Group Project

VHF Satellite Ground Station

Sławomir Figiel Tomasz Mrugalski Ewelina Omernik

Space and Satellite Technologies

Supervisor: prof. M. Moszyński, Ph.D D.Sc.

Technical supervisor: W. Siwicki, Ph.D.

2019-11-07



## 

1. Table of Contents

[1 Project overview 3](#_Toc24188443)

[1.1 Project goal 3](#_Toc24188444)

[1.2 Project participants 3](#_Toc24188445)

[1.3 Project Schedule 4](#_Toc24188446)

[2 Progress report 5](#_Toc24188447)

[2.1 Task 1: Feasibility study 5](#_Toc24188448)

[2.2 Task 2: Hardware acquisition 5](#_Toc24188449)

[2.3 Task 3: System Integration 6](#_Toc24188450)

[2.4 Bibliography 8](#_Toc24188451)

# Project overview

This is a group project being developed as part of the Space and Satellite Technologies studies held at ETI Faculty of Gdańsk University of Technology.

## Project goal

Take over the world, bwahahah! But we need to receive some signals from space first.

## Project participants

The are several participants involved in the project. **Tomasz Mrugalski** (TM) is a project lead, orbital mechanics specialist, logistics, and a reliability engineer. **Sławomir Figiel** (SF) is a geospatial data engineer, programmer, Raspberry Pi, and an OS specialist. **Ewelina Omernik** (EO) is a low-level software developer, integrated circuits specialist. Prof. **Marek Moszyński**, Ph.D D.Sc is a supervisor of the project. **Wojciech Siwicki**, Ph.D is a technical supervisor.

## Project Schedule

The overall project time boundaries are limited by the winter 2019/2020 semester terms. The detailed schedule has been proposed and after several iterations agreed on with all major participants. The current schedule is presented in Error: Reference source not found below.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Task** | **Deadline** | **Coordinator** |
| 1 | **Feasibility study** Research of available satellites, capabilities of existing SDR hardware, necessary SDR, antenna and LNA capabilities. | 2019-10-17 | SF |
| 2 | **Hardware acquisition** Selection of specific hardware type, vendor selection, purchasing process, shipment, hardware delivery. | 2019-11-07 | TM |
| 3 | **System integration** Hardware (computing unit, SDR, antenna, wiring), assembly, base software installation (OS, SDR drivers, SDR software) | 2019-11-14 | EO |
| 4 | **Software automation design**  Design of the automated data acquisition, processing pipeline, data deployment | 2019-11-21 | SF |
| 5 | **Software implementation**  Implementation of the design specified in task #4, developed software deployment | 2019-12-05 | TM |
| 6 | **Test campaign**  Test specification, experimental assessment of the system performance, test report, improvement suggestions, conclusions | 2019-12-19 | EO |

Table 1: Project schedule

## Project organization and code repository

To streamline the work, keeping tasks and manage the source code a Gitlab instance has been set up at <https://gitlab.klub.com.pl:30000/astro/satnog-gdn>. Gitlab software was designed to manage software projects and it offers many useful features suitable for a project such as this one. A git repository for the source code brings all benefits of git (version control, changes tracking, accountability, history, etc.). Another great feature of gitlab is a powerful issue tracking that offers discussions, easily formulated task lists, content (e.g. images) uploading, easy cross-references and more. Technical discussions are held on gitlab. Many sections of this report reference to gitlab issues (e.g. gitlab #5). To see specific issue, go to <https://gitlab.klub.com.pl:30000/astro/satnog-gdn/issues> and find the issue number. Note the issue may be closed already. You can go to specific issue directly by specifying it in the URL, e.g. <https://gitlab.klub.com.pl:30000/astro/satnog-gdn/issues/5>.

# Ground station development process

This section describes the process that ultimately led to creation of fully functional VHF ground station that is able to receive satellite transmissions automatically.

## Feasibility study

Engineer responsible: **Sławomir Figiel**.

The first task conducted was a determination whether the data reception from satellites is feasible by a group of students with modest budget. The key concern was whether the hardware required to reliably and repeatedly receive transmissions would be within our budget. Several existing projects were identified with reported repeated successes [1], [2], [3]. The typical radio hardware used was an inexpensive SDR (software defined radio) running on a PC, connected to VHF antenna. In some projects additional components, such as LNA (low noise amplifier) or more advanced directional antenna with tracking mechanism, were used.

Our team looked at various embedded computing platforms. The leading solution available on market is a Raspberry Pi. Its popularity comes from several factors – affordability (cost around 50-70 EUR), high performance (1.5GHz CPU, comparable to mid-level laptops), availability (sold by many vendors, hardware available in stored, including those in Poland), and extensibility (4 USB sockets for data, powered over USB, Ethernet, some models have PoE, some models have WiFi integrated, GPIO, HDMI output). Our earliest experiments used Raspberry Pie 1B+ model, which is 5 years old. While it has proven the general approach, it was difficult to work with due to low performance. We decided to use the latest RPi 4B model. For more details, see Gitlab #4.

Another researched aspect was the radio bands. Two most popular bands are VHF and UHF. We decided to use VHF due to being used by several Polish sats, available antennas and other factors. For more details, see Gitlab #2.

The third researched problem was the choice of antenna. We had to balance several factors here. First concern was he antenna availability. Since the project has strict deadlines imposed, we wanted to get the antenna as soon as possible. Second, the antenna should be reasonably simple to construct. The final aspect was financial. There are many high performance antennas, but their price is often prohibitive. Two final candidates were Winkler turnstile antenna and WiMo TA-1. The latter was slightly more expensive (90EUR, compared to 40EUR), but offered much better delivery options (shipment within 4 days rather than 28 working days).

The deliverable for this task is an analysis with set of specific hardware selected for purchase.

Status: **complete**

## Hardware acquisition

Engineer responsible is **Tomasz Mrugalski.**

The second task conducted after the feasibility study (see task #1) was to analyze the market from the perspective of available components. Our team looked at several vendors offering different Raspberry PI models via varied channels. Our process covered purchase of three elements: embedded computing platform, a Software Defined Radio component and an antenna.

**Embedded computing platform**. As determined in task #1 (see the text above), our platform choice was Raspberry Pi 4. It’s a very recent model with many powerful features. Our research uncovered stories of users complaining about RPi 4 stability. It seems the problem was faulty design of the USB used to power the solution. This was promptly fixed in an updated 4B versions. The RPi 4B comes with 1, 2 and 4GB memory variants. Since the price difference between models is not that great, we chose the most powerful model with 4GB of memory. Our rationale for this decision is to be able to run GUI software, such as gprx, gnu radio or gpredict on this configured RPi 4B. We also chose a kit that provided several essential hardware. The kit included the board itself, a robust case, micro-HDMI to HDMI connector, a USB-C power fully that can meet the power requirements (constant 3A, even under heavy load), a new micro SD card,and a reader for SD cards. The kit has been purchased on Allego, a popular sales platform in Poland, on Oct. 22nd and delivered on Oct. 28th.

**SDR platform**. As determined in task #1, we decided to purchase an SDR platform. By far, the most popular solution is based on two chipsets: RTL2832U + R820T2. Obviously, we needed to connect the SDR dongle to the computing platforms, so it must use USB connector. The model we chose also had a robust case, which protected the delicate hardware inside. The kit we chose came up with a telescope antenna, an SMA cable and a mini-tripod. While we understood the kit antenna is of poor quality, we decided to pay that little extra money to get it, so we could start doing experiments earlier, before the main antenna becomes available. The SDR kit was ordered on Oct. 12th and was delivered couple days later.

**Antenna**. The last missing element of a robust program was an antenna. The initial antenna we considered was Winkler turnstile. We discovered that the vendor requires 28 working days to build the antenna and ship it. This was a major problem, given our projects schedule. Fortunately, we were able to find WiMo TA-1 antenna. While is it significantly more expensive (c.a. 100EUR) as compared to Winkler antenna, it has a great benefit of being readily available. The vendor claims the antenna will be shipped within 4 working days. The order has been placed on Oct. 22nd and we received a tracking number for the shipment. As of Nov 6, the package has arrived to Wroclaw and is expected this week.

The deliverable of this task is to have all the hardware components received.

Status: **complete (2019-11-10).**

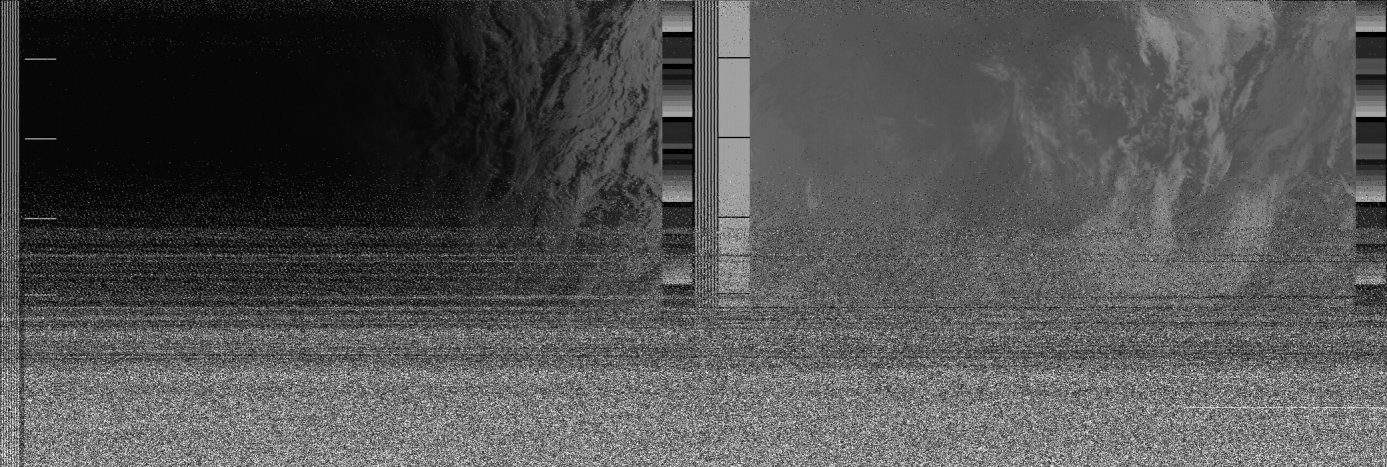
## System Integration

Engineer responsible is **Ewelina Omernik.**

During the week of Oct. 21st our team did not do any individual tasks. Instead, we met together and spent half a day assembling the system. We migrated to the new Raspberry Pi 4B platform, replaced old power supply with a new one using USB-C, also put the motherboard into a case. The hardware setup was assembled at Tomek’s apartment. One major problem to solve was how to deploy the system in a way that has good sky visibility from the antenna point of view, has Internet connectivity, has a power supply and the electronics is protected from the weather. After several attempts, we came up with a plan to house the system in the apartment close to a window. The SMA coax cable will go outside through not completely shut down window. The antenna will be deployed on a photographical tripod, standing on a balcony near the window. We also adapted the basic telescope antenna to work a V dipole (53,4cm length, 120 degrees angle). This provisional set-up will be replaced with the ultimate one once the antenna ordered arrives.

We installed several software pieces: GNU Radio, GQRX (both used to control SDR hardware), gpredict (a software that tracks satellites and informs about upcoming fly-overs), NOAA-APT (an open source alternative to wxtoimg software, it takes the recorded WAV audio file and attempts to extract image data from it).

We experimented with several fly-overs. We later determined that the initial failures were caused by an attempt to receive transmissions from fly-overs that were low on the horizon. This, together with a poor antenna and a poor weather (it was foggy and rainy), caused the signal to be too weak for our system. However, once we picked up a fly-over with high maximum elevation (almost crossed zenith), we were finally able to set up appropriate frequency for NOAA-18, record received transmission as audio and store it as WAV file. The file was then processed using NOAA-APT software and generated the following image:



The lower part of the image is garbled, because we went into NLOS (non line-of-sight) mode (part of the sky was obscured by the roof). Nevertheless, we consider this experiment a full success.

The one remaining task here is to install the new antenna once it arrives.

Status: **work in progress**, expected completion date **2019-11-14**

## Software automation design

Engineer responsible is **Sławomir Figiel**.

So far, the entire process of receiving satellite images is carried out manually. This requires our work at a specific time when the satellite is visible. It is burdensome, so we decided to automate the whole process.

The first step towards automation is to determine which tasks should be performed in which order to receive the photos. As we have already done this manually before, we know how such a process should take place and what components are going to be required.

**Satellite tracking component**:  
The satellites orbit the Earth, following a specific, predictable route. To capture a satellite signal, it must be in the visible range. Therefore we need a compon5ent that will track the movement of the chosen satellites and inform the system when the satellite will be in the field of visibility and when it will disappear. This component would be responsible for initiating the entire process and for stopping it.

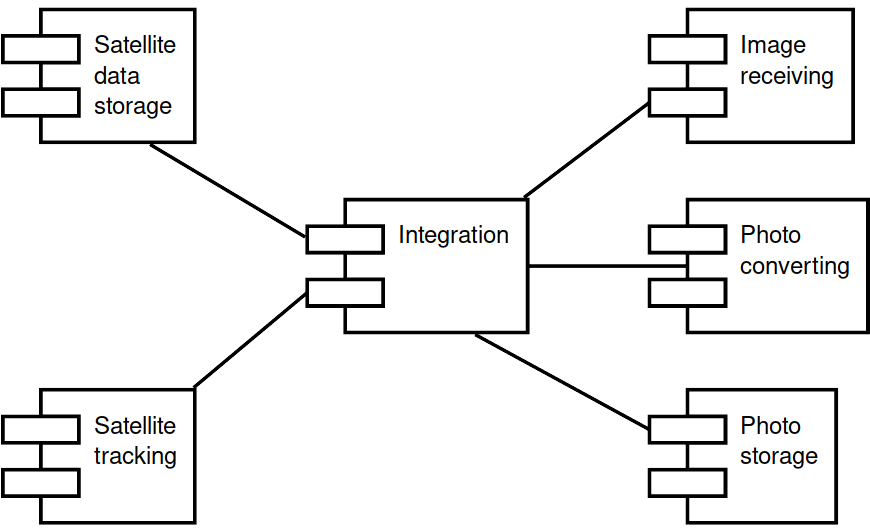
**Satellite data storage component**:  
The tracking component reports the upcoming flight. However, information about the signal frequency, bandwidths, etc. is also required for data reception. These data are found on the Internet and are generally constant. Each satellite has its unique frequencies on which it broadcasts. Exceptions may occur when one communication module is damaged or a new one is added. Due to the variability caused by the above being low, we came to the conclusion that we can download the data just once and then store it.

**Image receiving component**:  
Through the data obtained from the two components described above, we can start recording. After obtaining the signal about the end of the passage, the listening is completed and the "photo" saved, initially as an audio file (\*.wav). This is because we receive radio waves that are easiest to receive and save in this format.

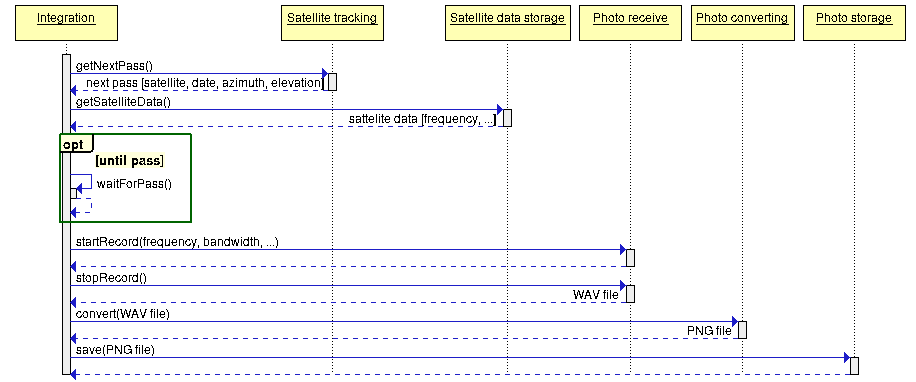
**Photo converting component**:  
The next step is to convert the received photo from an audio format to a graphic format. This component is going to carry out the conversion.

**Photo storage component:**  
Received photos need to be stored. We decided to delegate a separate component that would function as a database of received photos.

**Integration component**:  
We also need a component that connects everything together. It would be responsible for communication between all of the other components.

Figure 1: Components

## Software implementation

  
Figure 2: Control flow between components

The responsible engineer is Tomasz Mrugalski. TODO. See gitlab #16.

## Test campaign

The responsible engineer is Ewelina Omernik. TODO. See gitlab #17.

## Data utilization

The responsible engineer is TBD. TODO. See gitlab #9.

# Project summary

## Conclusions

## Next steps

## Bibliography

1. Satnogs project website, [https://satnogs.org](https://satnogs.org/), retrieved on 2019-10-14
2. Perun Rockets website, <http://perunrockets.net/posluchajmy-satelitow.html>, retrieved on 2019-10-14
3. Pobieranie zdjęć ziemi z satelity za pomocą anteny DIY i… tunera TV, <https://majsterkowo.pl/pobieranie-zdjec-ziemi-z-satelity-za-pomoca-anteny-diy-i-tunera-tv/>, retrieved on 2019-10-30.