Embedded and Connected Receiver System for nano-satellite in Low Earth Orbit (LEO)

F. Spies[1,2], S. Givron[2]
[1] FEMTO-ST Complex Computer System, Montbéliard, France
[2] Université de Franche-Comté, Montbéliard, France

Abstract

The objective of this work is to design a set of satellite signal reception, embedded, connected and low power consumption. This set must be simple to implement with the ambition of being widely deployed on a global scale to provide complete and continuous coverage so that each satellite transmission can be received at any time without loss. The altitude of the satellite orbit must allow the planet to be covered with less than a hundred reception stations on earth. The stations will be located mainly in universities with an *eduroam* connection to facilitate the transmission of information on a server.

1 Introduction

Communication systems with nano-satellites remain complex to implement. They must have a suitable receiving chain in order to be able to process small amplitude signals. The implementation altitude is around 500 km. The nano-satellite transmission chain must be as light as possible, including the antenna, power amplifier and communication board. The objective of nano-satellites is to be as light as possible with a reduced altitude to limit the cost of placing them in orbit as much as possible. Therefore, the ground station must be able to compensate for the transmission and reception constraints of nano-satellites. It is necessary to be able to amplify weak signals received with motorized directional antennas combined with power amplifiers. The objective of this work is to provide a ground-based reception solution consisting of a simple omnidirectional antenna and a Software Defined Radio (SDR) interface. The advantage of radio software is that it can receive several different nanosatellite signals and can be reprogrammed at any time. In addition, this solution can be updated over time to adapt to changes in modulation in the nano-satellites of next generations. Finally, a study of the dimensioning of the ground reception module will make it possible to reduce the unit cost so that a large-scale deployment can be envisaged.

2 Demodulation Chain

The frequencies mostly used by these nano-satellites are VHF/UHF amateur radio frequencies (145 and 435/438 MHz) or the S band (2.2 to 2.3 GHz). Modulations are generally of the FSK, BPSK or GMSK type. Baud rates are generally between 1200 and 9600 baud. The use of two symbols per period significantly reduces the reception threshold and thus increases the reception distance. One of the advantages of using software radio is that it allows you to reconfigure the demodulation chain for each passage of a nano-satellite. There are software programs such as *gpredict* that integrates the paths of all active in-orbit nano-satellites to predict the path times.

3 Picsat Demodulation

Picsat, a specific nano-satellite put into orbit in 2018, demodulated the signal transmitted in BPSK 1200 baud. The demodulation chain performed by [1] integrated all the demodulation functions necessary to interpret the information transmitted by the satellite and send it back to the server centralizing all the reception on the planet. Indeed, at this altitude, the flight times of an area are generally between 5 and 15 minutes. We understand the importance of being able to multiply the number of reception sites around the world in order to be able to increase the exchange times with the nano-satellite. If ground systems are connected to the Internet, it is possible to gradually move towards a permanent and continuous connection that would limit the redundancy of data transmitted from the satellite and thus increase its ability to transmit information to the ground. All you need is between 80 and 100 ground receiving systems evenly distributed to be in permanent communication with the nano-satellite. A maximum distance between 2 reception systems of about 3500 km (30 degrees) would allow this continuity of communication.

4 Systems Implementation

Before attempting to embed the receiving chain on low-cost electronic boards, it is preferable to validate oversized solutions to facilitate development. Thus, we used a laptop computer, a Raspberry Pi 3, a Raspberry Pi zero and a microcontroller ESP8266 [2] including a Wi-Fi interface to perform the signal demodulation. The signal reception was performed with a USRP B200 mini, an ADALM-Pluto and an RTL-SDR.

On laptops and Raspberry PIs that use a UNIX system, the use of gnuradio is possible. On Raspberry PI boards, computing power is limited, the USB interface works in version 2 and the use of command line gnuradio is preferable. On the microcontroller, the program is downloaded from the Arduino development interface in C/C++/LUA language.

5 Signal Generator

When designing a demodulation function, it is necessary to decode a good quality signal, but this is not enough. It is also necessary to be able to measure the qualities and characteristics of the algorithm designed to identify the potential of the reception sensitivity threshold obtained. If the reception sensitivity threshold is too high, decoding will only be possible when the nano-satellite is near the zenith of our position. If the sensitivity threshold is low enough, decoding can be carried out up to positions close to the horizon where the link budget is most unfavourable.

We have produced several types of files containing signals including increasingly large attenuations and small constant, linear and polynomial frequency shifts to gradually reproduce the Doppler effect of LEO orbits.

6 Decoding Tests

First, we validated that reception could be achieved with an omnidirectional antenna placed horizontally in the perpendicular of the satellite path connected to a USRP B200 mini interface. The whole thing was connected to a laptop PC with gnuradio/gr-picsat [1] to ensure sufficient computing power and memory capacity. The reception time is approximately 8 minutes out of a potential 10 to 12 minutes. This first result was considered sufficient to qualify the reception.

In a second step, we validated the computing and memory capacity of the Rapsberry Pi zero Wi-Fi (ARM 1 core 1GHz 512GB RAM) to demodulate the I/Q signals received in gnuradio/gr-picsat in command line mode.

In a third step, the demodulation was coded directly into the micro-code of the ESP8266 to allow the restitution of the transmitted data. Software radios have a USB port and can only be connected to the ESP8266 at a low speed equivalent to version 1.0 or 1.1 of about 1Mbps. We have chosen to use the radio software in SDR-IP mode (Fig. 1), i.e. through a Wi-Fi/IP connection to validate the processing capacity of the microcontroller. Data reception centered on the corrected frequency could be completely integrated, in purely integer functions for more efficiency. In order to limit the energy consumption by the entire station, it is only executed when a nano-satellite passes through and remains in deep sleep between two passes. The times of the visits are obtained on servers located on the Internet.

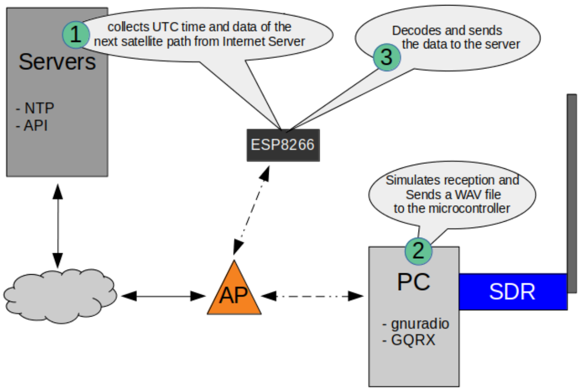

Figure 1: Global Architecture with the Wi-Fi microcontroller ESP8266.

7 Conclusion

Porting demodulation functions into a microcontroller has allowed us to improve the overall efficiency of the receiving chain. The Doppler effect correction remains to be integrated into the microcontroller to complete the demodulation chain. We finally chose to reintegrate the entire demodulation functions into gnuradio to reduce processing time and use a low-power Raspberry Pi zero that easily connects an SDR interface.

References

[1] *PicSat telemetry parser added to gr-satellites* at `https://destevez.net/2018/01/picsat-telemetry-parser-added-to-gr-satellites/`

[2] *ESP8266: Wi-Fi microchip with full TCP/IP stack and microcontroller* at `https://en.wikipedia.org/wiki/ESP8266`