



**ST150M**  
**MODULE**  
**USER MANUAL**

9150-0125-01 R-1



**Distribution Statements:**

GLOBALSTAR CONFIDENTIAL AND PROPRIETARY INFORMATION – All data and information contained in this document are confidential and proprietary to Globalstar, Inc. Information and instructions contained in this publication anticipate the user possesses and applies the knowledge, training, and experience commensurate with requirements to meet prerequisite certification. All handling and distribution of this document is subject to the terms & conditions of the Non-Disclosure Agreement currently in place between Globalstar, Inc. and the recipient.

RESTRICTIVE EXPORT NOTICE – This document may contain technical data whose export is restricted by the Export Administration Act of 1979, as amended, Title 50, U.S.C., App 2401 et seq. Violations of these export laws are subject to severe criminal penalties. Do not export, directly or indirectly, any technical data or defense information, or any products utilizing any such data, to any country for which the U.S. Government or any agency thereof, requires an export license, according to their laws and regulations or other Government approval, without first obtaining such license or written approval.

DESTRUCTION NOTICE – Destroy by any method that will prevent disclosure of contents or reconstruction of the document.

Published and printed in the U.S.A. by:

Globalstar, Inc. 1351  
Holiday Square Blvd.  
Covington, LA 70433

Copyright ©2022 Globalstar, Inc. All rights reserved.

Globalstar® is a registered trademark of Globalstar, Inc. All other brands, product name, trademarks and registered trademarks are properties of their respective owners. This material may not be reprinted, republished, broadcast, or otherwise altered without the publisher's written permission. This manual is provided without express, statutory, or implied warranties. The publisher will not be held liable for any damages caused by or alleged to be caused by use, misuse, abuse, or misinterpretation of the contents. Content is subject to change without notice.

Revision date: 04/28/2022

## Contents

<b>LIST OF FIGURES .....</b>	<b>7</b>
<b>1 INTRODUCTION.....</b>	<b>8</b>
1.1 PURPOSE .....	8
1.2 APPLICABLE DOCUMENTS.....	8
1.2.1 GLOBALSTAR DOCUMENTS.....	8
1.2.2 EXTERNAL REFERENCES .....	8
1.3 DESCRIPTION .....	9
1.3.1 Feature List .....	9
1.4 CERTIFICATIONS.....	11
<b>2 APPLICATION.....</b>	<b>12</b>
2.1 THEORY OF OPERATION FOR GLOBALSTAR COMMERCIAL IoT .....	12
2.2 QUICK START/CONFIGURATION GUIDELINES.....	13
2.3 BLOCK DIAGRAM .....	14
2.4 Core Components .....	14
2.4.1 Nordic SoC .....	14
2.4.2 ASIC satellite transmitter.....	14
2.4.3 GNSS receiver .....	15
2.4.4 Memory .....	15
2.4.5 Accelerometer .....	15
2.4.6 Program and debug .....	15
2.4.7 DC Supply .....	15
<b>3 Power Management .....</b>	<b>16</b>
3.1 Power Management High Level Overview.....	16
<b>4 PERIPHERALS.....</b>	<b>18</b>
4.1 Near-Field Communication (NFC).....	18
4.2 GPIO pin peripherals .....	18
4.2.1 GPIO .....	19
4.2.2 Comparator.....	19
4.2.3 PDM - Pulse Density modulation interface .....	20
4.2.4 PPI - Programmable Peripheral Interconnect .....	20
4.2.5 PWM - Pulse Width Modulation.....	20
4.2.6 QDEC - Quadrature Decoder.....	20
4.2.7 RTC - Real Time Counter.....	21
4.2.8 SPI - Serial Peripheral Interface .....	22
4.2.9 UART - Universal asynchronous receiver / transmitter .....	22
4.3 Using and Enabling Peripheral Devices.....	23
<b>5 HARDWARE LAYOUT/REFERENCE DESIGN .....</b>	<b>26</b>
5.1 Pin Assignments .....	26
5.2 Reference Circuitry .....	30
5.2.1 PCB Guidelines .....	30

5.2.2	Module Dimensions .....	30
5.2.3	PCB Layout Example .....	32
5.3	Reference Design Schematics .....	35
5.4	Transmit/Receive Antenna Requirements .....	36
5.5	BLE Antenna Requirements .....	36
5.6	Single Modular Approval Requirements .....	37
5.7	Trace and Cable Losses .....	37
<b>6</b>	<b>RECOMMENDED OPERATING CONDITIONS .....</b>	<b>38</b>
6.1	ABSOLUTE MAXIMUMS .....	38
6.2	HANDLING OF ELECTROSTATIC SENSITIVE DEVICES .....	38
6.3	WIRING .....	38
6.4	NOISE SOURCES .....	38
6.5	NORDIC PROCESSOR UNUSED PINS .....	39
6.6	OBSTRUCTIONS .....	39
6.7	POWER PERFORMANCE CHARACTERISTICS .....	39
<b>7</b>	<b>APPLICATION PROGRAMMING INTERFACE (API) .....</b>	<b>39</b>
7.1	AT Commands .....	39
7.1.1	ST150 Device Configuration .....	39
7.1.2	Configuration Security .....	39
7.1.3	General Configuration .....	40
7.2	URC Notification Settings .....	41
7.2.1	Transmitter Event Notification Settings .....	41
7.2.2	Set Transmitter Notify Flags via NUS, UART .....	41
7.2.3	Read Transmitter Notify Flags via NUS, UART .....	42
7.2.4	Set Transmitter Notify Flags via Configuration Service .....	42
7.2.5	Read Transmitter Notify Flags via Configuration Service .....	42
7.3	Message Queue Event Notification Settings .....	42
7.3.1	Set Message Queue Notify Flags via NUS, UART .....	42
7.3.2	Read Message Queue Notify Flags via NUS, UART .....	42
7.3.3	Set Message Queue Notify Flags via Configuration Service .....	42
7.3.4	Read Message Queue Notify Flags via Configuration Service .....	42
7.4	GPIO Event Notification Settings .....	43
7.4.1	Set GPIO Notify Flags via NUS, UART .....	43
7.4.2	Read GPIO Notify Flags via NUS, UART .....	43
7.4.3	Set GPIO Notify Flags via Configuration Service .....	43
7.4.4	Read GPIO Notify Flags via Configuration Service .....	43
7.5	Inputs/Count Event Notification Settings .....	43
7.5.1	Set Inputs/Count Notify Flags via NUS, UART .....	43
7.5.2	Read Inputs/Count Notify Flags via NUS, UART .....	43
7.5.3	Set Inputs/Count Notify Flags via Configuration Service .....	44
7.5.4	Read Inputs/Count Notify Flags via Configuration Service .....	44
7.6	Track events Notification Settings .....	44
7.6.1	Set Track Events Notify Flags via NUS, UART .....	44

- 7.6.2 Read Track Events Notify Flags via NUS, UART .....44
- 7.6.3 Set Track Events Notify Flags via Configuration Service .....44
- 7.6.4 Read Track Events Notify Flags via Configuration Service .....44
- 7.7 System OTA Event Notification Settings .....45
  - 7.7.1 Set OTA Events Notify Flags via NUS, UART .....45
  - 7.7.2 Read OTA Events Notify Flags via NUS, UART .....45
  - 7.7.3 Set OTA Events Notify Flags via Configuration Service .....45
  - 7.7.4 Read OTA Events Notify Flags via Configuration Service .....45
- 7.8 GPS Event Notification Settings .....46
  - 7.8.1 Set GPS Notify Flags via NUS, UART .....46
  - 7.8.2 Read GPS Notify Flags via NUS, UART .....46
  - 7.8.3 Set GPS Notify Flags via Configuration Service .....46
  - 7.8.4 Read GPS Notify Flags via Configuration Service .....46
- 7.9 Accelerometer Event Notification Settings .....47
  - 7.9.1 Set Accelerometer Event Notify Flags via NUS, UART .....47
  - 7.9.2 Read Accelerometer Event Notify Flags via NUS, UART .....47
  - 7.9.3 Set Accelerometer Event Notify Flags via Configuration Service .....47
  - 7.9.4 Read Accelerometer Event Notify Flags via Configuration Service .....47
- 7.10 Power Management Event Notification Settings .....47
  - 7.10.1 Set Power Management Notify Flags via NUS, UART .....48
  - 7.10.2 Read Power Management Notify Flags via NUS, UART .....48
  - 7.10.3 Set Power Management Notify Flags via Configuration Service .....48
  - 7.10.4 Read Power Management Notify Flags via Configuration Service .....48
- 7.11 External Flash Logging Event Notification Settings .....48
  - 7.11.1 Set External Flash Logging Notify Flags via NUS, UART .....48
  - 7.11.2 Read External Flash Logging Notify Flags via NUS, UART .....48
  - 7.11.3 Set External Flash Logging Notify Flags via Configuration Service .....48
  - 7.11.4 Read External Flash Logging Notify Flags via Configuration Service .....49
- 7.12 GPIO/Inputs Configuration .....49
  - 7.12.1 Getting a list of all the GPIO .....49
  - 7.12.2 Configuring inputs: .....50
  - 7.12.3 Mapping inputs: .....51
  - 7.12.4 Getting the input state of mapped inputs: .....51
  - 7.12.5 GPIO Notify settings: .....52
- 7.13 Enabling and Configuring the UART: .....53
- 8 SOFTWARE HIGH LEVEL OVERVIEW .....53**
  - 8.1 BLE Configuration Service and Interfaces .....53
    - 8.1.1 Device Advertising and Discoverability .....54
    - 8.1.2 External Interfaces .....54
    - 8.1.3 External Interface Security .....54
- 9 Unified API Modules .....55**
  - 9.1 Messages .....56
    - 9.1.1 Message Priority .....56

9.1.2	Burst Count .....	56
9.1.3	Min/Max Burst Interval Seconds.....	57
9.1.4	Interval and Time-Of-Day Message List.....	57
9.1.5	Transmitter .....	57
9.1.6	Message Preemption and Aborted Transmissions .....	57
9.1.7	GPIO .....	58
9.1.8	Special Function Inputs .....	58
9.1.9	UART .....	59
9.1.10	GPIO Events.....	59
9.1.11	Inputs .....	59
9.1.12	OTA Messages.....	59
9.1.13	Input Events.....	59
9.1.14	Track Functions .....	59
9.1.15	GPS Module.....	60
9.1.16	Accelerometer Module.....	61
9.1.17	Power Management .....	62
9.1.18	Temperature Notifications .....	62
9.1.19	Data Logging.....	63
9.1.20	BLE Settings .....	63
<b>10</b>	<b>TEST MODES.....</b>	<b>64</b>
10.1	Transmitter Test Modes .....	64
<b>11</b>	<b>ORDERING INFORMATION/DEVICE MARKINGS/ACTIVATION INFORMATION .....</b>	<b>65</b>
11.1	Device Marking .....	65
11.2	Box Labels .....	65
11.3	Minimum Order Quantity.....	<b>Error! Bookmark not defined.</b>
11.4	How to Order .....	66
11.5	Globalstar Product Certification .....	67
<b>12</b>	<b>GENERAL WARNINGS.....</b>	<b>68</b>
<b>13</b>	<b>REGULATORY.....</b>	<b>69</b>
<b>14</b>	<b>ACRONYM LIST .....</b>	<b>72</b>

## LIST OF FIGURES

---

FIGURE 1 - PHYSICAL DIMENSIONS	11
FIGURE 2 - LEO CONSTELLATION	12
FIGURE 3 - OVERALL MODULE SYSTEM	14
FIGURE 5 - PIN ASSIGNMENTS	26
FIGURE 6 - MODULE DIMENSIONS	31
FIGURE 7 - TOP AND BOTTOM LAYOUTS	32
FIGURE 8 - STX LAYOUT	33
FIGURE 9 - BLE LAYOUT	33
FIGURE 10 - GPS LAYOUT	<b>Error! Bookmark not defined.</b>
FIGURE 11 - TOP AND BOTTOM PCB VIEWS	34
FIGURE 12 - STX, BLE, AND GPS VIEWS (LEFT TO RIGHT)	34
FIGURE 13 - COPLANAR WAVEGUIDE	37
FIGURE 15 - UNIFIED API MODULES	55
FIGURE 16 - PCB LABEL	65
FIGURE 17 - BOX LABEL AND PACKING SLIP	65

DRAFT

## 1 INTRODUCTION

### 1.1 PURPOSE

This document describes the physical, electrical, and functional characteristics of the ST150 Satellite Transmitter Module, hereafter referred to as ST150M. The information contained in this document is intended to provide the VAR/Integrator with the necessary technical information required to configure and use the module in a custom application.

This document is intended to be used by engineers and technical management and assumes a general knowledge of basic engineering practices by the user. **NOTE:** Globalstar is only responsible for ensuring that the ST150M meets the published manufacturing specifications described herein.

### 1.2 APPLICABLE DOCUMENTS

#### 1.2.1 GLOBALSTAR DOCUMENTS

GS-07-1248 - REMOTE TELEMETRY SERVICE FREQUENCY PLAN FOR SIMPLEX TRANSMITTERS  
 GS-07-1247 - REMOTE TELEMETRY SERVICE SIMPLEX ANTENNA SPECIFICATION AND TEST PROCEDURE  
 GS-02-0795 - REMOTE SIMPLEX TELEMETRY SYSTEM DESCRIPTION AND APPLICATION NOTE  
 GS-09-1302 - GLOBALSTAR EQUIPMENT EVALUATION APPLICATION FOR REMOTE TRANSMITTER UNIT  
 9150-0126-01 ST150 INTEGRITY USER GUIDE  
 9150-0127-01- ST150M APP USER GUIDE  
 8545-0196-01 - GLOBALSTAR ST150M PRODUCT SUBMISSION GUIDE  
 8550-0002-01 - ST150M UNIFIED API REFERENCE  
 8550-0006-01 - ST150 SOFTWARE DESIGN GUIDE  
 8550-0003-01 - GLOBALSTAR ST150 BLUETOOTH SENSORS USER GUIDE  
 8550-0004-01 - GLOBALSTAR ST150 BLUETOOTH REFERENCE  
 8550-0010-01 - GLOBALSTAR IOT ASIC DFU SERVICE BLE SPECIFICATION  
 8550-0009-01 - GLOBALSTAR IOT CONFIGURATION SERVICE BLE SPECIFICATION  
 8550-0008-01 - GLOBALSTAR IOT SENSOR SERVICE BLE SPECIFICATION

#### 1.2.2 EXTERNAL REFERENCES

Wikipedia article on Intel Hex format: [https://en.wikipedia.org/wiki/Intel\\_HEX](https://en.wikipedia.org/wiki/Intel_HEX)

NMEA-0183 rev. 3.0 Specification

FCC Guidance for Use of Radio Modules KDB 996369 D03 & D04

- Nordic Programmer (Segger):  
<https://www.digikey.com/product-detail/en/segger-microcontroller-systems/8-08-00-J-LINK-BASE/899-1004-ND/2175882>
- ASIC Programmer (On Semi):  
<https://www.digikey.com/product-detail/en/on-semiconductor/AXDBG-2-GEVK/AXDBG-2-GEVK-ND/5820231>

Commented [JM1]: This list needs vetting.



### 1.3 DESCRIPTION

The ST150M is a transmit-only Satellite IoT module designed to send small packets of user defined data to a network of low earth orbiting (LEO) satellites using the Globalstar commercial IoT network. The received data is then forwarded to a user defined network interface that may be in the form of an FTP host, HTTPS host or HTTP host where the user will interpret the data for further processing.

The ST150M is designed to be integrated into an integrator's custom carrier board design. The ST150M can operate 100% autonomously once configured or can be controlled via external interfaces/AT commands, or can be modified with custom firmware leveraging the onboard Nordic 52840 as the applications processor.

The ST150M can be configured via the Globalstar Mobile App, through the BLE configuration service, or through a physical or BLE-based UART.

The ST150M includes value-added software modules that can be configured for tracking applications based on GPS and/or accelerometer movement detection, input-based count, accumulate, state, and transition events, interval or time-of-day events, and even Smart One Sensor Service data.

The ST150M consists of the Nordic NRF52840, External Flash, GPS, accelerometer, ASIC, and internal power controls. It does not have an onboard patch antenna. The Nordic and ASIC are shielded for modular certification. The ST150M has gold "fingers" around the perimeter of the PCB so that it can be SMT-installed onto a carrier board or another custom PCB.

#### 1.3.1 Feature List

##### Hardware

###### Attributes

- 54 pin SMT
- 2 x 1.1 inches
- 3.3V VCC
- Nordic 52840 MCU
- Onboard GPS and 3-axis accelerometer
- Onboard satellite ASIC transmitter
- Onboard/Offboard MCU programming
- Onboard/Offboard ASIC programming
- Integrated voltage regulation
- GPS RF signal
- ASIC RF signal
- BLE RF signal
- NFC RF signal
- 24 configurable I/O
- MCU USB Device
- |°C

###### Temperature

- Operating -40 to 85C

###### GPS

- Active and passive antennas
- 72-channel GPS/QZSS L1 C/A, GLONASS L10F, BeiDou B1I, Galileo E1B/C, SBAS L1 C/A, WAAS, EGNOS, MSAS, GAGAN
- Tracking: -167 dBm
- Cold Starts: -148 dBm
- Hot Starts -157 dBm
- Accuracy: 2.0m CEP

###### Accelerometer

- 3-axis acceleration
- Selectable +/-2G, +/-4G, +/-16G scales
- 6D orientation detection

###### Power

- Nominal: < 100uA
- Sat Burst: 1.6A
- GPS on: 50mA
- BLE connection: variable and customizable

Commented [JC2]: @John Mochring Your preference for word usage – "device" or "module"

Commented [JG3]: Technically should be °C

## Software

### Security

- Interface-based security policies for configuration commands
  - Always disabled
  - Always enabled
  - Passphrase auth required
  - Passphrase auth required with timeout
- Customizable user passphrase
- Encrypted BLE configuration interface

### Interface

- Modem AT command interface
  - UART
  - BLE Config (secure key)
  - BLE Nordic UART Service (NUS)
- Unsolicited response codes (URCs)
  - Customizable event filters
  - Selectable output GPIO interrupts
- BLE Smart One Sensor Service
  - Up to 10 BLE sensor nodes
  - Customizable security key

### Tracking

- Normal Track Mode
  - Previous track point option
  - Dock mode option
- Theft Alert Mode
  - Geofencing
- Integrated or external GPS
- Integrated or external accelerometer

### Software Inputs

- 8-Software Defined Count Inputs
  - GPIO mapping
  - Accelerometer event mapping
- Count
- Accumulated Time
- Transition Events
- Undesired State
  - Optional hysteresis

### OTA Messages

- 3 customizable priority settings
- Customizable dither and re-transmission attempts
- Legacy Message Compatibility
  - Raw Messages
  - Truncated Messages
  - Standard Messages
    - Optional previous track point
  - Speed and Heading
    - Optional previous track point
  - Smart One Sensor Raw and Truncated Sensor Messages
- Modular Messages
  - Powered-On
  - GPS
  - Timestamp
  - Inputs 1-8
- Health Check
- Interval or Time-Of-Day

### Managed OTA Message Transmit Queue

- Supports up to 10 pending OTA messages
- Pre-emption settings
  - Disabled
  - Priority
  - Time
  - Priority and Time

### Configurable GPIO

- Special Function
  - Line Power Detection
  - Advertising Enable Detection
  - Tamper Detection
- GPS Features
  - NMEA pass-through
  - Current and previous position information
- Power Management
  - Settable power states and track-mode behavior
    - Full-on
    - Partial Suspend
    - Full Suspend

The ST150M weighs ~0.27oz (7.6g) with shield (no battery, no solar panel), with the dimensions shown below:

Commented [JM4]: Image available?

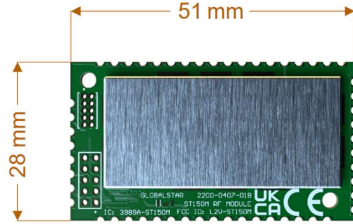


FIGURE 1 - PHYSICAL DIMENSIONS

#### 1.4 CERTIFICATIONS

Globalstar products comply with WEEE and RoHS. The ST150M has the following certifications:

- Bluetooth LE
- FCC
- ISED
- CE RED
- UKCA

DRAFT

## 2 APPLICATION

### 2.1 THEORY OF OPERATION FOR GLOBALSTAR COMMERCIAL IoT

The ST150M IoT Module operates on the Globalstar LEO satellite network. LEO (Low Earth Orbit) means that there are a number of satellites in low earth orbit that constantly orbit the planet and can communicate with Globalstar devices that are within range of its current position.



FIGURE 2 - LEO CONSTELLATION

Because the satellite position is constantly changing, commercial IoT devices on the ground will transmit (with no knowledge of any of the satellites locations) and the transmission may be received by one or more satellites. These satellites will then relay the message to the nearest satellite gateway (Figure 3). Once received by the satellite gateway, the simplex message will be delivered to the simplex gateway where redundant messages are discarded and the data from the message is sent to the OEM via the Internet.



FIGURE 3 - SIMPLEX IoT MESSAGING Data Flow

Commented [JC5]: Perhaps use "IoT module"?

Commented [JC6]: Graphics of Globalstar satellites are available, and would be more appropriate than generic satellite clipart.

Arrow is missing from Simplex\_Gateway-to-VAR/VAM

## 2.2 QUICK START/CONFIGURATION GUIDELINES

To get up and running, the following details both mandatory and optional steps:

1. Globalstar offers an ST150M Dev Kit that provides a quick way to power-up the module and communicate with modem AT command interface via physical UART, Globalstar Mobile App, or BLE Nordic UART Service (NUS)
2. Create an Account
  - a. Create a VAR account and access the Globalstar IoT Data Management Portal (Requires VAR/VAM BackOffice endpoint to route device traffic to)
  - b. Or, work with a Globalstar VAR to setup billing, activate your module, and create a user login account (Requires backoffice endpoint to route device traffic to)
  - c. Or, use Davra's IoT platform to setup billing, activate your module, and direct device traffic to your Davra account (Does not require separate backoffice endpoint)
3. Optionally, if using 2A or 2B, integrate backoffice APIs with the Globalstar Simplex API for programmatic API access to billing and device management APIs
4. Optionally, if using 2A or 2B, access Globalstar's Data Warehouse Portal for real-time debugging of device traffic
5. Get the Globalstar Mobile App and use your account credentials from step 2 to login, configure your device, and sync device activation status
6. Optionally, send a raw or truncated message via modem AT commands to initiate device OTA traffic

## 2.3 BLOCK DIAGRAM

The ST150M Block Diagram illustrates the overall module system.

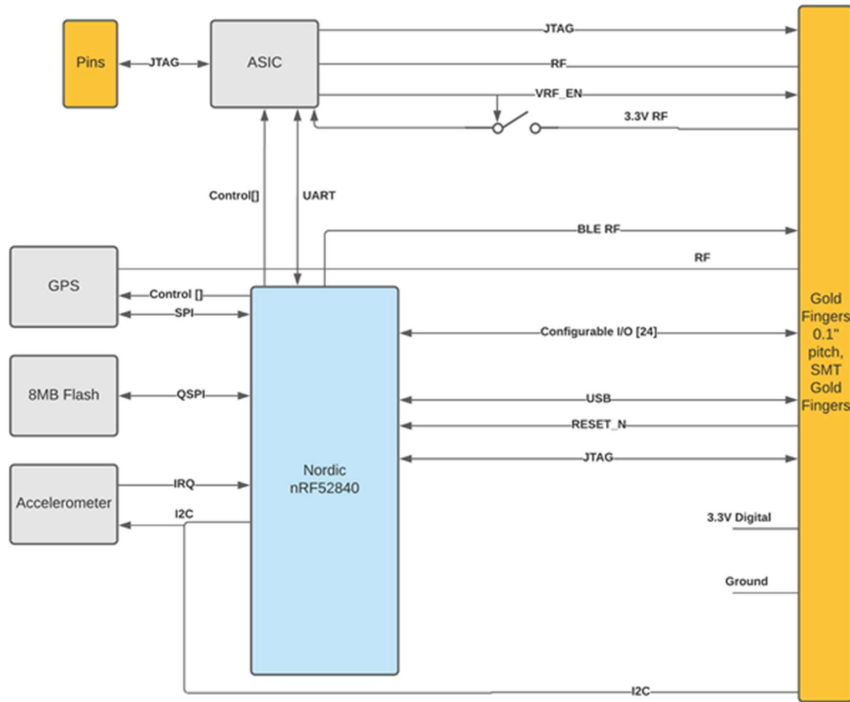


FIGURE 3 - OVERALL MODULE SYSTEM

## 2.4 Core Components

### 2.4.1 Nordic SoC

The ST150M contains a Nordic nRF52840, which utilizes an ARM Cortex-M4 processor with integrated communication protocols including Bluetooth 5.0, NFC, I<sup>2</sup>C, SPI, UART and Client USB.

Details on the nRF52840 SOC can be found in documentation from Nordic Semiconductor:

[https://infocenter.nordicsemi.com/index.jsp?topic=%2Fug\\_nrf52840\\_dk%2FUG%2Fdk%2Fintro.html](https://infocenter.nordicsemi.com/index.jsp?topic=%2Fug_nrf52840_dk%2FUG%2Fdk%2Fintro.html)

### 2.4.2 ASIC satellite transmitter

The ST150M contains a Globalstar proprietary ASIC used for all satellite transmission and related operations including signal modulation, channelization, and amplification.

Communication is handled over UART interface bus connecting to the MPU.

### 2.4.3 GNSS receiver

The ST150M contains a UBLOX ZOE-M8Q GNSS SiP (System in Package) module with integrated SAW filter and front-end LNA, plus additional external LNA control line. Both the GNSS RF input and the external LNA control output is available on ST150M's edge connector interface.

With a dual-frequency RF front-end the ZOE-M8Q can receive concurrently from three GNSS systems (GPS / Galileo together with BeiDou or GLONASS).

Communication is handled over GPS\_SPI interface bus connecting to the MPU.

### 2.4.4 Memory

The ST150M contains 64Mb (8MB) of ultra-low power flash memory accessed over QSPI bus. In addition, the nRF52840 has 1024 kB of flash and 256 kB of RAM located on the SoC.

### 2.4.5 Accelerometer

The ST150M contains a 3-axis accelerometer capable of 4098 LSB/g (// 16g), with scaling options of 2g, 4g, 8g and 16g, and high shock survivability of up to 10,000g.

The accelerometer can be communicated with over I<sup>2</sup>C using 16-bit data output and 32-level FIFO buffer. An interrupt pin connects to P0.17 of the nRF52840 for configurable motion detect signaling.

### 2.4.6 Program and debug

The ST150M has dedicated programming and debug ports for both the Nordic SoC and the ASIC.

Nordic programming and debug occurs over signals SWDCLK, RESET, SWDIO which are fed to gold finger pads 46, 47, 48 and to 0.5 MIL (1.27mm) header J1 (depopulated).

ASIC programming occurs over signals SAT\_DBG\_CLK, SAT\_DBG\_EN, SAT\_RST\_N, SAT\_DBG\_DATA which are fed to gold finger pads 50, 51, 52, 53 and to 1 MIL (2.54mm) header J2 (depopulated).

### 2.4.7 DC Supply

Using separate DC sources can help to reduce noise on the system, maintain efficiency, and improve signal integrity for transmissions. ST150M divides power input between VRF\_IN and V33\_IN.

VRF\_IN powers the ASIC RF circuitry.

V33\_IN powers everything else on the ST150M including ASIC digital circuitry and GNSS receive.

VRF\_EN is an active-high 0-3V control output to indicate if currently transmitting. Connect VRF\_EN to the enable pin of VRF\_IN supply.

VRF\_IN and V33\_IN can be tied together using a single source if desired. VRF\_EN should remain floating if a single source is used.

### 3 Power Management

Signal	Parameter	Range	Nom.	Units	Notes
V33_IN	Input Voltage Range for Digital Inputs	3.1 - 3.6	3.3	Volts	50mV ripple max, 80uA - 200mA depending on BLE/GPS power and usage
I33_IN	Input Current for Digital Inputs	100 max	-	mA	
VRF_IN	Input Voltage Range for RF circuitry	3.0 -3.6	3.3	Volts	50mV ripple max, 1.6A bursts @ 25ms during transmit bursts. Integrator should ensure min voltage during bursts does not drop below 3V
IRF_IN	Input Current for RF Circuitry	500 max	-	mA	
VUSB	Input Voltage Range for USB Device	4.75 - 5.75	5.0	Volts	
VRF_EN	Voltage range of VRF_EN output	0 - 3	0	Volts	
V33_ripple	Ripple voltage requirements for V33_IN	0 - 50	0	mV pk-pk	
VRF_ripple	Ripple voltage requirements for VRF_IN	0 - 50	0	mV pk-pk	
I2C	Voltage range of I <sup>2</sup> C	0 - 3.3	3.3	Volts	

Note: -40° to +85°C operating temperature

#### 3.1 Power Management High Level Overview

The Power UAPI is responsible for handling the different power states of the UAPI and switching between them. The power states are based on configurable battery voltage thresholds. This module also handles the required action before moving into a powered down state.

##### Power States

There are five different power states that have different limitations on them to control the amount of power of the module:

- **Normal:** The device runs as normal with no restrictions.
- **Factory:** Same as normal, but if no solar input is detected for four hours, the device goes into the Full Off state. This is the default state of the device when the factory configuration is on the device. Once the user updates the configuration, this mode is inaccessible.
- **Partial Suspend:** The device is limited on the frequency of track messages it can send to reduce power consumption.
- **Full Suspend:** The device cannot send any track messages.
- **Full Off:** The device sends a command to the fuel gauge to fully disconnect the ST150 module from power. Only light on the solar panel can wake the device up from this state.

Commented [NV7]: Confirm

Commented [NV8]: Confirm

Commented [EJ9]: Need to make a state machine diagram.



### Default Thresholds for Different Power States

- Partial Suspend: 3.5V
- Full Suspend: 3.1V
- Full Off: 2.7V
- HW Protection Cutoff: 2.6V

To set different thresholds via AT command details refer UAPI document Section 21.5 **Voltage-Based Power Management** and Section 21.6 **State of Charge Power Management**.

A power state transition will trigger an event via the Event Queue UAPI so any application specific power logic can be added. Additionally, there is a callback when preparing to enter Full Off mode for application specific shutdown preparation.

### Power Management Type

To verify the power management transitions to different states it is important to select right power management type, either power measurement via voltage or by state of charge. By default, it's set to POWER\_MEAS\_TYPE\_SOC i.e., 1. For more details refer UAPI document Section 21.4 **Power Management Type**.

```
typedef enum
{
    POWER_MEAS_TYPE_VOLTAGE = 0,
    POWER_MEAS_TYPE_SOC
} power_meas_type_enum;
```

AT+PWRMEAS=< power\_meas\_type\_enum> Queries and selects between voltage or state-of-charge-based power management.

### Manual Power States

If desired the power UAPI can be set into a manual mode allowing for power state transitions to be triggered manually via AT command for API call. Manual power mode must be enabled before this feature will work. Once in manual mode, all automatic state changes are disabled, and only manual state change requests will cause state transitions.

1. To enable the manual transitions:

```
AT+PWRMANUAL=<manual_mode_enabled>
0 = Disable, 1 = Enable
```

For more details refer UAPI document section 21.3 **Manual Power State Mode**.

2. To make power transitions manually:

```
typedef enum
{
    POWER_STATE_NORMAL_OPERATION = 0,
    POWER_STATE_PARTIAL_SUSPEND,
    POWER_STATE_FULL_SUSPENDED,
    POWER_STATE_FULL_OFF,
    POWER_STATE_FACTORY
} power_state;
```

```
AT+PWRSTATE=<power_state>, [<delay_minutes>]
```

Note: [<delay\_minutes>] valid only when transitioning to Full Off Mode.

For more details refer UAPI document section 21.2 **Power State API**.

### Power Notifications

A variety of power notifications can be enabled including low voltage, Nordic MCU Temperature, transmitter temperature, battery pack temperature, and power state transitions. These settings and their thresholds can be enabled through the Config UAPI. For more details refer UAPI document section 21.15 **POWER EVENT NOTIFY SETTINGS**.

## 4 PERIPHERALS

### 4.1 Near-Field Communication (NFC)

There are two GPIO pins (P0.09/NFC1 and P0.10/NFC2) on the ST150M that can be configured to be a NFC listening device. The operating frequency is 13.56MHz and the bit rate is 106kbps.

There are many NFC antennas on the market, but one that's been tested with the ST150M is Nordic Semiconductor's flex PCB NFC antenna. More details can be found on Nordic's website. When doing PCB layout, the NFC traces should be routed as a differential pair.

### 4.2 GPIO pin peripherals

The following GPIO pins can be configured to whatever Nordic processor peripheral is desired. The peripherals are listed below and go into some detail.

P0.02/AIN0  
P0.03/AIN1  
P0.04/AIN2  
P0.05/AIN3  
P0.28/AIN4  
P0.29/AIN5  
P0.30/AIN6  
P0.31/AIN7  
P0.09/NFC1  
P0.10/NFC2  
P0.16  
P0.24  
P0.25  
P1.01  
P1.02  
P1.03  
P1.04  
P1.05  
P1.06  
P1.07  
P1.10  
P1.12  
P1.14  
P1.15

**4.2.1 GPIO**

GPIO has the following user-configurable features:

- Configurable output drive strength
- Internal pull-up and pull-down resistors
- Wake-up from high or low level triggers on all pins
- Trigger interrupt on state changes on any pin
- All pins can be used by the PPI task/event system
- One or more GPIO outputs can be controlled through PPI and GPIOTE channels
- All pins can be individually mapped to interface blocks for layout flexibility
- GPIO state changes captured on SENSE signal can be stored by LATCH register
- An event can be generated in each GPIO based on a rising edge, falling edge, or any changing edge.

Symbol	Description	Min	Typ	Max	Units
VIH	Input High Voltage	0.7VDD	-	VDD	V
VIL	Input Low Voltage	VSS	-	0.3VDD	V
VOH	Output High Voltage	2.9	-	VDD	V
VOL	Output Low Voltage	VSS	-	0.4	V
IOL,SD	Current at VSS+0.4 V, output set low, standard drive	1	2	4	mA
IOL,HDH	Current at VSS+0.4 V, output set low, high drive	6	10	15	mA
IOH,SD	Current at VDD-0.4 V, output set high, standard drive	1	2	4	mA
IOH,SD	Current at VDD-0.4 V, output set high, high drive	6	9	14	mA
RPU	Internal Pull-up Resistance	11	13	16	kOhm
RPD	Internal Pull-down Resistance	11	13	16	kOhm

**4.2.2 Comparator**

The comparator (COMP) compares an input voltage (VIN+) against a second input voltage (VIN-). VIN+ can be derived from an analog input pin (AIN0-AIN7). VIN- can be derived from multiple sources depending on the operation mode of the comparator.

Main features of the comparator are:

- Input range from 0 V to VDD
- Single-ended mode
- Fully flexible hysteresis using a 64-level reference ladder
- Differential mode
- Configurable 50 mV hysteresis
- Reference inputs (VREF): VDD
- External reference from AIN0 to AIN7 (between 0 V and VDD)
- Internal references 1.2 V, 1.8V and 2.4V
- Three speed/power consumption modes: low-power, normal and high-speed
- Single-pin capacitive sensor support
- Event generation on output changes
- UP event on VIN- > VIN+
- DOWN event on VIN- < VIN+
- CROSS event on VIN+ and VIN- crossing
- READY event on core and internal reference (if used) ready

Commented [JG10]: Word General is redundant to General Purpose Input Output

### 4.2.3 PDM - Pulse Density modulation interface

The pulse density modulation (PDM) module enables input of pulse density modulated signals from external audio frontends, for example, digital microphones. The PDM module generates the PDM clock and supports single-channel or dual-channel (Left and Right) data input. Data is transferred directly to RAM buffers using EasyDMA.

Listed here are the main features for PDM:

- Up to two PDM microphones configured as a Left/Right pair using the same data input
- 16 kHz output sample rate, 16-bit samples
- EasyDMA support for sample buffering
- HW decimation filters
- Selectable ratio of 64 or 80 between PDM\_CLK and output sample rate

### 4.2.4 PPI - Programmable Peripheral Interconnect

The programmable peripheral interconnect (PPI) enables peripherals to interact autonomously with each other using tasks and events independent of the CPU. The PPI allows precise synchronization between peripherals when real-time application constraints exist and eliminates the need for CPU activity to implement behavior which can be predefined using PPI.

On each PPI channel, the signals are synchronized to the 16 MHz clock, to avoid any internal violation of setup and hold timings. As a consequence, events that are synchronous to the 16 MHz clock will be delayed by one clock period, while other asynchronous events will be delayed by up to one 16 MHz clock period.

Note that shortcuts (as defined in the SHORTS register in each peripheral) are not affected by this 16 MHz synchronization, and are therefore not delayed.

### 4.2.5 PWM - Pulse Width Modulation

The pulse with modulation (PWM) module enables the generation of pulse width modulated signals on GPIO. The module implements an up or up-and-down counter with four PWM channels that drive assigned GPIOs.

The following are the main features of a PWM module:

- Programmable PWM frequency
- Up to four PWM channels with individual polarity and duty cycle values
- Edge or center-aligned pulses across PWM channels
- Multiple duty cycle arrays (sequences) defined in RAM
- Autonomous and glitch-free update of duty cycle values directly from memory through EasyDMA (no CPU involvement)
- Change of polarity, duty cycle, and base frequency possibly on every PWM period
- RAM sequences can be repeated or connected into loops

### 4.2.6 QDEC - Quadrature Decoder

The Quadrature decoder (QDEC) provides buffered decoding of quadrature-encoded sensor signals. It is suitable for mechanical and optical sensors.

The sample period and accumulation are configurable to match application requirements. The QDEC provides the following:

- Decoding of digital waveform from off-chip quadrature encoder.
- Sample accumulation eliminating hard real-time requirements to be enforced on application.
- Optional input de-bounce filters.
- Optional LED output signal for optical encoders.

The QDEC decodes the output from an incremental motion encoder by sampling the QDEC phase input pins (A and B). The off-chip quadrature encoder is an incremental motion encoder outputting two waveforms, phase A and phase B. The two output waveforms are always 90 degrees out of phase, meaning that one always changes level before the other. The direction of movement is indicated by which of these two waveforms that changes level first.

Invalid transitions may occur, that is when the two waveforms switch simultaneously. This may occur if the wheel rotates too fast relative to the sample rate set for the decoder. The QDEC decodes the output from the off-chip encoder by sampling the QDEC phase input pins (A and B) at a fixed rate as specified in the SAMPLEPER register. If the SAMPLEPER value needs to be changed, the QDEC shall be stopped using the STOP task. SAMPLEPER can be then changed upon receiving the STOPPED event, and QDEC can be restarted using the START task. Failing to do so may result in unpredictable behavior.

Symbol	Description	Min	Typ	Max	Units
Tsample	Time between sampling signals from quadrature decoder	128	-	131072	usec
Tled	Time from LED is turned on to signals are sampled	0	-	511	usec

#### 4.2.7 RTC - Real Time Counter

The Real-time counter (RTC) module provides a generic, low power timer on the low-frequency clock source (LFCLK).

The RTC module features a 24-bit COUNTER, a 12-bit (1/X) prescaler, capture/compare registers, and a tick event generator for low power, tickless RTOS implementation. The RTC will run off the LFCLK. The COUNTER resolution will therefore be 30.517 $\mu$ s. Depending on the source, the RTC is able to run while the HFCLK is OFF and PCLK16M is not available. The software has to explicitly start LFCLK before using the RTC.

Counter increment frequency:  
fRTC [kHz] = 32.768 / (PRESCALER + 1)

The PRESCALER register is read/write when the RTC is stopped. The PRESCALER register is read-only once the RTC is STARTed. Writing to the PRESCALER register when the RTC is started has no effect.

The PRESCALER is restarted on START, CLEAR and TRIGOVRFW, that is, the prescaler value is latched to an internal register (<<PRESC>>) on these tasks.

Examples:

- Desired COUNTER frequency 100 Hz (10 ms counter period)  
PRESCALER = round(32.768 kHz / 100 Hz) - 1 = 327  
fRTC = 99.9 Hz  
10009.576  $\mu$ s counter period
- Desired COUNTER frequency 8 Hz (125 ms counter period)  
PRESCALER = round(32.768 kHz / 8 Hz) - 1 = 4095  
fRTC = 8 Hz  
125 ms counter period

#### 4.2.8 SPI - Serial Peripheral Interface

The SPI master provides a simple CPU interface which includes a TXD register for sending data and an RXD register for receiving data. There are up to four SPI master/ three SPI slave with EasyDMA on the Nordic processor.

The SPI master does not implement support for chip select directly. Therefore, the CPU must use available GPIOs to select the correct slave and control this independently of the SPI master. The SPI master supports SPI modes 0 through 3.

Mode	Clock Polarity	Clock Phase
SPI_MODE0	0(Leading)	0(Active High)
SPI_MODE1	0(Leading)	1(Active Low)
SPI_MODE2	1(Trailing)	0(Active High)
SPI_MODE3	1(Trailing)	1(Active Low)

The different signals SCK, MOSI, and MISO associated with the SPI master are mapped to physical pins. This mapping is according to the configuration specified in the PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers respectively. If the CONNECT field of a PSEL.xxx register is set to Disconnected, the associated SPI master signal is not connected to any physical pin. The PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers and their configurations are only used as long as the SPI master is enabled, and retained only as long as the device is in ON mode. PSEL.SCK, PSEL.MOSI, and PSEL.MISO must only be configured when the SPI master is disabled.

To secure correct behavior in the SPI, the pins used by the SPI must be configured in the GPIO peripheral as described in GPIO configuration on page 399 prior to enabling the SPI. The SCK must always be connected to a pin, and that pin's input buffer must always be connected for the SPI to work. This configuration must be retained in the GPIO for the selected IOs as long as the SPI is enabled. Only one peripheral can be assigned to drive a particular GPIO pin at a time, failing to do so may result in unpredictable behavior.

For more detailed info, refer to the Nordic Semiconductor nRF52840-CKAA datasheet.

#### 4.2.9 UART - Universal asynchronous receiver / transmitter

There are two UARTs on the Nordic processor. One of the UARTs is not available, as it is used by the Nordic processor and the satellite transmitter.

The UART implements support for the following features:

- Full-duplex operation
- Automatic flow control
- Parity checking and generation for the 9th data bit

As illustrated in UART configuration on page 500, the UART uses the TXD and RXD registers directly to transmit and receive data. The UART uses one stop bit.

**Note:** External crystal oscillator must be enabled to obtain sufficient clock accuracy for stable communication. See CLOCK — Clock control on page 82 for more information.

The different signals RXD, CTS (Clear To Send, active low), RTS (Request To Send, active low), and TXD associated with the UART are mapped to physical pins according to the configuration specified in the PSEL.RXD, PSEL.CTS, PSEL.RTS, and PSEL.TXD registers respectively.

If the CONNECT field of a PSEL.xxx register is set to Disconnected, the associated UART signal will not be connected to any physical pin. The PSEL.RXD, PSEL.CTS, PSEL.RTS, and PSEL.TXD registers and their configurations are only used as long as the UART is enabled, and retained only for the duration the device is in ON mode. PSEL.RXD, PSEL.CTS, PSEL.RTS and PSEL.TXD must only be configured when the UART is disabled.

To secure correct signal levels on the pins by the UART when the system is in OFF mode, the pins must be configured in the GPIO peripheral as described in Pin configuration on page 500. Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior. If flow control is not enabled, the interface will behave as if the CTS and RTS lines are kept active all the time.

**UART pin Direction Output Value**

RXD	Input	N/A
CTS	Input	N/A
RTS	Output	1
TXD	Output	1

There are 18 configurable baud rates:

1200  
2400  
4800  
9600  
14400  
19200  
28800  
31250  
38400  
56000  
57600  
76800  
115200  
230400  
250000  
460800  
921600  
1000000

### 4.3 Using and Enabling Peripheral Devices

It is possible to re-assign the general purpose GPIO pins provided on the ST150M module to peripherals such as SPI ports and I<sup>2</sup>C ports using custom code. One existing I<sup>2</sup>C port is brought out to the external pins on the module and an I<sup>2</sup>C HAL is provided in the Globalstar SDK for sending data over that module.

#### Using the external I<sup>2</sup>C pins

I<sup>2</sup>C port pins are available on the module as follows:

- Pin 45 - CLK
- Pin 44 - SDA

A HAL in the SDK supports the following functions for transferring data to and from I<sup>2</sup>C devices via those pins:

```
bool i2c_is_hal_initialized(void);
bool i2c_initialize_hal(i2c_config_t* ptr_i2c_config);
void i2c_shutdown(void);
```

```
ghal_ret_code_t i2c_simple_read(uint8_t u8_address, uint8_t* ptr_rx_data, size_t rx_length);
ghal_ret_code_t i2c_byte_offset_read(uint8_t u8_address, uint8_t u8_offset, uint8_t* ptr_rx_data, size_t rx_length);
ghal_ret_code_t i2c_word_offset_read(uint8_t u8_address, uint16_t u16_offset, uint8_t* ptr_rx_data, size_t rx_length);
ghal_ret_code_t i2c_dword_offset_read(uint8_t u8_address, uint32_t u32_offset, uint8_t* ptr_rx_data, size_t rx_length);
ghal_ret_code_t i2c_simple_write(uint8_t u8_address, uint8_t* ptr_tx_data, size_t tx_length);
ghal_ret_code_t i2c_byte_offset_write(uint8_t u8_address, uint8_t u8_offset, uint8_t* ptr_tx_data, size_t tx_length);
ghal_ret_code_t i2c_word_offset_write(uint8_t u8_address, uint16_t u16_offset, uint8_t* ptr_tx_data, size_t tx_length);
ghal_ret_code_t i2c_dword_offset_write(uint8_t u8_address, uint32_t u32_offset, uint8_t* ptr_tx_data, size_t tx_length);
```

Commented [JM11]: Verify appropriate placement for this content as 4.3

To initialize the HAL, call the function "i2c\_initialize\_hal(NULL);"

This will initialize the pins on the I<sup>2</sup>C port to those provided on the module.

Advanced users can tweak the settings, or assign other module GPIO pins (after removing those pins from the GPIO list) by calling the function above with a pointer to a structure like the following (default values shown):

```
static i2c_config_t m_i2c_hal_default_config =
{
    .u32_scl_pin      = I2C_1_SCL_PIN,
    .u32_sda_pin      = I2C_1_SDA_PIN,
    .frequency        = I2C_1_FREQUENCY,
    .u8_interrupt_priority = APP_IRQ_PRIORITY_LOW,
};
```

Assigning pins for peripheral interface.

To assign pins for use by peripheral devices (SPI ports, I<sup>2</sup>C ports, UARTS, etc), they should first be removed from the GPIO list in the file: UAPI\_GPIO\_list.c

Then, the pins you intend to assign should be reserved in the GPIO library with a call to the function:

```
int32_t g_library_set_pin_pinmap_by_pin_number(uint32_t u32_pin_no);
```

This function takes the Nordic pin number as an argument and will return an error code if the pin is already reserved. The function is found in GSTAR\_Library\_gpio.c and its definition can be included in your source file by including GSTAR\_Library\_gpio.h

To check if a pin is already reserved, use:

```
int32_t g_library_get_pin_pinmap_by_pin_number(uint32_t u32_pin_no);
```

Example of re-assigning an I<sup>2</sup>C peripheral

- Remove the pins you intend to use for the I<sup>2</sup>C peripheral from the GPIO list in UAPI\_GPIO\_list.c (as described in the previous section).
- You could, optionally, add GPIO definitions to the "default" I<sup>2</sup>C pins that you are re-assigning.
  - I2C\_CLK = 43 (1.11)
  - I2C\_SDA = 45 (1.13)
- Use the function "int32\_t g\_library\_set\_pin\_pinmap\_by\_pin\_number(uint32\_t u32\_pin\_no);" to reserve the pins you intend to move the I<sup>2</sup>C peripheral to.
- Create a structure as shown below, with the pin assignments you intend to use.
- Then call:
  - bool ret\_val = i2c\_initialize\_hal(&m\_i2c\_hal\_my\_config)

```
i2c_config_t m_i2c_hal_my_config =
{
    .u32_scl_pin      = I2C_1_SCL_PIN,
    .u32_sda_pin      = I2C_1_SDA_PIN,
    .frequency        = I2C_1_FREQUENCY,
    .u8_interrupt_priority = APP_IRQ_PRIORITY_LOW,
};
```

Setting up a SPI master peripheral

The Nordic nRF52840 can support up to 4 SPI master interfaces. At the Nordic SDK level, the SPI Master interfaces are identified via "instances" 1 – 4. The ST150M uses instance 1 to communicate with its on board GPS receiver. External devices cannot be added to this SPI port since its signals are dedicated to the GPS and are not brought out to external pins. This leaves instances 2 – 4 available for other uses via custom code.



The basic steps in setting up an additional SPI instance is as follows:

- o You will need 4 pins for MOSI, MISO, CLK and CS. If you plan to drive multiple SPI Slave devices off of the same master device instance, you will need an additional pin for each additional CS line, and your HAL will be more complex than the example in the Globalstar ST150 SDK.
- o Remove the 4 (or more) pins you plan to use from the GPIO list in UAPI\_GPIO\_list.c. Note that the pin definition constants that you are setting below will reserve those pins for your SPI port via the GPIO library when the initialization function for your new library is called, if the initialization function retains the lines of code that reserve the pins from the original template.
- o Make a copy the source file "GSTAR\_HAL\_spi\_1.c" and add the new file to the project. A recommended name is "GSTAR\_HAL\_spi\_n.c" where "n" is the instance number that you are using.
- o Edit the new file to change the instance number to the one desired (e. g. "2").
- o In "GSTAR\_HAL\_spi.h" copy the block below:

```
#define SPI_1_CLK_PIN      12 //28
#define SPI_1_MOSI_PIN    NRF_GPIO_PIN_MAP(1,9) //29
#define SPI_1_MISO_PIN    NRF_GPIO_PIN_MAP(1,8) //30
#define SPI_1_CS_PIN     11 //NRF_DRV_SPI_PIN_NOT_USED
#define SPI_1_FREQUENCY   NRF_DRV_SPI_FREQ_125K
#define SPI_1_MODE        NRF_DRV_SPI_MODE_0
#define SPI_1_BIT_ORDER   NRF_DRV_SPI_BIT_ORDER_MSB_FIRST
#define SPI_1_IRQ_PRIORITY NRFX_SPI_DEFAULT_CONFIG_IRQ_PRIORITY
```

- o And make it look something like this:
 

```
#define SPI_2_CLK_PIN    <your clock pin>
#define SPI_2_MOSI_PIN  <your MOSI pin>
#define SPI_2_MISO_PIN  <your MISO pin>
#define SPI_2_CS_PIN    <your CS pin>
#define SPI_2_FREQUENCY <your selection of standard NRF SPI frequencies>
#define SPI_2_MODE      <your SPI mode>
#define SPI_2_BIT_ORDER <your devices bit order> Note: MSB first is by far the most common for SPI
#define SPI_2_IRQ_PRIORITY <your desired IRQ priority> Note: For most cases, the "default" is most appropriate, as above.
```

- o Create the following struct in your new HAL source file:

```
static nrf_drv_spi_config_t m_spi2Config =
{
    .sck_pin      = SPI_2_CLK_PIN,
    .mosi_pin     = SPI_2_MOSI_PIN,
    .miso_pin     = SPI_2_MISO_PIN,
    .ss_pin       = SPI_2_CS_PIN,
    .irq_priority = SPI_2_IRQ_PRIORITY,
    .orc          = 0xFF,
    .frequency    = SPI_2_FREQUENCY,
    .mode         = SPI_2_MODE,
    .bit_order    = SPI_2_BIT_ORDER
};
```

- o Modify the remaining functions in the HAL to point to your new instance of SPI Master port.

Adding a SPI Slave port:

The Nordic nRF52840 can support one SPI slave port. Configuring such a port allows a host processor that may be on your carrier board to send instructions to the Nordic on the ST150M board as a master.

- o You will need to remove the 4 pins you wish to use for the SPI Slave port from the GPIO list as in the sections above.
- o You will need to write a custom HAL for the SPI Slave device.
- o The HAL for the SPI Master device in the Globalstar ST150M SDK should be useful as a template for this HAL. However, you will need to consult the Nordic SDK documentation as a reference how to link your new HAL into the SPI Slave control functions.

## 5 HARDWARE LAYOUT/REFERENCE DESIGN

### 5.1 Pin Assignments

Commented [AW12]: @Kevin McCullough please check if this is current

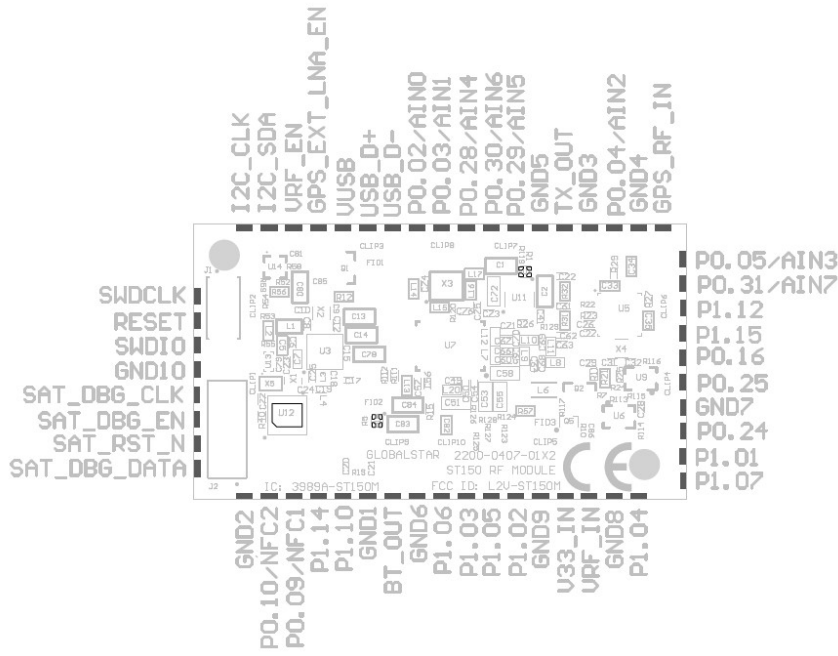


FIGURE 5 - PIN ASSIGNMENTS

Pin	Function	Description	Recommended Usage
SWDCLK	Debug	Serial wire debug clock input for Nordic debug and programming	3.3V level
RESET	Debug	Reset input for Nordic debug and programming	3.3V level
SWDIO	Debug	Serial wire debug I/O for debug and programming	3.3V level
GND10	Power	Ground	–
SAT_DBG_CLK	Debug	Serial wire debug clock input for ASIC debug and programming	3V level
SAT_DBG_EN	Debug	Debug enable input for ASIC debug and programming	3V level

Pin	Function	Description	Recommended Usage
SAT_RST_N	Debug	Reset input for ASIC debug and programming	3V level
SAT_DBG_DATA	Debug	Serial wire debug I/O for debug and programming	3V level
GND2	Power	Ground	–
P0.10/NFC2	Digital I/O / NFC input	GPIO / NFC antenna	3.3V level, Standard drive, low frequency I/O only
P0.09/NFC1	Digital I/O / NFC input	GPIO / NFC antenna	3.3V level, Standard drive, low frequency I/O only
P1.14	Digital I/O	GPIO	3.3V level, Standard drive, low frequency I/O only
P1.10	Digital I/O	GPIO	3.3V level, Standard drive, low frequency I/O only
GND1	Power	Ground	–
BT_OUT	RF	Single-ended Bluetooth antenna connection	–
GND6	Power	Ground	–
P1.06	Digital I/O	GPIO	3.3V level, Standard drive, low frequency I/O only
P1.03	Digital I/O	GPIO	3.3V level, Standard drive, low frequency I/O only
P1.05	Digital I/O	GPIO	3.3V level, Standard drive, low frequency I/O only
P1.02	Digital I/O	GPIO	3.3V level, Standard drive, low frequency I/O only
GND9	Power	Ground	–
V33_IN	Power	+3.3V digital supply input	–
VRF_IN	Power	+3.3V RF supply input	–
GND8	Power	Ground	–

Pin	Function	Description	Recommended Usage
P1.04	Digital I/O	GPIO	3.3V level, Standard drive, low frequency I/O only
P1.07	Digital I/O	GPIO	3.3V level, Standard drive, low frequency I/O only
P1.01	Digital I/O	GPIO	3.3V level, Standard drive, low frequency I/O only
P0.24	Digital I/O	GPIO	3.3V level
GND7	Power	Ground	–
P0.25	Digital I/O	GPIO	3.3V level
P0.16	Digital I/O	GPIO	3.3V level
P1.15	Digital I/O	GPIO	3.3V level, Standard drive, low frequency I/O only
P1.12	Digital I/O	GPIO	3.3V level, Standard drive, low frequency I/O only
P0.31/AIN7	Digital I/O / Analog input	GPIO / Analog input	3.3V level, Standard drive, low frequency I/O only
P0.05/AIN3	Digital I/O / Analog input	GPIO / Analog input	3.3V level
GPS_RF_IN	RF	Single-ended GPS antenna connection	–
GND4	Power	Ground	–
P0.04/AIN2	Digital I/O / Analog input	GPIO / Analog input	3.3V level
GND3	Power	Ground	–
TX_OUT	RF	Single-ended ASIC antenna connection	–
GND5	Power	Ground	–
P0.29/AIN5	Digital I/O / Analog input	GPIO / Analog input	3.3V level, Standard drive, low frequency I/O only
P0.30/AIN6	Digital I/O / Analog input	GPIO / Analog input	3.3V level, Standard drive, low frequency I/O only

Pin	Function	Description	Recommended Usage
P0.28/AIN4	Digital I/O / Analog input	GPIO / Analog input	3.3V level, Standard drive, low frequency I/O only
P0.03/AIN1	Digital I/O / Analog input	GPIO / Analog input	3.3V level, Standard drive, low frequency I/O only
P0.02/AIN0	Digital I/O / Analog input	GPIO / Analog input	3.3V level, Standard drive, low frequency I/O only
USB_D-	USB	Differential USB data connection	–
USB_D+	USB	Differential USB data connection	–
VUSB	Power	USB device +5V input	–
GPS_EXT_LNA_EN	Digital Output	Control signal for external GPS LNA	3.3V level
VRF_EN	Digital Output	Signal for indicating ASIC transmission	3V level
I2C_SDA	Digital I/O	I <sup>2</sup> C serial data	3.3V level
I2C_CLK	Digital Output	I <sup>2</sup> C serial clock	3.3V level

Commented [JG13]: Technically should be I<sup>2</sup>C

Commented [JG14]: Technically should be I<sup>2</sup>C

DRAFT

## 5.2 Reference Circuitry

### 5.2.1 PCB Guidelines

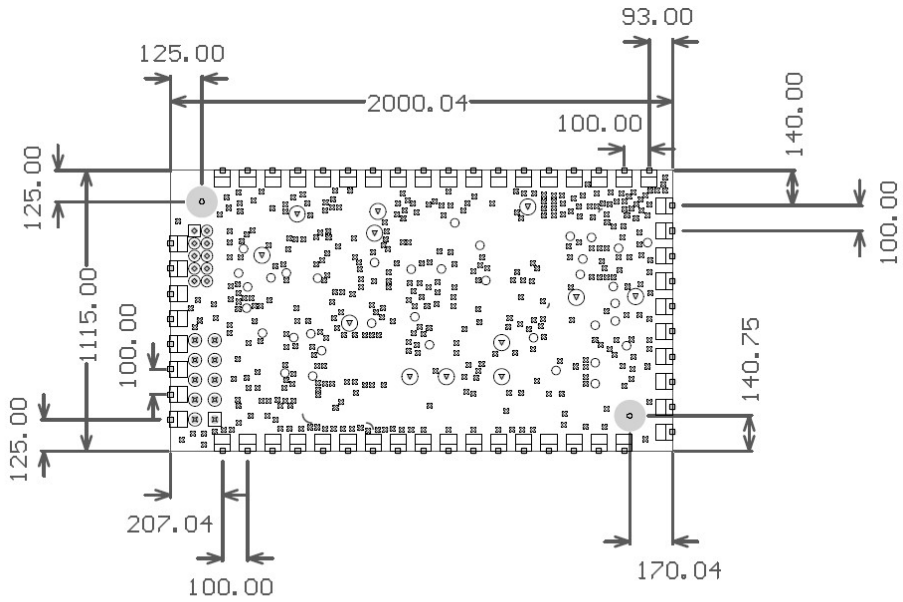
OEM designs are required to follow certain guidelines with respect to the antenna configuration and layout. Physical layout, PCB size and stack-up, component part numbers, etc. may vary. However, the following conditions must be met in order to comply with the ST150M modular grant.

1. Total EIRP of the device shall not exceed the EIRP limitation listed on the Modular Grant. (EIRP: **xx.xx** dBm)
2. Globalstar will manufacture the ST150M with a default conducted output power which allows up to a +5.1 dBi antenna gain (+10% tolerance) before reaching the max EIRP limitation.
3. Globalstar will review all module embedded designs and ensure tuning meets Globalstar's Network specifications without exceeding grant limitations.  
~~The antenna peak gain must not exceed 5.1dB.~~
- ~~3-4.~~ The antenna used must be a patch antenna.
- ~~4-5.~~ The antenna must be passively connected to the RF output of the ST150M using a 50-ohm nominal impedance printed circuit trace. No connectors are allowed on the antenna connection.

If any of the above conditions are not met, the OEM device must be submitted for FCC Part 25 testing (and any additional certification testing required in the intended geographical service area).

In addition, it is strongly recommended that the OEM design include ESD protection circuitry. The circuit should provide a DC blocking capacitor as well as a shunt inductor to short any static discharge from the antenna to ground.

### 5.2.2 Module Dimensions



Dimensions in mils

FIGURE 6 - MODULE DIMENSIONS

DR

### 5.2.3 PCB Layout Example

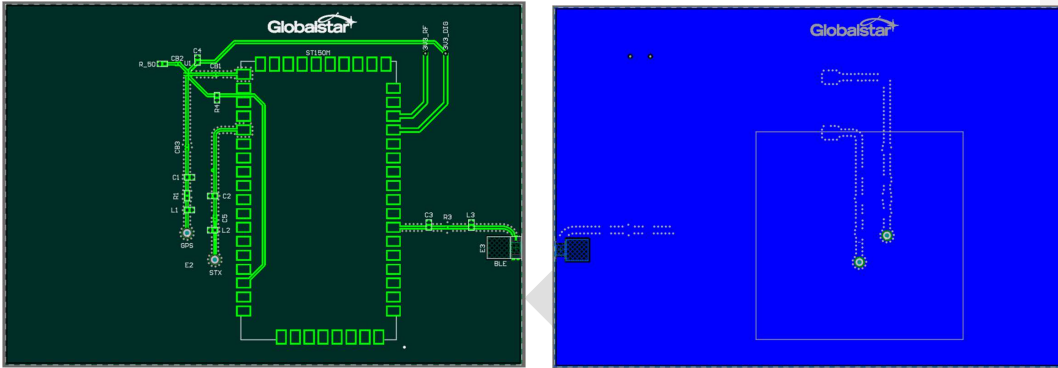


FIGURE 7 - TOP AND BOTTOM LAYOUTS

DRAFT



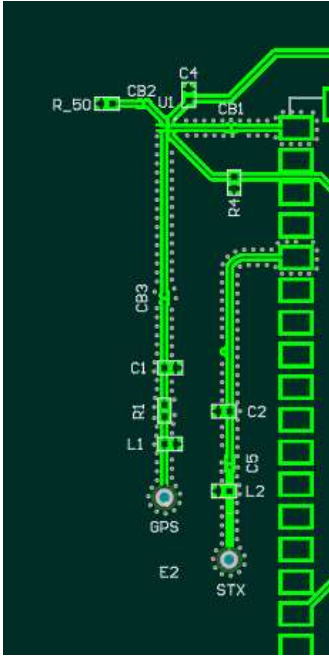


FIGURE 8 - STX/GPS LAYOUT



FIGURE 9 - BLE LAYOUT

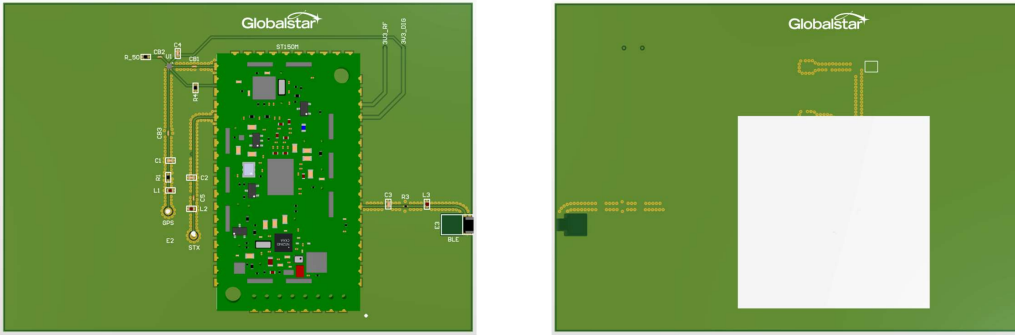


FIGURE 11 - TOP AND BOTTOM PCB VIEWS

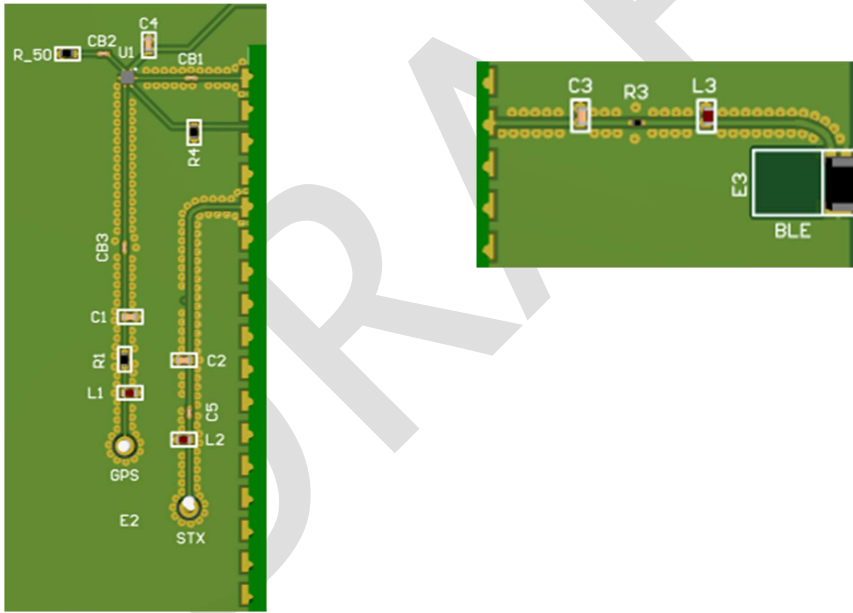
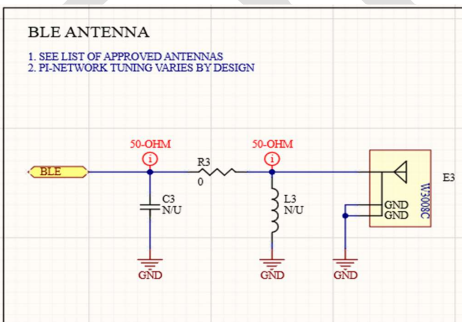
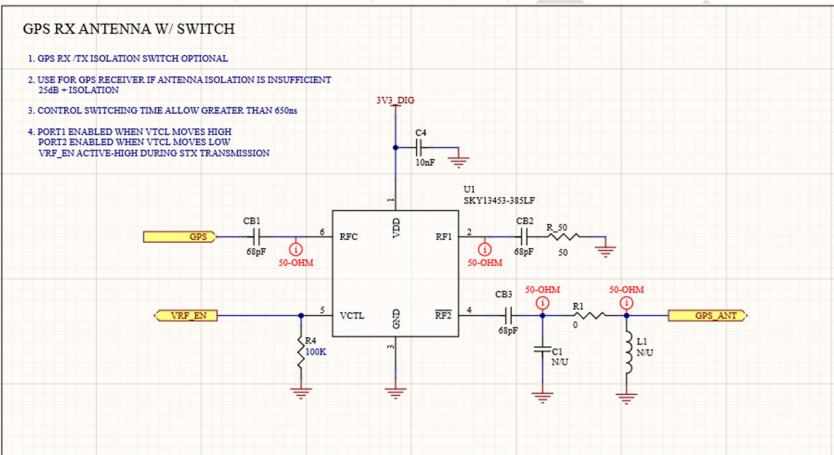
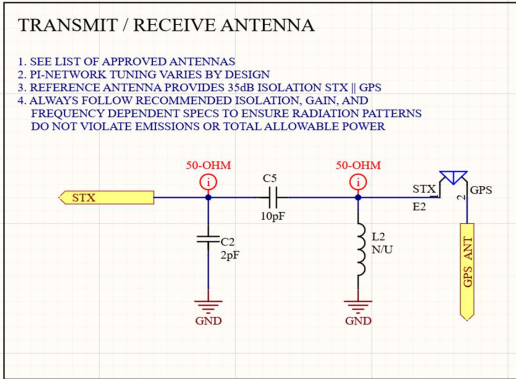


FIGURE 12 - STX/GPS, BLE VIEWS (LEFT TO RIGHT)

### 5.3 Reference Design Schematics



#### 5.4 Transmit/Receive Antenna Requirements

The ideal antenna is a ceramic patch that has left hand circular polarization. This antenna should have a gain ~~of between -1 dB and 5.1 dB~~ in the spatial area between 25 degrees and 90 degrees elevation, with a maximum gain that, when combined with the conducted output of the module at the antenna output, does not exceed the EIRP limitations specified within the Modular Grant Certification (EIRP xx.xx dBm). A candidate antenna meeting these requirements is the TAOGLAS SP-1615-25.4.A.02. See "Single Modular Approval Requirements" and "Globalstar Product Certification" sections below for more details.

Formatted: Highlight

The ST150M has Modular Certification from the FCC (USA) and Industry Canada (Canada) when used with an antenna that meets the specifications above. This means that, in general, no further FCC Part 25 testing or IC testing is required when using an approved antenna in accordance with the Globalstar-specified implementation requirements listed in this document. Note however, that regulatory requirements are subject to change at any time, and it is the responsibility of the integrator to ensure that the product is fully compliant with all applicable regulations in whatever markets the integrator is selling the product.

Final installation must be in compliance with 25.213. The antenna installation and operating configuration of this transmitter must satisfy MPE categorical Exclusion Requirements of 2.1091. The antenna used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be collocated or operating in conjunction with any other antenna or transmitter.

#### 5.5 BLE Antenna Requirements

Globalstar recommends the following antenna, or similar, for optimal BLE performance:

Manufacturer: Pulse Larsen Antennas  
 Part Number: W3008C  
 Frequency: 2400-2483.5MHz  
 Max Gain: +1.3 dBi  
 Nominal Impedance: 50Ω  
 Power Handling: 5W

Note: it will be the responsibility of the VAR/developer to ensure the tuning of the Bluetooth antenna in its intended final installation meets the tuning requirements for the end application.

### 5.6 Single Modular Approval Requirements

The ST150M module provides surface mount pads for the RF output and BLE antenna which must be internally attached passively via the PCB to the approved antennas (referenced above). No antenna connector or any other antenna may be used. In order for any device to obtain certification from Globalstar for use under this modular approval, this condition must be met.

These requirements are enforced as part of the Globalstar Certification process whereby any device utilizing the ST150M module must be submitted to Globalstar for approval prior to being approved for provisioning for operation on Globalstar's network. Adherence to these requirements is an integral part of the Globalstar Certification process.

Since the intent of Globalstar Certification is to ensure that only Globalstar approved antennas are used, the Globalstar certification and provisioning process constitutes a more stringent enforcement of sections 15.203, 15.204(b), 15.204(c), 15.212(a), and 2.929(b) than merely requiring a "unique connector".

In summary, Globalstar will not certify ANY device using the ST150M module UNLESS the approved layout and antennas for satellite and BLE transmission are passively connected to the ST150M on a single circuit board. This meets the requirements of the aforementioned part 15 clauses.

### 5.7 Trace and Cable Losses

The Antenna port (Pin 32) is designed to interface to a 50 Ohm antenna. This connection should feed to the antenna by means of a 50 Ohm impedance trace on the circuit board. The following example impedance calculation is for a co-planar waveguide trace on the top layer of a standard 0.063" FR-4 2-layer board material (60-mil core material) with a dielectric  $\epsilon_r \approx 4.6$  and 1/2 oz. Copper.

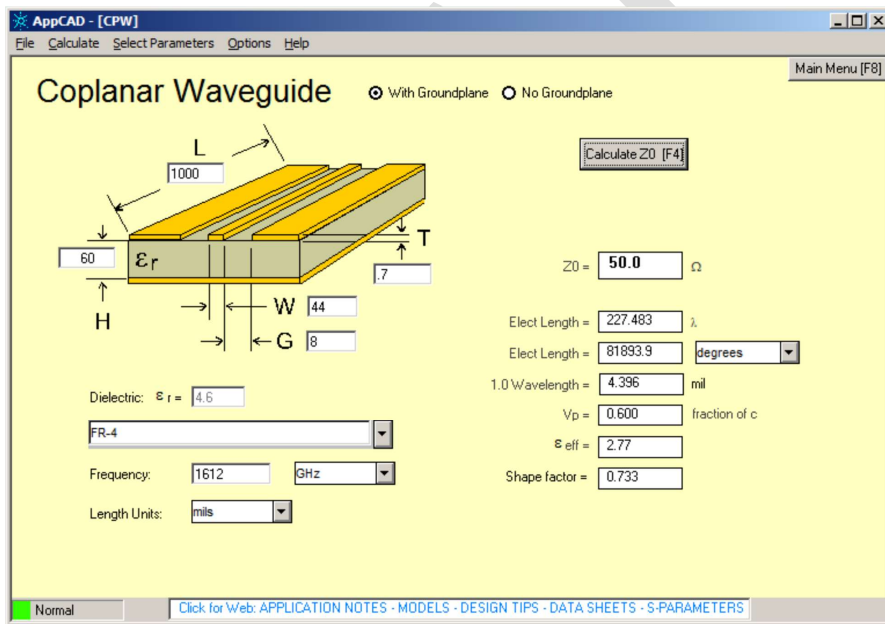


FIGURE 13 - COPLANAR WAVEGUIDE

Commented [JM15]: Appropriate dimensions?

Commented [JM16R15]: Moved to Reference Circuitry 5.2.2

## 6 RECOMMENDED OPERATING CONDITIONS

### 6.1 ABSOLUTE MAXIMUMS

Symbol	Parameter	Max	Units
V33_IN	Input Voltage Range for Digital Inputs	3.6	Volts
I33_IN	Input Current for Digital Inputs	100	mA
VRF_IN	Input Voltage Range for RF circuitry	3.6	Volts
IRF_IN	Input Current for RF Circuitry	500	mA
VUSB	Input Voltage Range for USB Device	5.25	Volts
PRF_IN_GPS	Input Power for GPS Receiver	0	dBm
PRF_IN_BLE	Input Power for Bluetooth Transceiver	+10	dBm
PRF_OUT_BLE	Output Power for Bluetooth Transceiver	+8	dBm
PRF_OUT_TX	Output Power for Satellite Transmitter	+24	dBm
TA	Operating Ambient Temperature Range	85	°C
TS	Storage Temperature Range	85	°C
	Isolation between PRF_OUT_TX and PRF_IN_GPS	25	dB

Commented [JM17]: Verify content and presentation.

Commented [JM18R17]: Resolved by Vindhya email 05-12-2022

### 6.2 HANDLING OF ELECTROSTATIC SENSITIVE DEVICES

Some components on the ST150M PCB are sensitive to electrostatic discharge (ESD). The following general guidelines can help reduce damage due to ESD.

1. Use a grounding bracelet if possible, to minimize charge build-up on personnel.
2. Handle the ST150M PCB by the edges without touching components or printed circuit paths.
3. Store the ST150M PCB in static sensitive packaging whenever possible.

**NOTE:** Failure to adhere to ESD guidelines will void the product warranty.

### 6.3 WIRING

Proper installation and field wiring practices are important to follow. Users and installers should familiarize themselves with the requirements of all applicable codes, laws, and standards.

### 6.4 NOISE SOURCES

Electrical noise is an important consideration in any installation of the ST150M. When possible, sufficient physical separation should be maintained between electrical noise sources and the ST150M.

## 6.5 NORDIC PROCESSOR UNUSED PINS

Due to its sensitivity to IR light, the Nordic processor is covered with opaque, light-blocking epoxy.

Tests have shown that if unused pins on the Nordic processor are set as input without internal pulldowns enabled, the epoxy will cause higher current draw in low power mode.

It is strongly advised to not leave unused Nordic pins at high impedance (input without internal pulldown).

## 6.6 OBSTRUCTIONS

For best performance the ST150M should have an unobstructed view of the sky, either outdoors or in a glass-enclosed area such as a vehicle dashboard.

## 6.7 POWER PERFORMANCE CHARACTERISTICS

# 7 APPLICATION PROGRAMMING INTERFACE (API)

## 7.1 AT Commands

AT commands are list of commands giving accesses to APIs that can configure runtime settings on the device. All end users accessing the device will leverage APIs at the Unified API level. The ST150 Security Library is a UAPI level library intended to prevent access to device configuration APIs via external access.

### 7.1.1 ST150 Device Configuration

ST150 device supports different type of AT commands to configure all supported software components over the NORDIC UART SERVICE (NUS), UART and Configuration service interface.

To access the AT command list and corresponding authentication levels details please refer to document:

8550-0002-01 Globalstar ST150 Unified API Reference

### 7.1.2 Configuration Security

The Configuration security settings include individual settings via three supported interfaces. A user can set the settings such that the module never allows access to auth commands, does so temporarily, or does so all the time. An AT auth command is issued through the command "AT+SECAUTH={1/0}, {passphrase}, where the first parameter is set to 1 if auth permissions are being requested, and a non-null passphrase must be set on the device if enabling auth. This command is only required if the policy is not set to AUTH\_PERMANENT.

For more details on security refer to Section 3 the Globalstar ST150 Unified API Reference.

Query state of authentication using AT+SECAUTH?

Security policy types are supported as described below:

- AUTH\_BASIC - 0  
AUTH\_BASIC never allows any elevated auth permissions.
- AUTH\_PERMANENT - 1  
AUTH\_PERMANENT always allows any commands that requires auth privileges to be passed through.
- AUTH\_TEMPORARY - 2  
AUTH\_TEMPORARY allows the auth command to set elevated auth permissions for the current thread, which will remain in effect until cleared by a subsequent auth command. This state will not persist across a device reset.
- AUTH\_TIMEOUT - 3  
AUTH\_TIMEOUT allows the auth command to set elevated auth permissions for the current thread, which will remain in effect until the elapsed seconds denoted by the applicable timeout setting occur.

Commented [AW19]: Will want @Vindhya Reddy to add in some details on security here

Commented [VR20R19]:

Commented [VR21]: Added details John as request !  
@John Moeberg verify and let me know anything missing.

The ST150 device configuration AT commands can be accessed via three interfaces:

- **ST150 Configuration BLE Service**

This interface requires usage of a 128bit shared AES key to encrypt/decrypt BLE traffic. Only the device owner can access the shared key, either via the Device Management APIs or using the Globalstar mobile application.

Specific details of the ST150 Configuration Service Implementation can be found in the document:

8550-0009-01 R-1 Globalstar IOT Configuration BLE Service Specification

- **Nordic UART BLE Service (NUS)**

NUS is essentially a UART over BLE, allowing AT commands to be used with the device. NUS is disabled by default and must be enabled by a user via the BLE configuration service.

NUS, just like a physical UART, will simply pass incoming commands to the AT handler. The AT handler will check with the security module to see if NUS has permissions to execute any auth command.

- **Physical UART**

The Physical UART handler passes commands and responses directly to the AT command handler. Similar to NUS, the security module can determine the incoming source of the command from the executing thread context.

### 7.1.3 General Configuration

Please follow below steps to configurate the device:

1. The first step of configuration is to get authentication to perform any configuration changes. Query security authentication using AT+SECAUTH? If Security policy is set to 2 or 3, get authentication with AT+SECAUTH = 1, <authorization password>, else send AT commands without any authentication.
2. Next, set device configuration version using below AT command to identify and track your different configuration details.

AT+CFGVER = <major>.<minor>

By default, Configuration version is set to 0.0 and device is considered in the FACTORY Power State. When device is in Factory Power State, device will automatically transition to Full Power off state after 4 hrs. To get more details on ST150 Power management please refer to Section 3.1 **Power Management High Level Overview**.

Set configuration version to non "0.0" version to transition device power state to configured live-state and to avoid automatic power off. To query the version use AT+CFGVER?

3. After setting the configuration version, configure all required parameters of respective component using configuration AT commands. These changes are saved in volatile storage of device.
4. To save configuration settings changed in step 2 to non-volatile storage i.e., Flash storage of device commit operation must be performed using below command:  
AT+COMMIT, Response: OK or ERROR
5. To have the device to perform with all configured setting, do system reboot AT+MCUREBOOT to have device powered up with new configuration settings on startup.
6. A factory reset operation can be run to restore all settings back to factory reset settings. The AT command AT+FACTORYRESET must be executed to succeed. A factory reset operation will also commit settings and reboot the device. In case of factory resets steps 3 and 4 can be avoided.

Commented [VR22]: 1

Commented [JM23]: Are 7.1.3, 7.1.4, and 7.1.5 supposed to be a list under 7.1.2 as the three interfaces? Or is there additional content needed here?

Commented [VR24R23]: 7.1.3, 7.1.4 and 7.1.5 are 3 interfaces John

Commented [VR25R23]:

Commented [JM26]: Verify



## 7.2 URC Notification Settings

URCs can be enabled for each module to allow asynchronous notifications of these system events through external interfaces, such as UART, Nordic UART Service (NUS) and CONFIG Service allowing an integrator to customize their application from an external source.

The following events support configurable notification events.

1. Transmitter
2. Transmitter Queue
3. General Purpose IO
4. Inputs Count Event
5. Track events
6. GPS module
7. Accelerometer
8. Power Management
9. Logging component
10. OTA Events App

URCs for supported component events can be enabled for visibility from external interfaces using following AT commands.

Main features:

1. All URC Notification settings will have distinct AT command to configure each component over UART and Nordic UART Service (NUS) with supported notification flags.
2. AT command to configure URC Notification settings of each component over Config Service (NUS) will have distinct AT command with same notification flags supported over UART & NUS interface settings.
3. Each event type associated with each component will have corresponding notification flags code to enable notification to capture URC's messages on event generation.
4. To enable multiple events notification of a component the addition of all notification flags code assigned is a valid Notify flags.
5. If trying to enable an un-supported notifications flag will return an INVALID PARAMS error code.

### 7.2.1 Transmitter Event Notification Settings

Supported Transmitter Event notifications code <transmitterNotifyFlags>:

- TRANSMITTER\_NOTIFY\_FAULT\_DETECTED = 1
- TRANSMITTER\_NOTIFY\_ACTIVATED = 2
- TRANSMITTER\_NOTIFY\_DEACTIVATED = 4
- TRANSMITTER\_NOTIFY\_RAS\_DETECTED = 8
- TRANSMITTER\_NOTIFY\_RAS\_CLEARED = 16
- TRANSMITTER\_NOTIFY\_TEMP\_THRESHOLD\_DETECTED = 32
- TRANSMITTER\_NOTIFY\_TEMP\_THRESHOLD\_CLEARED = 64
- TRANSMITTER\_NOTIFY\_TEMP\_UNDER\_THRESHOLD\_DETECTED = 128
- TRANSMITTER\_NOTIFY\_TEMP\_UNDER\_THRESHOLD\_CLEARED = 256
- TRANSMITTER\_NOTIFY\_TEMP\_PWR\_EXCP\_DETECTED = 512
- TRANSMITTER\_NOTIFY\_TEMP\_PWR\_EXCP\_CLEARED = 1024

### 7.2.2 Set Transmitter Notify Flags via NUS, UART.

```
AT+TXNSF=<transmitterNotifyFlags>
```

### 7.2.3 Read Transmitter Notify Flags via NUS, UART.

AT+TXNSF?

### 7.2.4 Set Transmitter Notify Flags via Configuration Service

AT+TAXNSF =<transmitterNotifyFlags>

### 7.2.5 Read Transmitter Notify Flags via Configuration Service

AT+TAXNSF?

Example:

- To receive Transmitter Activation and Temperature detected event notification, the valid command to activate notification AT+TXNSF = 34.
- To receive all Transmitter event notifications, the valid command to activate all notifications AT+TXNSF = 2047.
- AT+TXNSF = 2048 will return ERROR.

## 7.3 Message Queue Event Notification Settings

Supported Transmitter Event notifications code <messageQueueNotifyFlags>

- MESSAGE\_QUEUE\_NOTIFY\_FLAGS\_MESSAGE\_QUEUED = 1
- MESSAGE\_QUEUE\_NOTIFY\_FLAGS\_MESSAGE\_TRANSMITTED = 2
- MESSAGE\_QUEUE\_NOTIFY\_FLAGS\_MESSAGE\_DEQUEUED = 4
- MESSAGE\_QUEUE\_NOTIFY\_FLAGS\_RAS\_EXCEPTION = 8
- MESSAGE\_QUEUE\_NOTIFY\_FLAGS\_TEMP\_EXCEPTION = 16
- MESSAGE\_QUEUE\_NOTIFY\_FLAGS\_UNDERTEMP\_EXCEPTION = 32
- MESSAGE\_QUEUE\_NOTIFY\_FLAGS\_POWER\_EXCEPTION = 64

### 7.3.1 Set Message Queue Notify Flags via NUS, UART

AT+QNSF = <messageQueueNotifyFlags>

### 7.3.2 Read Message Queue Notify Flags via NUS, UART

AT+QNSF?

Response

+ QNSF: <messageQueueNotifyFlags>

### 7.3.3 Set Message Queue Notify Flags via Configuration Service

AT+QANSF = <messageQueueNotifyFlags>

### 7.3.4 Read Message Queue Notify Flags via Configuration Service

AT+QANSF?

Response

+ QNSF: <messageQueueNotifyFlags>

Example:

- To receive Message transmitted and Message dequeued event notification, the valid command to activate notification AT+QNSF = 6.
- To receive all Message Queue event notifications, the valid command to activate all notifications AT+QNSF = 127.
- To receive all Transmitter event notifications over configuration service, the valid command to activate all notifications AT+QANSF = 127.
- AT+QNSF = 128 will return ERROR.

## 7.4 GPIO Event Notification Settings

Supported GPIO Events notifications code <GPIONotifyFlags>

- GPIO\_EVENT\_ISR\_TRIGGERED = 1
- GPIO\_EVENT\_PROPERTY\_TRIGGERED = 2
- GPIO\_EVENT\_HARDWARE\_ERROR\_DETECTED = 4

### 7.4.1 Set GPIO Notify Flags via NUS, UART

AT+GPIONSF = <GPIONotifyFlags>

### 7.4.2 Read GPIO Notify Flags via NUS, UART

AT+GPIONSF?

Response

+ GPIONSF: <GPIONotifyFlags>

### 7.4.3 Set GPIO Notify Flags via Configuration Service

AT+GPIOANSF = <GPIONotifyFlags>

### 7.4.4 Read GPIO Notify Flags via Configuration Service

AT+GPIOANSF?

Response

+ GPIOANSF: <GPIONotifyFlags>

Example:

- To receive hardware error event notification, the valid command to activate notification AT+GPIONSF = 4.
- To receive all GPIO event notifications, the valid command to activate all notifications AT+GPIONSF = 7.
- To receive all GPIO event notifications over configuration service, the valid command to activate all notifications AT+GPIOANSF = 7.
- AT+GPIONSF = 8 will return ERROR.

## 7.5 Inputs/Count Event Notification Settings

Supported Inputs/Count Event notifications code <InputsCountNotifyFlags>

- COUNT\_NOTIFY\_FLAGS\_RESERVED\_1 = 1
- COUNT\_NOTIFY\_FLAGS\_RESERVED\_2 = 2
- COUNT\_NOTIFY\_FLAGS\_RESERVED\_3 = 4
- COUNT\_NOTIFY\_FLAGS\_INPUT\_COUNT\_MODULO = 8
- COUNT\_NOTIFY\_FLAGS\_INPUT\_ACCUMULATED\_MODULO = 16
- COUNT\_NOTIFY\_FLAGS\_INPUT\_TRANSITION = 32
- COUNT\_NOTIFY\_FLAGS\_INPUT\_TRANSITION\_HYSTERESIS = 64
- COUNT\_NOTIFY\_FLAGS\_CHANGED\_STATE = 128
- COUNT\_NOTIFY\_FLAGS\_POWER\_ON\_DELAY = 256
- COUNT\_NOTIFY\_FLAGS\_COUNT\_ROLLOVER = 512
- COUNT\_NOTIFY\_FLAGS\_ACCUMULATED\_ROLLOVER = 1024

### 7.5.1 Set Inputs/Count Notify Flags via NUS, UART

AT+CNSF = < InputsCountNotifyFlags>

### 7.5.2 Read Inputs/Count Notify Flags via NUS, UART

AT+CNSF?

Response

+ CNSF: < InputsCountNotifyFlags>

### 7.5.3 Set Inputs/Count Notify Flags via Configuration Service

AT+CAPPNSF = < InputsCountNotifyFlags>

### 7.5.4 Read Inputs/Count Notify Flags via Configuration Service

AT+CAPPNSF?

Response

+ CAPPNSF: < InputsCountNotifyFlags>

Example:

- To receive count modulus and input transition event notification, the valid command to activate notification AT+CNSF = 40 (32+8).
- To receive all Inputs/Count event notifications, the valid command to activate all notifications AT+CNSF = 2047.
- To receive all Inputs/Count event notifications over configuration service, the valid command to activate all notifications AT+CAPPNSF = 2047.
- AT+CNSF = 2048 will return ERROR.
- Configuring any reserved notification will return error.

## 7.6 Track events Notification Settings

Supported Track Event notifications code <TrackNotifyFlags>

- TRACK\_NOTIFY\_FLAGS\_UPDATE\_TRIGGERED = 1
- TRACK\_NOTIFY\_FLAGS\_RESERVED\_1 = 2
- TRACK\_NOTIFY\_FLAGS\_RESERVED\_2 = 4
- TRACK\_NOTIFY\_FLAGS\_RESERVED\_3 = 8
- TRACK\_NOTIFY\_FLAGS\_LOCATION\_ERROR = 16
- TRACK\_NOTIFY\_FLAGS\_IN\_GEOFENCE = 32
- TRACK\_NOTIFY\_FLAGS\_OUT\_GEOFENCE = 64
- TRACK\_NOTIFY\_TRACK\_STATE\_CHANGED = 128

### 7.6.1 Set Track Events Notify Flags via NUS, UART

AT+TNSF = < TrackNotifyFlags>

### 7.6.2 Read Track Events Notify Flags via NUS, UART

AT+TNSF?

Response

+ TNSF: < TrackNotifyFlags>

### 7.6.3 Set Track Events Notify Flags via Configuration Service

AT+TANSF = < TrackNotifyFlags>

### 7.6.4 Read Track Events Notify Flags via Configuration Service

AT+TANSF?

Response

+ TNSF: < TrackNotifyFlags>

Example:

- To receive Track in geofence and out geofence event notification, the valid command to activate notification AT+TNSF = 6.
- To receive all Track event notifications, the valid command to activate all notifications AT+TNSF = 96.
- To receive all Track event notifications over configuration service, the valid command to activate all notifications AT+TANSF = 255.
- AT+TNSF = 256 will return ERROR.
- Configuring any reserved notification will return error.

## 7.7 System OTA Event Notification Settings

Supported OTA Events notifications code <OTAEventNotifyFlags>

• OTA_EVENT_NOTIFY_FLAGS_RESERVED_1	=	1
• OTA_EVENT_NOTIFY_FLAGS_RESERVED_2	=	2
• OTA_EVENT_NOTIFY_FLAGS_RESERVED_3	=	4
• OTA_EVENT_NOTIFY_FLAGS_INPUT_COUNT_MODULO	=	8
• OTA_EVENT_NOTIFY_FLAGS_INPUT_ACCUMULATED_MODULO	=	16
• OTA_EVENT_NOTIFY_FLAGS_INPUT_TRANSITION	=	32
• OTA_EVENT_NOTIFY_FLAGS_INPUT_TRANSITION_HYSTERESIS	=	64
• OTA_EVENT_NOTIFY_FLAGS_CHANGED_STATE	=	128
• OTA_EVENT_NOTIFY_FLAGS_POWER_ON_DELAY	=	256
• OTA_EVENT_NOTIFY_FLAGS_COUNT_ROLLOVER	=	512
• OTA_EVENT_NOTIFY_FLAGS_ACCUMULATED_ROLLOVER	=	1024

### 7.7.1 Set OTA Events Notify Flags via NUS, UART.

AT+CANSF = < OTAEventNotifyFlags >

### 7.7.2 Read OTA Events Notify Flags via NUS, UART.

AT+CANSF?

Response

+ CANSF: < OTAEventNotifyFlags >

### 7.7.3 Set OTA Events Notify Flags via Configuration Service

AT+CCNSF = < OTAEventNotifyFlags >

### 7.7.4 Read OTA Events Notify Flags via Configuration Service

AT+CCNSF?

Response

+ CCNSF: < OTAEventNotifyFlags >

Example:

- To receive count modulus and input transition event notification, the valid command to activate notification AT+CANSF = 40 (32+8).
- To receive all system OTA event notifications, the valid command to activate all notifications AT+CANSF = 2047.
- To receive all system OTA event notifications over configuration service, the valid command to activate all notifications AT+CCNSF = 2047.
- AT+CANSF = 2048, AT+CANSF = 1 , AT+CANSF = 2, AT+CANSF = 4 , AT+CANSF = 3 etc will return ERROR i.e. configuring any reserved notification will return error.

## 7.8 GPS Event Notification Settings

Supported GPS Event notifications code <GPSNotifyFlags>

• GPS_NOTIFY_FLAGS_FIX_STARTED	= 1
• GPS_NOTIFY_FLAGS_FIX_COMPLETED_SUCCESS	= 2
• GPS_NOTIFY_FLAGS_FIX_COMPLETED_ERROR	= 4
• GPS_NOTIFY_FLAGS_HARDWARE_ERROR	= 8
• GPS_NOTIFY_FLAGS_CHANGE_IN_LOCATION_DETECTED	= 16
• GPS_NOTIFY_FLAGS_NMEA_SENTENCE	= 32
• GPS_NOTIFY_TIMEOUT_WITH_FIX	= 64
• GPS_NOTIFY_2D_FIX_ACHIEVED	= 128
• GPS_NOTIFY_3D_FIX_ACHIEVED	= 256
• GPS_NOTIFY_GPS_POSITION_MANUALLY_SET	= 512
• GPS_NOTIFY_FLAGS_GEOFENCE_EVENT_DETECTED	= 1024
• GPS_NOTIFY_FLAGS_GEOFENCE_REENTRY_DETECTED	= 2048

### 7.8.1 Set GPS Notify Flags via NUS, UART.

AT+GNSF = <GPSNotifyFlags>

### 7.8.2 Read GPS Notify Flags via NUS, UART.

AT+GNSF?

Response

+GNSF: <GPSNotifyFlags>

### 7.8.3 Set GPS Notify Flags via Configuration Service

AT+GANSF = <GPSNotifyFlags>

### 7.8.4 Read GPS Notify Flags via Configuration Service

AT+GANSF?

Response

+GNSF: <GPSNotifyFlags>

Example:

- To receive GPS Fix completed and timeout with fix event notification, the valid command to activate notification AT+GNSF = 66.
- To receive all GPS event notifications, the valid command to activate all notifications AT+GNSF = 4095.
- To receive all GPS event notifications over configuration service, the valid command to activate all notifications AT+GANSF = 4095.
- AT+GNSF = 4096 OR GREATER will return ERROR.

## 7.9 Accelerometer Event Notification Settings

Supported Accelerometer Event notifications code <AccelerometerNotifyFlags>

- MESSAGE\_QUEUE\_NOTIFY\_FLAGS\_MESSAGE\_QUEUED = 1
- ACCELEROMETER\_NOTIFY\_FLAGS\_HARDWARE\_ERROR\_DETECTED = 2
- ACCELEROMETER\_NOTIFY\_FLAGS\_IN\_MOTION = 4
- ACCELEROMETER\_NOTIFY\_FLAGS\_NO\_MOTION = 8
- ACCELEROMETER\_NOTIFY\_ACCELEROMETER\_TRIGGERED = 16
- ACCELEROMETER\_NOTIFY\_ORIENTATION\_CHANGE\_TRIGGERED = 32
- ACCELEROMETER\_NOTIFY\_VIBRATION\_TRIGGERED = 64
- ACCELEROMETER\_NOTIFY\_ORIENTATION\_MATCH = 128
- ACCELEROMETER\_NOTIFY\_ORIENTATION\_NOMATCH = 256
- ACCELEROMETER\_NOTIFY\_ORIENTATION\_SELF\_TEST = 512

### 7.9.1 Set Accelerometer Event Notify Flags via NUS, UART.

AT+ANSF = <AccelerometerNotifyFlags>

### 7.9.2 Read Accelerometer Event Notify Flags via NUS, UART.

AT+ANSF?

Response

+ ANSF: <AccelerometerNotifyFlags>

### 7.9.3 Set Accelerometer Event Notify Flags via Configuration Service

AT+AANSF = <AccelerometerNotifyFlags>

### 7.9.4 Read Accelerometer Event Notify Flags via Configuration Service

AT+AANSF?

Response

+ ANSF: <AccelerometerNotifyFlags>

Example:

- To receive accelerometer vibration triggered and in motion event notification, the valid command to activate notification AT+ANSF = 68.
- To receive all Accelerometer event notifications, the valid command to activate all notifications AT+ANSF =511
- To receive all Accelerometer event notifications over configuration service, the valid command to activate all notifications AT+AANSF = 511.
- AT+ANSF = 512 or greater will return ERROR.

## 7.10 Power Management Event Notification Settings

Supported Power Management Event notifications code < PowerManagementNotifyFlags>

- POWER\_NOTIFY\_MCU\_RESET = 1
- POWER\_NOTIFY\_TX\_ASIC\_RESET = 2
- POWER\_NOTIFY\_FLAGS\_POWER\_STATE\_CHANGED = 4
- POWER\_NOTIFY\_FLAGS\_TEMP\_ALERT = 8
- POWER\_NOTIFY\_FUEL\_GAUGE\_ERROR = 16
- POWER\_NOTIFY\_TX\_TEMP\_THRESH\_ALERT = 32
- POWER\_NOTIFY\_VOLT\_ALERT = 64

### 7.10.1 Set Power Management Notify Flags via NUS, UART.

AT+PNSF = < PowerManagementNotifyFlags >

### 7.10.2 Read Power Management Notify Flags via NUS, UART.

AT+PNSF?

Response

+ PNSF: < PowerManagementNotifyFlags >

### 7.10.3 Set Power Management Notify Flags via Configuration Service

AT+PANSF = < PowerManagementNotifyFlags >

### 7.10.4 Read Power Management Notify Flags via Configuration Service

AT+PANSF?

Response

+ PNSF: < PowerManagementNotifyFlags >

Example:

- To receive ASIC RESET and POWER STATE CHANGE event notification, the valid command to activate notification AT+PNSF = 6.
- To receive all Power Management event notifications, the valid command to activate all notifications AT+PNSF = 127.
- To receive all Power Management notifications over configuration service, the valid command to activate all notifications AT+PANSF = 127.
- AT+PNSF = 128 will return ERROR.

## 7.11 External Flash Logging Event Notification Settings

Supported External Flash Logging Event notifications code <ExternalFlashLoggingNotifyFlags>

- MESSAGE\_QUEUE\_NOTIFY\_FLAGS\_MESSAGE\_QUEUED LOGGING\_NOTIFY\_QSPI\_ERROR = 1
- LOGGING\_NOTIFY\_INTERNAL\_FLASH\_ERROR = 2
- LOGGING\_NOTIFY\_GPS\_POSITION\_TRACK = 4

### 7.11.1 Set External Flash Logging Notify Flags via NUS, UART.

AT+LNSF = < ExternalFlashLoggingNotifyFlags >

### 7.11.2 Read External Flash Logging Notify Flags via NUS, UART.

AT+LNSF?

Response

+ LNSF: < ExternalFlashLoggingNotifyFlags >

### 7.11.3 Set External Flash Logging Notify Flags via Configuration Service

AT+LANSF = < ExternalFlashLoggingNotifyFlags >



#### 7.11.4 Read External Flash Logging Notify Flags via Configuration Service

```
AT+LANSF?
Response
+ LNSF: < ExternalFlashLoggingNotifyFlags >
```

##### Example:

- To receive INTERNAL FLASH ERROR and GPS POSITION TRACK event notification, the valid command to activate notification AT+LANSF = 6.
- To receive all External Flash Logging event notifications, the valid command to activate all notifications AT+LANSF = 7.
- To receive all External Flash Logging event notifications over configuration service, the valid command to activate all notifications AT+LANSF = 127.
- AT+LANSF = 8 or greater will return ERROR.

### 7.12 GPIO/Inputs Configuration

#### 7.12.1 Getting a list of all the GPIO.

To get a list of all GPIO, send the command: "AT+GPIOLIST=0"

The meaning of the single argument is "GPIO type" where 0 = All, 1 = Available, 2 = Reserved.

The response format is as follows:

```
+GPIOLIST:<key>,<property>,<pin>,<is enabled>,<is fixed>,<type>,<set state>,<read state>,<is active high>,<ISR received>,<description>
```

The following is the meaning of the fields:

- Key = An identifier of the pin used in commands that reference it.
- Property: 0 = GPIO, 1 = IRQ, 2 = Tamper detect, 3 = line power detect, 4 = advertising enable, 5 = uart TX, 6 = uart RX.
- Pin = The Nordic pin designation.
- Is enabled = Boolean value . Is the pin enabled.
- Is fixed = Boolean. It is an actual physical pin.
- Type: 0 = input, 1 = output
- Set state: High or low if output.
- Read State: High or low if input.
- Is active high: Boolean. True = a high on the pin is the "exception" or "alert" condition for the input.
- ISR Received.
- Description: A string description of the GPIO purpose.

The output of a module to “AT+GPIOLIST=0” with none of the GPIO reserved yet is shown below. This shows all inputs available.

```
+GPIOLIST:0,0,2,0,1,0,0,0,0,0,P0.02
+GPIOLIST:1,0,3,0,1,0,0,0,0,0,P0.03
+GPIOLIST:2,0,4,0,1,0,0,0,0,0,P0.04
+GPIOLIST:3,0,5,0,1,0,0,0,0,0,P0.05
+GPIOLIST:4,6,16,1,1,0,0,0,1,0,P0.16
+GPIOLIST:5,0,24,0,1,0,0,0,0,0,P0.24
+GPIOLIST:6,5,25,1,1,1,0,0,1,0,P0.25
+GPIOLIST:7,0,28,0,1,0,0,0,0,0,P0.28
+GPIOLIST:8,0,29,0,1,0,0,0,0,0,P0.29
+GPIOLIST:9,0,30,0,1,0,0,0,0,0,P0.30
+GPIOLIST:10,0,31,0,1,0,0,0,0,0,P0.31
+GPIOLIST:11,0,33,0,1,0,0,0,0,0,P1.01
+GPIOLIST:12,0,34,0,1,0,0,0,0,0,P1.02
+GPIOLIST:13,0,35,0,1,0,0,0,0,0,P1.03
+GPIOLIST:14,0,36,0,1,0,0,0,0,0,P1.04
+GPIOLIST:15,0,37,0,1,0,0,0,0,0,P1.05
+GPIOLIST:16,0,38,0,1,0,0,0,0,0,P1.06
+GPIOLIST:17,0,39,0,1,0,0,0,0,0,P1.07
+GPIOLIST:18,0,42,0,1,0,0,0,0,0,P1.10
+GPIOLIST:19,0,44,0,1,0,0,0,0,0,P1.12
+GPIOLIST:20,0,46,0,1,0,0,0,0,0,P1.14
+GPIOLIST:21,0,47,0,1,0,0,0,0,0,P1.15
+GPIOLIST:22,2,0,0,0,0,0,0,0,0,Tamper
+GPIOLIST:23,3,0,0,0,0,0,0,0,0,Line Pwr
+GPIOLIST:24,4,0,0,0,0,0,0,0,0,Adv Enable
OK
```

### 7.12.2 Configuring inputs:

To reserve one of the inputs in the list above, use:

```
AT+GPIOCFGIN=<reserved>,<key>,[<is_active_high>]
```

Examples:

Reserve key 0 as active high  
AT+GPIOCFGIN=1,0,1

Reserve key 1 as active low:  
AT+GPIOCFGIN=1,1,0

Reserve key 2 as active low (note, omitting the “active high” Boolean defaults to active low):  
AT+GPIOCFGIN=1,2

To get a list of enabled inputs, use the command “AT+INLIST=<source\_type>”.

For input source type: 0 = All, 1 = GPIO, 2= BLE, 3 = Accelerometer.

Example of issuing "AT+INLIST=0" after issuing the GPIOCFGIN commands shown above:

```
AT+INLIST=0
+INLIST 3,48,0,Accel Motion Event
+INLIST: 3,49,0,Accel Vibration Event
+INLIST: 3,50,1,Accel In-Motion
+INLIST: 3,51,1,Accel Orientation Match
+INLIST: 3,52,0,Accel Orientation Changed Event
+INLIST: 1,2,1,P0.02
+INLIST: 1,3,1,P0.03
+INLIST: 1,4,1,P0.04
```

### 7.12.3 Mapping inputs:

To map inputs into the on air messages.

Up to 8 inputs can be mapped.

The input mapping command:

```
AT+INMAP=<input_source_id>,<input_num>
```

Where:

<input\_source\_id> is the Nordic pin number (not the "key") of the input you want to map. There are also "virtual pins" available with the following pin numbers:

- 48:Accel Motion Event
- 49:Accel Vibration Event
- 50:Accel In-Motion
- 51:Accel Orientation Match
- 52:Accel Orientation Changed Event

Examples:

Map Accel Motion event to input 1 in OTA messages:

```
AT+INMAP=48,1
```

Map actual pin 3 to input 2 in OTA messages:

```
AT+INMAP=3,2
```

"AT+INMAP?" Returns the mapped inputs. For example, issuing this query after doing the mappings listed above results in:

```
AT+INMAP?
+INMAP: 3,2
+INMAP: 48,1
OK
```

### 7.12.4 Getting the input state of mapped inputs:

The command "AT+GETINSTATE=<input number>" gets the state of the input. The argument is the OTA message input field that it is mapped to, not the Nordic input number or the key.

For example, in the above examples, virtual input 48, "ACCEL In Motion" is mapped to input 1, so the following response, shows us that we are not in motion:

```
AT+GETINSTATE=1
+GETINSTATE: 1,0
OK
```

### 7.12.5 GPIO Notify settings:

The GPIO API can be configured to generate notifications in the form of URC messages upon GPIO input events.

The available events are:

```
#define GPIO_NOTIFY_ISR_TRIGGERED           0x00000001
#define GPIO_NOTIFY_PROPERTY_TRIGGERED     0x00000002
#define GPIO_NOTIFY_HARDWARE_ERROR_DETECT 0x00000004
```

Use the following command to turn on the desired URC notifications:

```
AT+GPIONSF =<gpioNotifyFlags>
```

Where <gpioNotifyFlags> is a bitmask of the above values.

The active flags can be queried via:

```
AT+GPIONSF?
```

The notification enumeration is as follows:

```
typedef enum
{
    GPIO_EVENT_TYPE_INVALID = 0,
    GPIO_EVENT_ISR_TRIGGERED,
    GPIO_EVENT_PROPERTY_TRIGGERED,
    GPIO_EVENT_HARDWARE_ERROR_DETECTED,
} gpio_event_types_enum;
```

The URC notifications will come back in the standard URC format, e. g:

**+URC: "GPIO",<MessageSource\_t><GPIOEventType>,eventspecificparams"**

The most common event to be reported with be the ISR event, which put simply, means that the state of that input. The format of the ISR event URC is:

```
+URC: "GPIO",<MessageSource_t>,<GPIOEventType>,<key>,<pin>,<readState>
```

In one of the examples above, we reserved the pin who's key is "0", and the Nordic pin number is "2". When this pin transitions from low to high, the URC generated will be:

```
+URC: "GPIO",17,1,0,2,1
```

### 7.13 Enabling and Configuring the UART:

Use the following command to configure the UART:

```
AT+UART2PROP=<is_enabled>,<tx_gpio_key>,<rx_gpio_key>,<baud_rate_enum>]
```

This is the baud rate enumeration:

```
typedef enum
{
    GHAL_UART2_BAUD_1200 = 0,
    GHAL_UART2_BAUD_2400,
    GHAL_UART2_BAUD_4800,
    GHAL_UART2_BAUD_9600,
    GHAL_UART2_BAUD_14400,
    GHAL_UART2_BAUD_19200,
    GHAL_UART2_BAUD_28800,
    GHAL_UART2_BAUD_31250,
    GHAL_UART2_BAUD_38400,
    GHAL_UART2_BAUD_56000,
    GHAL_UART2_BAUD_57600,
    GHAL_UART2_BAUD_76800,
    GHAL_UART2_BAUD_115200,
    GHAL_UART2_BAUD_230400,
    GHAL_UART2_BAUD_250000,
    GHAL_UART2_BAUD_460800,
    GHAL_UART2_BAUD_921600,
    GHAL_UART2_BAUD_1000000,
} ghal_uart2_baud_t;
```

Use "AT+UART2PROP?" To query the configuration.

Here is a query showing the factory default configuration for the ST150 Module. Note that the UART is enabled by default. Default baud rate is 9600 baud.

```
AT+UART2PROP?
+UART2PROP:1,6,4,3
OK
```

## 8 SOFTWARE HIGH LEVEL OVERVIEW

This section serves to illustrate the high-level software functionality of the ST150 Module. For detailed software description and design guidance please refer to the following documentation:

- ST150 Unified API
- ST150 Software Design Guide

### 8.1 BLE Configuration Service and Interfaces

BLE Configuration Service is an encrypted BLE service used by the Globalstar Mobile App to provide trusted user authentication and access to all device configuration settings. It is necessary to first create a user account, as outlined in the Quick Start section, in order to securely access the device settings.

The BLE Configuration Service serves as the root of trust for device configuration changes. Through the Configuration Service, permissions can be set for each external interface, physical UART and BLE UART (NUS). It is possible to permanently disable, permanently enable, or temporarily enable user authentication access for each interface with the following options.

The BLE Configuration Service can, optionally, be disabled if device discovery is not wanted. However, in doing so, provision should be made to be able to recover an un-advertising device via AT commands through one of the following External Interfaces, or through a GPIO Special Function. Disabling the BLE Configuration Service will also disable Nordic DFU Service (needed for Nordic firmware updates) as well as ASIC DFU Service (needed for ASIC firmware updates).

### 8.1.1 Device Advertising and Discoverability

If NUS is enabled, the device will advertise and remain discoverable, however, the BLE Configuration Service, Nordic DFU Service, and ASIC DFU Service will not be accessible. In order to completely prevent any device advertising and discoverability, BLE Configuration Service and NUS must both be disabled.

### 8.1.2 External Interfaces

Device is supported by two Interfaces currently as following:

- Physical UART
  - Can be enabled/disabled via GPIO Module AT commands
- BLE Nordic UART Service (NUS), BLE-UART
  - Can be enabled/disabled via BLE Settings AT commands

### 8.1.3 External Interface Security

Device external interfaces are protected with various security configurations:

#### 8.1.3.1 Permanently Disabled

Only read-only commands can be executed

#### 8.1.3.2 Permanently Enabled

All read and write configuration commands can be executed without requiring an authentication passphrase to be entered.

#### 8.1.3.3 Temporarily Enabled

All read-only command can be executed any time, however, write command require a pre-set passphrase to be entered via an auth command prior to executing a write command.

The auth command enables write-access, however, is not saved to persistent storage and will reset when the device resets or when an auth disable command is executed.

#### 8.1.3.4 Temporarily Enabled with Timeout

All read-only command can be executed any time, however, write command require a pre-set passphrase to be entered via an auth command prior to executing a write command. Additionally, elevated authorization status will timeout automatically after a pre-set configurable time.

The auth command enables write-access, however, is not saved to persistent storage and will reset when the device resets or when an auth disable command is executed.

## 9 Unified API Modules

The Unified API is Globalstar’s SDK that incorporates the same API command whether used externally over UART, or through source code directly. All AT commands rely on documented C APIs that are available to system integrators writing custom firmware applications. The C layer relies on FreeRTOS as the embedded RTOS.

The Unified API abstracts all hardware, platform-specific, and value-added EDGE features through an abstracted API as shown below. An event-driven architecture makes it possible to catch global module events at the UAPI or App level, and easily customize a new application.

Similarly, URCs (Unsolicited Response Codes) can be enabled for each module to allow asynchronous notifications of these system events through external interfaces, such as UART, allowing an integrator to customize their application from an external source.

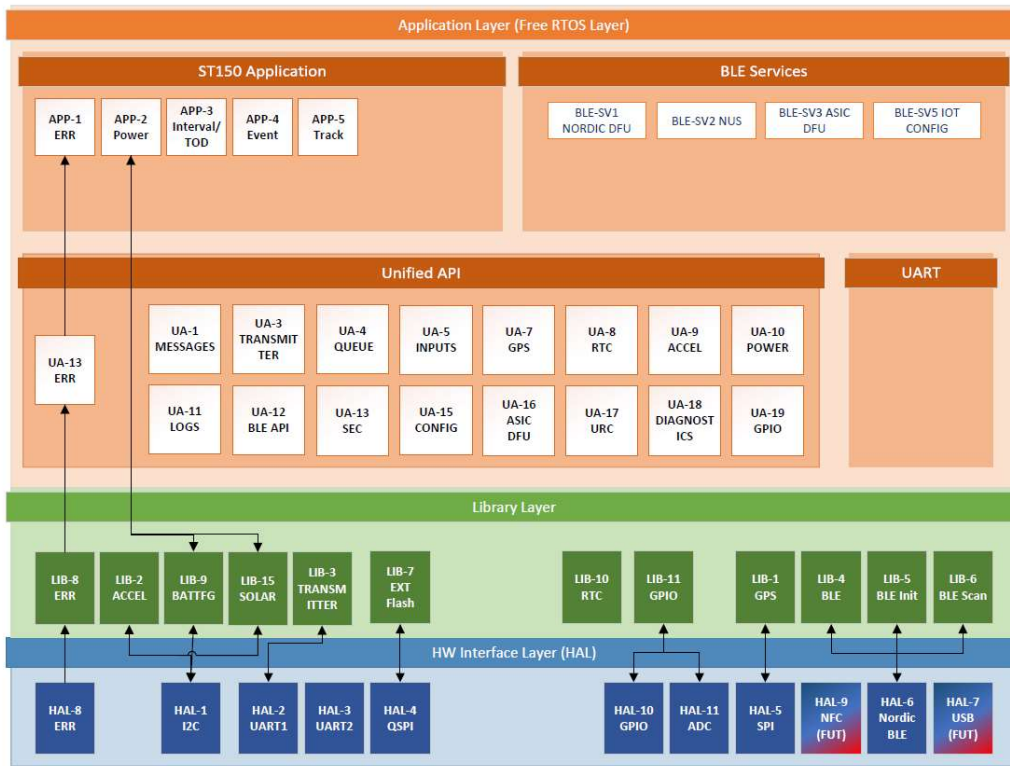


FIGURE 15 - UNIFIED API MODULES

Commented [JM27]: Per Michael Pergola

This needs to be updated as the modules have been updated/deleted. For instance, there is no UA-12 BLE API or BLE-SV7 or BLE-SV8 available yet.

Commented [VR28R27]:

Commented [VR29R27]: @Michael Pergola BLE API's on UAPI are implemented, but not the sensors support.

Commented [VR30R27]:

## 9.1 Messages

The Messages module is used to read and set properties on all over-the-air (OTA) message profiles. OTA message profiles are consumed by the application and map UAPI module events and sequences of events into OTA messages.

Legacy messages are intended for backwards compatibility with previous Globalstar products. Modular messages allow for future scalability and payload customization by use of a configuration byte for each modular message profile. For example, a Device Reset event can send a Modular Turned On Message that includes UTC time from the Modular Message UTC Time profile and custom raw payload. It could also be configured to send a device turned on message that includes the Modular Message GPS Position data with speed/heading/accuracy data.

Modular messages may include more than one contained message type, and always includes a status byte for each contained message to indicate how to parse the payload for each contained message. When decoding the message on the backoffice, it is necessary to interrogate the status byte of each included message to properly decode the payload that follows.

The available OTA message profiles include the following:

- Type 0 Subtype 0-5 Standard Message: Legacy GPS/Input 1 & 2/Vibration info
- Type 0 Subtype 6 Speed and Heading: Legacy Speed/Heading/Input1 & 2 info
- Type 1 Truncated: Legacy GPS + raw message
- Type 1 Subtype 59 Sensor Truncated No Timestamp: Smart 1 Sensor Message with GPS info
- Type 1 Subtype 63 Sensor Truncated With Timestamp: Smart 1 Sensor message with GPS info and timestamp
- Type 2 Raw Message: Raw message
- Type 2 Subtype 59 Sensor Raw No Timestamp: Smart 1 Raw Sensor Message
- Type 2 Subtype 63 Sensor Raw With Timestamp: Smart 1 Raw Sensor Message With Timestamp
- Type 3 Subtype 24 Count Accumulate: Legacy count/accumulate info for Input 1/2
- Type 3 Subtype 0-20 Modular Messages:
  - Mod Msg Turned On: Device Turned On Message
  - Mod Msg UTC Time: UTC timestamp message
  - Mod Msg UTC Time Sensor: Reserved for future use
  - Mod Msg GPS Position Track: Used by the track application
  - Mod Msg GPS Pos Track Non Maskable: Used by non-maskable track application (ie: SOS mode)
  - Mod Msg GPS Pos General: General GPS position profile that can be used by non-track and non-sensor uses, such as for a device turned-on message, inputs 1-8, and time-of-day/interval messages
  - Mod Msg GPS Pos Sensor: Reserved for future use
  - Mod Msg Input Config 1-8: Customizable input OTA messages that can be mapped to software inputs 1-8 and include count, accumulate, state, and transition info
- Type 3 Subtype 31 Health Check: Used to send diagnostics information about the device. Message is required to send at least once a month, but may be sent more frequently. If only one message is sent per month, no usage charge is incurred.

### 9.1.1 Message Priority

Each message type has a configurable priority, 1 (lowest) – 3 (highest), with the exception of raw and truncated messages. Raw and truncated message priorities are specified on a per-message basis for the message being queued into the transmit queue (See Transmit Queue)

### 9.1.2 Burst Count

The burst count specifies how many times a message should attempt transmission. This is required for transmit-only IoT satellite transmissions since message delivery success cannot be determined by the transmitter.



### 9.1.3 Min/Max Burst Interval Seconds

The min/max burst interval determines the scheduled time at which queued messages attempt re-transmission, and in combination with other queued messages sent as an interleaved burst (see Transmit Queue).

### 9.1.4 Interval and Time-Of-Day Message List

The interval and Time-Of-Day-Message List allows for either interval or time-of-day messages to be used. Both options allow for selecting many of the message types discussed in the Messages component to send as OTA messages.

Interval messages specify a fixed interval that the selected message type should send, whereas time-of-day messages specify a specific time of the day the selected message type should send.

Minimum time spacing constraints between messages is required.

### 9.1.5 Transmitter

The transmitter will not transmit any OTA messages until it has been activated. In order to activate a transmitter, the steps described in the Quick Start section should be followed.

#### 9.1.5.1 End Date

When the transmitter is activated, an end date is set that will cause the transmitter to stop transmitting once that date is reached.

#### 9.1.5.2 Channel Selection

The transmitter is required to switch channels depending on its global location. The ST150M can automatically channel-steer the ASIC when a GPS position is obtained or input, or the user can manually set the operating channel.

#### 9.1.5.3 Transmission Exceptions

Occasionally, a transmission exception may occur and prevent a queued message from attempting transmission. This can be due to an over-temperature condition or RAS exception. If the device is located in close proximity to a network ground station, federal regulations prohibit transmission.

Transmission exceptions are logged to the device's OTA message logging section.

#### 9.1.5.4 Transmitter Events

Transmitter events can be sent via URCS to indicate asynchronous events that are occurring, and can be very useful for debugging.

#### 9.1.5.5 Transmit Queue

The transmit queue can hold up to 10 pending and current messages.

#### 9.1.5.6 Interleaved Messages

Messages in the transmit queue are automatically scheduled to send in an interleaved pattern to increase throughput of messages. The number of interleaved messages sent at a time can be adjusted.

### 9.1.6 Message Preemption and Aborted Transmissions

Once the transmit queue is full, adding additional messages can pre-empt existing ones or return an error depending on the queue pre-emption settings.

#### 9.1.6.1 Preemption Disabled

No message queue requests will preempt any existing messages. User must explicitly abort a message or wait for queued messages to complete transmitting.

#### 9.1.6.2 Preemption by Priority

In the event the queue is full, the oldest message found at the lowest priority that can be aborted will be removed from the send queue to make room for the next incoming message. If no lower priority messages are found, an error will be

returned.

#### **9.1.6.3 Preemption by Time and Priority**

In the event the queue is full, the oldest message with lowest priority found that can be aborted will be removed to make room for the next incoming message.

#### **9.1.6.4 OTA Message Logging**

All messages that are added to the transmit queue are logged into device flash for diagnostic retrieval. These entries contain information about the message, which includes timestamp, message UMN (unique message number), message size, and any flags that indicate a transmission failure. Transmission failures can occur due to reasons defined in the Transmitter section, or due to a message being aborted or pre-empted.

#### **9.1.6.5 Transmit Queue Events**

Transmit Queue events can be sent via URCs to indicate asynchronous events that are occurring, and can be very useful for debugging.

#### **9.1.7 GPIO**

Available GPIOs can be setup for inputs, module event interrupt outputs, special input functions, or for supported peripheral usage.

##### **9.1.7.1 Reserved vs Non-Reserved GPIO Entries**

Once a GPIO is assigned to a role, it is considered reserved. A reserved GPIO cannot be assigned to another role without being un-reserved first

##### **9.1.7.2 Inputs**

When a GPIO is setup as an input, the GPIO component will generate events on edge transitions.

##### **9.1.7.3 Event Interrupts**

When a GPIO is setup as an Event Interrupt, the corresponding pin can be shared amongst other module IRQs as long as the active high/low polarity is the same.

This feature makes it possible to have module interrupts assert and wake an external MCU or device. Each event that generates an interrupt can be read via the Interrupt AT command and doing so will also clear all active interrupts.

#### **9.1.8 Special Function Inputs**

Special Function Inputs make it possible to add device-specific features in the application and will send events that describe input transitions for these features. These functions include the following, and can be mapped to Software Inputs 1-8 (See Inputs section):

##### **9.1.8.1 Tamper Detection**

Used to trigger a tamper event

##### **9.1.8.2 Line Power Detection**

Used for line-power detection status. Line Power status is described in the monthly Health Check OTA message, but can also be mapped to an available software input.

##### **9.1.8.3 Advertising Enable**

Used to re-enable BLE Configuration Service advertising temporarily if it has been disabled. This makes it possible to re-connect the Globalstar Mobile App if the BLE Configuration Service is normally disabled.

##### **9.1.8.4 Fixed and Non-Fixed Special Functions**

Special functions can be assigned to un-reserved pins if the special function entry is not fixed. If the special function entry is fixed, it must use the fixed pin assignment.

### 9.1.9 UART

The physical UART can be enabled/disabled with a custom baud rate and configured to use available GPIOs for its transmit and receive pins.

The UART currently only supports disabled and enabled modes of operation, where the receive line is always powered and listening.

### 9.1.10 GPIO Events

Transmit Queue events can be sent via URCs to indicate asynchronous events that are occurring, and can be very useful for debugging.

### 9.1.11 Inputs

8 software inputs are available that can be used for count and accumulate functions on physical GPIOs, accelerometer events, and other abstracted event types.

GPIOs configured as inputs and special functions can be mapped to software inputs.

### 9.1.12 OTA Messages

OTA messages can include count, accumulate, undesired state, and transition events from inputs that support both open and closed states. Inputs that are based on singular events (where there is only an active state), will only support count and transition events.

Mod Msg Input Config 1-8 OTA messages make it possible to describe inputs and inputs states through OTA messages.

Up to 64 bytes of user-defined payload can accompany OTA messages when utilizing Modular Msg Input Config 1-8 OTA profiles (See Messages section).

### 9.1.13 Input Events

Input events can be sent via URCs to indicate asynchronous events that are occurring and can be very useful for debugging.

### 9.1.14 Track Functions

Supported track modes include the following:

#### 9.1.14.1 Normal Track Mode

In Normal Track Mode, location updates are sent periodically based on the at-rest update interval setting when the unit is "At-Rest", and at the in-motion update interval when the unit is "In-Motion".

Power states can reduce the in-motion reporting interval (see Power Management).

#### 9.1.14.2 Dock Mode

Dock mode is similar to Normal Track Mode, however, requires the accelerometer to report In-Motion based on the continuous-vibration and continuous-lacking-vibration settings as verification for movement in addition to the GPS's change-of-location event. This track mode adds resiliency by utilizing two sources for movement verification.

#### 9.1.14.3 Theft-Alert Mode

Theft Alert mode creates a geofence around the device after power-up.

If inside the geofence, OTA position messages are sent the same as in Normal Track Mode.

If outside the geofence, OTA position messages are sent more frequently at the in-motion interval.

#### 9.1.14.4 Enabling/Disabling Track Modes

Track modes can be enabled and canceled/disabled via Track Mode AT commands.

#### **9.1.14.5 Previous Track Point**

Previous Track Point is a feature that helps compensate for any missed tracks by sending the delta latitude/longitude of the last two positions at preset intervals. This makes it possible to re-construct a track up to two reporting intervals beyond the initial reported location.

#### **9.1.14.6 OTA Messages**

OTA messages can include legacy standard and speed/heading messages, or Modular GPS Position messages.

Up to 64 bytes of user-defined payload can accompany OTA messages when utilizing the Mod Msg GPS Position Track OTA profile (See Messages section).

#### **9.1.14.7 Track Events**

Track events can be sent via URCS to indicate asynchronous events that are occurring, and can be very useful for debugging.

#### **9.1.14.8 Non Maskable Track Mode**

Non Maskable Track Mode is similar to regular Track Mode, except it is intended to be used like an SOS track mode, where a priority 4 OTA message will pre-empt any and all messages in the transmit queue (See Transmit Queue section).

Non Maskable Track Mode contains its own modular message profile, or can be used with legacy standard and speed/heading messages (see Messages section).

#### **9.1.15 GPS Module**

The GPS module can use the internal GPS chipset or leverage an external GPS via the set GPS position AT commands. The GPS generates events that are used by the application's track mode.

##### **9.1.15.1 Stale Time**

GPS stale time is the amount of elapsed time allowed to occur while utilizing the current GPS fix in OTA messages that send GPS positions.

If the stale time has elapsed, a new fix will be attempted. Requesting new GPS fixes consumes considerable power, so the stale time can provide a balance for optimizing this tradeoff.

##### **9.1.15.2 Fix Timeout**

The GPS fix timeout refers to the maximum amount of time the GPS module will wait to obtain a position before giving up and putting the GPS back into a low-power state.

##### **9.1.15.3 Fix Settle Time**

The GPS fix settle time refers to the maximum amount of time the GPS module will allow to try to meet the minimum accuracy requirements before giving up. The fix settle time begins once a fix is obtained or the fix timeout occurs.

##### **9.1.15.4 GPS Accuracy**

The GPS accuracy refers to the minimum estimated horizontal accuracy that should be met in order to declare the current fix position. The GPS fix settle time will wait for this condition to be met or timeout.

##### **9.1.15.5 Geofence Position**

The Geofence position is a configuration setting that can store the root of the current geofence. GPS events can notify the application when the device has transitioned in or out of the geofence, when a gps fix occurs.

The geofence can be set to a circle or square with configurable radius/half-width.

#### **9.1.15.6 Change of Location Detection**

The Geofence position is a configuration setting that can store the current position. GPS events can notify the application when the device has transitioned out of the stored position, when a gps fix occurs. This feature is particularly useful for the application's tracking module, to be able to offload this calculation to the GPS module.

The change of location detection is based on a circle with configurable radius.

#### **9.1.15.7 Power Settings**

The GPS supports cold and hot starts but allows a warm selection for compatibility. The time-to-fix (TTF) can be reduced by using warm starts but requires additional low-power consumption by the GPS to maintain ephemeris data.

#### **9.1.15.8 NMEA Pass Through**

The NMEA sentences captured can be passed to the application or external interfaces via URC messages, if the corresponding GPS event flag is enabled.

#### **9.1.15.9 Manual Mode**

In manual mode, set GPS Position AT commands can be used to set the GPS position reported by the GPS module, as well as synchronize the RTC clock.

#### **9.1.15.10 RTC Clock**

The RTC clock is automatically updated when a GPS fix occurs

#### **9.1.15.11 RTC Module**

The RTC module saves and synchronizes the current UTC date time whenever a GPS fix is obtained internally or via external APIs.

#### **9.1.15.12 RTC Stale Time**

The RTC time may drift over time, and thus, an adjustable stale time setting will cause the RTC Module to request a GPS fix.

#### **9.1.15.13 Set RTC Time**

A set RTC time API is available. Additionally, the RTC will be synchronized to UTC time if the set GPS Position API is used (see GPS Module Section).

#### **9.1.16 Accelerometer Module**

The accelerometer can detect acceleration changes independent from orientation changes. Either or both of these changes can be enabled to generate a motion event.

##### **9.1.16.1 Motion/Vibration/In-Motion Events**

Additionally, it is possible to require consistent motion for a period of time before declaring the device is "In-Motion". This is accomplished with the vibration continuous time setting. When this setting is non-zero, successive motion events are used to meet the continuous vibration setting requirement.

Likewise, it is possible to require consistent no-motion to occur to transition out of the "In-Motion" state, which is accomplished with the vibration lacking continuous time seconds setting.

### 9.1.16.2 External Accelerometer

It is possible to use an external accelerometer and set instantaneous motion via the set motion API. This API can be used if the accelerometer mode is set to the manual state. In this state, the internal accelerometer will be placed in a low power state.

Reading motion via the get motion API does **not** return instantaneous motion as the set command does, but rather the “In-Motion” state of the device. Thus, in order for the get motion command to return a non-zero value, the set motion API must be called repetitively, once every two seconds at a minimum, until the continuous vibration requirement is met. Only after that condition is met, will the get motion API return a non-zero value.

### 9.1.16.3 Accelerometer Sensitivity

The accelerometer can use a 2G, 4G, or 16G scale reference for acceleration detection, in addition to a sensitivity setting, 1-10, which is proportional to that scale. So, a 2G scale with a 1 sensitivity will yield the most sensitive motion detection possible, whereas a 16G scale with a 10 sensitivity will yield the least sensitive setting.

Changes to accelerometer sensitivity occur in real-time, and events can be used to enable URC messages to quickly test the results of sensitivity changes.

### 9.1.16.4 Orientation Detection

6D orientation match detection can be specified such that transitions in or out of configurable orientation states can be identified.

### 9.1.16.5 Input and OTA Events

Accelerometer events can be mapped to Software Input functions (See Inputs Module Section). Events that support active and inactive states can be mapped to count, accumulate, transition, and undesired state functions. Events that do not support open and closed states, such as a motion event, can be mapped to count and transition functions.

Mod Msg Input Config 1-8 OTA messages make it possible to describe accelerometer events and states through OTA messages.

### 9.1.17 Power Management

Power management supports 3 power states as follows:

#### 9.1.17.1 Normal Operation

In Normal Operation, tracking occurs at the at rest and in-motion tracking intervals.

#### 9.1.17.2 Partial Suspended

In the Partial Suspended state, tracking occurs at the low battery update interval, if the low battery update interval is less than the normal or in-motion interval respective of the current tracking state.

#### 9.1.17.3 Full Suspended

In the full suspended state, the tracker and transmitter are suspended and cannot be used.

#### 9.1.17.4 Power Management State Changes

Power management state changes can be set with the Power Management Set Power State API. Based on the user's implementation and power architecture, the ST150M's power state can be directed to the appropriate state.

### 9.1.18 Temperature Notifications

Temperature notification events can be configured for high/low temperatures on the ASIC and MCU. URCs for these events can be enabled for visibility from external interfaces.

### 9.1.19 Data Logging

Data logging includes logging of GPS track positions, OTA message metadata, and controllable diagnostic logging from the device's Unified API modules.

#### 9.1.19.1 GPS Position Logging

GPS Position Logging can optionally be disabled to prevent sensitive location data from persisting on the device.

#### 9.1.19.2 OTA Metadata Logging

OTA Metadata Logging logs information about all OTA messages that enter the Transmit Queue (See Transmit Queue section for logging details).

#### 9.1.19.3 Diagnostic Logging

Diagnostic logging can be used to enable various levels of debug verbosity from each UAPI module, and have those logs save to flash for retrieval later. This can be very useful for troubleshooting an issue, and being able to review logs after-the-fact via the Globalstar Mobile App.

### 9.1.20 BLE Settings

#### 9.1.20.1 Advertising Prefix

The advertised name of the device can have up to 3 preceding characters customized with the Advertising Prefix setting

#### 9.1.20.2 BLE Power

The transmit BLE power can be adjusted with the BLE Power setting

#### 9.1.20.3 BLE Advertising Interval

The BLE Advertising Interval can be customized. The advertising interval may be disrupted or delayed due to BLE connections or Smart One Sensor operations (see Smart One Sensor Service Section)

#### 9.1.20.4 BLE Config Service Advertising Enabled

The BLE Config Service Advertising Enabled controls whether or not the config service, Nordic DFU (firmware update) service, and ASIC DFU services are advertised.

See BLE Configuration Service and Interfaces section for additional details

#### 9.1.20.5 Nordic UART Service (NUS) Enabled

Nordic UART Service is essentially a UART over BLE, and can be optionally enabled or disabled. See BLE Configuration Service and Interfaces section for additional details

#### 9.1.20.6 Smart One Sensor Service Settings

Smart One Sensor Service is a BLE service the ST150M offers where up to 10 sensor nodes may advertise data to be sent over-the-air when data is available and needs to send. The ST150M will switch from peripheral mode to central mode at the Sensor Service Scan Interval below, and look for any advertising sensor nodes.

Up to 10 sensor nodes can work in conjunction with a single ST150M.

When a sensor advertisement is discovered, the sensor's data is captured and sent as a Type 59 or Type 63 Sensor OTA Message (See Messages Section), and the sensor becomes whitelisted for a minimum of several seconds.

ECDSA is used in combination with a configurable AES public key to provide authentication of messages. Additionally, the device's ESN or UID can be used to authenticate sensor nodes meant only for the intended ST150M.

## 10 TEST MODES

### 10.1 Transmitter Test Modes

The following 3 Test Modes are supported to verify Transmitter capabilities:

#### CW Test Mode (0x01)

- **AT+TXTESTMODE = 1**

CW\_MODE Mode // Send out an un-modulated carrier for 30 seconds

The CW Test Mode command requests the device to enter the CW test mode. This will cause the device to output a CW signal for 30 seconds after which it will turn off the RF output and return to sleep mode.

- **Send Test Packet (0x02)**

AT+TXTESTMODE = 2

TEST\_PACKET Mode // Send out a single 9-byte test packet

The Send Test Packet command requests the device to transmit a single test packet. This test packet consists of the following payload: 0x80AAFOFOAAFOFOFO

- **Modulation Test Mode (0xFC 0x03)**

AT+TXTESTMODE = 3

3 = MOD\_MODE // Send out a modulated carrier for 30 seconds

The Modulation Test Mode command requests the device to enter the Modulation test mode. This will cause the device to output back-to-back test packets continuously for 30 seconds after which it will turn off the RF output and return to sleep mode.



## 11 ORDERING INFORMATION/DEVICE MARKINGS/ACTIVATION INFORMATION

### 11.1 Device Marking

The ST150 Module is marked with a PCB label as shown in the following figure:

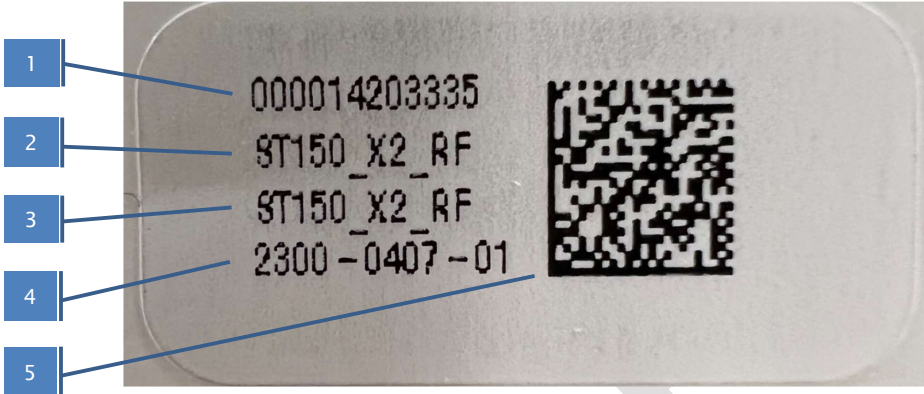


FIGURE 16 - PCB LABEL

1. 12 Digit UID. 12<sup>th</sup> digit is a Luhn checksum of the previous 11 digits used to prevent incorrect data entry during activation/device management
2. Hardware SKU - Used by Globalstar to identify different base hardware configurations
3. Product SKU - Used by Globalstar to identify different products using a specific Hardware SKU as the base module
4. Globalstar Part Number
5. Scannable DataMatrix Code - Contains product information in the format <UID>,<Hardware SKU ID>,<Hardware SKU String ID>,<Product SKU ID>,<Product SKU String ID>,<Globalstar Part Number>

### 11.2 Box Labels

The following Figures show the box labels and packing slips used for the ST150 Module:

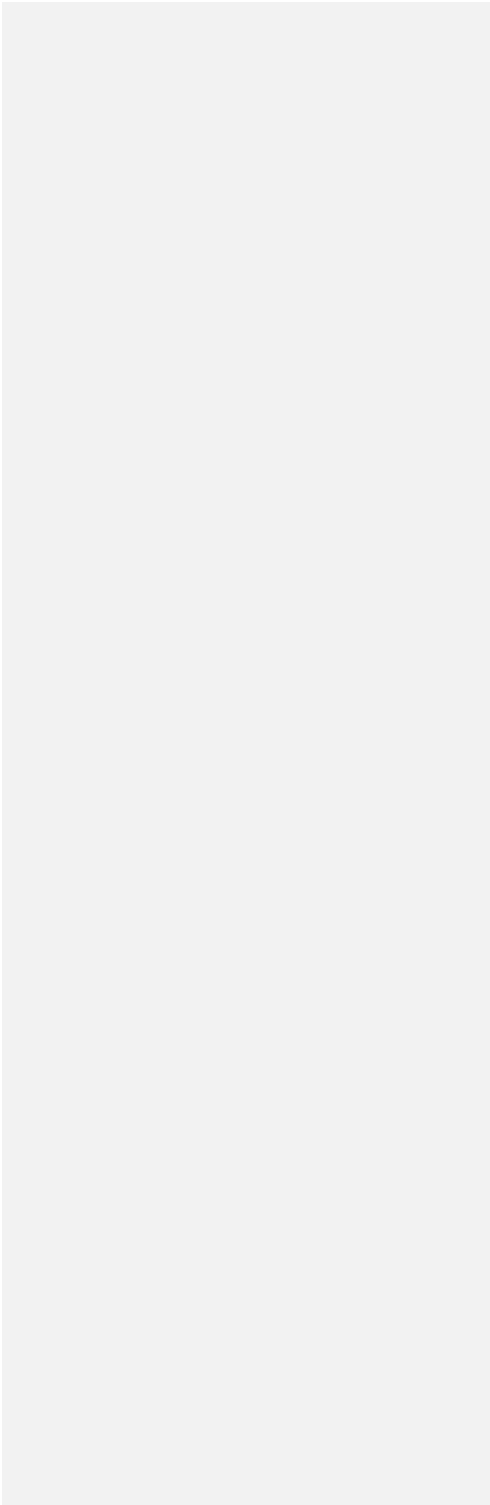


FIGURE 17 - BOX LABEL AND PACKING SLIP

### 11.3 How to Order

Contact your RSM or go to [www.globalstar.com](http://www.globalstar.com) for ordering information.

DRAFT



## 11.4 Globalstar Product Certification

Before a customer's end product can be used on the Globalstar network, it must receive a Globalstar Product Certification. The certification process ensures that the customer's product meets Globalstar's internal system standards and has received the applicable regulatory approvals for the countries it will be operating in.

Typical certification process flow:

- Customer designs their product using the ST150M.
  - NOTE:** Over the Air (OTA) testing strongly recommended during design phase to catch performance bugs before submission for certification.
- Customer notifies their Globalstar Regional Sales Manager (RSM) that they want to start the Globalstar certification process.
- Customer completes ST150M Product Submission Form and Equipment Details Form
  - The application provides the list of items needed for certification and the product's configuration.
- Globalstar will require 2 - 3 units for antenna tuning/tuning verification and OTA testing at Globalstar.
  - Potential issues are highlighted/corrective actions made.
- Customer obtains all applicable regulatory approvals for end product.

**NOTE:** Typical reasons for delays in the certification process include:

- Problems found during Antenna tuning:
  - No housing submitted with circuit.
  - Housing submitted is missing pieces, not in the final shape or uses materials other than what will be used in final version of housing.
- Problems found, especially during OTA testing.
- Lack of required documentation.
- Regulatory certificate issues.

## 12 GENERAL WARNINGS

**Warning – Modifications:** Changes or modifications to the ST150M not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

**Warning – Blasting Area:** To avoid interference with blasting operations, turn your ST150M off when in a "Blasting Area" or in areas posted "Turn off two-way radio." Obey all signs and instructions.

**Warning – Potential Explosive Atmosphere:** Turn off the ST150M or any product with the ST150M installed when in any area with a potentially explosive atmosphere and obey all signs and instructions. (End products specifically designed to work in Explosive Atmospheres are to follow their certification guidelines.)

**Warning – Pacemakers:** The Health Industry Manufacturers Association recommends that a minimum separation of six (6") inches be maintained between the ST150M and a pacemaker to avoid potential interference with the pacemaker.

**Warning – Hearing Aids:** Some digital wireless devices may interfere with some hearing aids. In order to prevent such interference, you may want to consult the manufacturer of your hearing aid.

DRAFT

## 13 REGULATORY

### Radio Astronomy Site Avoidance

The ST150M complies with FCC CFR25.213. The customer end-product must comply with the requirements for Radio Astronomy avoidance. See Globalstar document GS-07-1248 REMOTE TELEMETRY SERVICE FREQUENCY PLAN FOR SIMPLEX TRANSMITTERS for more details.

### Regulatory Notices

The ST150M Board has received Federal Communications Commission authorization under FCC Rules Part 25 and Part 15C as a modular transmitter. Final installation must be in compliance with 25.213 (see section 6 above). Integrators must complete final end-product certifications dependent upon their end application's use. The installation and operating configurations of this transmitter must satisfy MPE categorical Exclusion Requirements of 2.1091. The antenna used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be collocated or operating in conjunction with any other antenna or transmitter.

The ST150M module will satisfy FCC/IC modular transmitter requirements only when used with the antenna type specified in Section 2.6. No power amplifiers may be used under the terms of this modular approval. No trace antennas or any other type of antenna are approved for use under the terms of this modular approval. It is permissible to use different antenna manufacturers provided the same antenna type that has similar in-band and out-of-band radiation patterns and antenna gain (equal to or less than the antenna gain listed above) is used. The module provides a surface mount pad for the RF output which must be internally attached passively via the PCB to the approved antenna. No antenna connector, cable, or any other type of antenna may be used. In order for any device to obtain certification from Globalstar for use on the Globalstar satellite network under this modular approval, all conditions listed above must be met. Use of any antenna other than those listed in (LIST SECTION/ANNEX) will require proof of compliance (peak gain). This device contains license-exempt transmitter(s)/receiver(s) that comply with Innovation, Science and Economic Development Canada's license-exempt RSS(s). Operation is subject to the following two conditions:

- (1) This device may not cause interference.
- (2) This device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

- (1) l'appareil ne doit pas produire de brouillage, et
- (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

**NOTE:** Any additional host device the module is integrated into will require regulatory testing and certification (C2PC for FCC and C4PC for ISED).

The ST150M BOARD module has been labeled with its own FCC and Industry Canada (IC) ID numbers, and if the FCC/IC ID numbers are not visible when the module is installed inside another device, then the outside of the finished product into which the module is installed must also display a label referring to the enclosed module:

Contains Transmitter Module FCC ID: L2V-ST150M IC: 3989A-ST150M

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

The VAR's user's manual shall include the following statements:

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy, and if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment OFF and ON, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

**WARNING:** Changes or modifications not expressly approved by Globalstar may render the device non-compliant to FCC and other regulatory body standards for operation and may void the user's authority to operate the equipment.



This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This Class B digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de classe B est conforme à la norme NMB-003.

Hereby, Globalstar declares that this ST150M is in compliance with the essential requirements and other relevant provisions of Directive 2014/53/EU. The declaration of conformity may be consulted at [www.Globalstar.com/Regulatory](http://www.Globalstar.com/Regulatory).

**NOTICE:** This equipment complies with FCC, IC and CE RF Exposure Limits. A minimum of 20 centimeters (8 inches) separation between the device and the user and all other persons should be maintained.

AVIS: Cet équipement est conforme aux RSS-102 Limites d'exposition RF. Un minimum de 20 centimètres (8 pouces) entre l'appareil et l'utilisateur et toutes les autres personnes devrait être maintenue.	
Transmit Frequencies: 1611.25 Mhz - 1618.75 Mhz (4 Channels) Max Power Out: 25.82 dBm EIRP	
FCC ID: <a href="#">5/BL2V-ST150M</a> CAN ICES-3(B)/NMB-3B IC: 3989A-5/8-ST150M  Complies with FCC standards. FOR HOME OR OFFICE USE	
The ST150M has been so constructed that the product complies with the requirement of Article 10(2) as it can be operated in at least one Member State as examined and the product is compliant with Article 10(10) as it has no restrictions on being put into service in all of the EU except Ireland. The ST150M cannot be marketed in Ireland.  <b>IE Not to be Marketed in Ireland</b>	
Anatel Certification stamp for operation in <a href="#">Brazil</a>	
TBD	
Portuguese	Produto não acabado, de uso interno, cuja integração em outros equipamentos pode requerer nova certificação.
English	Unfinished product, for internal use, the integration of which into other equipment may require further certification.

**Commented [JM31]:** Per Michael Pergola: If this is not going to be certified in Brazil, I suggest to remove this until it is certified.

**Commented [JM32R31]:** Per Matthew Kapral: "ST150 is currently being certified, but ST150M has not been started yet. I will follow up with the Brazil team to see if they can add the module to their currently open project. Either way, I think this can stay as TBD, as it shouldn't be long to obtain certification."

## 14 ACRONYM LIST

<b>ACK</b>	Acknowledgement
<b>APP</b>	Application
<b>ASCII</b>	American Standard Code for Information Interchange
<b>ASIC</b>	Application-Specific Integrated Circuit
<b>BLE</b>	Bluetooth Low Energy
<b>C</b>	Celsius
<b>CE RED</b>	Conformité Européene Radio Equipment Directive (Europe)
<b>CPU</b>	Central Processing Unit
<b>CRC</b>	Cyclic Redundancy Check
<b>dBm</b>	Decibel-Milliwatts
<b>DC</b>	Direct Current
<b>DFU</b>	Device Firmware Update
<b>EIRP</b>	Effective, or Equivalent, Isotropically Radiated Power
<b>ESD</b>	Electrostatic Discharge
<b>ESN</b>	Electronic Serial Number
<b>F</b>	Fahrenheit
<b>FCC</b>	Federal Communications Commission
<b>FTP</b>	File Transfer Protocol
<b>GATT</b>	Generic Attribute Profile
<b>GLONASS</b>	Global Navigation Satellite System
<b>GND</b>	Ground
<b>GPIO</b>	General-Purpose Input/Output
<b>GPS</b>	Global Positioning System
<b>HTTP</b>	Hypertext Transfer Protocol
<b>HW</b>	Hardware
<b>I<sup>2</sup>C</b>	Inter-Integrated Circuit
<b>IC</b>	Industry Canada
<b>iOS</b>	iPhone Operating System
<b>ISED</b>	Innovation, Science and Economic Development (Canada)
<b>JTAG</b>	Joint Test Action Group
<b>LEO</b>	Low Earth Orbiting
<b>mA</b>	Milliamps
<b>mAh</b>	Milliamp Hour

Commented [JG33]: Technically should be I<sup>2</sup>C



---

<b>MBR</b>	Master Boot Record
<b>mm</b>	millimeter
<b>mW</b>	milliwatt
<b>N/A</b>	Not Applicable
<b>NAK</b>	Negative-Acknowledgement
<b>NMEA</b>	National Marine Electronics Association
<b>OEM</b>	Original Equipment Manufacturer
<b>OTA</b>	Over the Air
<b>PCB</b>	Printed Circuit Board
<b>PLL</b>	Phase-Locked Loop
<b>POST</b>	Power-On Self-Test
<b>RF</b>	Radio Frequency
<b>RoHS</b>	Restriction of Hazardous Substances
<b>RTU</b>	Remote Transmitter Unit
<b>RX</b>	Receive
<b>SAR</b>	Specific Absorption Rate
<b>SCL</b>	Serial Clock
<b>SDA</b>	Serial Data
<b>SDK</b>	Software Development Kit
<b>SMA</b>	Sub-Miniature Version A
<b>STU</b>	Satellite Transmitter Unit
<b>TX</b>	Transmit
<b>UART</b>	Universal Asynchronous Receiver/Transmitter
<b>V</b>	Volts
<b>VAM</b>	Value Added Manufacturer
<b>VAR</b>	Value Added Reseller
<b>VDC</b>	Voltage Direct Current
<b>VNA</b>	Vector Network Analyzer
<b>WEEE</b>	Waste Electrical & Electronic Equipment (Europe)



9150-0125-01 R-1