GSE
GLOBAL SATELLITE ENGINEERING

GSatMicro

Operations Manual

sales@gsat.us : gsat.us

GSatMicro Operations Manual

Specifications

- Dimensions: 55.5mm x 49.25mm x 89mm including battery, modem and antenna
- Dimensions: 41mm x 45mm x 17mm (OEM version)
- SiRFstarIV GPS with an amazing -163dBm sensitivity
- AES 256-bit encryption supported (optional)
- Rechargeable Lithium Polymer battery (2.4 Ah, up to 1000 position reports)
- Integrated omnidirectional helical antenna (high gain ceramic patch antenna for OEM model)
- Over the air configuration of terminal
- USB, RS-232 and Bluetooth connections for programming
- Full 2 way communications network
- Bluetooth 4.0 (LE) smartphone connectivity for messaging and configurations
- Truly global coverage with the Iridium satellite network.

Interfaces

- USB DC Input (4.5V to 5.5V)
- High Voltage DC Input (7.5V to 36V) (protected to 40V DC)
- USC Interface (Power, Serial Console/Logging, Firmware Update)
- RS232 Interface (Serial Console/Logging)
- Bluetooth 4.0 Low Energy
- 2 Relay Outputs @ 250Ma (Open Drain)
- 2 Analog Inputs (0V to 30V DC)

Getting Started

Out of the box, your unit should be in off or storage mode. In order to power it up and start transmitting you will need to hold down the power button until power and GPS lights "fade up" (dark to bright). When you release the button it will start a position report cycle. The states of which are described below:

While the unit is powered on, it will go through progressive stages of acquiring, transmitting, and sleeping, as indicated by the Power and Message LED's.

1) Upon power up or wakeup from sleep, the Power LED will "fade up" until initialization is complete, after this the LED will indicate battery state (flashing up to 5 times if the battery is full).

2) Next, when GPS is enabled the GPS LED will fade up until satellites are acquired, at which point the number of flashes indicates the number of satellites acquired (up to 5) until a fix achieved, then the LED will stay on, solid.

3) Next, the Iridium modem will be enabled and the Satellite LED will fade up until signal is acquired. Once iridium signal is acquired, the LED will flash 1-5 times indicating signal level. The LED will stay on solid once a transmission has been successful.

4) If only one report or message was pending the unit will then sleep until the next transmission. On battery all LEDs will turn off. If plugged into an external power source, LEDs will reflect last state achieved. The last status can be displayed by briefly pressing the power button if the unit is on battery.

If transmission was successful your unit should appear on the map at its current location.

See the "Terminal Behavior" section below for more details on button and LED behavior.

Terminal Behavior

The front interface panel has three buttons and 5 lights. Behavior is described below.

Button Behavior

Button	Description
⏻ Power	<1 second: flash LEDs showing last state before sleep >=1 second: From off/storage, resume transmit/sleep (Power and GPS LEDs will "fade up" to indicate this will happen when button is released). From sleep or wake, enter off mode (power and GPS LEDs will "fade down") >=10 seconds: From any mode. Enter storage mode (Power, GPS, Satellite and Message LEDs will "fade down" to indicate the mode is selected)
✓ Check-in	Set the check-in bit for next transmission.
⚙ Configuration	Hold for >=10 seconds (message LED will fade up) to enable Bluetooth pairing. Hold for >=10 seconds while in pairing mode to disable pairing (led will fade down)
⚙ ✓ Check-in+ Configuration	If ALERT mode is disabled, enable (Alarm will flash), transmit immediately. If ALERT is enabled, disable.

When you depress the Power button for more than 1 second, the unit will either turn Off if it is currently On and go into a deep sleep without transmitting, or if it is currently sleeping, will wake the unit and immediately transmit a position.

When you depress the Check-In button WHILE the unit is on, the next successful transmission report will contain a bit flag indicating that it is a Check In message. This mode will be cleared once a successful transmission is made. The MSG light will begin flashing slowly.

When you depress the Configuration + Check In button together for ALERT mode, the unit will IMMEDIATELY begin transmitting it's current location (even if it was sleeping). The MSG light will begin flashing rapidly. The unit will also use change the sleep interval to the value specified by "ALERT sleep" time in seconds indicated in the "settings" function below. To exit ALERT mode, depress the Power + Check In button together again and the MSG light will turn off again.

LED modes

⚡ **Power**	**Description**
1-5 flashes	1: 0-19% battery 2: 20-39% battery 3: 40-59% battery 4: 60-79% battery 5: 80-100% battery When charging between flashes LED will glow with diminished brightness. When not charging, LED is off between flashes. When on external power and battery >=95% power LED will be solid on.
Fade up	Powering up
Fade down	Powering Down
Off	Off

GPS **GPS**	**Description**
Slow flash	Powered, no fix
1-4 flashes	1-4 GPS satellites
5 flashes	>=5 GPS satellites
On	GPS fix acquired
Off	Off

🛰 **Iridium/ Satellite**	**Description**
Fade up	Radio on, no signal
1-5 flashes	Iridium signal acquired, flash count corresponds to signal. More flashes, better signal
On	Transmitted message
Off	Off, or powered with no radio

! Alert (!)	Description
Fast Flash	ALERT mode, cleared on successful transmit
Off	Not on ALERT mode

✉ Message	Description
Medium Flash	Check-in mode, cleared on successful transmit
Off	Not in check-in mode
Fade up	Bluetooth pairing enabled (overrides check-in)
Fade down	Disabling Bluetooth check-in

Lua Functions

These options can be sent to the terminal via the "Send Command" function within the tracking platform, or entered through the RS232 serial connection, or the USB cable via CDC mode.

NOTE: Commands sent from the tracking platform will not show command output anywhere other than on the serial console. However, messages can be composed using the tracking platform’s functions and the Iridium transmit function can be used to send them.

GPS functions

```
// GPS Power
gps.setpower( {mode} )
NOTE: Iridium modem must be powered on for GPS to get signal

mode:
1: on and initialize
0: off

// Get GPS Time
{second}, {minute}, {hour}, {day}, {month}, {year} = gps.gettime()
{epochseconds} = gps.getunixtime()

// Get Position
{latitude}, {longitude}, {speed}, {altitude}, {course}, {hdop} = gps.getpos()
```

Messaging functions

```
// Encode position report in Palmer format
{string} = gsattrack.encposition({latitude}, {longitude}, {speed}, {course},
                                  {battery percent}, {minute}, {altitude},
                                  {epochseconds} )

// Encode email/text message
{string} = gsattrack.enctext( {destination}, {message} )

// Encode settings report
{string} = gsattrack.encsettings()
```

Iridium functions

```
// Power Iridium Modem
iridium.setpower( {mode} )

mode:
1: on
0: off

NOTE: Must be called with mode 1 to enable signal for GPS functionality

// Enable Iridium Radio
iridium.setradio( {mode} )

mode:
1: on
0: off

iridium.transmit( {string} )

string: arbitrary string to transmit over SBD

Example:
iridium.transmit(gps.encode(lat,lon,speed,course,batp,min,alt))
```

GPIO functions

```
{ variable } = gpio.wakereason()

values:
gpio.UNKNOWN: no matching wake reason
gpio.POWERUP: unit was just powered on
gpio.RESETPIN: unit was reset using the reset button
gpio.WAKEPIN: unit was woken using the wake pin
gpio.RTC: unit was woken by RTC
gpio.POWERCONNECTED: unit was woken when power was connected
```

Compass functions

```
// Compass heading in degrees
{ variable } = compass.getheading()

// Monitor compass and accelerometer values
{ variable } = compass.setmonitor( {mode} )

mode:
1: on
0: off
```

```
Example output:
A#51,-120,-157,-213,-288,48,17168

A#<heading>, <mag. field x>, <mag. field y>, <mag. field z>, <accel x>, <accel y>, <accel z>
```

Accelerometer functions

```
// Temperature in degrees Celsius
{ variable } = accel.gettemp()

// Unit orientation
{ variable } = accel.getorientation()

values to compare with:
accel.UP_SX (0x44): Y-low
accel.UP_DX (0x42): X-high
accel.DW_SX (0x41): X-low
accel.DW_DX (0x48): Y-high
accel.TOP (0x60): Z-high
accel.BOTTOM (0x50): Z-low
accel.NA (0x00) : ??
```

NOTE: need to add diagram here showing orientation of unit and maybe change the names of these to be more intuitive.

Battery functions

```
//Battery percentage integer value from 0 to 100
print(bat.getpercent())

//Battery voltage in millivolts IE: 4000 = 4.0V
print(bat.getvoltage())

// Enable/disable periodic battery state logging
bat.setmonitor( {mode} )

mode:
1: on
0: off
```

Settings functions

These settings can be configured through the "Send Command" function on the GSatTrack site (found under the down arrow menu for the GSatMicro asset of interest). The "Get Parameters" function from this screen can request the current parameters stored on the device. Sending parameter requests

```
through the site to the unit will not automatically update fields on the site. "Get Parameters" can be
used
to confirm the current configuration state of the unit.


//Set variable into flash
settings.flash( {setting}, {value} )

//Read variable from flash
{variable} = settings.flash( {setting} )

setting:
settings.sleep // Seconds to sleep when in NORMAL mode (between cycles) (range: 0 - ( 2^31 - 1 ) )
settings.sos_sleep //Seconds to sleep when in SOS mode (between cycles) (range: 0 - ( 2^31 - 1 ) )
settings.g_timeout //GPS Timeout in seconds, default 120 (range: 0 - ( 2^31 - 1 ) )
settings.g_hdop //GPS Horizontal Dilution of Precision needed x10, default 20 (range 0 - 500)
settings.g_settle // Seconds to settle GPS after getting a fix, default 15 (range: 0 - ( 2^31 - 1 ) )
settings.i_tx_timeout //Iridium transmit timeout, default 60 (range: 0 - ( 2^31 - 1 ) )
settings.i_signal_timeout //Iridium signal timeout, default 60 (range: 0 - ( 2^31 - 1 ) )
settings.i_tx_retries //Iridium transmit attempts, default 3 (range: 0 - ( 2^31 - 1 ) )
settings.sleep_w_power // Set to 1 to sleep when on external high voltage
                         // (does not sleep on USB), default 0 (range: 0=sleep,1=run, count down to
next cycle)
settings.led_mask // mask to mask off LED's, default 0xFF (range: 0 - 255, 0=all off, 255=all on)
  Membrane Panel Individual Bits:
  bit0: GPS
  bit1: Message
  bit2: Power
  bit3: Satellite
  bit4: Alarm
  bit5-7: ignored
settings.i_rx_always // keeps radio awake always to receive commands, default 0 (range: 0=off,1=on)
settings.tx_altitude // transmit altitude with position reports (range: 0=off,1=on)

Example:
//Set sleep time between reports to 10 minutes
settings.flash(settings.sleep, 600)

//Print current sleep time
print(settings.flash(settings.sleep))
```

LEDs

```
NOTE: These functions are currently overridden during normal transmit behavior by built-in functions.

//Set LED
led.led( {led}, {led mode}, {cycles} )

// Set LED with cycles set to continuous
led.led( {led}, {led mode} )
```

```
//Read current LED mode
{variable} = led.led( {led} )

led:
led.pwr: Power
led.gps: GPS
led.sat: Satellite/Iridium
led.alrm: Alarm
led.msg: Message

led mode:
led.off: off
led.on: on (solid)
led.fadeoff: fade from bright to dark
led.fadeon: fade from dark to light
led.slow: slow flash
led.medium: medium flash
led.fast: fast flash
led.1: single flash
led.2: two flashes
led.3: three flashes
led.4: four flashes
led.5: five flashes

cycles:
The number of times to repeat the mode sequence. 255 = forever.

Example:
//Set LED 0 to ON permanently
led.led(led.pwr, led.on, 255)
```

Interrupts / Event Handlers

A set of Lua functions can be defined for servicing various state changes on the GSatMicro.

```
cpu.set_int_handler( {interrupt type}, {function} )

interrupt type:
cpu.INT_IRIDIUM_SIGNAL: Iridium modem has signal from satellite network
cpu.INT_IRIDIUM_TX_OK: SBD transmit completed successfully
cpu.INT_IRIDIUM_TX_FAIL: SBD transmission failed (after settings.i_tx_retries retries)
cpu.INT_IRIDIUM_TIMEOUT: Iridium timeout waiting for signal from satellite network
cpu.INT_GPS_VALID: GPS fix is valid and settled
cpu.INT_GPS_TIMEOUT: timeout waiting for GPS fix
cpu.INT_BOOT: called at boot
cpu.INT_CONTENTION: Iridium modem is already receiving when a radio command was issued
```

```
Example:
function  iridium_signal()
    lat,lon,speed,alt,course=gps.getpos()
    sec,min=gps.gettime()
    if(speed < 5) then
      course = compass.heading()
    end
    batp = bat.percent()
    iridium.transmit(gps.encode(lat,lon,speed,course,batp,min,alt))
end

cpu.set_int_handler( cpu.INT_IRIDIUM_SIGNAL, iridium_signal)
```

COCOM Limits

COCOM limits are a GPS limitation by international law to disable GPS chipsets at high altitudes or high speeds.

The GSatMicro will stop reporting if the unit exceeds 1,000 knots AND exceeds 18,000 meters. Both of these limits must be exceeded for COCOM limits to be enabled which will disable the internal GPS receiver until one of the two limits returns to normal.

http://en.wikipedia.org/wiki/CoCom

Pinout v.1

1. RS232 - TX
2. RS232 - RX
3. USB - V+ (4.5-5.5v DC)
4. USB - D-
5. USB - D+
6. OUTPUT: 0
7. OUTPUT: 1
8. INPUT: 0
9. INPUT: 1
10. GROUND
11. VCC (7.5v-40v DC)

Pinout v.7

Name	Flex Pin	Hirose Pin	Cable Color
ADC IN 0	15	1	
ADC IN 1	14	5	
OUTPUT 1	13/12	2	
OUTPUT 0	11/10	3	
V_USB	9	7	yellow
USB D-	8	6	white
USB D+	7	10	green
RS232 TX	6	12	blue
RS232 RX	5	11	grey
VCC	4/3	8	red
GND	2/1	4	black

Message Formats

Mobile Terminated (MT)

Future versions will make use of a header byte with values ranging from 0 - 31 to define message type.

Mobile Originated (MO)

Position Report:

Text Message:

```
header (byte): 1
destination length (byte): <0 - 255>
destination (byte array): <destination length bytes>
message (byte array): <byte array bounded by end of MO message>
```

Settings Dump:

```
header (byte): 2
interface version (byte): <version, currently: 1>
settings version (byte): <version, currently: 1>
setting 0 (signed 32 bit integer): <value of first setting>
...
setting N-1 (signed 32 bit integer): <value of Nth setting>

Current order:
0: defaultv
1: g_hdop
2: g_timeout
3: i_tx_timeout
4: i_signal_timeout
5: i_tx_retries
6: sleep
7: sos_sleep
8: sleep_w_power
9: led_mask
10: i_rx_always
11: tx_altitude
12: g_settle
13: t_adc_id
14: t_adc_threshold
15: status_line
16: low_bat_off
```