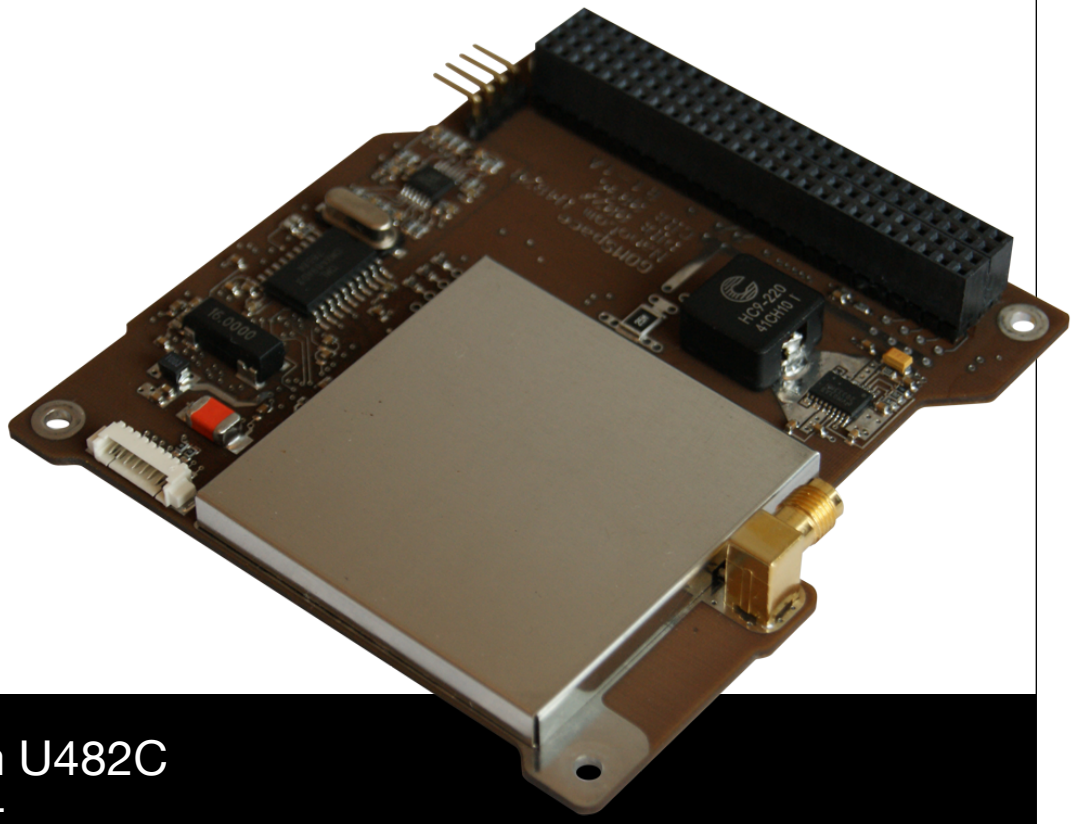
GOMSPACE



NanoCom U482C
Datasheet
Tentative

UHF packet transceiver system for space applications

NanoCom U482C

The NanoCom U482C offers a reliable space-link to small satellite platforms with its flight-proven transceiver and simple yet fully-featured architecture. Mission success highly reliant on the space-link quality, and with its sensitive receiver and error-correcting code, NanoCom U482C ensures a good link margin in any low-earth orbit.

Feature Overview

- Half-duplex UHF narrow-band FM transceiver
  - Designed for 435-437MHz operation
  - Super heterodyne receiver
  - Packet reception down to -125dBm at 1200bps
  - Transmitter with 30-34dBm at > 45 % PAE
- MSK baseband
  - Uplink: 1200-4800 baud
  - Downlink: 1200-9600 baud
- CCSDS frame format with:
  - 32 bit ASM header
  - Fixed frame length (256 bytes)
  - Optional Virterbi FEC r=1/2, K=7
  - Optional Reed-Solomon FEC (223,255)
  - Optional Randomization
- On-board measurement of:
  - PCB and PA temperatures
  - Battery voltage
  - RSSI and RF-error on receiver
- Autonomous audio morse beacon when idle
  - User-defined timeout, interval and WPM
  - Configurable content (text and measurements)
- Simple CSP based I²C interface at 400kbps
- High-effeciency buck-converter for transmitter supply
- Compatible with the Cubesat Kit "PC104" formfactor
- Operational temperature: -30 C to +60 C
- Dimensions: 95.40 mm x 90.15 mm x 18.00 mm
- PCB material:
  Glass/Polyamide IPC 6012C cl. 3/A

Applications

- Single CubeSat satellites
- Triple CubeSat satellites
- Nano Satellites

Compatibility

- GomSpace products
- CubeSat Kit products
- Innovative Solutions in Space products
- ClydeSpace products

Flight Heritage

- The RF design is based on heritage from the SSETI-Express satellite and is strongly based on the AAUSAT-II transceiver, with the exception improved oscillators, matching plus a power amplifier with better performance. The baseband is an enhanced version of the AAUSAT-II system while the uC interface and its software is a new generation with greatly improved performance and robustness.

Functional Description

The NanoCom U482 communication system is designed to provide the space-segment-part of a space-link for nano- and pico-satellites by means of a half-duplex UHF transceiver operating in the amateur radio 70cm band. The internal interface is an I²C bus allowing command and data exchange with the NanoCom board via a simple protocol.

All packet framing plus the optional Viterbi encoding/decoding is performed by the NanoCom so only the raw data packets need to be transferred over I²C bus allowing for swift integration both of

the hardware and the communication protocol. Vital housekeeping measurement points are sampled at user-defined intervals and stored for retrieval via the I²C bus or spacelink upon request. An FM morse beacon is activated when the system is idle which may morse any combination of the housekeeping elements in human-readable format.

The unit uses one or two power supplies depending on configuration: VCC and VCC_TX. VCC powers all the digital and analog circuitry plus the receiver. VCC_TX powers the transmitter power amplifier. These supplies may be connected to various pins in the stack connector, and VCC_TX can use an on-board high-effeciency buck-converter.

The recommended configuration is to use the buck-converter for VCC_TX and supply this from a voltage higher than the required supply for the transmitter. In this way, the current through the stack connector is minimized which may be necessary as the power amplifier can draw up to several amps - more than the rating of the stack-connector.

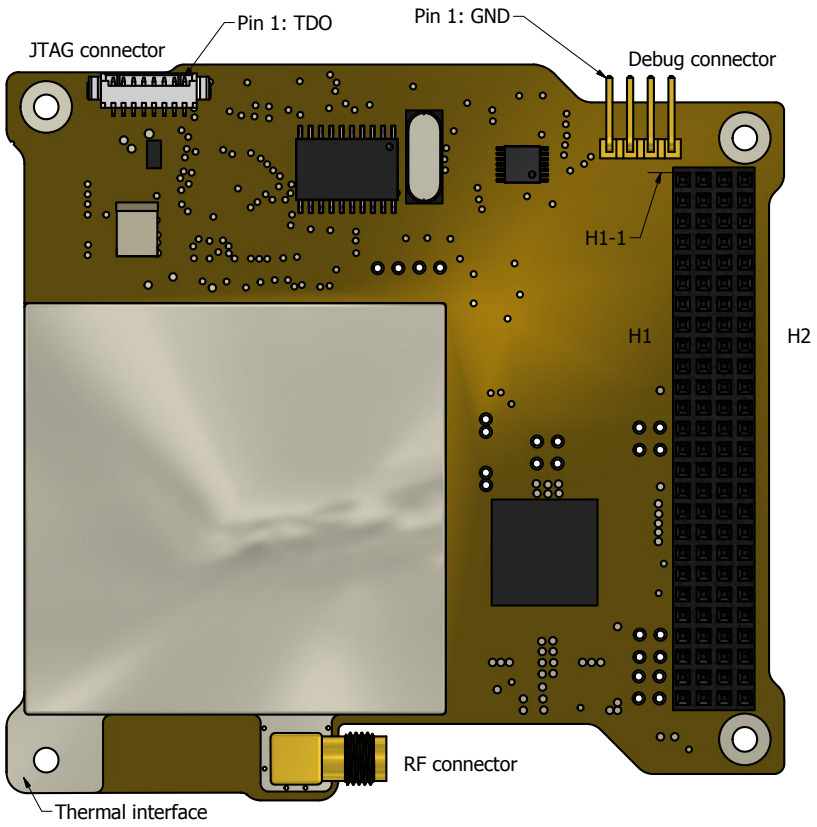
Getting Started

Be sure to use observe antistatic handling procedures at all times when handling the NanoCom U482 transceiver. Before applying power to the board, make sure that the following conditions are in order:

- The SMA RF connector is connected to a 50 oOhm load rated at >4W.
- The ground connection is properly connected to ground on the power supply.
- The thermal power dissipated by the transmitter can be conducted away from the board. In 1 atm. pressure the transmitter is operational up to 60 deg. C. While the power amplifier can be operated up to +80 deg. C. *Do not exceed these temperatures.*
- If operated in vacuum, the heat must be conducted away from the board by means of a solid thermal connection to the corner-holes or by soldering a heat-strap directly onto the ground-plane below the power amplifier. *Contact gomspace for further instructions.*
- All four corner holes are connected to GND so make sure there are no short-circuit from these holes to other electrical potentials.
- If operated at temperatures below 0 deg. C in atmospheric air make sure the board is cleaned properly with IPA before so that the unavoidable condensing upon de-freezing does not cause electrically conductive ionised water to form on the circuits.

Connectors

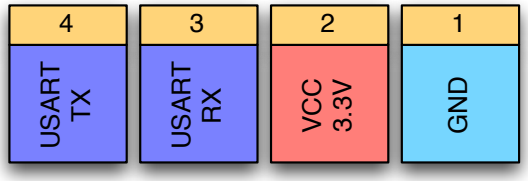
The U482C is equipped with the standard CubesatKit Stack Connector (H1, H2) plus a debugging connector, RF connector and a JTAG connector for firmware upgrades.



Debug Connector

The Debug connector can be used to power the unit and to access a console interface to control and monitor the system - mainly used by GomSpace for testing and check-out. The functionality is self-explanatory and changes a bit with different software revisions as it is mainly meant for testing. The interface is a 3.3V (CMOS level) UART (serial port) running at 500 kbaud.

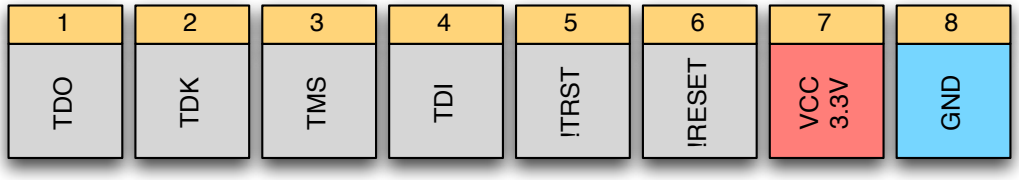
Debugging Connector Pinout



JTAG Connector

A Molex PicoBlade connector is used for firmware upload. No further documentation is provided unless for some reason it is necessary to upload firmware without the presence of a GomSpace engineer.

JTAG Connector Pinout



RF Connector

The RF connector is a right-angle SMA (TYCO ELECTRONICS 5-1814400-1).

Stack Connector

The following table shows the pinout for the CubeSat Kit Connector H1 and H2 (see page 7). Pins with red dots are optional (to be agreed upon time of order placement). Some pins are shown multiple times as they can be configured to either of multiple connections.

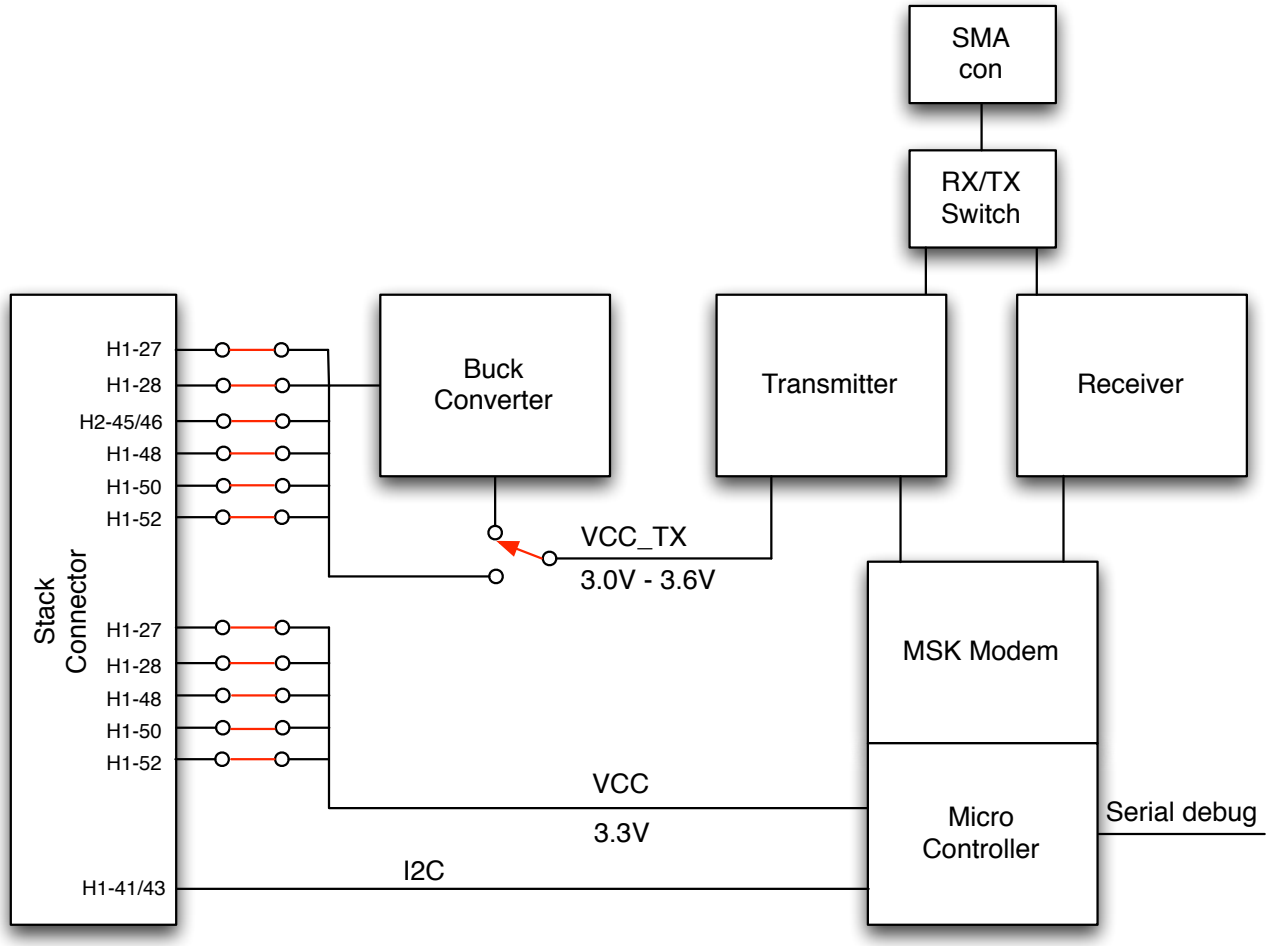
Pin#	Mnemonic	Description	Opt
H1-41	I2C-SDA	I2C serial data	
H1-43	I2C-SCL	I2C serial clock	
H1-48	VCC	3.3V supply	●
H1-50	VCC	3.3V supply	●
H1-52	VCC	3.3V supply	●
H2-27	VCC	3.3V supply	●
H2-28	VCC	3.3V supply	●
H1-48	VCC_TX	Transmitter supply	●
H1-50	VCC_TX	Transmitter supply	●
H1-52	VCC_TX	Transmitter supply	●
H2-27	VCC_TX	Transmitter supply	●
H2-28	VCC_TX	Transmitter supply	●
H2-45	VCC_TX	Transmitter supply	●
H2-46	VCC_TX	Transmitter supply	●
H2-29	GND	Power ground	
H2-30	GND	Power ground	
H2-32	GND	Power ground	
H2-45	V_BAT	Battery voltage for measurement and optional dc-dc converter	
H2-46	V_BAT	Battery voltage for measurement and optional dc-dc converter	

Block Diagram

Below are the principle block diagrams showing the architecture of the transceiver and the baseband/digital circuits.

Main Block

The main block diagram gives an overview of supply and signal lines in the U482C. The red lines are configurable during manufacture and should be specified by the customer using the options sheet provided prior to ordering.



Receiver

The receiver is a super heterodyne receiver with two intermediate frequencies of 21.4MHz and 455kHz. The first IF of 21.4MHz is high enough to give good image frequency suppression by the 437MHz filter after the LNA, while the second IF at 455kHz ensures good channel separation due to a steep +/-6kHz (or optionally +/- 17.5kHz) filter. The FM detector is an integrated device, SA606, with high sensitivity which provides the audio output plus an RSSI measurement. An indication of the RF error on the input is taken from the DC-level of the filtered audio-signal before its AC-coupling.

Transmitter

The direct-FM transmitter consists of a VCO built around a silicon tuning diode which allows the TX frequency to be adjusted by a few kHz by moving the DC level of the AF input up or down. After the VCO is a filter and a buffer leading the low-distortion signal into the power amplifier which outputs 30-34dBm with an efficiency around 50% depending on the temperature. The transmitter frequency is temperature-dependent in a range of maximum +- 10 kHz, so the ground station operator needs to take the frequency drift into account.

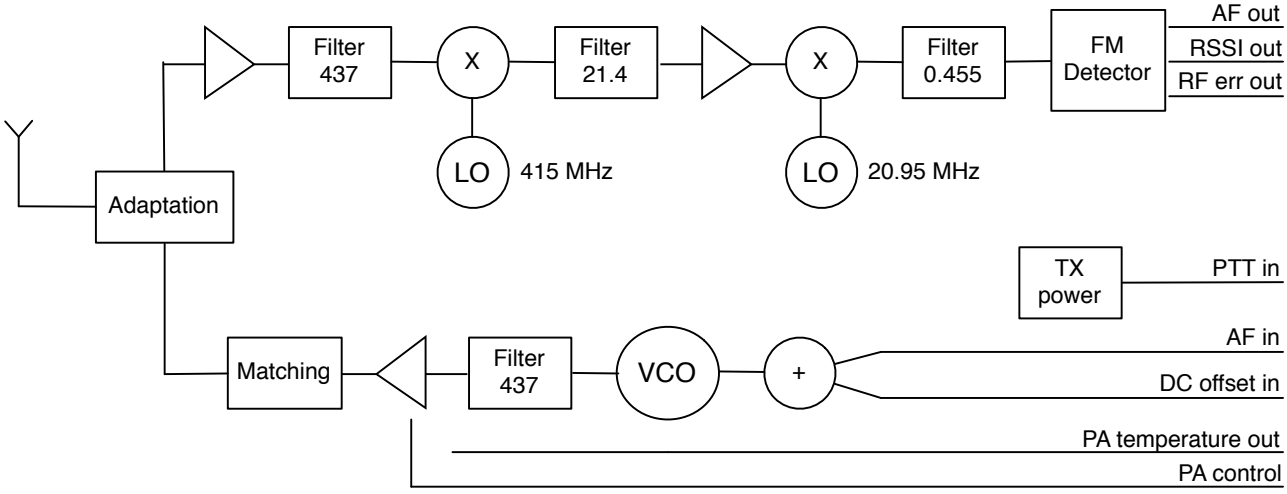


Figure: Transceiver block

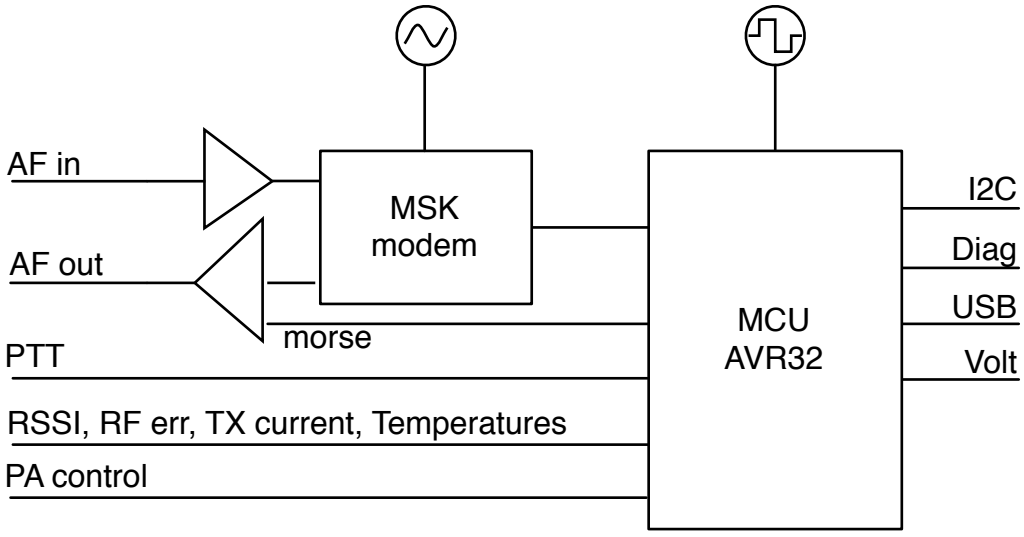


Figure: Baseband and uC

Baseband

The baseband processing is done by an MSK modem allowing 1200, 2400 and 4800 baud half-duplex operation. The transmitter supports 9600 baud. Only use this option if you are sure that your FM receiver has a large enough baseband bandwidth to carry a 15 kHz signal without distortion.

Microcontroller

The heart of the NanoCom system is a 32 bit 60MHz microcontroller, Atmel AVR32, which controls the modem and the transceiver and facilitates communication with other on-board systems via its I^2C interface. When the modem detects a carrier in the baseband signal, the uC starts clocking in the bits that the modem detects searching for a sequence marking the beginning of a packet. In the CCSDS frame format this sequence is called the ASM (attached sync marker) and once detected, the uC receives 256 bytes of data which is passed on to further processing while the modem receive unit goes back to searching for the next ASM. The received 256-byte frames may be decoded by the Viterbi decoder if this option is enabled. If the checksum is correct a CSP packet is extracted from the CCSDS frame and routed to the destination on the internal network.

Temperatures, RSSI and RF error are sampled by the internal 10 bit ADC in the uC and stored such that the user can retrieve the values upon request.

The Morse beacon signal is generated by sinusoidal modulation of a PWM channel by the uC which give an 800Hz tone which is fed directly to the amplifier/mixer along with the MSK output signal. In morse mode, the modem output is always disabled.

Connections

For compatibility with the majority of the products on the pice-satellite market, the system uses a 104-pin stack-connector composed by two SAMTEC ESQ-126-39-G-D connectors (optionally other types can be mounted) and all connections to other on-board systems (except the antenna) is done through this connector.

The NanoCom system is supplied by either a single 3.3Vdc supply or by two supplies: 3.3Vdc for the uC, baseband and receiver plus another regulated supply for the transmitter of up to 3.6Vdc. The system also supports running the power amplifier from an unregulated 5-28Vdc supply on the Vbatt connector, using an on-board switch-mode dc-dc converter. A common ground connection is used. The I^2C bus uses two wires and has optional pull-up resistors on-board. A total pull-up resistance of max 1kOhm is recommended in order to ensure reliable communication at 400kbps.

The RF connection is an sturdy right-angle SMA from TYCO ELECTRONICS providing a reliable connection to the antenna.

Beacon Mode

When the transceiver has been idle for 60 seconds (adjustable), it sends out a morse signal every 20 seconds (adjustable) containing a user-defined text string and a number of housekeeping measurements also defined by the user. The modulation is on-off keying but with a constant carrier to ease tracking of the satellite from ground; instead of CW, an 800Hz audio signal is keyed on and off to produce the morse symbols. Standard timing is 20 WPM (Paris Standard) but can be changed by the user. The period of the beacon can be set by a command.

Electrical Characteristics

Parameter	Condition	Min	Typ	Max	Unit
VCC	Supply voltage	3.20	3.30	3.40	V
VCC Tx	Supply voltage to transmitter	3.00	3.30	3.60	V
Current Consumption, VCC	Power consumption VCC, 3.3V				
- standby	Listening			70	mA
- rx	Receiving			70	mA
- tx	Transmitting			40	mA
Power Consumption, VCC_TX	Power consumption VCC_TX				
- standby	Listening			0.1	mA
- rx	Receiving			0.1	mA
- tx	Transmitting	2000	5000	5500	mW

Command and Data Interface

The I²C interface is used for commanding NanoCom and for receiving housekeeping and status messages. NanoCom operates on the I²C bus in multi-master node, that is either as "slave receiver" or as "master transmitter". A “frame” is defined as a single I²C transmission of any number of bytes between a START and STOP condition (REPEATED_START may be used also).

In a multi-master I2C network, your controller should only enter master-mode (set the clock output) when it wants to send something to another node. Otherwise it should be in slave-receive mode and wait for incoming messages, for example from the radio. This means that you will not have to poll the radio for incoming data, the radio will transmit received frames directly on the I2C bus when it is received.

This mode of operation resembles a typical computer network, and therefore each message is prepended with a packet-header, in order to see its source and destination.

CSP Packet Header

NanoCom implements a network-layer protocol called Cubesat Space Protocol. This protocol defines a simple 32-bit network header, in which the address and destination port must be set correctly for NanoCom to accept the message. The first 4 bytes of any I²C frame must contain this header. A detailed description is found on www.libcsp.org and on http://en.wikipedia.org/wiki/Cubesat_Space_Protocol

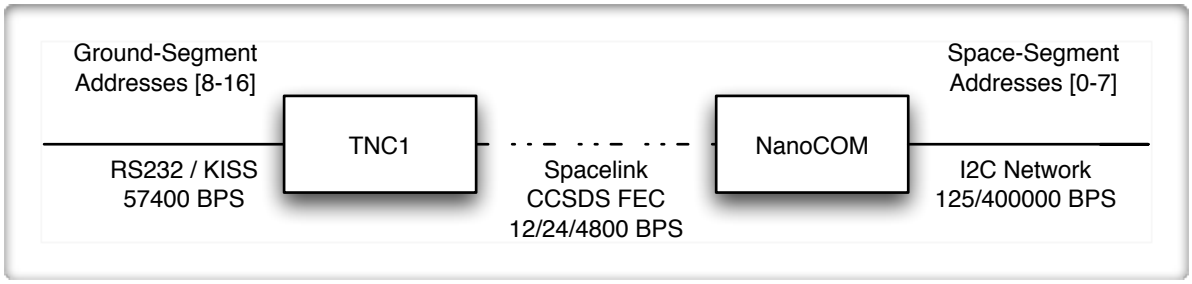
The most important fields to notice is the source and destination, which is used as an OSI layer 3 network header for correct routing of messages, and the port numbers which are used as OSI layer 4 transport header.

Each node will have an address and a set of port numbers it is listening on.

Routing traffic with the NanoCom radio

The 5 bit network address gives up to 32 nodes on the network with addresses from 0 to 31. These addresses has been divided into a space-segment and a ground-segment. These segments are bridged together by the NanoCom radio and Gomspace TNC devices. The default configuration is to divide the network such that all addresses 0 to 7 is space-segment, and addresses 8-15 are ground-segment. (The upper 16 addresses are unused)

So in order to transmit a message from the satellite to the ground station, you must send a valid CSP packet with a destination address from 8-15 to the NanoCom I2C address. On the other hand, in order to transmit a message from the ground-segment and back to the satellite, a CSP packet with a destination address of 0-7 must be sent to the TNC1 device over the KISS serial interface. This is also shown in the figure below:



The advantage of using the NanoCOM device as a router, is that any node on the I2C bus can accept a message from the ground-segment directly, thereby bypassing the on-board computer, for direct sub-system to sub-system communication. This facilitates both the use of a centralized and decentralized architecture.

CSP header and NanoCom data is Big-Endian

Since CSP has been designed to work with a variety of different processors and architectures, all nodes on the network has to agree on a byte-order. The byte order chosen is Big-Endian, which is also used in TCP/IP and other networks. It is often just referred to as Network-byte-order.

If your microprocessor is a Little-Endian platform, you need to byte-swap all incoming data-types larger than one byte. This means that a uint16_t, which is two-bytes, needs to be swapped from big-endian to little-endian before it can be used as a regular 16-bit integer. This of course introduces a bit of processing overhead on the network-code, but it ensures that all nodes can communicate.

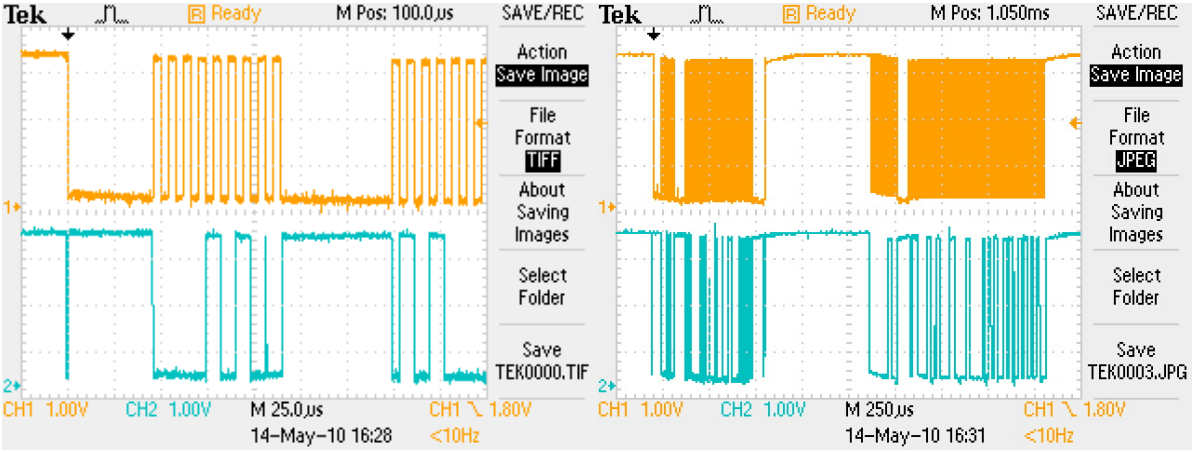
Example:
Here is an example from the Client-library for the NanoPower. This function is used to set the photovoltaic reference.

```
int eps_set_pv_volt(uint16_t pv1, uint16_t pv2, uint16_t pv3) {
        uint16_t pvolt[3];
        pvolt[0] = htons(pv1);
        pvolt[1] = htons(pv2);
        pvolt[2] = htons(pv3);
        return csp_transaction(PRIO_HIGH, NODE_EPS, EPS_PORT_SET_PV_VOLT,
                        10*configTICK_RATE_HZ, &pvolt, 6, NULL, 0);
}
```

Notice that before the request is transmitted, the three integer types is converted into network-byte-order by the *htons* macro. This is a typical and trivial task, but required to keep the transactions compatible.

I2C Examples:

Two examples of the I2C communication is shown below, left: Start of I2C frame to the NanoCom radio. Right: Full CSP ping request and reply.



Notice in the picture to the left, that the I2C address is binary 0000101 which is only 7 bits, the 8'th bit is a 0 which means I2C Write. The following 9'th bit is the acknowledge bit from the NanoCom. Also notice that there are some delays between the start condition and the first byte, and between the address and the data-bytes. This is perfectly normal and is caused by the slower execution speed of the software that must prepare the data for transmission. (This is called clock stretching and may be done both by master and slave when they are not ready).

CSP Addresses:	
COM	5
TNC	9
EPS	2
OBC	1

Command Overview

The following table shows a list of commands accepted by the NanoCom radio. The table should be read as follows: First a mnemonic, or command name is given. This corresponds directly to a specific CSP port number to which the request must be transmitted. For each port, a request and a reply is specified. The row is split into two, the upper-row is the request, and the lower is the reply. For both, a data-type, a data-size and a description is given. Some ports take arguments as simple data-types, some accept complex structures. Below the table, a list of data-structures is given.

This interface is subject to change, and may be further extended and/or improved in the future. Always refer to the specific documentation version delivered with your NanoCom radio.

Mnemonic	Port	Request/Reply Data	Bytes	Description
GET_CONF	7	empty	0	Empty request
		com_config_t;	33	Full radio configuration (see below)
SET_CONF	7	com_config_t;	33	Full radio configuration (see below)
		none	n/a	No reply, can be verified with a GET_CONF
GET_STATUS	8	empty	0	Empty request
		com_status_t;	34	Full radio status (see below)
GET_LOG_HK	9	empty	0	Empty request
		log_hk_t[8];	10*8	Array of housekeeping data elements
GET_LOG_RSSI	10	empty	0	Empty request
		log_rssi_t[10];	8*10	Array of RSSI log elements (see below)
SET_RADIO_PTT	11	uint8_t value	1	ASCII capital ‘U’ for UP, ‘D’ for down.
		none	n/a	no reply to this command.
GET_CALIBRATION	13	empty	0	Empty request
		com_calibrate_t	8	Returns, RSSI and RF-Error.
SET_CONF_RESTORE	14	empty	0	Empty request, will restore configuration
		none	n/a	No reply, can be verified with a GET_CONF

Client library and debug interface

GomSpace provides a C-library to use together with CSP to command the NanoCom radio. This is available in a Ground-Station application called CSP-Terminal, and also as a library for use on the NanoMind OBC.

Furthermore the NanoCom U482 has a serial debugging interface where these client commands are available by typing ‘help’. Access the serial port interface by using 500000 baud 8n1, with no flow control. The debugging connector is the same 4-pin connector used for NanoMind OBC boards.

Detailed Command Specification

The different NanoCom commands are explained in detail below:

GET/SET_CONF

The NanoCom device listens on port 7 for incoming requests to either GET or SET the configuration. If the request is empty, it is interpreted as a GET command, if it contains a valid com_config_t data-structure, it is seen as a SET command. The data-structure is formatted like this:

```
typedef struct __attribute__((packed)) {
        uint8_t do_rs;                  // 1 = turn on reed-solomon FEC, 0 = off
        uint8_t do_random;              // 1 = turn on CCSDS randomization, 0 = off
        uint8_t do_viterbi;             // 1 = turn on viterbi K=7 FEC, 0 = off
        uint8_t tx_baud;                // TX baud, [12,24,48] sent as int.
        uint8_t rx_baud;                // RX baud, [12,24,48] sent as int.
        uint8_t morse_byte;             // 7 first bits configure MORSE output
        uint8_t morse_inter_delay;      // Delay between beacons, in seconds.
        uint8_t morse_pospone;          // Delay before first beacon after activity
        uint8_t morse_wpm;              // WPM after paris standard, dfl = 20
        uint8_t morse_text[20];         // Text to morse, terminate with '\0'
        uint16_t morse_bat_level;       // Minimum battery level, int 500 = 5.00V
        uint16_t hk_interval;           // Interval between HK sampling in seconds.
} nanocom_conf_t;
```

In the above structure the morse_byte, is a bit mask indicating which housekeeping elements to include in the morse beacon. A binary "1" at a given position indicates that the corresponding item is included in the beacon. Bit mask definition (MSB first, LSB last):

- voltage: Measured voltage on pin H2-45

- rf_err: Carrier frequency error in last reception

- rssi: RSSI of last reception

- rx_count: Number of received packets

- temp_a: Temperature of sensor A (CPU)

- temp_b: Temperature of sensor B (PA)

- tx_count: Number of transmitted packets

- N/A: Reserved

Default: 0x80 (="voltage" enabled)

Notice that the two last elements, morse_bat_level and hk_interval are 16 bits which mean they must be transmitted BIG-endian, that is, most significant byte first.

When sending this command the values are written to the run-time configuration and saved to the non-volatile FLASH memory, and restored upon reboot.

GET_STATUS

Transmitting any message to port 8 is replied with a data structure which looks like this:

```
typedef struct __attribute__((packed)) {
        uint32_t bit_corr_tot;              // Total bits corrected by the viterbi algorithm
        uint32_t byte_corr_tot;             // Total bytes corrected by the RS(223,255) alg.
        uint32_t rx;                        // Total packets detected
        uint32_t rx_err;                    // Total packets with error
        uint32_t tx;                        // Total packets transmitted
        int16_t last_temp_pa;               // Last PA temperature in [C]
        int16_t last_temp_txo;              // Last TXO temperature in [C]
        int16_t last_rssi;                  // Last detected RSSI [dBm]
        int16_t last_rferr;                 // Last detected RF-error [Hz]
        int16_t last_txcurrent;             // Last TX current [mA]
        uint32_t bootcount;                 // Total bootcount
} nanocom_data_t;
```

Again, all fields of the data-structure is converted to Big-endian before transmitting on the network.

GET_LOG_HK

The NanoCom Radio records Housekeeping information at specific set intervals. This means that in addition to the “current” status information obtained though the GET_STATUS command, a set of historical samples can be obtained from the GET_LOG_HK command. Due to a limit on message size per packet, 8 samples is returned. The datatype looks like this:

```
typedef struct __attribute__((packed)) {
        uint32_t time;                      // CPU Timestamp (processor Ticks since boot)
        int16_t temp_a;                     // Temperature of sensor A in [C]
        int16_t temp_b;                     // Temperature of sensor B in [C]
        uint16_t batt_volt;                 // Battery voltage, int 712 = 7.12 V
} hk_t;
```

GET_LOG_RSSI

Every time the receiver detects the beginning of a frame, it logs the RSSI and RF-error to a ring-buffer. The contents of this buffer can be obtained by the GET_LOG_RSSI command. The answer will be 10 log_rssi_t types:

```
typedef struct __attribute__((packed)) {
        uint32_t time;                      // CPU Timestamp (processor Ticks since boot)
        int16_t rssi;                       // Measured RSSI [dBm]
        int16_t rferr;                      // Measured RFerr [Hz]
} log_rssi_t;
```

SET_RADIO_PTT

The RADIO_PTT port is used to key-up the transmitter during debugging. Send an ASCII uppercase ‘U’ to Key-up the transmitter, send anything else to turn it off again.

CALIBRATE

Request a sampling of the RSSI and RF-Error right now, and return both the RAW ADC values and the converted values for checking. This command is used by GOMspace during development to calibrate your RSSI and RF-error reading. However it can be used afterwards to request sampling of background noise. The datatype is:

```
typedef struct __attribute__((packed)) {
        int16_t raw_rssi;                       // The RAW ADC value of the RSSI
        int16_t con_rssi;                       // The converted value of the RSSI [dbm]
        int16_t raw_rferr;                      // The RAW ADC valued of the RF-ERR
        int16_t con_rferr;                      // The converted value of the RF-ERR [Hz]
} com_calibrate_t;
```

RF Transmission Format

NanoCom uses CCSDS to transmit and receive packets over the spacelink. When initiating transmission, a preamble of alternating 101010101... must be sent for at least 40ms. NanoCom uses 200ms preamble as default.

CCSDS Frame Format

ASM	Data
4 bytes	256 bytes

The first 4 bytes of each packet is an Attached Sync Marker containing 0x1ACFFC1D used to synchronize the beginning of each packet. The data payload is always 256 bytes when the full error correcting codes are enabled. Note: The system uses a microcontroller hardware carrier detect, and therefore the ASM is not encoded by the Viterbi.

Forward Error Correction

If the optional Viterbi, Reed solomon or Randomization is enabled, this only applies for the Data field. The sync-word is transmitted in big-endian byte order without any modification. Thereby allowing to sync regardless of which FEC format is used. However, a bit-error in the sync word may render a frame undetected. Therefore it is recommended to use a correlator with a tolerance of up to 2-3 bit-errors in the sync word. If Reed Solomon is used, which is always recommended, the algorithm will prove if it was a valid frame or not.

Typical performance improvement is 5 dB SNR, for example improving the sensitivity from -115 dBm to -120 dBm in 1200 baud mode, if full FEC is enabled.

If you do not use the Gomspace TNC1 device at your ground-station, and wish to implement the FEC decoder yourself, please contact Gomspace for specific information about the Viterbi symbol table and encoder polynomial.

Reduced FEC modes

If the optional Viterbi coding is disabled, but the reed-solomon code left enabled, a reduced FEC performance is obtained. There are several different modes available depending on the number of features enabled. Any of these features can be enabled/disabled: Viterbi, Reed-Solomon, Randomization and CRC32. This gives the opportunity of 16 different frame formats, which each has its own advantages and disadvantages.

Automatic system restore

In case the NanoCom does not talk with anybody for a period of time, the system will reboot itself. If this has happened several times, the system assumes that it is malfunctioning and will reset to factory defaults. This function can be adjusted by setting the intervals or completely disabling it. (must be specified compile time). In order to prevent the system from rebooting and thereby restoring the default configuration, a ping or any command must be sent from any on-board-computer, or from the ground-station, directly to the NanoCom subsystem (CSP id 0x05). In order to reset the reboot counter, and prevent a config restore, the SET_CONF command must be run at least once per day in order to maintain the configuration.

RF Specifications

Transmitter specifications:

Frequency: 437.???MHz (??? = defined by customer)
Output power: 27-34dBm
Spurious transmissions: < 0.025μW (except 2nd and 3rd harmonic)
TX-on delay: < 5ms
Modulation: FM
Frequency response: 1 Hz ... 6kHz (-3dB)
Modulation voltage: 1Vss (350mVrms) for 3kHz deviation
Distortion: <2% at 3kHz deviation
Temperature range: -20°C ... +70°C (PA up to 80°C)
Temperature stability: <±12ppm

Receiver specifications:

Frequency: 437.??? MHz (??? = defined by customer)
Sensitivity: -125dBm for 1200 baud detection threshold (e.g. -122dBm for 2400 baud with FEC)
Image frequency: -65dB
Adjacent channel: -60dB
Blocking: 50dB
RSSI: Calibrated to 1dB accuracy
RX-on delay: <5ms
Modulation: FM
Frequency response: 1Hz ... 6kHz (-3dB)
AF output voltage: 250mVss (90mVeff) at 3kHz deviation
Distortion: <2% at 3kHz deviation
Temperature range: -20°C ... +70°C
Temperature stability: <±8ppm

Baseband specifications

Modulation: FFSK/MSK

Baud rates: 1200, 2400, 4800, 9600

Mark/space frequencies:

- 1200 baud: one cycle of 1200Hz represents a logic ‘1,’ one-and-a-half cycles of 1800Hz represents a logic ‘0.’
- 2400 baud: one-half cycle of1200Hz represents a logic ‘1,’ one cycle of 2400Hz represents a logic ‘0.’
- 4800 baud: one-half cycle of 2400Hz represents a logic ‘1,’ one cycle of 4800Hz represents a logic ‘0.’
- 9600 baud: one-half cycle of 4800Hz represents a logic ‘1,’ one cycle of 9600Hz represents a logic ‘0.’

Tone frequencies are phase continuous; transitions occur at the zero crossing point.

Required SNR: 10dB

Modem device: CMX469AD3.

Operation and Handling

Warnings:

- The NanoPower system employs components based on FETs and therefore requires anti-static handling precautions to be observed. Do not touch or handle the product without proper grounding!

Customization Options

As GomSpace realizes that different applications place different requirements to a communications system, options to be agreed upon time of order placement include:

- All pin-connections indicated with red dots
- Transmitter supply voltage
- Conformal coating using NASA approved CV-1152 silicone coating (at an extra cost)
- More options may be available at the customers request
- Specific software changes (at an extra cost)

Quality Assembly

GomSpace space hardware is hand-assembled in a procedure where all parts are cleaned with IPA and then soldered in an anti-static environment to “IPC-A-610 Class 3” specifications. All solder-work is done under a microscope with tin-lead 63/37 using rosin flux. All solder joints are re-checked for class 3 compliance and the PCB is finally cleaned with IPA and ready for testing.

Materials

The circuit board of a NanoCom is the single most critical part of the system and is not allowed to suffer a mechanical failure. Therefore, we use polyamide P97 as the base material in the PCB because of its excellent temperature tolerance resulting in very little thermal expansion and hence very little stress on the copper and the via platings when the board is subjected to even broad temperature cycles. This also means, that the soldering process does not effect the peel-strength of the tracks and pads resulting in the best possible mechanical base for the electronic components.

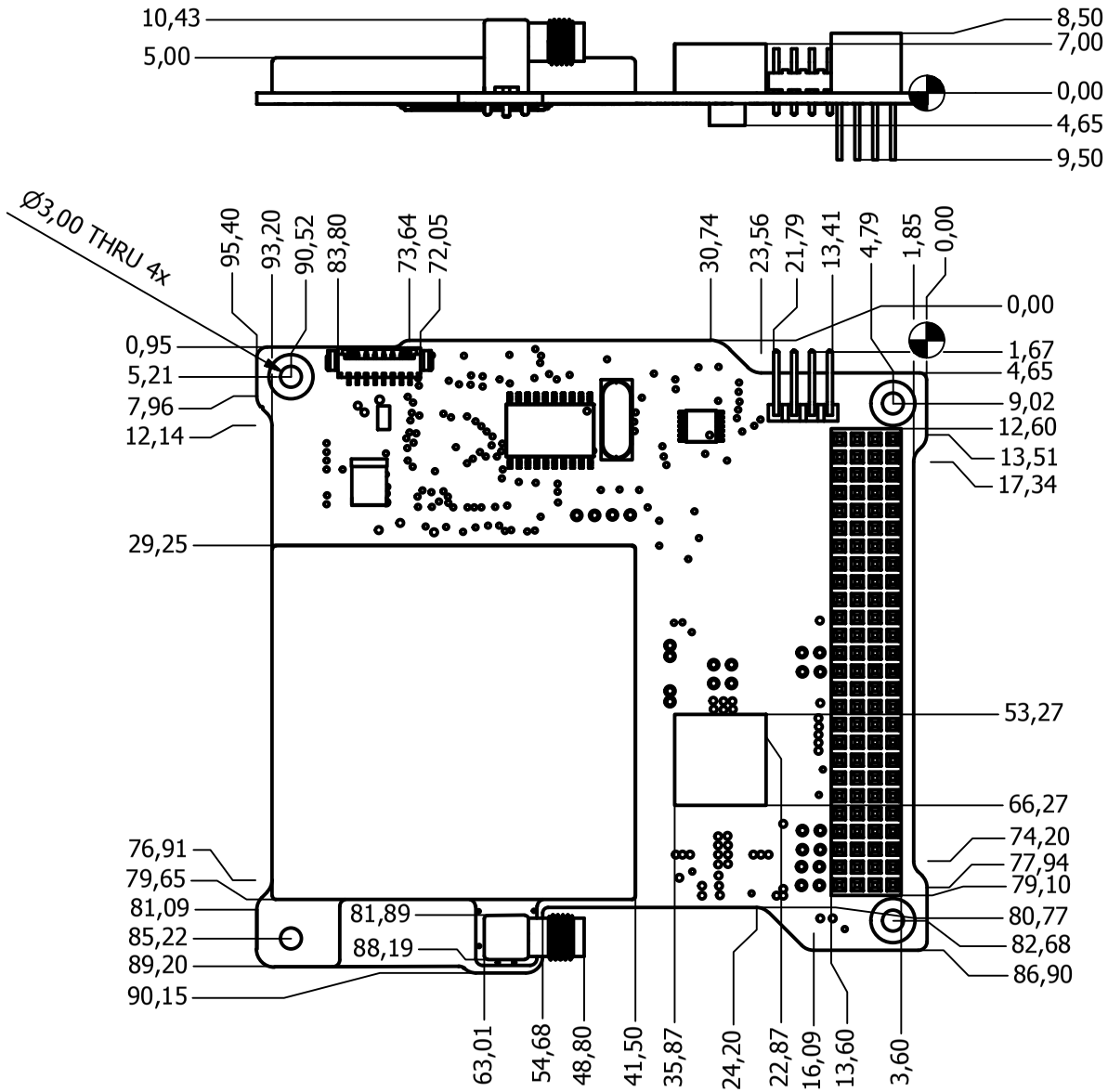
The surface of the conducting areas is covered in tin/lead for optimum solderability with allowance for small differences thermal expansion without the risk of exceeding the maximum tensile strength of joins. Lead-free solder is never allowed in a GomSpace flight-grade product.

The EMI shield on the transceiver is made from 0.2mm tin-plated steel TBD in order to ensure proper RF shielding and providing a level of radiation shielding as well. The shield also helps to serve as heat sink and a thermal radiator transport heat away from the transmitter power electronics.

Physical Dimensions

See last page. Dimensions are given in mm.

PCB-type is glass/polyamide, 6 layer, tin-lead fused surface, 1.60 mm thick. Mass: 75g.



Designed by	Checked by	Approved by	Date		Date	
KKL					12-10-2011	

GomSpace ApS Niels Jernes Vej 10 DK-9220 Aalborg East Denmark	NanoCom U482C		
	PID: 0037 CID: 0977	Edition	Sheet 1 / 1