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.

2019-12-12



Table of Contents

[1 Table of Contents 2](#_Toc26997757)

[1 Project overview 3](#_Toc26997758)

[1.1 Project goal 3](#_Toc26997759)

[1.2 Project participants 3](#_Toc26997760)

[1.3 Project Schedule 3](#_Toc26997761)

[1.4 Project organization and code repository 4](#_Toc26997762)

[2 Ground station development process 4](#_Toc26997763)

[2.1 Feasibility study 4](#_Toc26997764)

[2.2 Hardware acquisition 5](#_Toc26997765)

[2.3 System Integration 6](#_Toc26997766)

[2.4 Migrating to WiMO TA-1 Antenna 8](#_Toc26997767)

[2.5 Software automation design 9](#_Toc26997768)

[2.6 Software implementation 11](#_Toc26997769)

[2.7 Test campaign 11](#_Toc26997770)

[2.8 Data utilization 11](#_Toc26997771)

[3 Project summary 11](#_Toc26997772)

[3.1 Conclusions 11](#_Toc26997773)

[3.2 Next steps 11](#_Toc26997774)

[3.3 Bibliography 11](#_Toc26997775)

1. 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.

* 1. Project goal

The overall goal of this project is to design, build and then operate a satellite ground station that will be able to receive VHF transmissions from satellites.

* 1. 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 project supervisor. **Wojciech Siwicki**, Ph.D is a technical supervisor.

* 1. 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 Table 1 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 | 2019-12-19 | TM |
| 6 | **Test campaign** Test specification, experimental assessment of the system performance, test report, improvement suggestions, conclusions | 2020-01-09 | EO |

**Table 1**: Project schedule

* 1. Project organization and code repository

A Gitlab instance has been set up at [https://gitlab.klub.com.pl:30000/astro/satnog-gdn](https://gitlab.klub.com.pl:30000/astro/satnog-gdnT) to streamline the work, keeping tasks and manage the source code. The 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>.

1. 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.

* 1. Feasibility study

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 many stores, including those in Poland), and extensibility (4 USB sockets for data, powered over USB, Ethernet, some models have PoE, some 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. A decision has been made to migrate to 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 satellites, 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. The status of this task is **complete.** The engineer responsible for this task is **Sławomir Figiel.**

* 1. Hardware acquisition

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 was defined as to have all the hardware components received. The status of this task is **complete.** The engineer responsible for this task is **Tomasz Mrugalski**.

* 1. System Integration

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 photographic tripod, standing on a balcony near the window.

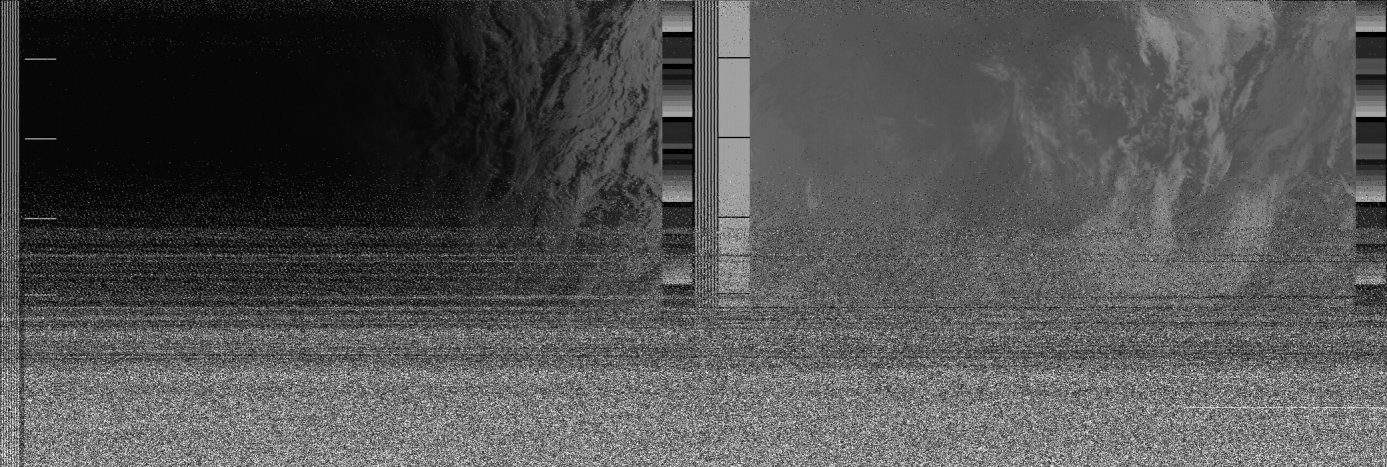
The initial set-up was using a cheap, low quality antenna included in the SDR package. This was a telescope in-door antenna that was originally intended to receive radio and TV broadcasts. The antenna was repurposed as dipole. The telescopic arms were extended to 53.4cm and set up 120 degrees apart on the horizontal plane. The antenna tripod with flexible arms has been used to hold the antenna in place, on top of a reasonably sturdy photographic tripod. The antenna in its assembled state is presented in Fig. 1.



**Fig. 1**: V dipole antenna

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 on the day of experiment), 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:



**Fig. 2**: First received and decoded 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 deliverable of this activity is an integrated, functional ground station. The status of this task is **complete**. The engineer responsible for this task is **Ewelina Omernik.**

* 1. Migration to Crossed Dipole Antenna

The new antenna has arrived in mid-November 2019. This is a crossed-dipole, circularly polarized antenna, specifically designed for satellite reception. Major parameters of the antenna are presented in Table 2 below.

|  |  |
| --- | --- |
| Parameter | Value |
| Vendor | WiMO |
| Model | TA-1 |
| Frequency | 135-152 [MHz] |
| Polarization | Circular right-handed |
| Impedance | 50 [Ohm] |
| Standing Wave Ration (SWR) | Less than 2.0 |
| Dimensions | 1300 x 1065 x 1065 [mm] |
| Weight | 2 [kg] |

**Table 2**: Major properties of the WiMO TA-1 antenna

While much more powerful, the WiMO TA-1 antenna posed a number of challenges. First, it required assembly. This wouldn’t be much of an issue, but one specific design flaw posted a major complication. The antenna consists of a hollow mast made of an aluminum pipe with two horizontal 8mm pipes (acting as reflectors) that are supposed to go through holes in the mast. However, the holes were not wide enough and the reflectors were extremely tight and had to be pushed hard to squeeze in. The excessive force used caused the reflector pipe to bend slightly.

When a pipe is bent, even slightly, changes its cross section shape to an ellipsis, which never returns to its circular shape, even if bent back. This caused the reflector to get stuck in a very shallow position. After over an hour of futile attempts to remove it (that involved, a vice, a hammer, a power drill, sand paper and even a blow torch), a decision has been made to cut the last 3 cm.

Once the reflector had been removed, a power drill has been used to widen the holes slightly. The antenna has been assembled and mounted on a balcony. Due to more permanent (or at least long term) nature of this antenna, a recommendation has been made by dr Siwicki to use an N-connector, as much more robust regarding resistance to adverse weather conditions. Fully assembled is presented in **Fig. 3**.



**Fig. 3**: Assembled crossed dipole antenna

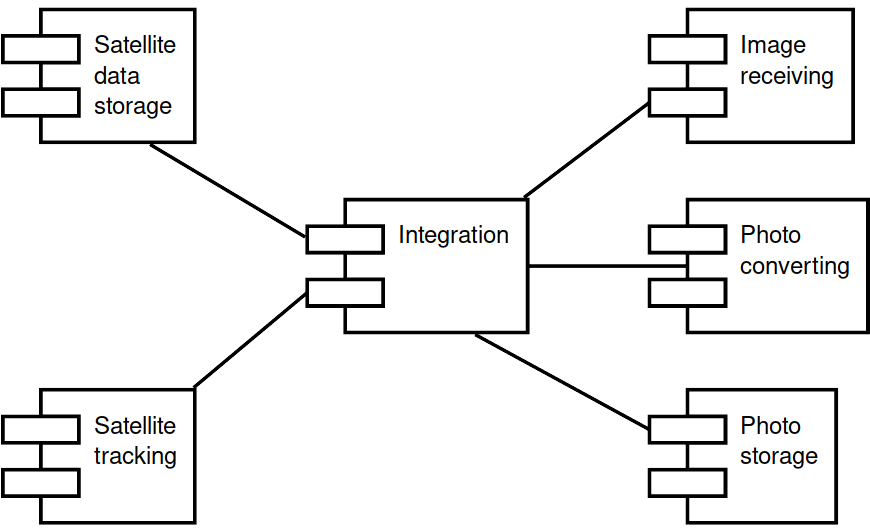
* 1. Software automation design

As of November 2019, the entire process of receiving satellite images is carried out manually. This requires project staff involvement at a specific time when the satellite is visible. It is burdensome and sometimes even impossible to conduct (e.g. when the fly-over happens during late night or early morning). Therefore an obvious decision has been made to automate the whole process.

The first step towards automation is to determine which tasks should be performed in which order to obtain the satellite images. 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. A high level architecture of the software is presented in **Fig. 4**.

**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 component 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 changing very slowly. The change is related to a slow orbital drift due to atmospheric dragging, solar wind pressure and similar low impact factors. 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. Assuming a long term evolution of the project, the system may be extended to do infrequent periodic updates of the orbital geometry. For the time being we assume this is out of scope.



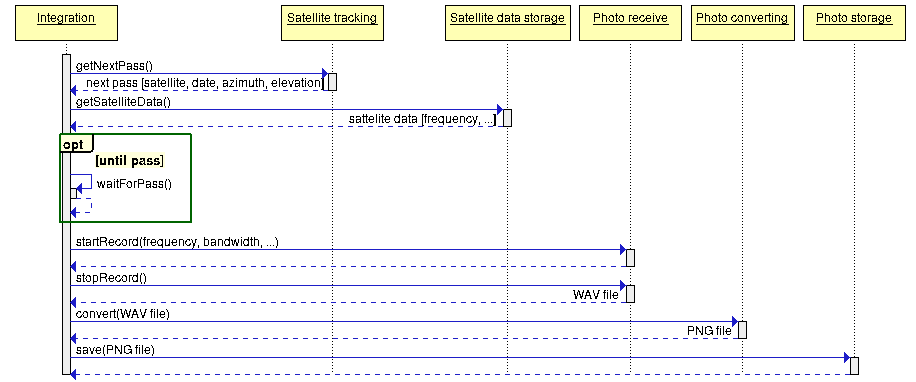
**Fig. 4**: High level software architecture

**Transmission reception component**. The data provided by the components mentioned above will be used to control the actual transmission reception component. This component will use the SDR to tune in to specific frequency, decode the transmission using FM modulation and record it using a standard WAV audio file format.

**Data processing component.** The image data is encoded using Automatic Picture Transmission (APT) mode. It was used by several US and Russian satellites, but as of 2019 there are only three active NOAA satellites that use it. For more details, see [5] and [6]. There are several tools that offer APT decoding. For some time, the most popular was wxtoimg, however, due to the software being abandoned by its developer and unclear licensing status, we chose to not use it. Instead, noaa-apt, an open source alternative has been used instead [7]. The software is able decode wav files and extract image data from it. The output generated is a regular PNG file.

**Photo repository component.** Received photos need to be stored somewhere. We decided to designate a separate component that would provide the functionality of a repository for the received photos. One specific aspect of consideration here is that the platform chosen (Raspberry Pi) has limited disk space available. The SD card used in 16GB, but part of it is already taken by the operating system and installed software, so the actual capacity available for the data storage is smaller. Since there is a powerful NAS storage available in the LAN this ground station is connected to, a network mounted storage will be used.

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



**Fig. 5**: Control flow between components

An interaction between different components will be conducted sequentially. The control flow will start and end with Integration, which will call specific routines from other components. The control flow is presented in **Fig. 5** above.

A separate document “Image reception automation” has been dedicated to this task. See appendix A3 for details.

The deliverable for this task is to have a high level design for the automation solution. The task is **complete**. The engineer responsible for this task is **Sławomir Figiel**.

* 1. Software implementation

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

* 1. Test campaign

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

* 1. Data utilization

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

1. Project summary
   1. Conclusions
   2. Next steps
   3. Bibliography
2. Satnogs project website, [https://satnogs.org](https://satnogs.org/), retrieved on 2019-10-14
3. Perun Rockets website, <http://perunrockets.net/posluchajmy-satelitow.html>, retrieved on 2019-10-14
4. 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.
5. Simple NOAA/Meteor Weather Satellite Antenna: A 137 MHz V-Dipole, <https://www.rtl-sdr.com/simple-noaameteor-weather-satellite-antenna-137-mhz-v-dipole/>, retrieved on 2019-12-11
6. Automatic Picture Transmission(APT), [https://www.sigidwiki.com/wiki/Automatic\_Picture \_Transmission\_(APT)](https://www.sigidwiki.com/wiki/Automatic_Picture%20_Transmission_(APT)), Signal ID wiki, retrieved on 2019-12-11
7. Automatic Picture Transition, <https://en.wikipedia.org/wiki/Automatic_picture_transmission>, wikipedia, retrieved on 2019-12-11
8. Noaa-apt, project website, <https://noaa-apt.mbernardi.com.ar/>, retrieved on 2019-12-11
9. T.Mrugalski, S.Figiel, E.Omernik, SATNOGS-GDN project homepage, [https://gitlab.klub.com.pl:30000/astro/satnog-gdn](https://gitlab.klub.com.pl:30000/astro/satnog-gdnT), retrieved on 2019-12-11
   1. Appendices
10. Weekly report – 2019-10-24
11. Weekly report – 2019-10-31
12. Image reception automation (a feasibility study)
13. Crossed Dipole Antenna Evaluation (preliminary report)