# PyNGHam: A Python library of the NGHam protocol

Gabriel Mariano Marcelino [1,2]

1 Space Technology Research Laboratory (SpaceLab), Universidade Federal de Santa Catarina **2** Senai Institute for Innovation in Embedded Systems (ISI-SE)

Summary

The PyNGHam library is a Python implementation of the original NGHam protocol library written in C by Jon Petter Skagmo (LA3JPA) (Skagmo, 2014). The NGHam protocol is an amateur radio protocol developed to be a modern version of the AX.25 protocol, with the main improvement of introducing a forward error correction (FEC) algorithm, which considerably improves the robustness of the communication link. One of the main applications of this protocol is on small satellite projects (specifically CubeSats), supporting a reliable communication link between the ground station and the satellite. This Python implementation enables easier integration and use of this protocol in computers and embedded devices.

Statement of Need

The NGHam protocol was originally developed in the context of a CubeSat development at the Norwegian University of Science and Technology (NTNU) (Løfaldli & Birkeland, 2016). It was later used by the Space Technology Research Laboratory (SpaceLab) team, from Universidade Federal de Santa Catarina (Brazil), on the FloripaSat-1 CubeSat, and is being used by all satellites of the group so far. The known satellites that have used or plan to use the NGHam protocol are:

- **FloripaSat**-**1** (Marcelino et al., 2020)
- **GOLDS**-**UFSC** (a.k.a. **FloripaSat**-**2**) (SpaceLab, 2022a)
- **Catarina**-**A1** (Catarina Constellation) (Seman et al., 2022)
- **PION**-**BR1** (PION Labs, 2022)
- **Aldebaran**-**1**
- **NUTS**-**1** (Løfaldli & Birkeland, 2016)

The first three satellites in the list are satellites developed (or in development) by the same research group, the SpaceLab, where this library was developed.

From the knowledge of the author, there is no Python implementation of the protocol in the literature. An implementation in a high-level language facilitates the development of user applications to communicate with objects in orbit. The intention is to make PyNGHam usable in end-user applications, like satellite decoders used in real satellite missions. It is an alternative to the GNU Radio (GNU Radio - the Free and Open Software Radio Ecosystem, 2022) blocks such as gr-nuts (Løfaldli, 2022), but independent from the GNU Radio ecosystem. It is also useful as a simulation or research/education tool that can be used in simple Python scripts.

This library is already being used in the design of satellites by SpaceLab, as a part of the ground station software (SpaceLab, 2022b, 2022c) responsible for the uplink of telecommands and downlink of telemetry/payload data.

NGHam Protocol

The NGHam (Next Generation Ham Radio) protocol is a link protocol partly inspired by AX.25 (Loveall, 2022), with the intention to be used in Ham radio packet communications, but using a FEC algorithm on a well-defined packet structure.

For the FEC algorithm, the Reed-Solomon code (RS) is employed (Reed & Solomon, 1960). Figure 1 presents a diagram with the fields of a NGHam packet.
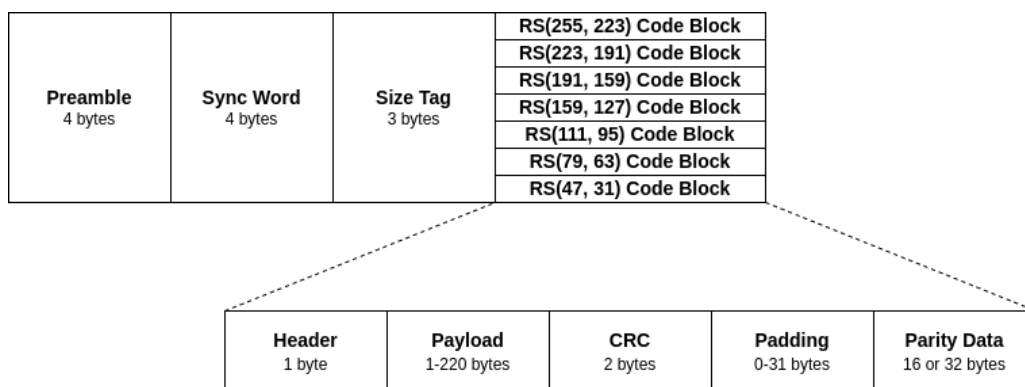

Figure 1: Fields of a NGHam packet.

For a GMSK (Gaussian Minimum - Shift Keying) modulation at 9600 bps, a typical preamble sequence would be 0xAAAAAAAA (a simple alternate of ones and zeros).

The size tag field has seven different options, each corresponding to a unique packet size as described in Table 1.

Table 1: NGHam packets sizes.

Size Num.	Tag	Reed-Solomon Config.	Max. Data Size
1	59, 73, 205	RS(47, 31)	28 bytes of data
2	77, 218, 87	RS(79, 63)	60 bytes of data
3	118, 147, 154	RS(111, 95)	92 bytes of data
4	155, 180, 174	RS(159, 127)	124 bytes of data
5	160, 253, 99	RS(191, 159)	156 bytes of data
6	214, 110, 249	RS(223, 191)	188 bytes of data
7	237, 39, 52	RS(255, 223)	220 bytes of data

Following the seven possible packet sizes, after the size tag field, there is the data field with the Reed-Solomon code block, with seven different schemes, one for each size tag. As can be seen in Figure 1, the used RS configurations are: (47, 31), (79, 63), (111, 95), (159, 127), (191, 159), (223, 191), and (255, 223), respectively.

The code block consists of two types of fields: packet data and parity data. The parity data is the byte sequence generated by the Reed-Solomon algorithm. The packet data is the information presented in the packet and is divided into four fields: header, payload, CRC (Cyclic-Redundancy Code), and padding.

Scrambling

Before transmitting a packet, the RS code block is scrambled by making a logical xor operation with a pre-generated table based on the polynomial $x^8 + x^7 + x^5 + x^3 + 1$ (defined in the CCSDS 131.0-B-4 standard, Section 10.4.1 (CCSDS, 2022)).

When the receiver receives a packet, it also performs the same operation to descramble the RS code block, ultimately retrieving the original content of the RS part of the packet. By scrambling the packets using the bit transition, long sequences of ones or zeros are mitigated (Section 8.3 of CCSDS (2022)).

Serial Protocol

While the NGHam protocol can be used to support wireless communications, there is also the possibility to use the protocol in physical serial interfaces with a slightly different packet structure. In the serial protocol, no FEC algorithm is involved, but merely a checksum field to indicate whether an error occurred during the transmission can be integrated. Both wireless and serial options can be integrated into a device.

Extensions

The payload of an NGHam packet can also contain subpackets, called extension packets. Each NGHam extension packet has a separate header describing the type and size of the subsequent data. A payload can contain multiple extension packets, each containing information such as position, callsign, timing information, statistics, destination, repeating information, and others. The use of extensions is optional.

The Python Implementation

This Python implementation of the protocol was partly inspired by the structure of the original implementation in C, but with the main difference being the use of an object-oriented language. This way, a class for each of the three main possible uses of the protocol is available: the normal NGHam packets, the serial port packets, and the use of the extensions. All of these three classes are implemented with a similar approach, using at least two methods for packet encoding and decoding. The encoding and decoding processes generate a list of integers, representing the bytes of an NGHam packet/payload data. More information about the PyNGHam library can be found in the documentation: https://mgm8.github.io/pyngham/.

Conclusion

The objective of this library is to offer an alternative to the original NGHam library, purely written in Python. In this manner, such a package can easily be used to carry out simulations, develop packet encoding and decoding tools, and introduce telecommunication concepts to students, among others. It is useful especially for satellite communications, space experiments, and for general use in the amateur radio community.

Acknowledgements

References

CCSDS. (2022). *TM synchronization and channel coding*. The Consultative Committee for Space Data Systems. https://public.ccsds.org/Pubs/131x0b4.pdf

GNU Radio - the free and open software radio ecosystem. (2022). https://www.gnuradio.org

Løfaldli, A. (2022). *gr-nuts - GNU Radio module for the NUTS project groundstation.* https://github.com/lofaldli/gr-nuts

Løfaldli, A., & Birkeland, R. (2016, June). *Implementation of a software defined radio prototype ground station for CubeSats.* https://doi.org/10.13140/RG.2.1.1806.0408

Loveall, P. (2022). *AX.25 Layer 2.* http://www.ax25.net/

Marcelino, G. M., Vega-Martinez, S., Seman, L. O., Kessler Slongo, L., & Bezerra, E. A. (2020). A critical embedded system challenge: The FloripaSat-1 mission. *IEEE Latin America Transactions*, *18*(02), 249–256. https://doi.org/10.1109/TLA.2020.9085277

PION Labs. (2022). *Information for communicating with PION-BR1.* https://github.com/pion-labs/PION-BR1/wiki/Radio-Information

Reed, I. S., & Solomon, G. (1960). Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, *8*(2), 300–304. https://doi.org/10.1137/0108018

Seman, L. O., Ribeiro, B. F., Rigo, C. A., Filho, E. M., Camponogara, E., Leonardi, R., & Bezerra, E. A. (2022). An energy-aware task scheduling for quality-of-service assurance in constellations of nanosatellites. *Sensors*, *22*(10). https://doi.org/10.3390/s22103715

Skagmo, J. P. (2014). *NGHam protocol.* https://github.com/skagmo/ngham

SpaceLab. (2022a). *GOLDS-UFSC mission.* https://github.com/spacelab-ufsc/floripasat2-doc

SpaceLab. (2022b). *SpaceLab packet decoder.* https://github.com/spacelab-ufsc/spacelab-decoder

SpaceLab. (2022c). *SpaceLab packet transmitter.* https://github.com/spacelab-ufsc/spacelab-transmitter