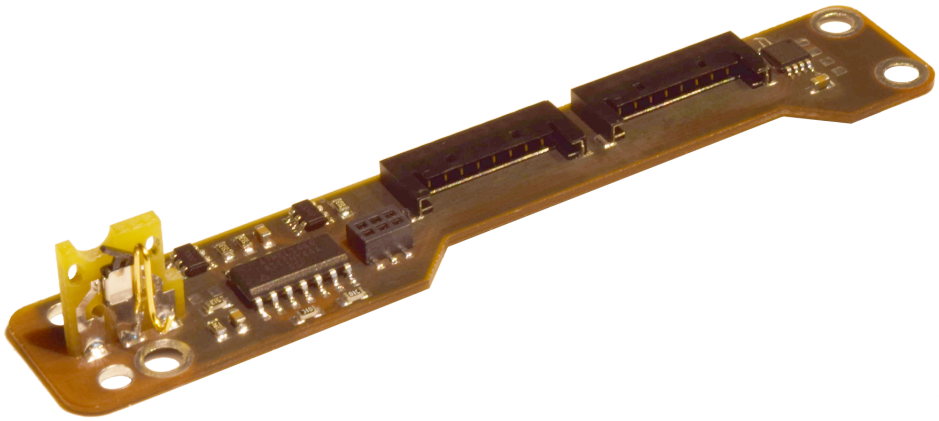
GOMSPACE



NanoUtil

AR6

Manual

NanoUtil Antenna Release Mechanism

GomSpace A/S | Langagervej 6 | DK-9220 Aalborg East | Denmark
gomspace.com | sales@gomspace.com | T: +45 71 741 741

Table of Contents

1. Changelog

Table 1.1: Changelog

Date Description	Revision	Author	Description
23/01-2018	3.1.0	JESM/ANM	First release

2. Introduction

The primary function of the GomSpace NanoUtil AR6 is to provide a release mechanism for VHF antennas in GomSpaces 6U structure.

The AR6 is placed on the 6U side of a structure. On each side, there are four slots allocated for the AR6'es. All can be set in series through the connectors on the PCB.

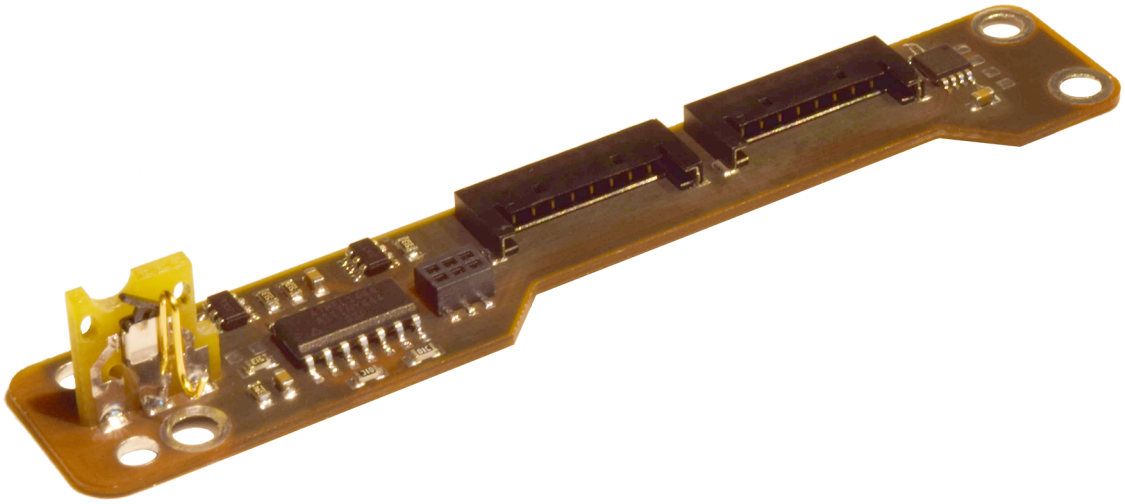


Fig. 2.1: NanoUtil AR6

2.1 Unpacking and handling precautions

Warning: The AR6 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.

Warning: While tying down the antennas from ANT-6F to the AR6, be very cautious about the switch. Applying a very small force in the wrong direction, to the arm on the switch, will result in breaking the switch. Since the switch is glued down to the PCB, it is not an easy fix.

3. Hardware

3.1 Mounting

This section describes how to mount the AR6 on the 6U nanosatellite structure.

3.1.1 Standard Tightening Torque for Screws

Below is shown a table for GomSpace suggested standard tightening torque for screws.

Screw Diameter [mm]	**Torque** [Nm]
2.5	2.0

3.1.2 Mounting the AR6 on the NanoStructure 6U

The AR6 is placed on the 6U side of a structure. On each side, there are four slots allocated for the AR6'es - see Fig.3.1. All can be set in series through the connectors on the PCB.

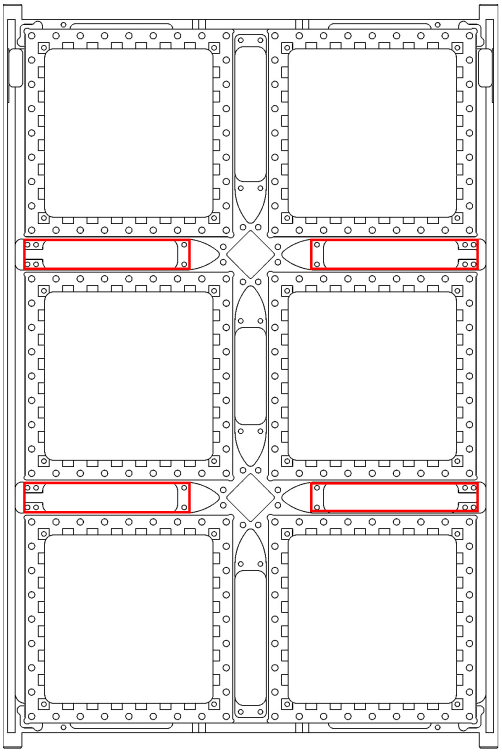


Fig. 3.1: Where to mount the AR6s on the 6U structure

Note: Usually, each antenna is only held in place by one AR6. So, there is a total number of four AR6 on a 6U structure. It is highly recommended to use the slot which has the biggest distance to the antenna board to hold the antenna rods at their ends.

Fig.3.2, shows how to mount the AR6 board on the 6U structure.

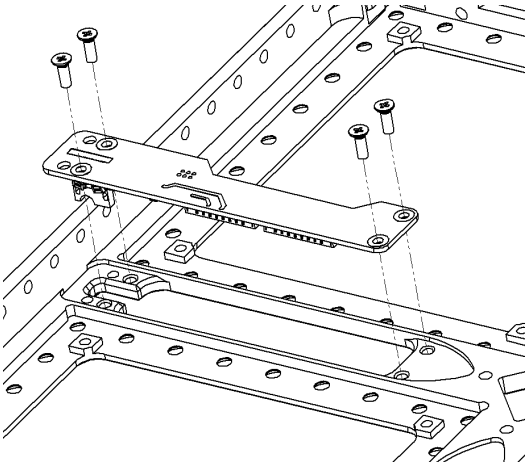


Fig. 3.2: Schematic view of how to mount the AR6 on the 6U structure

The bottom of the AR6 PCB contains the countersunk holes for mounting. Except for the antenna release, the whole PCB will be covered from the outside by solar panels.

Fig.3.3 and Fig.3.4 show an AR6 mounted in a 6U structure.

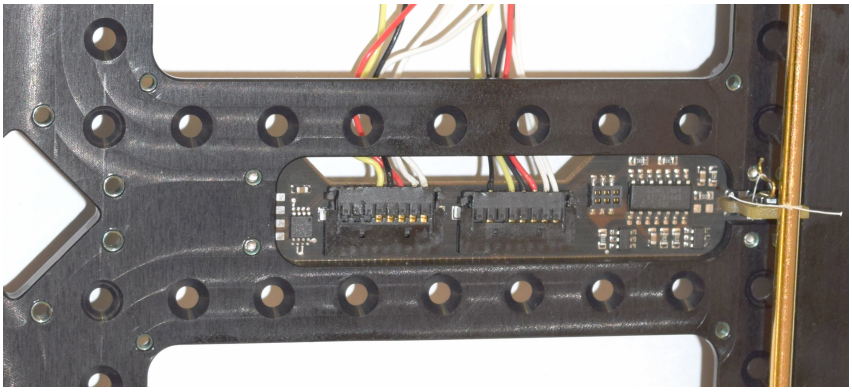


Fig. 3.3: AR6 mounted in the 6U structure (top)

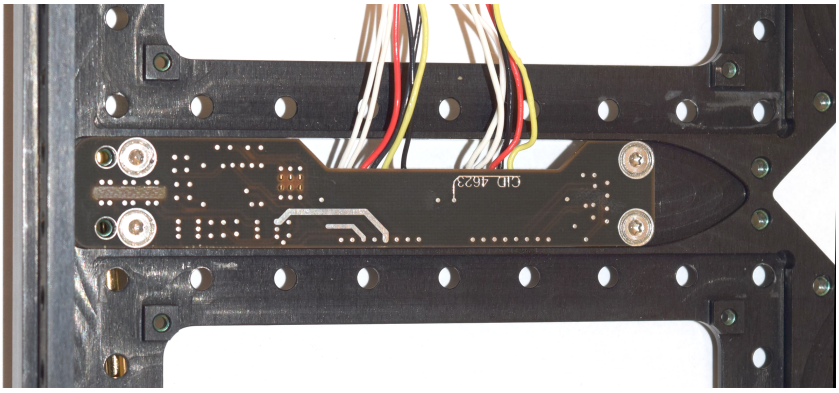


Fig. 3.4: AR6 mounted in the 6U structure (bottom)

4. Software

The NanoUtil AR6 is intented to be used together with the libgssb software library. A part of libgssb comes with the AR6. It implements the GSSB protocol and functions on top of a I^2C layer.

In addition, GomSpace has developed a console-like interface called GomSpace Shell (GOSH), which provides a simple but extensive debug and configuration interface. GOSH is a general feature present on several of GomSpace's products.

Note: The AR6 board itself does not have GOSH installed. The AR6 GSSB GOSH commands can however be run via the NanoPower P60 Dock.

The console provides a text-interface to a given input/output stream such as a serial port. GOSH is described in detail in the GOSH manual. If this manual has not been provided upon purchase, please contact GomSpace support.

4.1 Getting Started

This section contains a brief description of how to establish a connection with AR6 by using GOSH.

Note: The AR6 can be connected to either a NanoPower P60 Dock, NanoCom ANT6F-UHF or another NanoUtil AR6.

4.1.1 Communicating via the P60 Dock

Follow the steps below to communicate with the AR6.

1) Connect the AR6 to the P60 Dock

In order to gain access to GOSH, one has to connect the AR6 to the P60 Dock - see the AR6 and P60 Dock datasheets.

2) Connect a serial cable to the P2 connector of the P60 Dock

3) The serial communication program (minicom/tio) should be configured as follows:

- baudrate: 500.000 baud, 8n1
- disable flow-control

4) Run the `gssb ar6` command to list all sub-commands:

```
p60-dock # gssb ar6

  burn
  stop
  status
  reset_count
  settings
  reboot
  temp_internal
```

5) Run the `gssb ar6 status` command to get telemetry data:

```
p60-dock # gssb ar6 status
```

4.2 Interfaces

The AR6 supports GSRB over I^2C. The AR6 commands can be run directly via GOSH or the AR6 Client API can be used to develop custom mission code.

4.2.1 GSRB over I^2C

The AR6 is a GSRB/GSSB compatible device. It uses GSRB (GomSpace Release Bus) over I^2C, which is a variant of GSSB (GomSpace Sensor Bus). The bus carries the supply for AR6 in form of GND and 3.3 V and the two I^2C signals SDA and SCL are used for communication.

Note: The AR6 has been tested with standard mode (100 kbit/s) and full speed (400 kbit/s)

GomSpace has developed a library called 'libgssb', which makes it easy to communicate with GSSB compatible devices. The library implements commands for operating the AR6.

4.2.2 GOSH

The GOSH commands are accessible through the GSSB client, which gives access to all AR6 sub-commands:

Note: The examples given in this section are based on the AR6 being connected to the P60 Dock. See the P60 Dock and AR6 Datasheets for how to connect the two devices.

```
p60-dock # gssb ar6

  burn
  stop
  status
  reset_count
  settings
  reboot
  temp_internal
```

Note: The following GOSH commands require running the `gssb setaddr` command first:

- gssb ar6 burn
- gssb ar6 status
- gssb ar6 reset_count
- gssb ar6 reboot
- gssb ar6 temp_internal

Example: If an AR6 is located at I^2C address 5, run `gssb setaddr 5` before running `gssb ar6 status`.

Below is a description of AR6 sub-commands.

Antenna Deployment System (burn)

The GOSH command to burn on an AR6 board is:

```
p60-dock # gssb ar6 burn 3
```

The `burn` command only takes one argument, which is burn time in seconds. Running the above command will burn on the AR6 located at the I^2C address given in the `gssb setaddr` command.

Warning: The max burn time is 6 seconds for lab testing. Setting a wrong burn duration can damage the resistors used on the thermal knifes or ICs on the PCB.

It is not recommended using the burn command multiple times before launching the satellite in to space.

Stop burning (stop)

Stops burning immediately on the AR6 at the I^2C address given in the `gssb setaddr` command.

Seeing the status (status)

It is possible to check the status of the AR6 board. There are three status functions, which each gives different telemetry. The telemtry data can be access by running the GOSH command:

```
p60-dock # gssb ar6 status
```

The normal response for this command looks like the following example.

```
Antenna release
        Burn state:                             Idle
        Burn time left:                         0
        Antenna state:                          Released
        Burn attempts:                          0
Backup
        State:                                  Waiting to deploy
        Seconds to backup deploy:               3594
Board
        Seconds since boot:                     6
        Number of reboots:                      2
```

The telemetry data is described in the table below.

Table 4.1: Telemetry

Parameter	Description
Burn state	Can be *Idle* or *Burning*
Burn time left	Number of seconds until burning stops
Antenna state	Can be *Released* or *Not released*
Burn attemps	Number of deploy attemps
Backup	Backup deployment status
State	Can be 'Not active', 'Waiting to deploy', 'Deploying', 'Finished successful' or 'Finished unsuccessful'
Board	Board status
Seconds since boot	Number of seconds since powered on
Number of reboots	Number of reboots since last reboot count reset

Reset (reset_count)

Resets burn counters and reboot counter on AR6.

Configuration (settings)

Warning: The max burn duration is 10 seconds. Setting a wrong max burn duration can damage the resistors used on the thermal knifes or ICs on the PCB. If in doubt, contact GomSpace Support.

The AR6 is configured with the GOSH command:

```
p60-dock # gssb ar6 settings
```

If no arguments are supplied then the current configuration is fetched and displayed. Below is the normal response shown.

```
Backup deploy settings:
    Minutes until deploy:                   60
    Backup deployment:                      Not active
    Max burn duration:                      10
```

The following command shows how to modify the settings:

```
p60-dock # gssb ar6 settings 60 1 10
```

The parameters are described in the following table:

Table 4.2: Settings

Parameter	Description
Minutes until deploy	Minutes until the board will burn for max burn duration on each thermal knife [0-5000]
Backup deployment	Can be 'Not active (0)' or 'Active (1)'
Max burn duration	The burn duration for backup deploy and also general max burn duration

Note: The AR6 board should be rebooted for changes to take effect.

Soft reboot (reboot)

Resets burn counters and reboot counter on AR6.

Sensors (temp_internal)

The AR6 comes with a temperature sensor placed in the MCU on the board. It is read on a AR6 board with the GOSH command below:

```
p60-dock # gssb ar6 temp_internal
```

The output is in degree Celcius.

4.2.3 AR6 Client API

The AR6 Client API can be used to control the AR6 from custom mission code. The Client API can be found in `gssb_ar6.h`.

All the GOSH commands in chapter *GOSH* are based on the AR6 Client API.

The AR6 API header is shown below:

```
1  #ifndef _GS_GSSB_AR6_H_
2  #define _GS_GSSB_AR6_H_
3
4  #include <gs/gssb/gssb_dev.h>
5
6  /**
7     Get ar6 internal temperature.
8
9     Gets the ar6 temperature inside MCU.
10
```

```
11      @param[in] addr I2C address.
12      @param[in] timeout_ms Timeout in milliseconds.
13      @param[out] temp_out Pointer temperature.
14  */
15  gs_error_t gs_gssb_ar6_get_internal_temp(uint8_t addr, uint16_t timeout_ms, int16_t *␣
    ↪temp_out);
16
17  /**
18      Burn ar6.
19
20      Burns on the ar6.
21
22      @param[in] addr I2C address.
23      @param[in] timeout_ms Timeout in milliseconds.
24      @param[in] duration Duration in seconds to burn.
25  */
26  gs_error_t gs_gssb_ar6_burn(uint8_t addr, uint16_t timeout_ms, uint8_t duration);
27
28  /**
29      Stop ar6 burn.
30
31      Stops burning.
32
33      @param[in] addr I2C address.
34      @param[in] timeout_ms Timeout in milliseconds.
35  */
36  gs_error_t gs_gssb_ar6_stop_burn(uint8_t addr, uint16_t timeout_ms);
37
38  /**
39      Get ar6 release status.
40
41      Gets the ar6 release status.
42      It tells if burning.
43      It tells burn time left.
44      It tells release state.
45
46      @param[in] addr I2C address.
47      @param[in] timeout_ms Timeout in milliseconds.
48      @param[out] channel_status Pointer to release status.
49  */
50  gs_error_t gs_gssb_ar6_get_release_status(uint8_t addr, uint16_t timeout_ms, gs_gssb_ar6_
    ↪release_status_t * status);
51
52  /**
53      Get ar6 backup status.
54
55      Gets the ar6 antenna backup status.
56      It tells backup deployment state.
57      It tells time left until backup deploy.
58
59      @param[in] addr I2C address.
60      @param[in] timeout_ms Timeout in milliseconds.
61      @param[out] channel_status Pointer to backup status.
62  */
63  gs_error_t gs_gssb_ar6_get_backup_status(uint8_t addr, uint16_t timeout_ms, gs_gssb_
    ↪backup_status_t * backup_status);
64
65  /**
66      Get ar6 board status.
67
68      Gets the ar6 board status.
69      It tells reboot count.
70      It tells uptime.
```

```
71
72      @param[in] addr I2C address.
73      @param[in] timeout_ms Timeout in milliseconds.
74      @param[out] board_status Pointer to board status.
75  */
76  gs_error_t gs_gssb_ar6_get_board_status(uint8_t addr, uint16_t timeout_ms, gs_gssb_board_
    ↪status_t * board_status);
77
78  /**
79      Get ar6 backup settings.
80
81      Gets the ar6 backup settings.
82      It tells backup deployment state.
83      It tells backupdeployment delay.
84      It tells maximum burn time, for both backup burn and normal burn.
85
86      @param[in] addr I2C address.
87      @param[in] timeout_ms Timeout in milliseconds.
88      @param[out] settings Pointer to settings.
89  */
90  gs_error_t gs_gssb_ar6_get_backup_settings(uint8_t addr, uint16_t timeout_ms, gs_gssb_
    ↪backup_settings_t * settings);
91
92  /**
93      Set ar6 backup settings.
94
95      Sets the ar6 backup settings.
96      It activates/deactives backup deployment.
97      It sets backupdeployment delay.
98      It sets maximum burn time, for both backup burn and normal burn.
99
100     @param[in] addr I2C address.
101     @param[in] timeout_ms Timeout in milliseconds.
102     @param[in] settings Pointer to settings.
103 */
104 gs_error_t gs_gssb_ar6_set_backup_settings(uint8_t addr, uint16_t timeout_ms, gs_gssb_
    ↪backup_settings_t settings);
105
106 /**
107     Reset ar6 counters.
108
109     Resets burn counters and reboot counter on ar6.
110
111     @param[in] addr I2C address.
112     @param[in] timeout_ms Timeout in milliseconds.
113     @param[out] settings Pointer to settings.
114 */
115 gs_error_t gs_gssb_ar6_reset_count(uint8_t addr, uint16_t timeout_ms);
116
117 #endif
```

5. Disclaimer

The information in this document is subject to change without notice and should not be construed as a commitment by GomSpace. GomSpace assumes no responsibility for any errors that may appear in this document.

In no event shall GomSpace be liable for incidental or consequential damages arising from use of this document or the software and hardware described in this document.