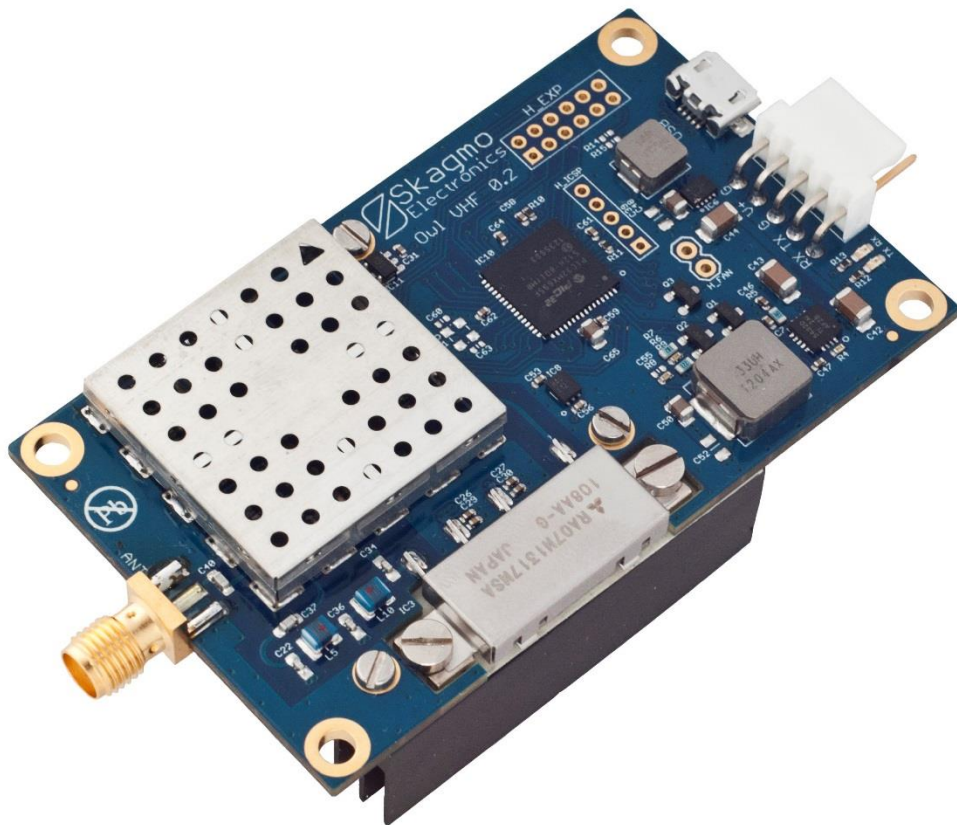


Jon Petter Skagmo, LA3JPA



NGHam protocol description

November 5, 2014

Table of contents

1. Introduction.....	3
2. Radio protocol	3
3. Serial port protocol.....	5

1. Introduction

NGHam is a protocol set for packet based amateur radio, intended to resolve some issues with currently used protocols AX.25 and KISS. Its key features are:

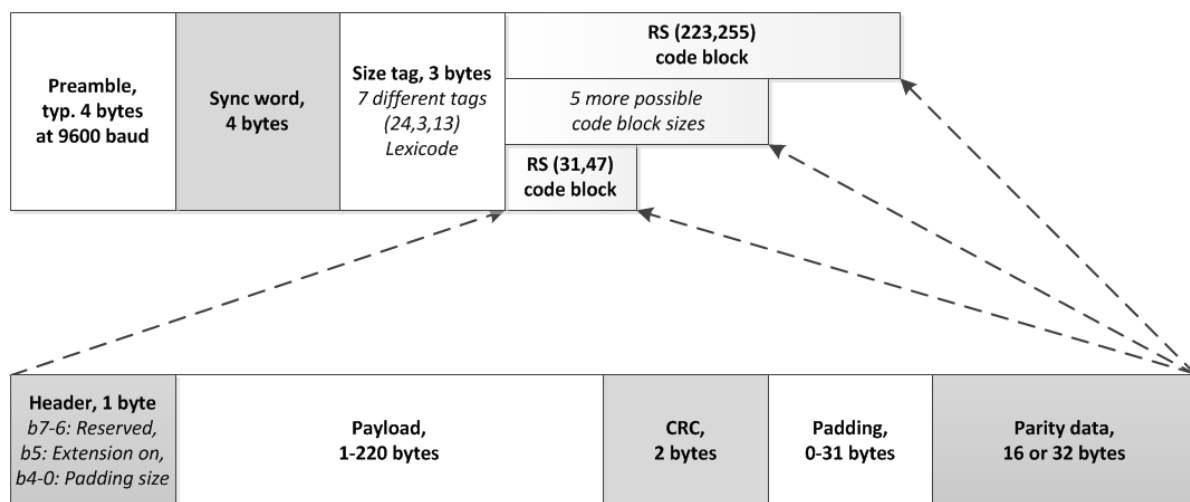
Very robust: Allows decoding with much lower SNR compared to 1200 baud AFSK AX.25 due to the use of a sync word with good autocorrelation properties and Reed Solomon FEC over the whole remaining packet

High throughput: Due to the short preamble and increased chance of successful packet reception, the practical throughput will be much higher than when using AX.25

Better spectral efficiency: Significantly better spectral efficiency than 1200 baud AFSK, and somewhat better than 9600 baud G3RUH modulation due to the reduced deviation.

2. Radio protocol

The structure of an NGHam radio packet is shown below.



NGHam radio protocol – LA3JPA 2014

Preamble: The preamble is short, and is transmitted while the PA is ramping up and the receiver AGC is stabilizing. Will vary with symbol rate, but 4 bytes is typically used for 9600 baud.

Sync word for bit- and packet-synchronization: A 32-bit correlation tag / sync word allows both bit-synchronization and detection of packet start in very short time and in very low S/N conditions. This means the preamble can be drastically reduced from the order of 100 ms as we typically see for AX.25 today, down to under 5 ms - depending on how fast the hardware can switch to transmission. This also increases the chance of detecting the packet start substantially.

Size tag: A 24 bit tag identifies one out of 7 possible code word sizes. This is made very robust by keeping a hamming distance of 13 bits between all vectors.

Code block: NGHam uses Reed Solomon block error correction to make it robust against bursts of errors. NGHam has no inner convolutional error correction, and this was decided to increase throughput at the cost of a somewhat higher SNR margin. Reed Solomon is a robust error correction scheme with no restrictions on usage (as opposed to eg. many turbo codes), and Phil Karn, KA9Q, provides an open source C-implementation with LGPL licensing (<http://www.ka9q.net/code/fec/>). Reed Solomon has a fixed block size, so some different methods for allowing variable packet sizes

were evaluated. Initially a preceding 12 bit length field with a (12,24) golay FEC allowing a variable packet length down to a single byte was tested, but this turned out to reduce the robustness due to it only being capable of correcting single bit errors. The final solution was to allow seven differently sized code blocks identified by a size tag. The transmitter chooses the smallest code block that can fit the payload to be sent, and the remaining bytes are padded. This allows higher robustness than the first method at a cost of a 0-31 byte overhead.

Scrambling: K9NG polynomial scrambling has served well for 9k6 AX.25 packet in many years, but unfortunately it cannot be used with most FEC-schemes as errors propagate throughout the packet. XOR-ing the data with a fixed sequence at encoding and decoding allows zero delay, synchronization with packet start, and no propagating errors. The polynomial used for generating the table is $x^8 + x^7 + x^5 + x^3 + 1$ (as of CCSDS 101.0-B-3).

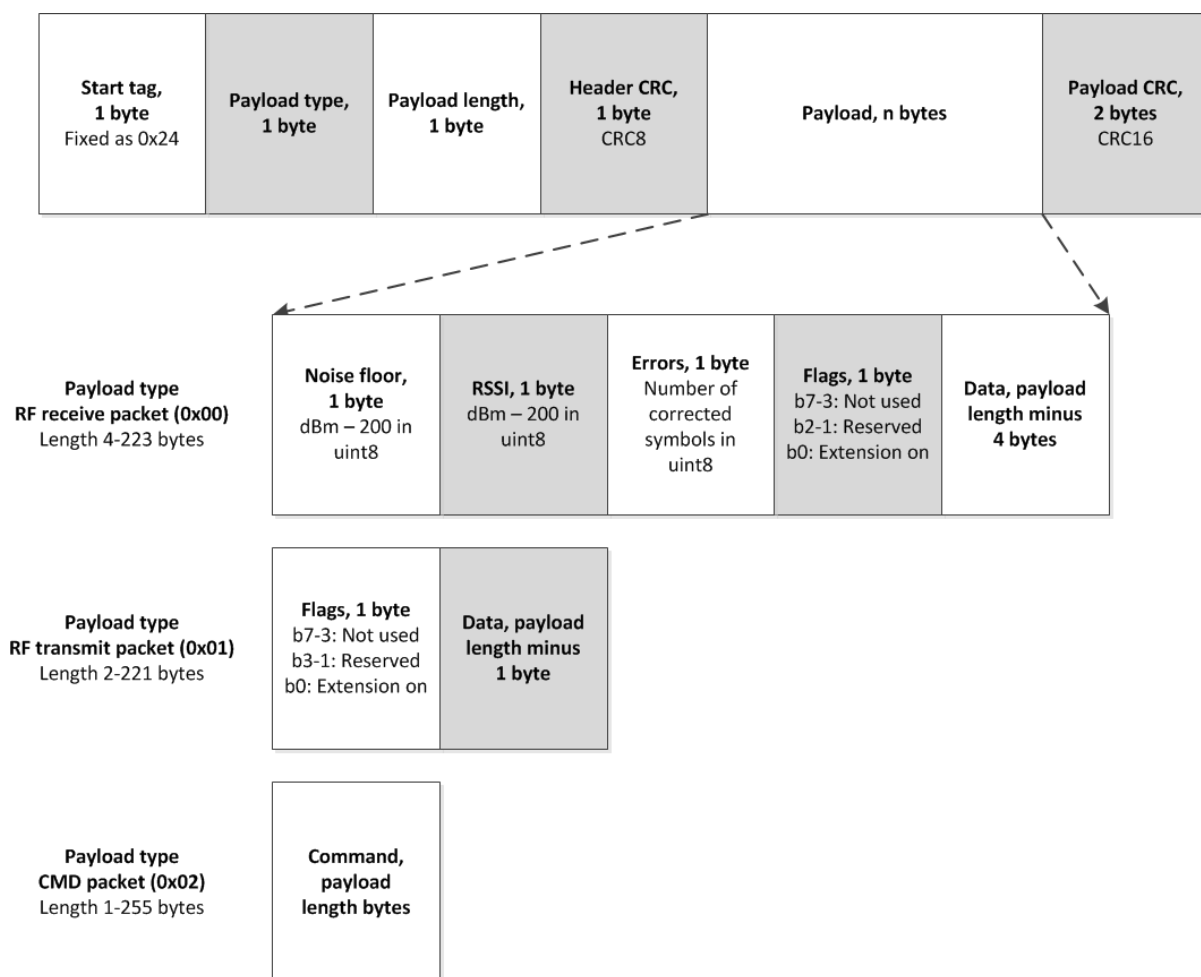
Modulation: GMSK modulation with $bt=0.5$ should be used. 9600 baud 2-GMSK is the default rate, but combinations of 4800, 9600 and 19200 baud symbol rate at 2-GMSK and 4-GMSK are all valid. At 4-GMSK the preamble, sync word and size tag should be transmitted using 2-GMSK.

Channel access: Due to the very different uses, CSMA should work as the default access scheme. However, the channel access could be changed depending on the next layer protocol. For example, a positioning protocol with regular short transmissions of pretty much the same length, such as APRS, should use a TDMA-derived scheme. This will allow a lot more efficient channel usage than CSMA. For transferring bigger amounts of data between a few people, a MACA-scheme with RTS/CTS packets as described by Phil Karn could be the most efficient method, but none of this is implemented as of today.

Continuous transmission: All NGHam devices should support continuous transmission and reception (ring buffers are used where needed) by stacking packets end-to-end.

3. Serial port protocol

This chapter describes the protocol used to transfer data and commands between the transceiver and the serial port host.



NGHam serial port protocol – LA3JPA 2014

General format

Name	Size (byte)	Notes
Start tag	1	Fixed as '\$'
Type	1	0x00=RF receive packet, 0x01=RF transmit packet, 0x02=Command, 0x13=XOFF (stop transmission), 0x11=XON (resume transmission)
Length	1	Length of data field, <i>n</i>
Header CRC	1	CRC8 of all the above except the start tag
Payload	<i>n</i>	This is the actual data - depends on packet type
Payload CRC	2	CRC16 of the payload field, only present if data field is present

Type RF receive packet (from radio to host)

Data received from RF link. Length from 4 to 223. The table below describes what is put into the payload of the general packet format.

Name	Size (byte)	Notes
Noise floor	1	In dBm - 200. Eg. 0x50 = -120 dBm

RSSI	1	In dBm - 200.
Symbol errors	1	Number of corrected Reed Solomon symbols.
Flags	1	Bit 0: NGHAm extension enabled flag.
Data	$n-3$ B	Received data.

Type RF transmit packet (from host to radio)

Data to be transmitted on RF link. Length from 1 to 220. The table below describes what is put into the payload of the general packet format.

Name	Size (byte)	Notes
Flags	1	Bit 0: NGHAm extension enabled flag.
Data	n B	Data to be transmitted.

Type CMD

This type of packet supports a standard NGHAm command as typed in the command-line interpreter, except it is not terminated by LF/CR/CRLF. The table below describes what is put into the payload of the general packet format.

Name	Size (byte)	Notes
Command	n B	Non-terminated command

Examples on command transmission to an Owl VHF

Host to radio: "\$\x02\x0DFREQ 144800000\xxx\xxx"

Radio to host: "\$\x02\x02FREQ 144800000 OK\xxx\xxx"

Flow control examples

host->radio: RF transmit packet

host->radio: RF transmit packet

host->radio: RF transmit packet

radio->host: XOFF

host waiting

radio->host: XON

host->radio: RF transmit packet

radio->host: RF receive packet

host->radio: Command