 15V power supplies. One of the power supplies is to supply the system (most components are supplied with 5V or 3.3V), the other is to power the heat sink resistors in the box. The interior has to be thermally stabilized around 45ºC because of the strong dependence of the radiometer's response to the internal temperature.

The box itself includes two antennas of 37 GHz and 18 GHz. The microwave signals from these antennas are then amplified by a pair of Low Noise Amplifiers (LNA). Then both signals are downconverted to 355 MHz (18GHz channel) and 1.307 MHz (37 GHz channel) by two mixers where the local oscillator is from an open loop frequency oscillator. Both signals then go through a bandpass filter and an IF amplifier, and finally go through a RMS true power detector to then go to the digitizer. The data is collected and stored in the SD Micro card of an ARDUINO pro micro module. While it stores the data, the radiometer saves its GPS location and it stores it with the data from the antennas. The ARDUINO also regulates the interior temperature controlling the heat sink resistors and an extracting fan. There is also another fan, which circulates the hot air inside the radiometer.

1.1. Parts of a radiometer

1.1.1. Low Noise Amplifier & DownConverter

Both Radiometers use an LNA after the signal goes through the antenna.

The 37 GHz uses a monolithic amplifier HMC1040 (from Analog Dev). This amplifier is very delicate when installed and can cause problems with the signal (which can be seen in section 1.5) so some special material is added around it to avoid oscillation and noise or wave interferences which could damage the signal.

The 18 GHz uses a Low Noise FET CE3520K3 (from NEC Semiconductors) which is not as delicate as the monolithic but special care has to be taken to avoid instabilities including absorbing material in its box.

Both signals are then downconverted using 2 drivers. HMC570LC5 for the 18 GHz and HMC6147 for the 37 GHz. Both downconverters use a Local Oscillator (f_{OL}) supplied by a frequency generator and a frequency power splitter. The frequency that goes to the 37 GHz is multiplied by 2 so that it can reach the desired frequency.

$$f_{FI18} = f_{RF} - 2 * f_{OL} = 355\ MHz$$

$$f_{FI37} = f_{RF} - 4 * f_{OL} = 1.307\ MHz$$

1.1.2. BandPass Filter

Once both signals have gone through the downconverters, a bandpass filter is then applied to both of them. BPF-A355+ (*Minicircuits*) for the 18 GHz and CBP-1307C+ (*Minicircuits*) for the 37 GHz. They are then amplified by a MAR-8ASM and an ERA-3SM (both are from *Minicircuits*).

1.1.3. RMS

In the RMS, the signal goes through a true power detector, where later the voltage is squared using an AD835 and that voltage is then detected through a voltage detector AD8361 (Analog Dev.). After that, the signal goes through the Arduino Micro Pro (passing through the ADC).

1.1.4. Temperature

The internal temperature of the radiometer system box is read by an LM35 which is situated close to the RMS detector. The heating system is made out of 10 heat sink resistors and 2 fans. One fan is always on to circulate the heat inside the box and the other is to extract air, the latter was installed mid-way through the project when it was deduced that a cooling method would ease the PI control.

1.1.5. GPS

The GPS is the Adafruit Ultimate GPS Breakout and it is connected to the Micro Pro.

1.1.6. Micro Pro

The Micro Pro is a small Arduino. This Arduino takes care of controlling the PI of the interior temperature, the signals being received by both antennas and the GPS coordinates which are all stored in the micro SD. This can be seen in figure 2.

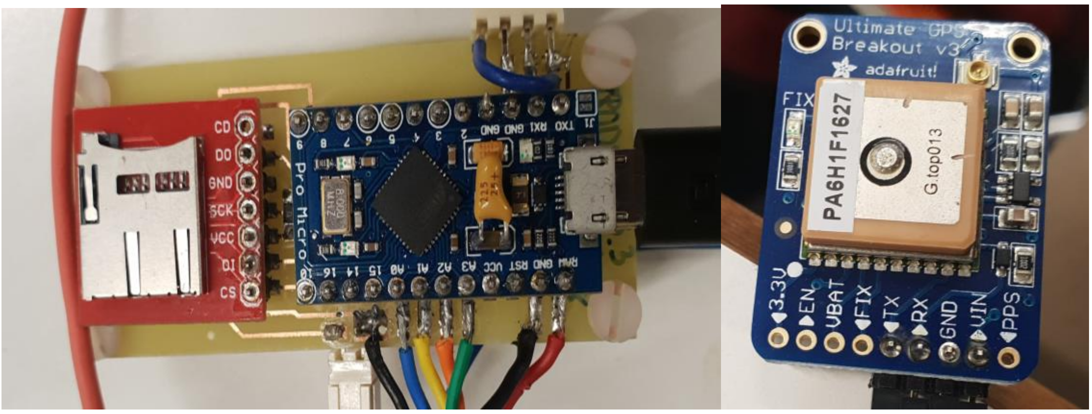


Figure 2. Circuit board

As seen below, it is the board which has the input/outputs of the Micro Pro. These contain the power supply at 3.3V, the analogue to digital converter, the heating and the GPS.

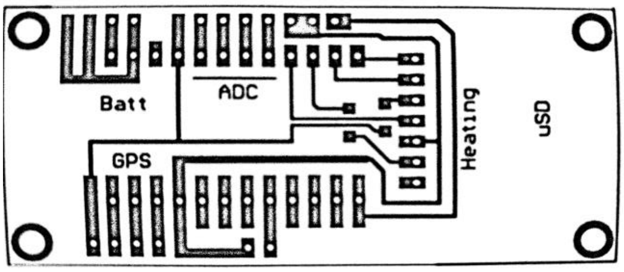


Figure 3. Schematic of the board

1.2. <u>Procedures</u>

1.2.1. PI temperature controller.

In mid-October (the week of the 13th), with trial and error, it was learnt that the thermal PI control inside the Radiometer could only be well programmed by using a cooling method. Without it, the results were not as desired, as it took too long to stabilize.

On the following week, although the PI worked perfectly, after doing some tests on the roof, it was discovered that applying a fan directly in front of the radiometer changed the outcome of the radiometer because of the cooling factor of the fan to the antenna. Therefore, the walls of the box were covered with absorbing material to be able to conceal the heat from the exterior. The antennas were covered with Tefflon to protect them from the wind.

As the two following figures show, the improvement was a success.

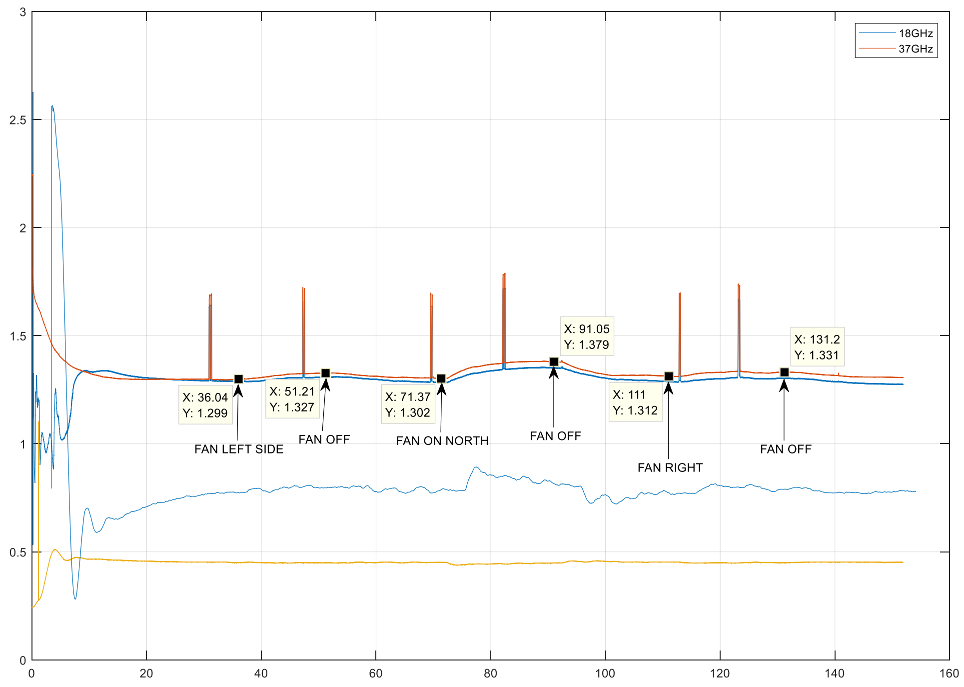


Figure 4. MWR without protected walls

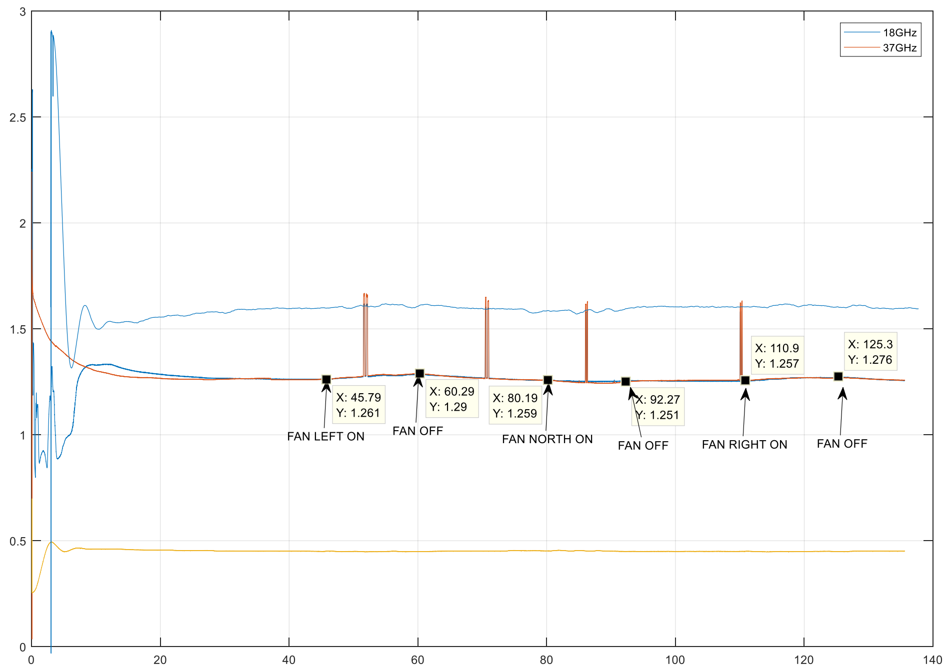


Figure 5. MWR with protected walls and covered antenna.

Each experiment carried out is left outdoors, on the roof of D3. A blackbody is put above the radiometer to make check points. Once finished, the data stored is processed by a MATLAB program which delivers a graph of the output from both antennas and 2 different types of interior temperature (one temperature reading being more accurate than the other).

Although most PIDs use the Zeiger-Nicole method, it was adopted but unsuccesful as the outcome gave an oscillating PI. After using trial and error, the PI was modified to have a 0.1ºC error difference.

1.2.2. Tipping Curve

Sky tipping Curve is the most important part of the radiometer because this is the method on how the voltage outcome of the radiometer can be transformed to brightness temperature. The details can be seen in 2.1.

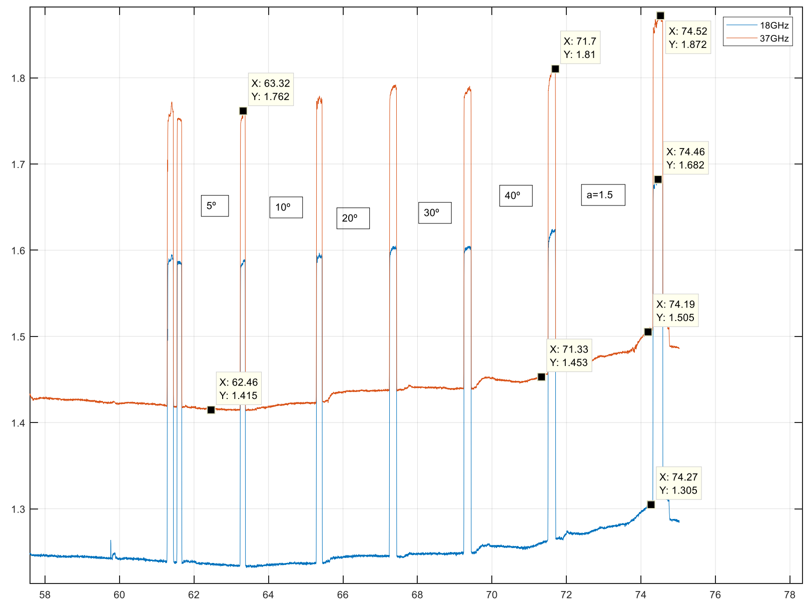


Figure 6. Inclination test

As seen in the figure above, a test was done by changing the angle of the radiometer until it reached air mass 1.5. As seen, a difference in temperature exists. However, a problem occurred with the value of the blackbody when placed. It was deduced that because it wasn't parallel to the radiometer, its value changed.

Once the inclination test was done, the sky-tipping curve was calculated but the results showed that the radiometer was not sensitive enough.

1.2.3. Drone Box

The radiometer had to be put on the drone to get it ready for flying. The best way to do it was to use a plastic box and a lid to secure the radiometer inside. But a problem occurred when placing the MWR in the box, the MWR did not get the desired readings. A test run was carried out and as seen in the figure below, after taking the lid off, the radiometer worked perfectly. However, with the lid, there were a lot of interferences.

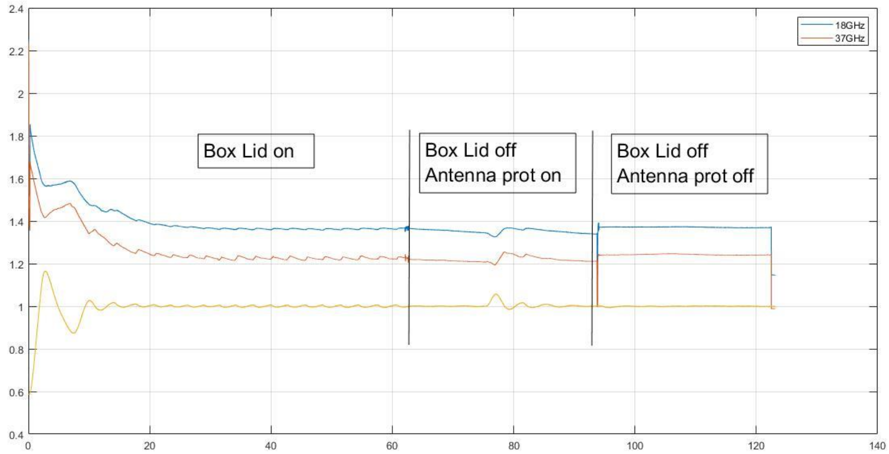


Figure 7. Box Lid experiment

Once the interference problem was solved, another problem arose. When the radiometer was moved, the LNA inside of the 18 GHz started giving problems due to a bad solder. When fixed, the LNA then started oscillating. This was also fixed but after another movement experiment, the 37 GHz started giving problems too. These problems can be seen in the following figure.

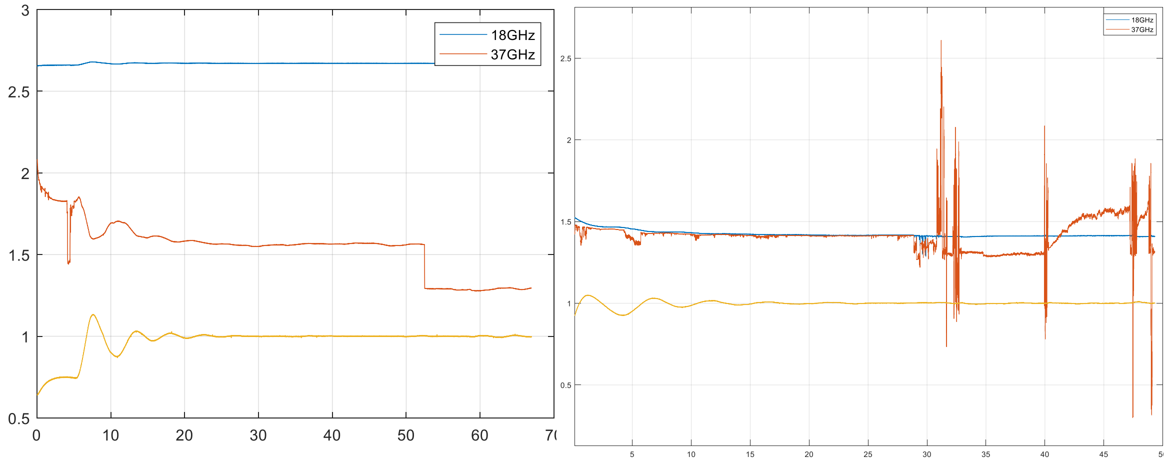


Figure 8. Left: 18 GHz LNA being saturated. Right: 37 GHz problems when moved

1.3. Milestones

- The 1st milestone was to understand the function of a radiometer and how it is used to calculate the SWE. In parallel with that, study the Arduino code used.
- The 2nd milestone was to fix the Arduino code and implement the PI properly with a cooler.
- The 3rd Milestone was to obtain a good reading and be able to get the sky tipping curve.
- 4th was securing the radiometer onto the drone.
- 5th Testing the radiometer on different surfaces.

1.3.1. Actual Milestones

- Studying the radiometer theory and radiometer code.
- Fix the PI thermometer.
  - Fridge experiment
  - Roof test with fan
  - Fridge with absorbing material
  - Roof test with fan
  - Fridge experiment with new fan
- Repair the 18GHz LNA that fell and broke.
- Sky tipping curve calibration
- Make a box to secure the radiometer on the drone.
- Fix the output so it works correctly.
- Fix the 18GHz LNA again because of a solder problem.
- Find the reason for the 37 GHz faulty readings.

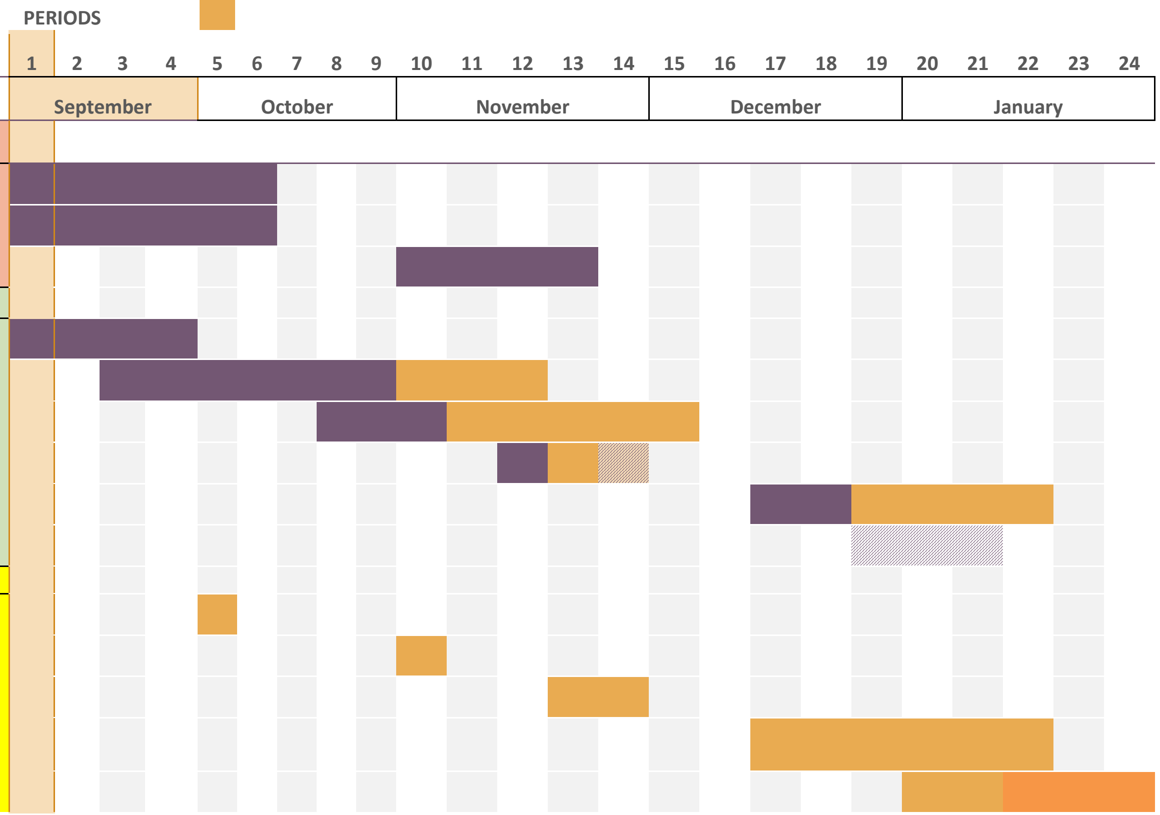
1.4. Gantt chart

Fall Semester 2019

Period Highlight: 1

Plan Duration | Actual Start | % Complete | Actual (beyond plan) | % Complete (beyond plan)

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
RESEARCH					
SWE	1	6	1	6	**100%**
Radiometer	1	6	1	6	**100%**
Tipping Curve	10	4	10	4	**100%**
Tasks					
Studying Code	1	4	1	4	**100%**
Doing PI	3	7	3	10	**100%**
Roof Tests	8	3	8	8	**100%**
Sky tipping Curve	12	1	12	3	**75%**
Drone Box	17	2	17	6	**100%**
Flight	19	3	0	0	**0%**
Incidences					
Broken Micro Pro	0	0	5	1	**1%**
Broken Micro Pro	0	0	10	1	**80%**
Broken LNA	0	0	13	2	**100%**
Box Calibration / Heat sinks	0	0	17	6	**100%**
Outdoor incident	0	0	20	4	**100%**



Gantt chart

As it can be seen in the Gantt chart above, if everything had gone to plan, the project would have ended in December and it would have had a flight test with the radiometer but because of the incidences in Section 1.5, those plans could not be carried out.

There were many setbacks: trying to adjust the Radiometer to become stable in perfect conditions, took longer than expected; from the start, the environment surroundings were not taken into consideration when calibrating the PI, it took time just to calibrate one radiometer but changing its interior led to another calibration and that meant more time spent on the same experiment. Therefore, a lot of time was wasted configuring the PI. Also, the fall of the radiometer set the project back a further week, as the broken micro pro and antenna had to be replaced/repaired. The night experiment gave a further delay to the progress of the project and then, the LNA situation occurred.

1.5. Incidences

Many incidences occurred during this project with the hardware. Due to the micro port on the Micro Pro being very sensitive, 3 of the Micro Pros had to be replaced.

In November, the radiometer fell to the floor. That broke the 18 GHz antenna and its LNA did not amplify as it should. The LNA was fixed by cleaning its box and adding some absorbing material so no jitters could damage the signal by oscillating. As shown in the next figure.

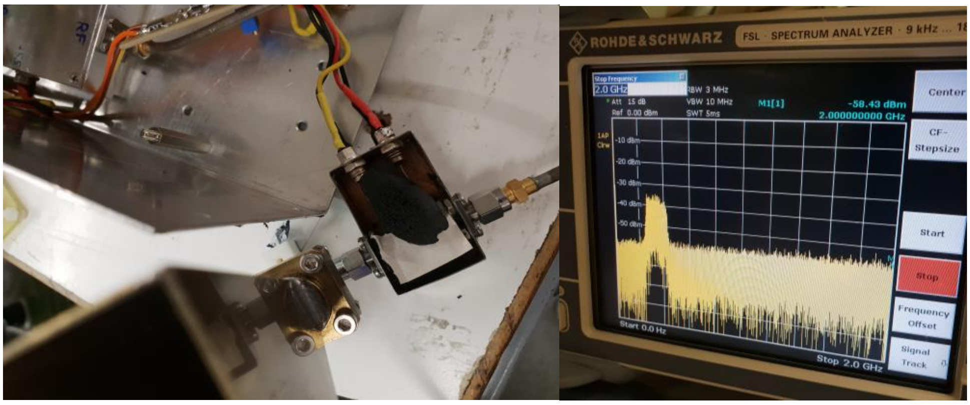


Figure 9. Fixing the LNA

In December, because the heating inside the box was too slow, new heat sink resistors were installed to quicken the heating process. 2 were placed on the antenna and the other were put near the LNA.

In January, during a night experiment, the board with the micro pro, the SD holder and the Micro SD got damaged because of the extreme humidity. Therefore, a new set of it all had to be made, verified and tested. The Micro SD card holder was not working properly because it made preposterous folders as seen in figure 10 and after solving that, the Micro itself did not save any of the data in the Micro SD.

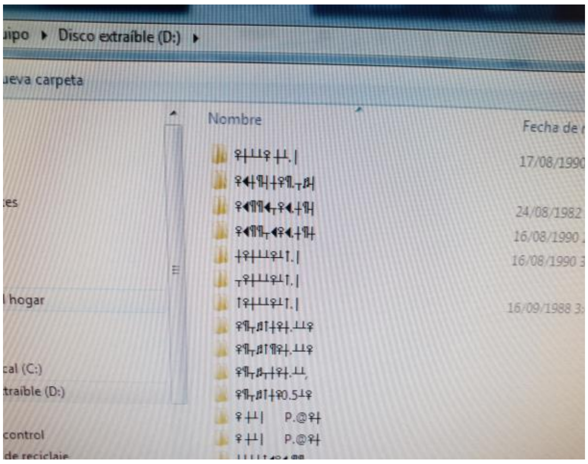


Figure 10. Corrupted Folders.

As seen in figure 8 in the procedures, on the last week of the deadline, both antennas malfunctioned with no time to fix both of them, the 18 GHz was able to be fixed by repairing the solder between the input of the signal and the board and to the stop the oscillating, some weight onto the LNA was added.

2. State of the art of the technology used or applied in this thesis:

2.1. Radiometer

A microwave radiometer is an instrument that measures the temperature of brightness, that is, the power emitted by a body per unit of solid and surface angle with high precision and resolution. The key parameter for observing this energy is the so-called brightness temperature and it is a term that represents the intensity of electromagnetic radiation emitted by the scene under observation. If an antenna points to a body, the power obtained at its output (expressed in terms of antenna temperature) is related to the brightness temperature of this body. A radiometer measures the power of the antenna delivered to the receiver.

2.1.1. TPR (Total Power Radiometer)

The TPR is the most used radiometer in the market. This is due to the simplicity of its topology and its high resolution. The block diagram of a radiometer consists of an antenna, a super heterodyne receiver which transfers the radio frequency (RF) signal to an intermediate frequency (FI), where a power detector and a low-pass filter perform the power measurement.

A low-pass filter is used as an integrator to increase the stability of the measurement. Therefore, the longer the integration time, the more stable the radiometer output would be. The radiometer output is a voltage that has two components, one of direct current V_{dc} corresponding to the average value of the input power, and one component of alternating current V_{ac}.

2.1.2. Dicke

The Dicke corrects the stability problems associated with the profit fluctuations that exist in the TPR. The Dicke radiometer (DR), instead of continuously measuring the antenna temperature such as TPR, it measures the temperature from the antenna for a certain time and then it measures a known reference temperature. With this method, thermal instability of the noise is filtered and the impact of gain variations is greatly reduced. Although the Dicke radiometer is more stable, its radiometric resolution worsens a factor of 2 with respect to the TPR.

2.1.3. NIR Noise Injection Radiometer

Noise Injection Radiometer (NIR) is a particular case of the Dicke radiometer; it is an optimized version so that its output is independent of both the gain fluctuations and the noise temperature of the receiver itself. To achieve this, the structure of an NIR is based on the feedback network. Its purpose is to balance the radiometer (as in the balanced Dicke radiometer) by injecting noise into the antenna line through a directional coupler.

2.2. Snow Water Estimate (SWE)

There are two types of snow, dry snow and wet snow. Dry snow is a dielectric medium consisting of ice-crystals in an air background, wet snow on the other hand is a mixture of ice-particles, water droplets and air.

In dry snow, the snow surface backscattering contribution may be neglected due to the dielectric mismatch at the air-snow boundary being small. It has been studied that the backscattering coefficient at 17 GHz gets saturated after a 40cm depth with a result of -2 dB and at 35 GHz, the backscattering saturates at 3dB after the 40cm.

With wet snow, the upper snow boundary is treated as a rough surface (compared to dry snow which is regarded as a planar interface). Wet snow causes the permittivity of the snow layer to increase and because of being water in the snow, there is an increase in the dielectric loss factor of the layer.

When studying the snow, it has been studied that with an incidence angle bigger than 20º, it is easy to discriminate areas covered with dry snow and areas with wet snow because of the large difference between the backscattering contribution. At a low frequency of 10 GHz, there is little sensitivity to the liquid water content but at 37 GHz, there can be a difference of 8 dB of backscattering difference between wet and dry snow. Melt water in the snowpack can raise the microwave brightness temperature since water droplets emit rather than scatter microwave radiation.

As seen in the figure below, an antenna of 37 GHz is more sensitive to read the SWE compared to an 18 GHz antenna. Therefore, the 18 GHz antenna acts like a reference of the soil background and snow because then once compared to the 37 GHz, it can read the SWE of the snow. [2] [3] [4]

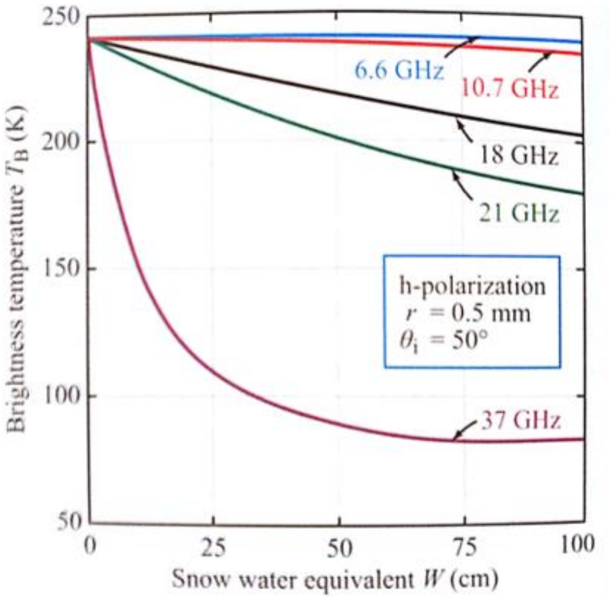


Figure 11. SWE with T_B

Furthermore, observation of the brightness temperature differences $\Delta T_B = T_{B18} - T_{B37}$ leads to the next figure that shows relation between the difference and the levels of grain diameter. Therefore, it could be said that SWE is $W = C_0(T_{B18} - T_{B37})$. If T_{B18} is less than the 37 GHz channel, then the snow depth and SWE is set to 0. To derive snow depth, SWE is simply divided by the snow density (300 kg m^{-3} is a common value with mature mid-winter snow packs in North America, therefore, c= 1.6 cm K^{-1}). [1]

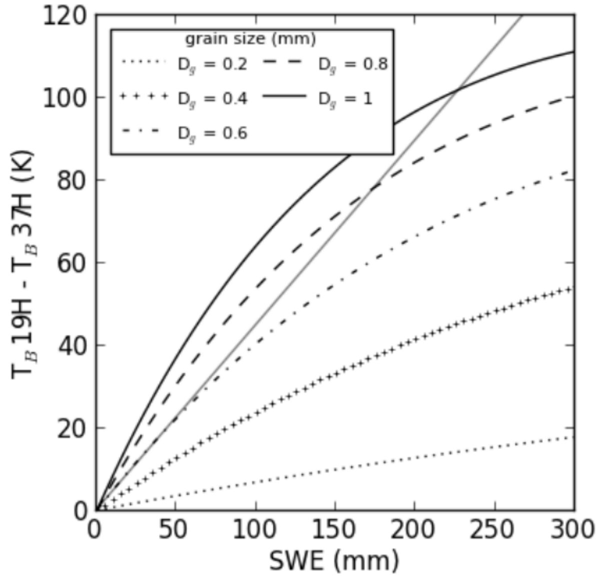


Figure 12. Brightness temperature difference as a function of SWE

2.3. PID Control

Proportional-integral-derivative (PID) controllers are used in industrial control systems because of the reduced number of parameters to be tuned. They provide control signals that are proportional to the error between the reference signal and the actual output (proportional action), to the integral of the error (integral action), and to the derivative of the error (derivative action).

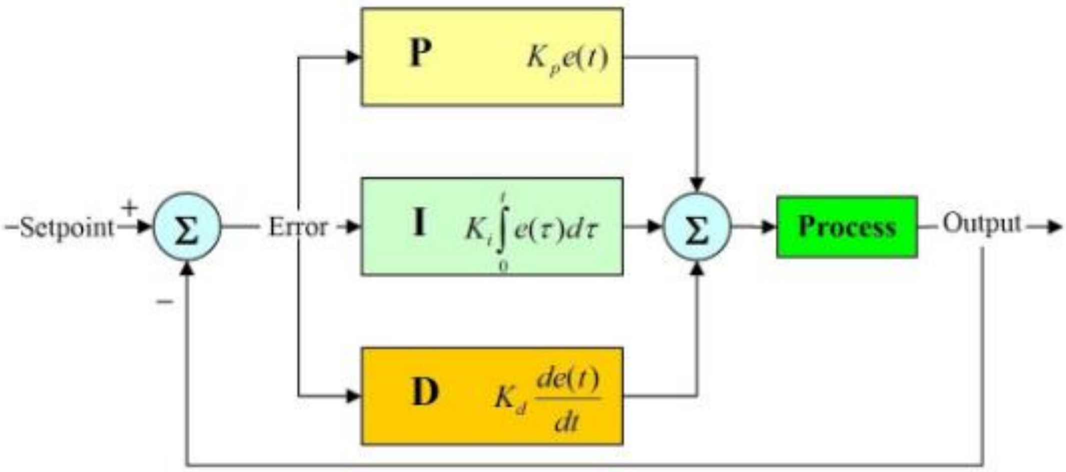


Figure 13. PID schematic

Therefore, the equation is as follows. $Out(t) = k_p e(t) + k_i \int_0^t e(\tau)d\tau + k_d \frac{de(t)}{dt}$. For a heating procedure, only PI is necessary because the task at hand is easy which means that the derivative action is not needed and the speed for it to stabilize is not that critical. Proportional Action: The P-action is the component mostly relevant to the dominant response of the system. Increasing the P gain K_p typically leads to shorter rise times, but also larger overshoots. Although it can decrease the settling time of the system, it can also lead to highly oscillatory or unstable behaviour. Integral Action: The integral action is typically employed to optimize the steady-state response of the system and shape its dynamic behaviour. Essentially, it brings memory to the system. Increasing the I gain K_i, leads to reduction of the steady-state error (often elimination) but also more and larger oscillations. [5]

CL Response	Rise Time	Overshoot	Settling Time	Steady-State Error
KP	Decrease	Increase	Small change	Decrease
KI	Decrease	Increase	Increase	Eliminate
KD	Small change	Decrease	Decrease	No change

Table 2. Summary of tuning tendencies

As seen in the table, if the values of K_p and K_i are increased too much, the system would overshoot, which in other terms would mean it would oscillate in time.

A quick way to calibrate the PI controller is using the Ziegler Nichols tuning method. It is performed by setting the K_i to zero. K_p is then increased (from zero) until it reaches the ultimate gain K_u, which is the largest gain at which the output of the control loop has stable and consistent oscillations; higher gains than the ultimate gain K_u have diverging oscillation. K_u and the oscillation period T_u are then used to set the P and I gains depending on the type of controller used and behaviour desired. The table underneath shows the method on how to do the Ziegler Nichols tuning method. This method was used but it made the PI oscillate so trial and error was used.

Control Type	K_p	T_I	T_D	K_I	K_D
P	$0.5\ K_u$	-	-	-	-
PI	$0.45\ K_u$	$\frac{T_u}{1.2}$	-	$\frac{0.54k_u}{T_u}$	$\frac{k_uT_u}{10}$
Classic PID	$0.6\ K_u$	$\frac{T_u}{2}$	$\frac{T_u}{8}$	$\frac{1.2k_u}{T_u}$	$\frac{3k_uT_u}{40}$
Pessen integrate Rule	$\frac{7k_u}{10}$	$\frac{2T_u}{5}$	$\frac{3T_u}{20}$	$\frac{1.75k_u}{T_u}$	$\frac{21k_uT_u}{200}$

Table 3. Ziegler Nichols tuning method

Trial and error meant finding Kp until it oscillated and it was then halved to incorporate the Ki.

2.4. Tipping Curve

Two calibration methods are often applied for radiometers: using a Hot-Cold calibration using liquid nitrogen LN_2 or a tipping calibration method that uses a clear atmosphere as a calibration reference. For the hot-cold calibration, two targets with well-known and constant, although strongly differing, temperatures are needed. One known interior hot target consisting of a blackbody load with a certain temperature and another blackbody on the outside filled with LN_2, an additional noise is injected to the system from an internal noise diode to calibrate the radiometer. The tipping calibration is used in this project due to not having a known interior blackbody nor a noise diode.

The radiometer takes measurements at two or more elevation angles in a horizontally-stratified atmosphere. The calibration is accomplished by adjusting a single numerical calibration parameter that is required by the system software until the outputs of the system comply with a known physical relationship. The aim of a tipping curve is to determine the opacity. With this information it is possible to determine the brightness temperature of the sky at zenith direction which can be used as a cold calibration target. The theoretical method it as follows:

Starting from the radiative transfer equation in the microwave region under the assumption of a one layered atmosphere with a mean temperature T_m, the brightness temperature under a zenith angle θ is given by:

$$T_b(\theta) = T_0 e^{-\tau(\theta)} + (1 - e^{-\tau(\theta)})Tm$$

T_0 is the cosmic background radiation (= 2.7 K). In addition to the sky measurement the hot calibration target is measured (U_{hot} is the Blackbody). As cold calibration target (U_{cold}) the sky at a high elevation angle is measured. The opacity can be found by iteration. The iteration starts with an initial estimation of the opacity τ_i (e.g. τ_i = 0.3). The iteration process works as follows:

- $T_{b,cold}$ is calculated using equation (1) and inserting τ_i.
- For zenith angles θ the tipping measurements are calibrated:
  - $T_{b,\theta} = \frac{U_\theta - U_{hot}}{U_{hot} - U_{cold}}(T_{hot} - T_{b,cold}) + T_{hot}$
- With the calibrated measurements the opacities can be calculated solving the first equation.
  - $\tau_\theta = \ln\left(\frac{T_m - T_0}{T_m - T_{b,\theta}}\right)$
- The slope of a linear regression of the opacities τ_θ against the airmass factors is equal to the new zenith opacity τ since it airmass A(90º) = 1. τ is taken as the new τ_i.

The iteration steps are repeated until the offset of the linear regression is smaller than 10^{-2}. The linear fit has to pass through the origin because the opacity for zero airmass has to be zero. The zenith opacity found by iteration is used to calculate the brightness temperature of the cold calibration load $T_{b,cold}$. With the brightness temperatures of the hot and the cold calibration load a hot-cold calibration can be performed. [4][6]

Because this procedure gave an erroneous result, a simpler approach was taken.

- Assuming that pointing to the sky (zenith), the antenna temperature is,

  $$T_{zenith} = T_{cosmic} + T_{atmosphere}$$

  where T_{cosmic} is 2.7K and T_{atmos} is unknown.
- If the radiometer points towards the sky with a tilt of 60º then:

$T_{60^0} = T_{cosmic} + 2 \cdot T_{atmosphere}$

- The output of the radiometer has the following relationship to $T_{antenna}$:

  $T_{rad} = a + b \cdot T_{antenna}$ where a and b are unkowns.

- Therefore, only with three different measurements, T_{hot} (pointing the radiometer to the blackbody = 300K), T_{zenith}, and T_{60^o}, the sky-tipping curve can be calculated.

  $T_{rad,hot} = a + b \cdot T_{hot}$

  $T_{rad,zen} = a + b \cdot (T_{cosmic} + T_{atmos})$

  $T_{rad,60^0} = a + b \cdot (T_{cosmic} + 2 \cdot T_{atmos})$

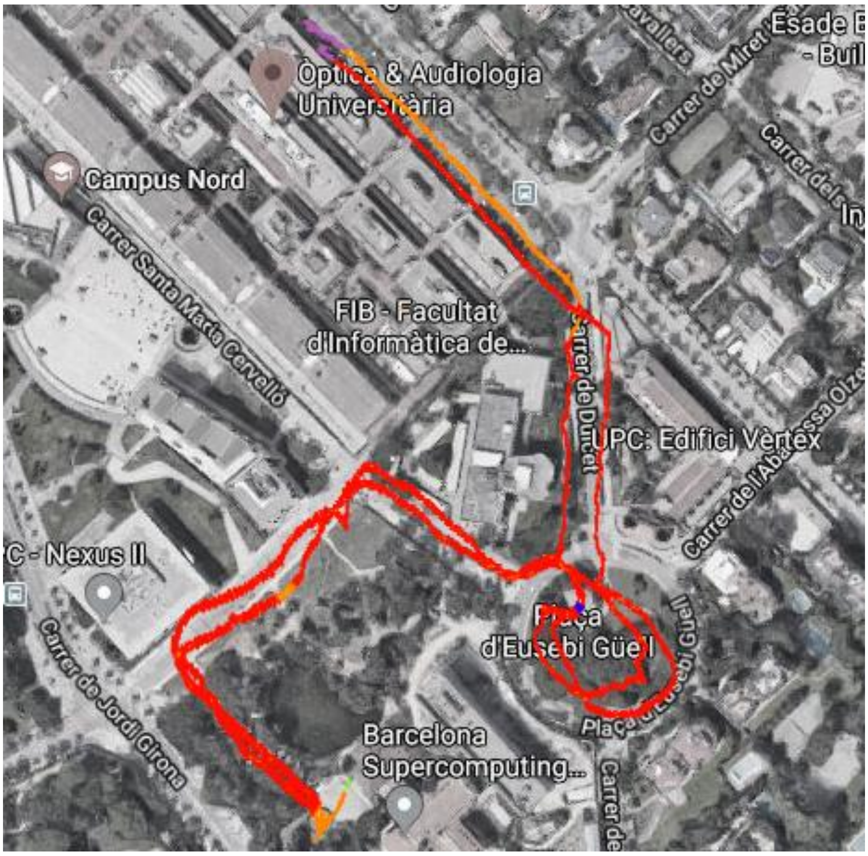
3. Methodology / Project development:

In the software methodology, the Arduino programme is used to implement C++ to the micro Pro. In the Annex section, the code used on the radiometer can be found.

Matlab is used to be able to plot the data stored from the MWR. Sky calibration is used to calibrate the brightness temperature received by the MWR. The Ziegler-Nichols method was used to try and calibrate the PID inside the MWR.

4. Results

Due to the problems with the LNA and the two antennas giving incorrect results, the project did not work. However, by a stroke of luck, the system worked for a demo run on the final day. In the figure, there is a clear difference between sky, water and ground brightness temperature and small differences between the ground surfaces (cement, grass and earth). Overall, the system works; the radiometer reads, GPS tracks the route and the data is stored correctly. However, it needs to be perfected.



Colour	Voltage average
Red	1.65 V
Orange	1.6 V
Green	1.55 V
Blue	1.45 V
Purple	1.25 V

Figure 14. Map of the radiometer. Table 4. Route Colours

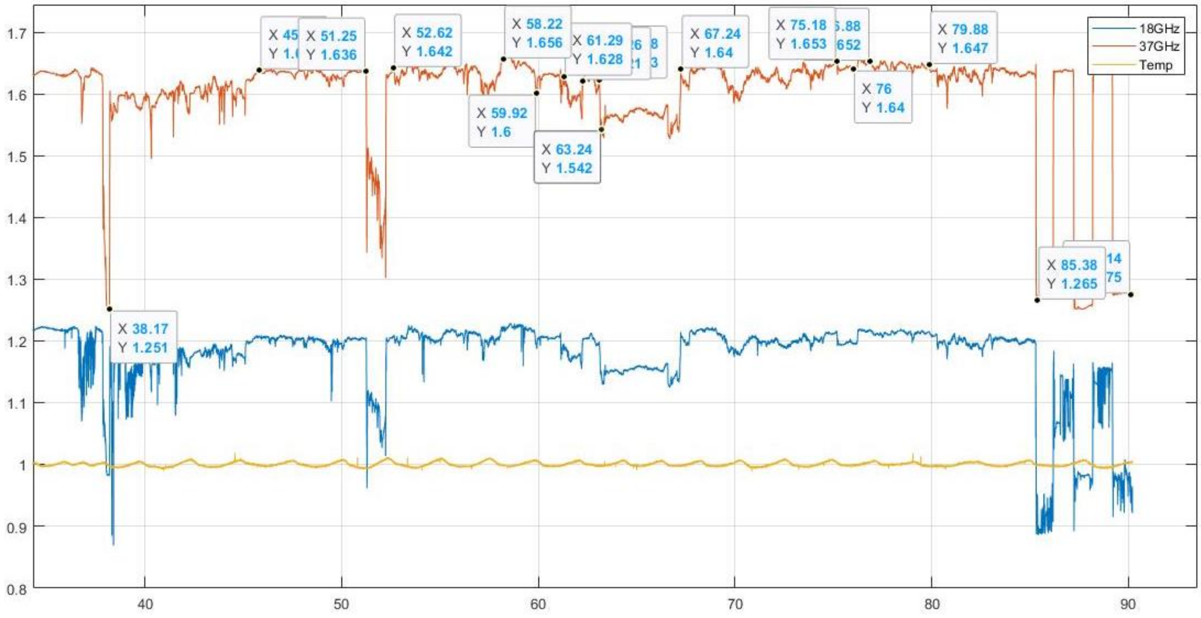


Figure 15. Radiometer route results

Time	Place
38	Sky
39	Shaded Road
41	Sunny Road
43	Shaded Road
45	Sunny Road
46	Grass
47	Shaded Grass
49	Red cement
51	Earth
52	Sky
53	Earth
53.5	Cement
58	Shadow route
61	Pebbles
62	Small Pond
63	Pebbles
64	Big Pond
68	Road
76	Earth
77	Red Cement
80	Grass
81	Cement
86	Sky x3

Table 5. Route Schedule

As seen in the graph, the first reads are quite noisy until the 45th minute mark but after that, small sensitive differences can be read with the surface terrains. The difference in ground and cement can be seen (49 - 50min & 75 – 76 min) and cement and ground at 80min. The 18 GHz is sensitive with rough movements (swinging the radiometer from ground to sky) but has more accurate reads. It makes sense because the 18 GHz does not focus on the small differences unlike the 37 GHz which does focus on them.

5. Budget

Devices	**Cost:**	**Web Page:**
Drone DJI S1000	1.700,00 €	https://www.google.com/search?q=DJI+S1000&rlz=1C1GCEU_esES872ES872&oq=DJI+S1000
LNA 37 GHZ HMC1040	52,00 €	https://www.mouser.es/
LNA CE3520K3	2,00 €	https://www.mouser.es/
DOWNCONVERTER HMC570	40,00 €	https://www.mouser.es/
DOWNCONVERTER HMC6147A	51,00 €	https://www.mouser.es/
Frequency Power Splitter ZX10-2-98-S+	40,00 €	https://www.minicircuits.com/WebStore/dashboard.html?model=ZX10-2-98-S%2B
BP Filter BPF-A355+	35,00 €	https://www.minicircuits.com/WebStore/dashboard.html?model=BPF-A355%2B
BP Filter CBP1307C+	30,00 €	https://www.minicircuits.com/WebStore/dashboard.html?model=CBP-1307C%2B
Amplifier MAR 8ASM+	1,25 €	https://www.minicircuits.com/WebStore/dashboard.html?model=MAR-8ASM%2B
Quadrant Multiplier AD835	17,89 €	https://www.mouser.es/Semiconductors/Logic-ICs/Multipliers-Dividers/_/N-4s6ga?Keyword=AD835&FS=True
Voltage Detector AD8361	7,70 €	https://www.mouser.es/Search/Refine?Keyword=AD8361
Temp Sensor LM35	0,35 €	https://www.mouser.es/_/?Keyword=LM35&FS=True
Antenna		
Metallic Box	30,00 €	
ikea plastic box	5,00 €	https://www.ikea.com/es/es/search/products/?q=plastico%20Caja%20transparente
Micro Pro	20 €	https://www.sparkfun.com/products/12640
Heat sink resistors	10€	https://www.aliexpress.com/w/wholesale-heat-sink-resistor.html
SparkFun Level Shifting microSD Breakout	5,50 €	https://www.sparkfun.com/products/13743
Adafruit Ultimate GPS	40 €	https://www.adafruit.com/product/746
Total:	**2.077,69 €**	

6. Environment Impact

Global warming has become a big concern, now more than ever. This project can demonstrate that global warming is true and how it is damaging the Earth because information about snow water equivalent is needed in applications such as flood forecasting, controlling the water level of power plant reservoirs, planning for forestry and crop irrigation and as input and control variable for many environment research purposes, including climate change research.

If perfected, this MWR drone would be the first of its kind and it would be able to study the SWE in real time and may help warn about the next climate disaster.

.

7. Conclusions and future development:

This project has further work to be done before it can be finished and therefore do what it was set out to do, i.e., find the SWE of the snow.

However, from working on this project I have learnt a valuable lesson, that all the parts given should be checked or rebuilt before doing anything else.

Future development would be to redo the interior of the box to make the heating process more even inside the radiometer, improve the cooling distribution, secure everything inside the box to make it fall-proof and change the Micro Pro for something less fragile when programming.

Bibliography:

[1] Foster, James & Sun, Chaojiao & Walker, Jeffrey & Kelly, Richard & Chang, Alfred & Dong, Jiarui & Powell, Hugh. (2005). Quantifying the uncertainty in passive microwave snow water equivalent observations. Remote Sensing of Environment. 94. 187-203. 10.1016/j.rse.2004.09.012

[2] Ulaby, Fawwaz & Long, David & Blackwell, William & Elachi, Charles & Fung, Adrian & Ruf, Christopher & Sarabandi, K. & Zyl, Jakob & Zebker, Howard. (2014). Microwave Radar and Radiometric Remote Sensing.

[3] Richardson, Mark & Davenport, Ian & Gurney, Robert. (2014). Global Snow Mass Measurements and the Effect of Stratigraphic Detail on Inversion of Microwave Brightness Temperatures. Surveys in Geophysics. 35. 10.1007/s10712-013-9263-x.

[4] Han, Yong & Westwater, Ed. (2000). Analysis and improvement of tipping calibration for ground-based microwave radiometers. Geoscience and Remote Sensing, IEEE Transactions on. 38. 1260 - 1276. 10.1109/36.843018.

[5] “A Quick Introduction to PID Control ”https://www.autonomousrobotslab.com/pid-control.html

[6] Microwave – Radiometer https://www.physik.unibe.ch/e41821/e41822/e140946/e148625/e270487/files270500/Mikrowellenradiometer_ger.pdf

Appendices:

Arduino Code for the Radiometer.

Main

```
#define USB_output 0
#define SD_card_output 1

#include <SD.h>
const int chipSelect = 10;
uint32_t syncTime = 0;
uint32_t syncTimeADC = 0;
uint32_t Past_Time = 0;
uint32_t Past_Time2= 0;
uint32_t Time_init_heat = 0;
uint32_t Time_actual = 0;
uint32_t Time_activacio = 0;
uint32_t Time_fan = 0;
float Time_activacio_f = 0;
float pTerm=1.0;
float iTerm=0.0;
float pOut=0.0;
float iOut=0.0;
float iOut2=0.0;
float iOut_ant=0.0;
float percen=0;
float dT = 2;
float kpp = 0.9;        //0.9 recent tests
float kii = 0.08;       //0.02 recent tests
int windUp = 600;
int enable_ctr_term=0;    //DELETE ITS A TIMER FOR
EXPERIMENTS
int enable_integr=1;
const int num_read = 10;  //array of 10
int reading[num_read];
int counter_mean=0;
int enable_mean=0;
String string_out2;  //SD CONTROL_TERMIC
String string_out4;
String string_out3;
char buffer_st[4];
int test=0;

uint32_t Time_anterior = 0;
int
Temp_actual,DTemp,DTemp_ant,DTemp_n,Temp_ac_
m;
float T_mean;
int ErrSum=0;
int Past_Err=0;
int Periode_Heat = 2000;
const int Temp_consigna= 8937; // Valor = Temperatura
x 198.6, 8000 = 40.28º 45º=8937
// cada grau son 198.6 comptes
const int graus1=198;
const int graus5=993;
const int graus10=1986;
int NO_ADC = 1;
int ACTIVE =0;
// per tant 5 graus  = 993 comptes
// per tant 10 graus  = 1986 comptes

File dataFile;

#define  ADS8344_CLK  21 // verd
#define  ADS8344_CS   20 // Taronja
#define  ADS8344_DIN   19 // Groc linia entrada al xip
ADS
#define  ADS8344_DOUT 18 // Blau  Dout del xip ADS,
per tant entra a l'arduino
#define  uSD_CS      10
#define  PinHeat      9 //Control Mosfet caldeig
#define  PinFan      5 //Fan
#define  PinLed      8 //Control LED
const int canal_0 = 0x8F;
const int canal_1 = 0xCF;
const int canal_2 = 0x9F;// actiu
const int canal_3 = 0xDF;// actiu
```

```
const int canal_4 = 0xAF;
const int canal_5 = 0xEF;
const int canal_6 = 0xBF;
const int canal_7 = 0xFF;
char buffer_gps[110];
int puntero_gps = -1;
char str1[6] = "     ";
char GPGGA[6] = "GPGGA";
int AD_counter = 0;
void setup() {
  // pinout of ADS8344
  pinMode(ADS8344_CLK, OUTPUT);
  pinMode(ADS8344_CS, OUTPUT);
  pinMode(ADS8344_DIN, OUTPUT);
  pinMode(ADS8344_DOUT, INPUT);
  pinMode(PinHeat, OUTPUT);
    digitalWrite(PinHeat,LOW);//aturar bit de caldeig   --Set to low
  digitalWrite(PinLed,LOW);//aturar bit de caldeig



  Serial1.begin(9600);  // GPS Serial Port

  // initialize the SD card
  pinMode(chipSelect, OUTPUT);
  if (!SD.begin(uSD_CS)) {
   //Serial1.println("Card failed, or not present");
    return;
  }

  // create a new file
  char name[] = "LOG00.TXT";
  for (uint8_t i = 0; i < 100; i++) {
    name[3] = i / 10 + '0';
    name[4] = i % 10 + '0';
    if ( ! SD.exists(name) ) break;
  }
  delay(1000);
  dataFile = SD.open(name, O_CREAT | O_WRITE);
  //Serial.print("Writing to: ");
  //Serial.println(name);
  // digitalWrite(LED,HIGH);

  syncTime=millis();   //time program started
  syncTimeADC=syncTime;
  Time_init_heat=syncTime;
  Time_activacio=0;

Serial1.print("$PMTK314,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*29"); //nomes GGA
  delay(1000);
  Temp_actual=leer_ADS8344(canal_1);
  DTemp_ant=Temp_consigna-Temp_actual;
}

void loop() {
  // GPS reading and processing
  while (Serial1.available()) {
    int inByte = Serial1.read();
    procesa_nmea(inByte);
  }
  procesa_ADC();

}
```

LECTURA ADS8344

```
void procesa_ADC()
{
 String string_out,str_AD,str_millis,channel2,channel3,channel1;
Time_actual=millis();
str_millis=String(Time_actual);
 if((millis()-syncTimeADC) > 10){ //temps entre lectures minim
 syncTimeADC=millis();
string_out=String("#RD,");
string_out.concat(str_millis);
string_out.concat(",");

// llegir CH2
```

```
channel2=String(leer_ADS8344(canal_2));
string_out.concat(String(channel2));
string_out.concat(",");
// llegir CH3
channel3=String(leer_ADS8344(canal_3));
string_out.concat(String(channel3));
string_out.concat(",");

// llegir CH1
Temp_actual=leer_ADS8344(canal_1);
str_AD=String(Temp_actual);
string_out.concat(String(str_AD));
string_out.concat(",");

//Serial.println((String)"#RD,"  +str_millis+","  +channel2+
","+channel3+","+str_AD);

//****************************CONTROL TERMIC
 Control_Termic();
//****************************CONTROL TERMIC

string_out.concat(String(int(Time_activacio)));
string_out.concat(",");
string_out.concat(String(DTemp));

//Serial.println(string_out);

 if ( SD_card_output ){dataFile.println(string_out);
          if ((millis()-syncTime) > 5000){
           syncTime=millis();
           dataFile.flush();
           }
           }

} //del if temporitzador
 }
```

GPS

```
void procesa_nmea(int inByte){

  String string_out,str_millis;

  switch (inByte){
  case '$': //CAS INICI DE LA LINIA DE GPS
    puntero_gps=0;
    buffer_gps[puntero_gps]=inByte;
    break;

  case 10: // in case of reading the last char of the GPS
sentence
    if (puntero_gps > 0){
      buffer_gps[puntero_gps]=char(0);
      if ( check_GPGGA(buffer_gps) ){
        buffer_gps[puntero_gps-3]=char(0);
        string_out=String("#GPS,");
        str_millis=String(millis());
        string_out.concat(str_millis);
        string_out.concat(buffer_gps+6);
        if ( USB_output ){ Serial.println(string_out); }
        if ( SD_card_output ){
          dataFile.println(string_out);
          if ((millis()-syncTime) > 5000){
            syncTime=millis();
            dataFile.flush();
            }
        }
      }
      puntero_gps=-1;
    }
    break;

  default: //GENERAL CASE
    if (puntero_gps >= 0){
      puntero_gps=puntero_gps+1;
      buffer_gps[puntero_gps]=inByte;
```

```
    }
    break;
  }
}

int check_GPGGA(char *s){

  int chk=1;
  char M[6]="GPGGA";
  char HEXA[17]="0123456789ABCDEF";

  int N;
  int i=0;
  char chk1,chk2;
  byte cs,cs1,cs2;


  // CCheck if GPGGA
  for (int i=0;i<5;i++){
    if (s[i+1] != M[i]) {
      chk=0;
    }
  }
  // Check the checksum
  if (chk){
    chk=0;
    N=0;
    while( s[N] != 0){
      N++;
    }
    chk1=s[N-2];
    chk2=s[N-1];

    cs=byte(s[1]);
    for (i=2;i<(N-3);i++){
      cs=cs^byte(s[i]);
    }
    cs2=(cs & byte(15));
    cs1=(cs >> 4)& byte(15);


    if ( ( HEXA[cs1]==chk1 ) && ( HEXA[cs2]==chk2 )){
      chk=1;
    }
  }
  return chk;
}
```

TERMIC CONTROL

```
void Control_Termic()
{

  if( (DTemp<0) && (test==0)){                //RESET TIMER
    iTerm=0;
    test=1;
  }
  if (Time_actual > Time_anterior + 2000){ //PUT TO 2000 in TESTS
    T_mean = 0;
    reading[counter_mean ] = Temp_actual;      //MEAN


      if (enable_mean == 1) {
        for (int i = 0 ; i < num_read ; i++)  {          //AVERAGE
          T_mean = T_mean + reading[i];
```

```
            //Serial.print((String)"read "+i+" "+reading[i]);
          }
        }

    Temp_ac_m = T_mean / (num_read) ;

    if ((enable_mean == 0)) {          //BEFORE ARRAY FULL
        Temp_ac_m = Temp_actual;                    //TEMPERATURE
      }

    //Serial.println((String)"T_act "+Temp_actual+" Temp_ac_m "+Temp_ac_m);

    Time_anterior = Time_actual;
    DTemp = Temp_consigna - Temp_ac_m;

    Serial.print((String)"TIME  " + Time_actual + "  kpp " + kpp + "  ki ");
    Serial.print(kii, 4);
    Serial.println((String)"  Temp actual " + Temp_actual / 198.6 + "  Dtemp " + DTemp);

      if ( ((Time_actual - Past_Time) > 10000) && enable_integr != 0) {              //ITERM
         Past_Time = Time_actual;
         ErrSum = (DTemp + Past_Err) / 2;
         iTerm = kii * ErrSum + iTerm;
         Past_Err = DTemp;
      }

    iOut = kpp * DTemp + iTerm;                    //ERROR
    DTemp_ant = DTemp;

    Time_activacio = abs(iOut * 255);              //O
    Time_activacio = Time_activacio / 1986;

      if ( Time_activacio > 255) {
        Time_activacio = 255;
      }
      if ((iOut>= 0)){
        Time_fan = 0;
      }                                            //U

Time_fan=255;
Time_activacio = 0;

      if ((iOut < 0)){
          if (abs(Time_activacio)<125){
             Time_fan = abs(Time_activacio)+125;
          }
           else{
             Time_fan = abs(Time_activacio);
          }
          Time_activacio = 0;
      }
      if ((Time_fan > 255)){
        Time_fan = 255;
      }                                            //T

    Serial.println("Time act     Percentage       iOut      iTerm " );
    Serial.println((String)"  " + Time_activacio + "         " + Time_activacio / 2.55 + "          " + iOut + "           zz," + iTerm + " T_F=" +Time_fan); //analogwrite stopped to check on TEmp

    analogWrite(PinFan, Time_fan);
    analogWrite(PinHeat, Time_activacio);
    //digitalWrite(PinHeat,HIGH);//activar bit de caldeig
```

```
//string_out2=((String)"#CT_T,"+Time_actual+","+kpp+",
"+kii+","+Temp_actual/198.6+","+DTemp+","+Time_activ
acio+","+Time_activacio/2.55+","+iTerm);

  string_out2 = ((String)"#CT_T," + Time_actual + "," +
kpp + ",");

  string_out3 = dtostrf(kii, 1, 4, buffer_st);

  string_out4 = ((String)"," + Temp_actual / 198.6 + "," +
DTemp + "," + Time_activacio + "," + Time_activacio /
2.55 + "," + iTerm);

  string_out2 = string_out2 + string_out3 + string_out4;


    if ( USB_output ) {

      Serial.println(string_out2);

    }

    if ( SD_card_output ) {

      dataFile.println(string_out2);

      if ((millis() - syncTime) > 5000) {

        syncTime = millis();

        dataFile.flush();

      }

    }

  counter_mean = counter_mean + 1;

  //Serial.println(counter_mean);


      if ( counter_mean == num_read) {

        counter_mean = 0;
//MATH MEAN

        enable_mean = 1;

      }


      if (enable_integr==0 &&
DTemp<0){                              //ENABLE INT
AFTER FIRST DROP

        enable_integr=1;

      }

}

  //  if((Time_actual>1764370) &&
(Time_actual<5400000)){   //TIME EXPERIMENT

  //          Time_activacio=0;

  //          analogWrite(PinHeat,Time_activacio);

  //          kpp=0.5;

  //  }

} // del void
```