SWARM

Swarm M138 Modem
PRODUCT MANUAL

Revision 1.20 · February 2022

Legal Notices

This User Manual provides user information and is provided “as is.” Swarm Technologies and its affiliated companies, directors, officers, employees, agents, trustees or consultants (“Swarm”) assume no responsibility for any typographical, technical, content or other inaccuracies in this User Manual. Swarm reserves the right to revise this User Manual or withdraw it at any time without prior notice. You can find the current version of this Swarm Modem User Manual at www.swarm.space.

SWARM MAKES NO GUARANTEES, CONDITIONS, WARRANTIES, OR REPRESENTATIONS, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED REPRESENTATIONS, GUARANTEES, CONDITIONS OR WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, SATISFACTORY QUALITY, NON-INTERFERENCE, ACCURACY OF INFORMATIONAL CONTENT, OR ARISING FROM A COURSE OF DEALING, LAW, USAGE, OR TRADE PRACTICE, USE, OR RELATED TO THE PERFORMANCE OR NONPERFORMANCE OF ANY PRODUCTS, ACCESSORIES, FACILITIES, SERVICES OR USER INFORMATION, EXCEPT AS EXPRESSLY STATED IN THE LIMITED WARRANTY. ANY OTHER STANDARDS OF PERFORMANCE, GUARANTEES, CONDITIONS AND WARRANTIES ARE HEREBY EXPRESSLY EXCLUDED AND DISCLAIMED TO THE FULLEST EXTENT PERMITTED BY LAW. THIS DISCLAIMER AND EXCLUSION SHALL APPLY EVEN IF THE EXPRESS LIMITED WARRANTY PROVIDED IN THE LEGAL INFORMATION BOOKLET FAILS OF ITS ESSENTIAL PURPOSE. FURTHER, THE LIMITED WARRANTY COVERS THE SWARM M138 MODEM (INCLUDING HARDWARE, SOFTWARE AND/OR FIRMWARE) AND ACCESSORIES ONLY, AND NO WARRANTY IS MADE AS TO USER INFORMATION AND/ OR COVERAGE, AVAILABILITY OR GRADE OF SERVICE PROVIDED BY SWARM SEPARATELY FOR SWARM SATELLITE SERVICES.

IN NO EVENT SHALL SWARM BE LIABLE, WHETHER IN CONTRACT OR TORT OR ANY OTHER LEGAL THEORY, INCLUDING WITHOUT LIMITATION STRICT LIABILITY, GROSS NEGLIGENCE OR NEGLIGENCE, OR IMPLIED WARRANTY, FOR ANY DAMAGES IN EXCESS OF THE PURCHASE PRICE OF THE SWARM M138 MODEM (INCLUDING HARDWARE, SOFTWARE AND/OR FIRMWARE) AND/OR ACCESSORIES AND/OR THE COST OF SWARM SATELLITE SERVICES PROVIDED, NOR SHALL SWARM BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, OR PUNITIVE DAMAGES OF ANY KIND, OR LOSS OF REVENUE OR PROFITS, LOSS OF BUSINESS, LOSS OF PRIVACY, LOSS OF USE, LOSS OF TIME OR INCONVENIENCE, LOSS OF INFORMATION OR DATA, SOFTWARE OR APPLICATIONS OR OTHER FINANCIAL LOSS CAUSED BY THE SWARM MODEM-M138 (INCLUDING HARDWARE, SOFTWARE AND/ OR FIRMWARE) AND/OR ACCESSORIES AND/OR THE SWARM SATELLITE SERVICES, OR ARISING OUT OF OR IN CONNECTION WITH THE ABILITY OR INABILITY TO USE THE SWARM MODEM-M138 (INCLUDING HARDWARE, SOFTWARE AND/OR FIRMWARE) AND/OR ACCESSORIES AND/OR THE SWARM SATELLITE SERVICES , TO THE FULLEST EXTENT THESE DAMAGES MAY BE DISCLAIMED BY LAW AND REGARDLESS OF WHETHER SWARM WAS ADVISED OF THE POSSIBILITIES OF SUCH DAMAGES.

Third Party Information

This User Manual might refer to third party sources of information, hardware or software, products or services and/or third party web sites ("third party information"). Swarm does not control, and is not responsible for, any third party information, including without limitation the content, accuracy, copyright compliance, compatibility, performance, trustworthiness, legality, decency, links, or any other aspect of third party information. The inclusion of such third party information does not imply endorsement by Swarm of the third party information. ANY THIRD PARTY INFORMATION THAT IS PROVIDED WITH SWARM'S FACILITIES, SERVICES, PRODUCTS OR USER INFORMATION IS PROVIDED "AS IS.'' SWARM MAKES NO REPRESENTATIONS, GUARANTEES OR WARRANTIES IN RELATION TO THIRD PARTY INFORMATION AND SWARM SHALL NOT BE LIABLE FOR ANY LOSSES, DAMAGES, LIABILITIES, JUDGMENTS, FINES, AMOUNTS PAID IN SETTLEMENT, EXPENSES OR COSTS OF DEFENSE SUSTAINED IN RELATION TO ANY SUCH THIRD PARTY INFORMATION.

Intellectual Property, Trade Secret, Proprietary or Copyrighted Information

To protect Swarm proprietary and confidential information and/or trade secrets, this User Manual may describe some aspects of Swarm technology in generalized terms. Swarm products may include copyrighted Swarm and third party software. Any such copyrighted software contained in Swarm products may not be modified, reverse engineered, distributed or reproduced in any manner to the extent provided by law. The purchase of any Swarm products shall not be deemed to grant either directly or by implication or otherwise, any license under copyrights, patents, or patent applications of Swarm or any third party software provider, except for the normal, nonexclusive, royalty free license to use that arises by operation of law in the sale of a product.

Content Copyright

You are exclusively responsible for the use of the Swarm Modem-M138, including proper use of third party copyrighted materials. If you violate these terms you agree to defend, indemnify and hold Swarm harmless with respect to any claims or actions by third parties related to your improper use of copyrighted materials and to pay all costs, damages, fines and other amounts incurred by Swarm, or on its behalf, in the defense of any such claims or actions.

Export Compliance

The Swarm Modem-M138 is controlled by the export laws and regulations of the United States of America. The U.S. Government may restrict the export or re-export of this product to certain individuals and/or destinations. For further information, contact the U.S. Department of Commerce, Bureau of Industry and Security or visit www.bis.doc.gov

Revision History

Revision	Date	Comment	Firmware Version
1.00	11/11/21	Swarm M138 Modem Product Manual - Initial Release	
1.01	12/01/21	Corrected pinout table Updated $GN and $GS descriptions Updated Modem status message structure Updated date/time format to reflect YYYY-MM-DDTHH:MM:SS	
1.02	12/07/21	Updated Table 6 to reflect Figure 4	
1.10	01/12/22	Updated transmit current measurements Removed section 3.3 Updated T/R pin notes Updated $GP command description Updated M138 Modem images Updated regulatory information	
1.11	01/20/22	Added antenna guidance to Section 6	
1.20	02/15/22	Updated power characteristics for all M138 operating modes Updated RSSI_bkgnd operating range Updated GPIO1 mode description GPS Cold/Warm start flowchart and timings 1pps pin removed $MT L=U command option removed Updated capacitance and inductance table	2.0.0

Additional Resources:

Please visit our developer tools webpage for quickstart guides and other helpful resources:
https://swarm.space/developertools

1 Safety Information and Compliance

The Swarm M138 Modem is designed to comply with the standards for Radio Emissions Compliance and Electromagnetic Compatibility in the United States, Canada, Australia, New Zealand, United Kingdom, European Union, Brazil, as well as worldwide.

1.1 FCC Compliance

1.1.1 FCC Interference Statement (Part 15.105 (b))

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 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.

This device complies with part 15 of the FCC Rules. Operation is subject to the condition that this device does not cause harmful interference.

1.1.2 FCC Part 15 Clause 15.21:

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

1.1.3 FCC Part 15.19(a):

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.

1.1.4 FCC ID:

The FCC ID for the Swarm M138 Modem is **2AVE9-M138**. All manufacturers integrating the Swarm M138 Modem into their products are required to provide a physical or e-label stating "Contains FCC ID: **2AVE9-M138**".

1.1.5 Part 15 Subpart B Disclaimer:

The final host product requires Part 15B compliance testing with the modular transmitter installed.

1.2 ISED Compliance

1.2.1 ISED RSS-Gen Notice CAN ICES-3 (B)NMB-3(B):

This device complies with Industry Canada's license-exempt RSSs. Operation is subject to the following two conditions:
(1) This device may not cause interference; and
(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;
2) l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

1.2.2 IC ID:

The IC ID for the Swarm M138 Modem is **25817-M138**. All manufacturers integrating the Swarm M138 Modem into their products are required to provide a physical or e-label stating "Contains IC ID: **25817-M138**".

1.3 RF Exposure Guidance

In order to comply with FCC / ISED RF Exposure requirements, this device and antenna must be installed to provide at least 36 cm separation from the human body at all times.

Afin de se conformer aux exigences d'exposition RF FCC / ISED, cet appareil doit être installé pour fournir au moins 36 cm de séparation du corps humain en tout temps.

1.4 EU RED Certification

1.4.1 Declaration of Conformity:

Currently under review

1.4.2 Article 10(2)/(10):

Currently under review

1.5 Transceiver Regulatory Certification

The Swarm Modem is a regulatory approved modular transmitter that is designed to be integrated into an enclosed host system. With appropriate external connections, the host can be designed to meet full regulatory tests and sold as a regulatory certified product that meets FCC, IC, and CE requirements. **Table 1** below is a partial list of regulatory approvals.

Regulatory Approvals	Radio Tests	EMC Tests	Safety Tests
FCC	FCC CFR47 Parts 2, 15, and 25		
IC	Industry Canada RSS170 Issue 2 - March 2011		
CE	ETSI EN 301 721 V2.1.1 (2016-05) ETSI EG203 367 V1.1.1	CISPR 16-23:2010/A1:2010 EN 55032:2012 EN 6100-4-2/EN55024:2010 EN 6100-4-3/EN55024:2010 EN 6100-4-8/EN55024:2010	EN 62368-1:2014/A11:2017

Table 1: Overview of the Swarm M138 Modem regulatory approvals.

2 Product Overview

The Swarm (Model: M138) satellite data modem transmits and receives data to and from Swarm's space network and is designed to be embedded into a third-party product. Swarm backend systems support the delivery of customer data via a REST API or Webhook to the cloud service of each user's choice.

The Swarm Modem is a module suitable for a variety of low-bandwidth use cases: from connecting people and tracking vehicles, ships, or packages to relaying sensor data for agriculture, energy, and industrial IoT applications.

The Swarm Modem is a Mini-PCI Express Card that can be easily integrated into any new or existing PCB design. The Swarm Modem communicates via a standard 3.3V CMOS serial UART interface or a PC interface with a USB-to-serial converter.

<table>
<tr><th>Category</th><th>Description</th></tr>
<tr><td>Components</td><td>GPS, VHF radio with integrated T/R switch, U.FL connector for GPS and VHF antenna, indicator LEDs, 3.3V serial interface, 3.3V GPIO</td></tr>
<tr><td>Onboard Sensors</td><td>uBLOX GPS (lat/lon/alt), CPU Temperature</td></tr>
<tr><td>Dimensions and Mass</td><td>51.0 mm x 30.0 mm x 5.3 mm, 9.6 g
See detailed description in the Mechanical Specification section.</td></tr>
<tr><td>Power</td><td>
<table>
<tr><th rowspan="2">Mode</th><th colspan="2">Typical</th><th colspan="2">Peak</th></tr>
<tr><th>3.3V</th><th>5V</th><th>3.3V</th><th>5V</th></tr>
<tr><td>Sleep</td><td>-</td><td>-</td><td>80µA</td><td>110µA</td></tr>
<tr><td>Receiver Active</td><td>26mA</td><td>25mA</td><td>40mA</td><td>45mA</td></tr>
<tr><td>Transmitter On</td><td>850mA</td><td>550mA</td><td>1000mA</td><td>600mA</td></tr>
</table>
</td></tr>
<tr><td>Protocol</td><td>Modified NMEA two-letter command set</td></tr>
<tr><td>Bit rate</td><td>1 kbps. Maximum packet size is 192 bytes</td></tr>
</table>

Table 2: Overview of the Swarm Modem.

3 Mechanical Specification

3.1 Modem Dimensions

The overall dimensions of the Modem and its weight are summarized below.

Parameter	Value
Length	51.0 mm
Width	30.0 mm
Height	5.3 ±0.1 mm
Weight	9.6 g

Table 3: Modem Mechanical Dimensions and Weight.

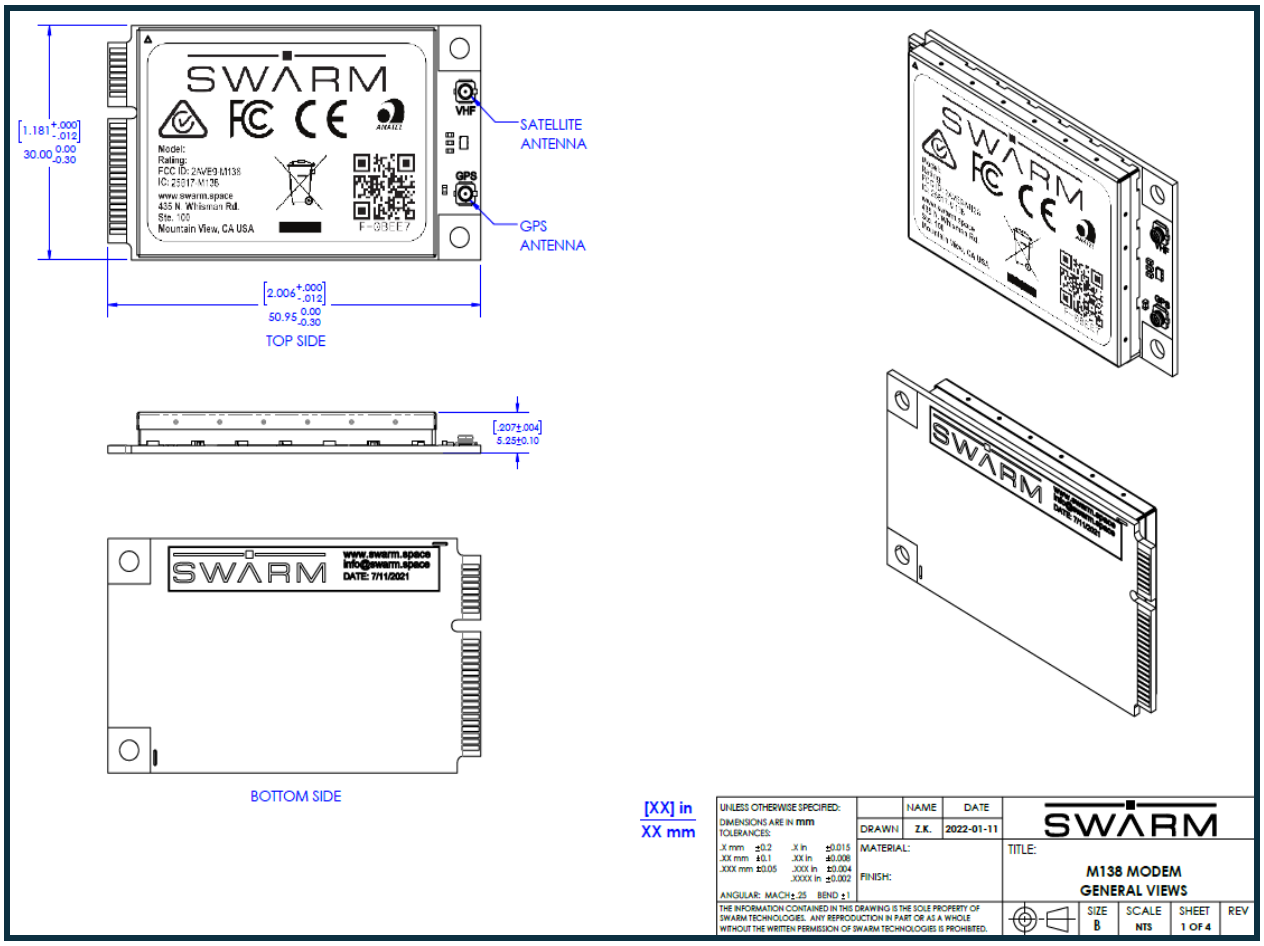


Figure 1: Modem front and back views.

3.2 Environmental

The environmental specifications of the Modem are summarized below. The Swarm Modem is not conformally coated, and as such the user needs to provide any weatherproofing for their application.

Parameter	Value
Operating Temperature Range	-40 °C to +85 °C
Storage Temperature Range	-40 °C to +85 °C
Operating Humidity Range	0% to 95%, non-condensing
Storage Humidity Range	0% to 95%, non-condensing

Table 4: Environmental Specifications.

4 Electrical Interfaces

The following subsections contain information for the electrical interfaces of the Modem.

4.1 User Host Device

The user host device provides the following connections to the Modem:

- DC power supply input: 3.0V to 5.0V, (1000 mA_{peak} at 3.3V)
    - EMI/RFI shielding CAN is ***required*** if using a switching DC power supply in order to contain radiated emissions
    - Additional filtering of the power supply is highly recommended
        - See Design Guidance for best practices
- Data interface
    - 3.3V Serial Data Interface
- Satellite signal (Use the provided U.FL connector labeled VHF)
- GPS signal (Use the provided U.FL connector labeled GPS)
- GPIO1 (optional)

4.2 Modem Pin Allocation

The pin numbering scheme of the Swarm Modem is shown in **Figure 4**. All pins are located on the card edge of the Swarm Modem and are designed to fit in a standard mPCIE card connector. The pin function assignment is given in **Tables 6** and **7**. Multiple supply grounds are provided and all power pins / supply grounds are required to be connected to the power supply in order to limit the current on any one pin. Multiple signal grounds are provided to reduce cross-talk. Many pins are intentionally left empty, and must be left unconnected.

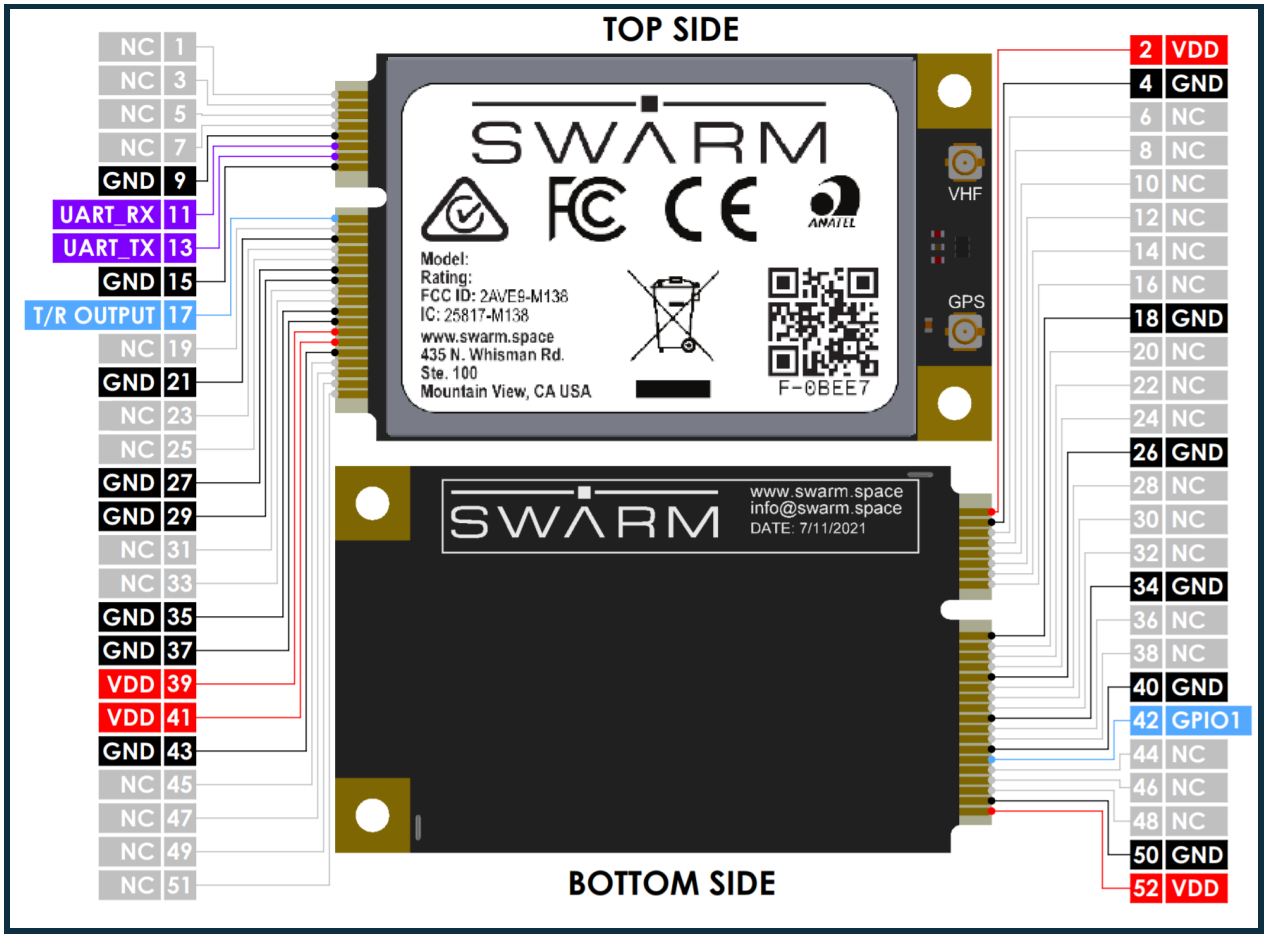


Figure 4: Modem pinout and pin number

Pin Number	Name	Type	Description
1,3,5-8,10,12, 14,16,19-20, 22-25,28,30-33, 36,38,44-49, 51	NC	NC	No Connection
2,39,41,52	VDD	Power	3.3V/5V at up to 1000mA/600mA
4,9,15,18,21,26, 27,29,34,35,37, 40,43,50	GND	Ground	Ground
11	UART_RX	Comms	3.3V serial receive
13	UART_TX	Comms	3.3V serial transmit
17	T/R OUTPUT	Digital Output	Transmit/Receive indicator ***leave unconnected if not used***
42	GPIO1	Digital Output/Analog Input	General purpose input/output

Table 6: Modem pin numbers and descriptions.

Additional Notes

Pin Number	Note
42	Connection is unbuffered and connected directly to a GPIO on the Modem processor. Configuration will be provided via serial commands. GPIO1 pin is 3.3V tolerant and open drain, with a sink current limit of 8 mA (20 mA with a relaxed VOL/VOH)
2, 39, 41, 52	The VDD connection points are in parallel with one another and power the Modem. If the designer wants to enable a complete power off mode, a load switch can be provided here
17	HIGH when transmitting LOW when receiving The state will not change faster than 500µs prior to the Modem beginning to transmit and 500µs after the Modem has finished transmitting. ***leave open if not used***

Table 7: Additional notes on pin numbers.

4.3 DC Power Interface

The DC power interface consists of the DC power inputs as summarized below. The power requirements apply to DC power measured at the Swarm Modem user connector input and not at the output of the power supply. It is required that users incorporate the required bypass capacitors for the supplied power input as can be seen in **Design Guidance**.

Name	Description	Min	Typ	Max	Unit
VCC	Module supply voltage	3.00	3.30	5.00	V
VCC Ripple	Module supply voltage ripple	-	-	75	mV_{pp}
VCC Limits	Module supply voltage **absolute** limits	3.00	-	5.00	V
	Current consumption - Sleep Mode	-	-	80	µA
	Current consumption - Receiver Active	23	26	40	mA
	Current consumption - GPS Acquisition Mode*	40	45	55	mA
ICC (3.3V)	Current consumption - Transmitter on	750	850	1000	mA
	Current consumption - Sleep Mode	-	-	100	µA
	Current consumption - Receiver Active	23	26	40	mA
	Current consumption - GPS Acquisition Mode*	40	45	55	mA
ICC (4.2V)	Current consumption - Transmitter on	640	700	800	mA
	Current consumption - Sleep Mode	-	-	110	µA
	Current consumption - Receiver Active	23	26	45	mA
	Current consumption - GPS Acquisition Mode*	40	45	55	mA
ICC (5.0V)	Current consumption - Transmitter on	500	550	600	mA

Table 8: Power supply characteristics for 3.3, 4.2, and 5.0V supplies to the Modem.

*Includes satellite receiver active current with GPS in acquisition mode. The Modem enters into GPS acquisition mode for approximately 30 seconds after exiting from sleep mode, on powerup, or when the Modem needs to re-acquire a GPS fix (approximately once every 4 hours) while the Modem is continuously powered on and not in sleep mode.

An example power profile for a Modem powered with 3.3V can be found below in **Figure 5.**

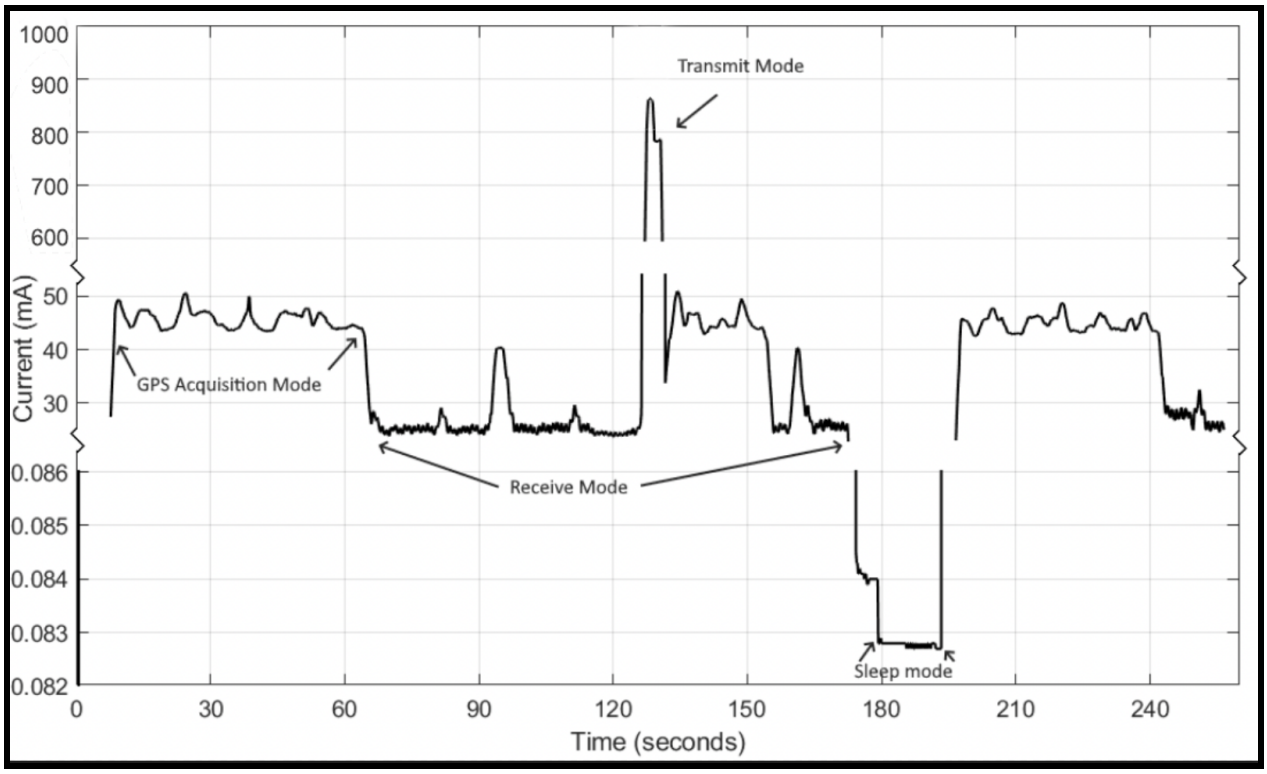


Figure 5: Example current use for a Modem with a 3.3 V input from wake-up, GPS acquisition, Transmit, Receive mode, and then sleep mode. Note the two breaks in the y-axis scale.

Transmissions from the Modem consist of short periods of high current draw conforming to the following characteristics:

Parameter	Value	Note
Transmission Length	3.7 s	Length of transmission for sending a 192 byte packet to space. Actual length may be shorter with a smaller packet size
Transmission Energy	3.4 mW-hrs (12.24 J)	Total amount of energy required for 1 transmission to space (192 bytes), when using 3.3V
Recovery Time	0.4s	When sending multiple messages, this is the minimum amount of time that will pass before the Modem will attempt another transmission

Table 9: Sample Modem transmission characteristics for sending a user payload of 192 bytes. Modem provided with 3.3V input.

Other electrical characteristics of the Modem can be found below:

Parameter	Value
Maximum power at antenna connector	1.122 W
Sum total of all capacitance	255 µF
Sum total of all inductance	16790 nH
Largest capacitor	47 µF
Largest inductor	15000 nH
Total Input Capacitance	57 µF

Table 10: Other electrical characteristics for the Modem

4.4 Power On/Off Control

The Modem can be powered down with the use of the **$PO*1F** command. Once powered down, it can be externally switched on/off by a user-supplied load switch on the VDD power rail.
After power to the Modem has been removed, power should not be restored until at least 1 second has elapsed.
If a Modem does not respond to software commands, remove power from the module, wait for at least 1 second, and then power it back on. If the 1 second minimum wait time is not adhered to, the reset circuit may not operate and the Modem could be placed in a non-operational state. The state is not permanent and can be rectified by the above procedure.

4.5 Serial Data Interface

The serial data interface is a CMOS serial UART 3-wire (serial Rx, serial Tx, and ground) interface at 3.3V digital signal levels over which the Modem transfers commands, responses, and message data. The serial communication parameters can be found below.

Parameter	Value
Baud Rate	115200
Data Bits	8 Bits
Parity	None
Stop bits	1 Bit
Flow Control	None

Table 11: Serial communication parameters.

In addition, the electrical characteristics for SERIAL_RX, SERIAL_TX, and GPIO1 can be found below.

Symbol	Parameter	Min	Typ.	Max	Unit
VIL	I/O input low level voltage	-0.3	-	0.3*3.3V	V
VIH	I/O input high level voltage	0.7*3.3V	-	3.3V	V
RPU/RPD	Weak pull up/down equivalent resistor (for GPIO1)	25	40	55	kΩ

Table 12: Electrical characteristics for SERIAL_RX, SERIAL_TX, and GPIO1.

All customer messages, data, and settings are stored in non-volatile memory. As such, after a power cycle, customer settings as well as any messages that have not yet been transmitted over the Swarm network will be retained on the Modem. The read/write lifetime of the Modem memory exceeds 20 years.

4.6 LED Indicators

The Modem contains three indicator LEDs, whose function is described below. All LEDs are off when the Modem is placed into sleep mode.

LED	Function
Green	During the bootup sequence the green LED will be on solid for 3 seconds. During normal operation after bootup and before shutdown, the green LED will blink 100ms every 5 seconds while the Modem is powered on. This is a “heartbeat” indication that the Modem is working as expected.
Red	After power is applied until the Modem begins booting, the red LED will be on solid for 10 seconds. After bootup and while the Modem is acquiring a GPS fix, the red LED will flash quickly until a valid GPS fix has been found. Then the red LED will shut off during normal operation. After waking up from sleep, and before a fresh GPS fix has been acquired, there will be a single red LED flash every 5 seconds (following the green LED flash) until a fresh GPS fix has been found. Then the red LED will again shut off during normal operation. During the shutdown sequence, the red LED will be on solid until 3.3V power is removed from the board.
Blue	The blue LED will be on solid when the Modem is actively receiving from a Swarm satellite

Table 13: LED functions.

5 Design Guidance

This section describes the ***recommended*** design practices for successful integration of the Modem.

5.1 Input Connections

The Swarm Modem utilizes standard 51 mm x 30 mm mPCIe form factors that require an industry standard connector listed below. All voltage pins need to be connected in such a way to minimize ground loops, often down by using a via to a power/ground plane. A method of retaining the Modem is also required, whether it be a retaining clip or screws.

Below are example connectors, retaining clips, and their part numbers. Many vendors and styles exist from vendors such as DigiKey/Mouser/Newark.

Example Connectors	Example Retaining Clips
JAE: MM60-52B1-B1-R850	**JAE:** MM60-EZH039-B5-R850
TE Connectivity: 2041119-2	**TE Connectivity:** 1717832-2

Altium footprints with correct connector/retaining clip spacing can be found in our Developer Tools .

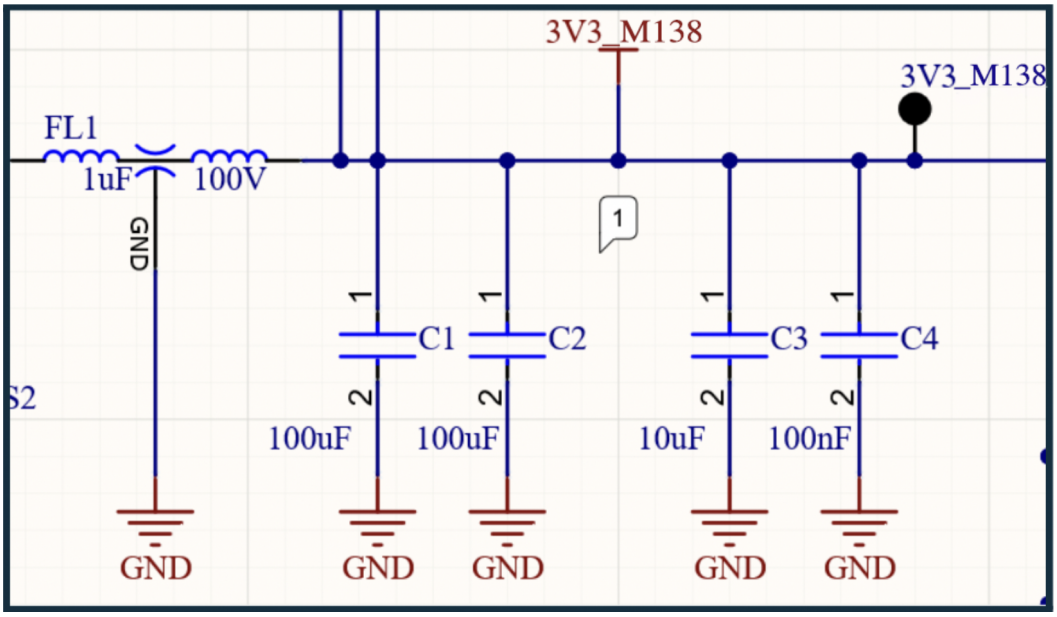
5.2 Decoupling and Feed-through Capacitors

We ***require*** a minimum amount of decoupling capacitance in order to reduce/eliminate any high frequencies from the input supply reaching the Modem input terminals.

Choose Ceramic capacitors rated for 16V or greater, X5R or better

Qty	Size
1	100 nF
1	10 µF
2	100 µF

In addition to the decoupling capacitors above, we also ***highly recommend*** a feed-through capacitor prior to decoupling to further clean up the supply voltage.



Example Feed-Through Capacitor
TDK: YFF31HC2A105MT000N

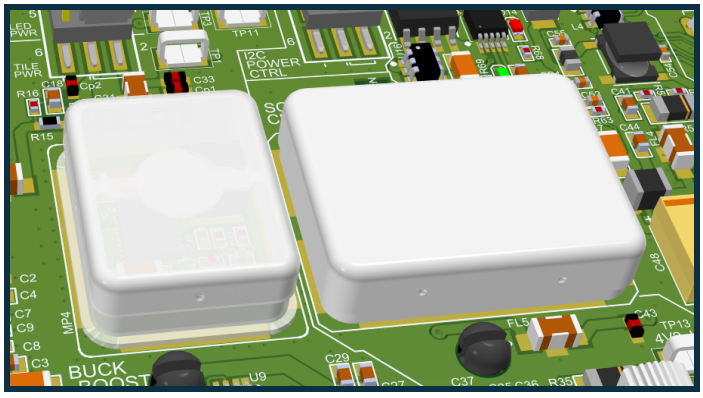
5.3 Input Voltage Examples

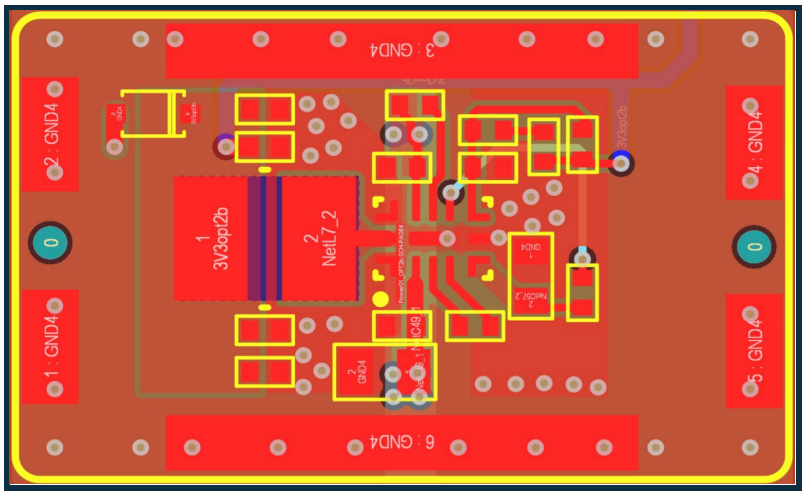
The Modem can be supplied via linear or switching regulator depending on the user's requirements. The Modem has an on-board boost regulator with an input voltage range of 3.0V to 5.0V. Most customers are operating remotely and use switching regulators for high efficiency and battery applications. We have provided example parts below. The customer can use these as a baseline for starting their own designs using other suppliers.

It's recommended that users pick a regulator capable of 2A for the modem alone; all other user peripherals should be in addition to the 2A regulator.

Example Regulator	Description
Analog Devices: LTC3113	3A Low Noise Buck-Boost DC/DC Converter
ST: LD39200	2A High PSRR Ultra low drop linear regulator with reverse current protection

Proper layout and design of switching regulators are ***critical*** to reduce conducted and radiated emission on-board. These are often coupled onto the power and antenna of the Modem increasing RSSI and impeding successful transmission. We ***highly recommend*** an RF CAN over any switching regulator to eliminate the risk of radiated emissions from the switcher. An example RF CAN and proper layout is pictured below.





5.4 Communications

A Serial Debug Header is included in the sample hardware design to upgrade the Modem firmware. It is ***critical*** that users include this, or similar, header in their design to provide serial access to the Modem, which can be used to upgrade to future firmware versions and for debugging.

The Serial Debug Header requires the following connections: SERIAL_RX, SERIAL_TX, and GND. It is ***highly recommended*** that the designer use a 6-pin connector to easily accommodate a COTS TTL (USB-A to UART) cable or Tag-Connect, for example:

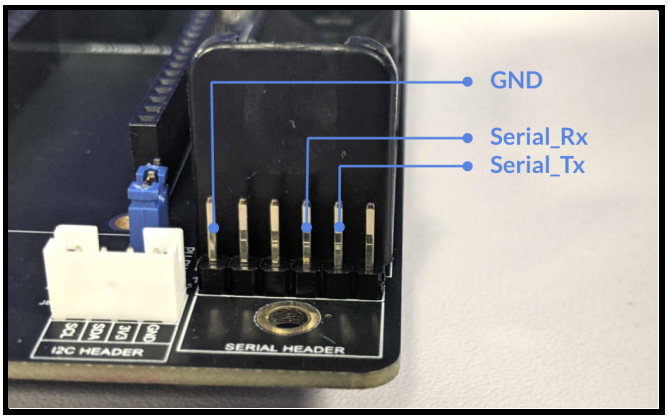
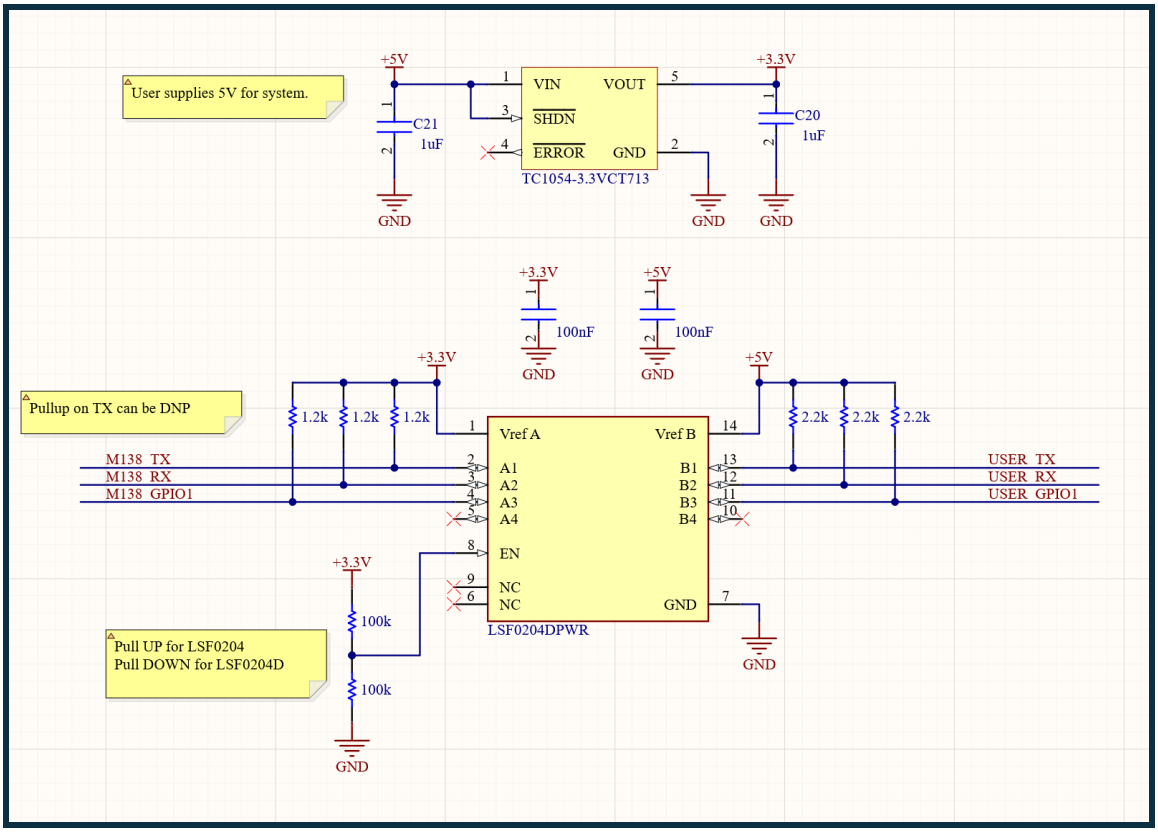


Figure 6: Example of a Serial Debug Header, shown on the Swarm Eval Board

5.4.1 Level Shifting

The Modem is 3.3V logic capable, so interfacing with higher or lower level logic systems will require level shifting. There are many methods to level shift a signal ranging from simple mosfets and resistors to dedicated IC's to handle the operation. Level shifting should be bi-directional.



5.5 Example Power Regulation Design

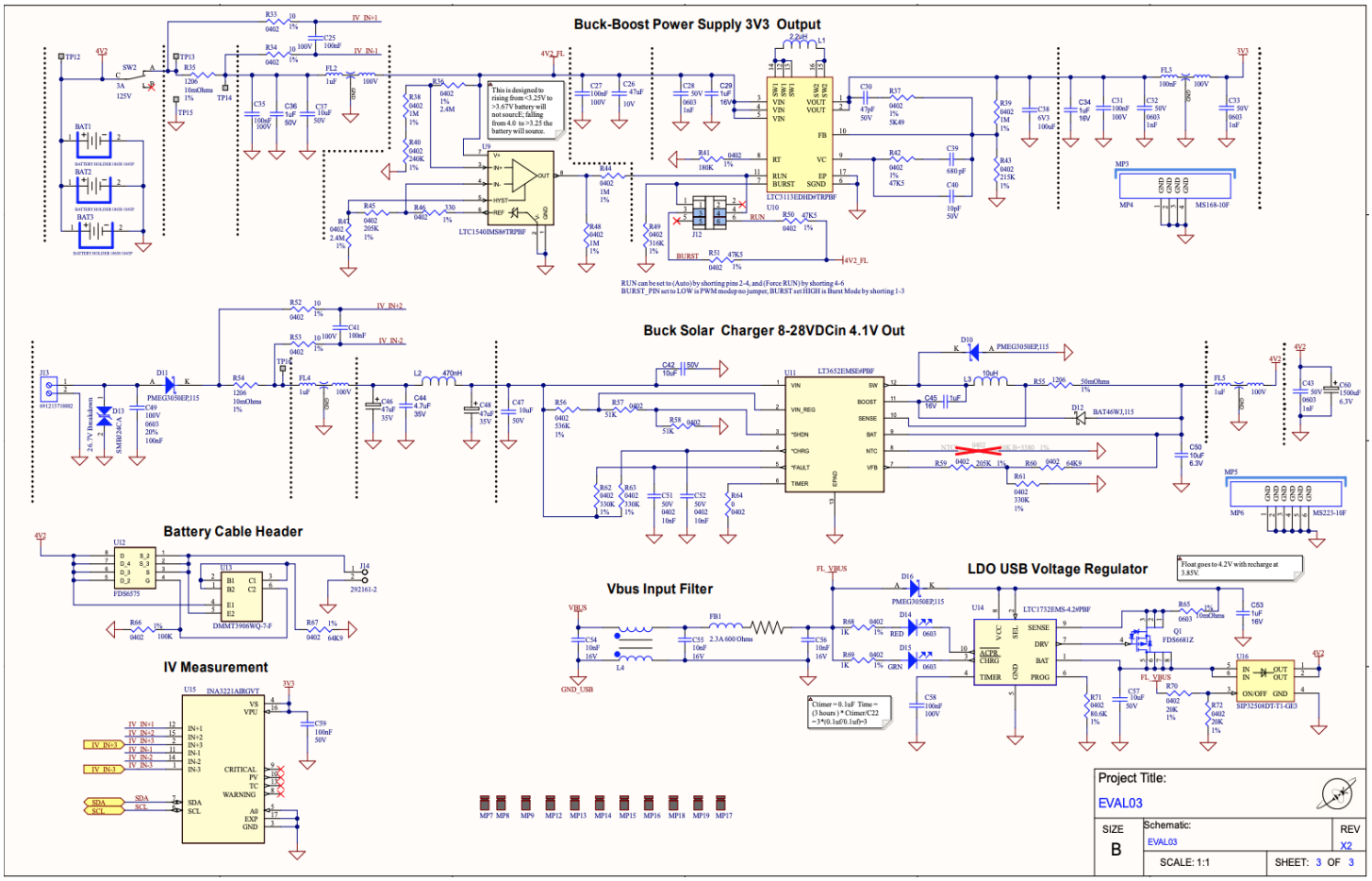


Figure 7: Schematic of a buck-boost-solar charger design for a Swarm Modem integration. Full –schematic for the Swarm Eval kit available at https://swarm.space/developertools/

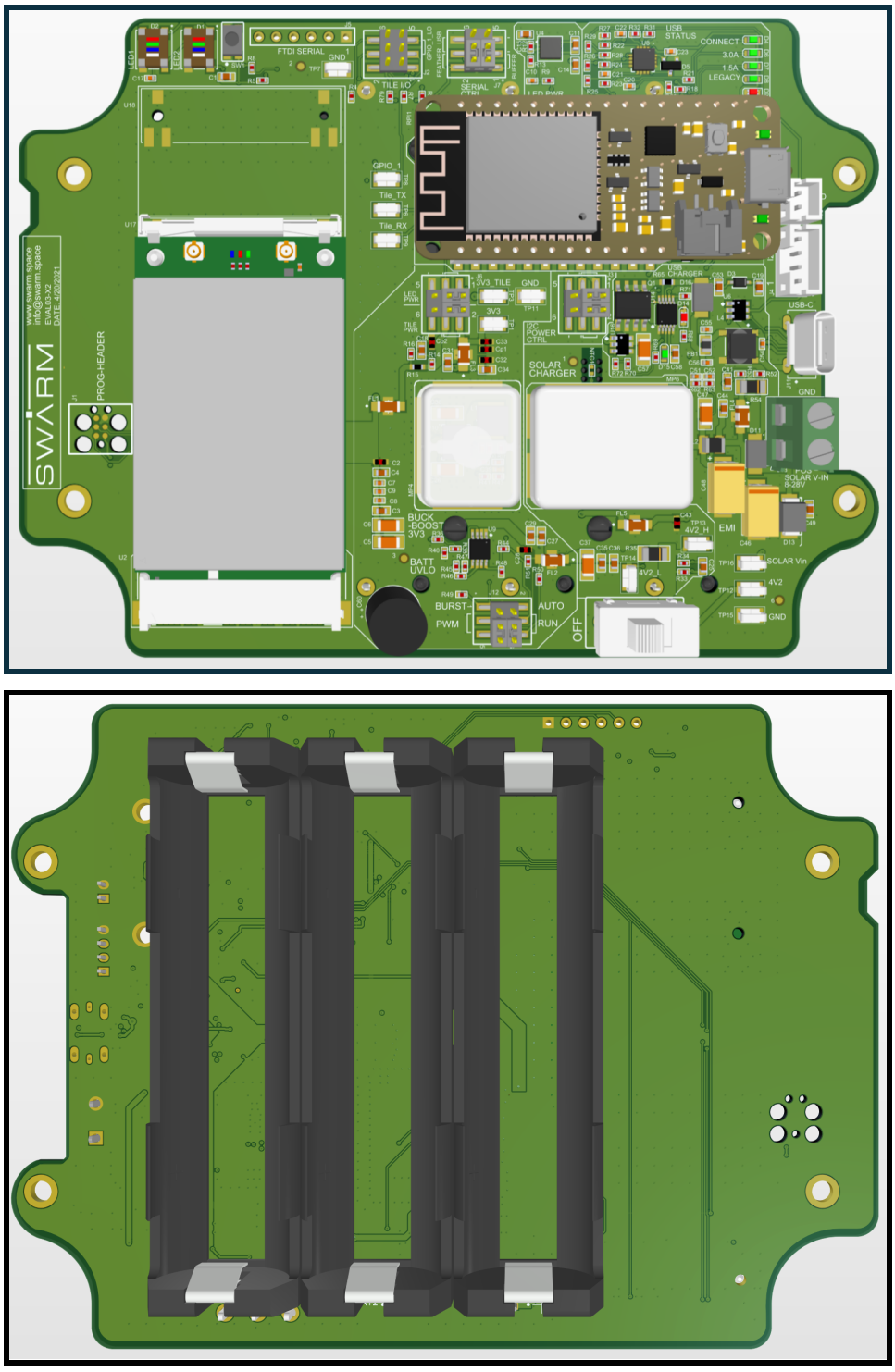


Figure 8: Sample hardware reference design integration for the Swarm Modem. This is a low noise buck-boost-solar charger example with the Swarm Modem (reference design files can be found at www.swarm.space/developertools/). A 6-pin serial programming header (upper-left of image) is highly encouraged so that the Modem firmware can be easily updated in the future. Qty 3 18650 lithium rechargeable batteries in parallel are used in this reference design. The circuit fits into BUD Industries IP68 waterproof case Model # PU-16533-C, with the use of 4 standoffs.

6 RF Interfaces

This section describes the physical characteristics of the RF connectors and specifications of the RF Interface.

6.1 RF Connectors

The Modem's satellite and GPS connectors are male U.FL connectors [TE Connectivity Part Number 1909763-1]. This is a surface mount connector that is directly attached to the Modem. A Swarm VHF antenna must be used to ensure that the antenna is tuned appropriately to make successful transmission to the Swarm satellites. Swarm antennas are tuned for VSWR (between Swarm Bands) with a max $\Delta < 0.5$.

6.2 RF Antenna

For illustrative purposes, a picture of the Swarm ¼-wave antenna is shown in **Figure 9.**

Figure 9: Swarm Coiled ¼ Wave Antenna.

6.2.1 Antenna Characteristics

The Modem is certified with the following antenna as described below. No power reduction compensation is required for use with this antenna.

Swarm Coiled ¼ Wave Antenna	
Parameter	**Value**
Length	22.0 cm
Diameter (Connector)	11.2 mm
Diameter (along major length)	7.6 mm
Weight	31.5 g
Operating Temperature	-55 °C to +130 °C
Operating Humidity	0-100% humidity, condensable
Impedance	50 Ohms nominal
Polarization	Linearly Polarized
VSWR (between Swarm Bands)	Max Δ < 0.5
Gain	2.0 dBi
Frequency	137.000-138.000 MHz (Rx) 148.000-150.000 MHz (Tx)
Connector	SMA male (U.FL to SMA female cable required)
Antenna Ground Plane	**Required**, see section 6.2.2
Antenna Height Requirement	1m above the ground/solid surfaces
Antenna Classification	Mobile, Fixed
Minimum separation distance from body	36 cm

Table 14: Antenna characteristics for Swarm Coiled ¼ Wave Antenna.

RF Antenna Guidance

The antenna gain shall be < 5 dB for an ideal 0 dBi radiator. If using an antenna other than the Swarm Coiled ¼ wave antenna, please refer to the notes below:

- An antenna that has a gain of > 5dB will not be compliant with current certifications.
- The end-user will have to get a Supplier's Declaration of Conformity (SDoC) and use Swarm's FCC/IC ID as included.
- If the antenna being used has > 2dBi of gain, then the antenna will have to be evaluated for SAR.

GPS Antenna Guidance

A GPS antenna is ***required for operation***. Any passive GPS antenna with an appropriate connection to the Modem's U.FL male connector or via the GPS_OUT pin is acceptable for use. One such example of a passive antenna is: [Molex Part Number 1461860300].

Active GPS antennas are supported at **3.3V** and up to **35mA**. Any active antennas that require a different voltage or maximum current must be powered by the user's host device.

Example u.fl to SMA cable	Example u.fl to u.fl cable
Taoglas: CAB.721	**Wurth**: 636201050100

6.2.2 Ground Plane Requirements

An antenna counterpoise or antenna ground plane is required for the Swarm Coiled ¼ Wave Antenna. Ground planes are electrically conductive surfaces that are connected to the ground conductor of the antenna that serve as a reflecting surface for radio waves.

To date, Swarm has reliably communicated with its satellites using a coiled ¼-wave antenna with a 50 cm long, 20 AWG (0.812mm diameter) wire descending from the ground of the SMA connector of the antenna mount. Swarm has also reliably communicated with its satellites using a solar panel off to one side as the ground plane of sufficient size: 30cm x 30cm x 0.3cm

Integrated solutions are recommended to be designed with the use of a symmetrical ground plane or similar RF grounding solution for the best results with a Swarm-provided antenna.

Figure 10: Example ¼-wave antenna with a (Swarm) 9W solar panel serving as the required 30 x 30 cm ground plane off to one side. Tested extensively in the field and works well.

6.3 Antenna Debugging

To quickly verify the antenna connection to Modem:

1. Power on the Modem.
2. After bootup, issue **$RT 1*17**. The noise floor the Modem hears (in dBm) will begin displaying at a rate of once per second.
3. While the Modem's LED is blinking red rapidly (indicates it has not yet acquired a GPS fix)
    a. Touch the Modem's antenna with your hand or with another object. The noise floor measurement should noticeably change.
    b. The Modem will not transmit until it has a GPS fix **and** it hears a Swarm Satellite. There is no risk of the Modem transmitting during this procedure if the red LED is blinking rapidly.

7 Software interface

Message types

There are two types of messages:

- **Unsolicited messages** - which include status messages, date/time and GPS information, and notifications that user data has been received. By default most of these are set to not appear, and user settings of these messages persist through restarts and power cycles.
- **Command responses** - which include responses to input commands, as well as notifications that user data has been sent or settings have been updated

General command structure

All messages sent and received are NMEA formatted sentences. NMEA sentences are terminated with a single newline character. Each sentence begins with a **$** and ends with ***xx** where **xx** is a two digit hexadecimal checksum of the characters between the **$** and ***xx**. The checksum conforms to the NMEA standard and does not include either the **$** or the ***xx**. Messages with a bad checksum will fail to be parsed and produce the following response.

```
$CMD? ERR,CMD_NMEACHECKSUMFAIL*21
```

A sample implementation of the NMEA checksum can be found below (written in C)

A **$** will never occur within a command, and may be used to reset the receiving state machine.

An ***** may occur within a command. The receiving state machine will verify the last three characters in the command are ***xx** after the **\n** is received and before the checksum is calculated. Each **x** may be any legal ASCII character in the range **0..9**, **A..F**, or **a..f**.

An example command is provided below to illustrate the command structure and a valid checksum. This command returns the most recent date/time message :

```
$DT @*70
```

```
$DT 20190408195123,V*41
```

Command timing

Once the **$** is received, all subsequent characters must occur within 250 milliseconds of the previous character. If the inter-character delay exceeds 250 ms, the command will be silently discarded, and the receiving state machine will consume and ignore any characters received until the next **$**.

Command responses

An **OK** response confirms that the input parameters have been updated in response to a command. An **ERR** response indicates that the command could not be parsed successfully, and an error type details the problem. Note that additional or invalid characters included between the two-character command designator and the ***** of the command can cause an error response. **ERR** responses have the following syntax:

$<cmd> ERR,<error_type>*xx

Value	Description
cmd	The two-character command designation
error_type	Type of error encountered during command processing

The follow error type messages will be provided:

Value	Description
CMD_BADPARAM	Unrecognizable parameter after command
CMD_BADPARAMLENGTH	A parameter has an incorrect length
CMD_BADPARAMVALUE	A parameter has a value that is out of range
CMD_INVALIDCHAR	A parameter has an invalid character
CMD_NOTIMPLEMENTED	The command is not recognized as valid
CMD_PARAMMISSING	A required parameter is missing
CMD_PARAMDUPLICATE	A parameter has been duplicated

Boot-up sequence

The user application should ignore any characters received during startup until the following NMEA sentence has been received:

`$M138 BOOT,RUNNING*2a`

The bootloader may output non-NMEA formatted messages during this time. These messages include, but are not limited to: status messages, firmware update progress messages, and error messages. These messages should be ignored and are for Swarm debugging purposes only.

GPS Date/Time Reference

The Modem will enter its GPS acquisition state once the boot-up sequence is complete. Please refer to **Figure 11** for estimates of how long the Modem will take to acquire a valid date/time reference. The user application should not send any transmit data ($TD) commands until the following NMEA sentence has been received:

`$M138 DATETIME*56`

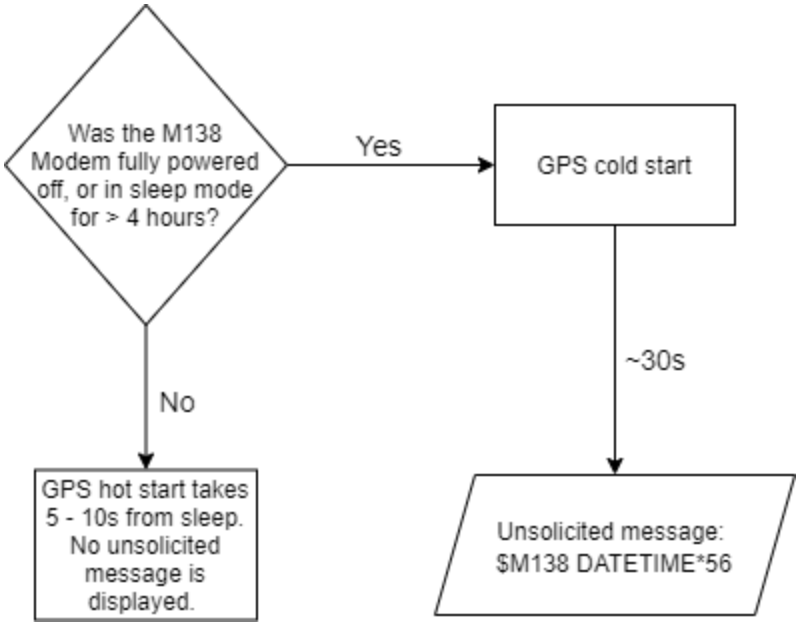


Figure 11: Time Estimates for GPS Date/Time Reference Update

Note that ALL time representations now use the T date and time delimiter as indicated in ISO 8901 *EXCEPT FOR THE $DT COMMAND.* Ex. 2022-01-26T08:30:00.

Messages

All messages sent to or from the Hive have an **Application ID** tag associated with them. The **Application ID** can be an arbitrary number from 0 to 64999, and may be used by the user to organize messages (ex: **Application ID** 1000 could be used for device telemetry, 2000 for commands to the device and 3000 for emergencies). Swarm reserves **Application ID** values 65000 to 65535 for internal use. Specifying an **Application ID** in the reserved range will result in unexpected operation and the messages may be lost.

See the $TD - Transmit Data command section in the command directory below for an example implementation of the Application ID.

Implementation of NMEA checksum in C

```
uint8_t nmeaChecksum (const char *sz, size_t len)
{
   size_t i = 0;
   uint8_t cs;

   if (sz [0] == '$')
       i++;

   for (cs = 0; (i < len) && sz [i]; i++)
       cs ^= ((uint8_t) sz [i]);

   return cs;
}
```

Table of Commands and Messages

Name	Description	Unsolicited Messages
$CS	Configuration Settings	
$DT	Date/Time Status	Yes
$FV	Firmware Version Read	
$GJ	GPS Jamming/Spoofing Indication	Yes
$GN	Geospatial Information	Yes
$GP	GPIO1 Control/Status	
$GS	GPS Fix Quality	Yes
$MM	Messages Received Management (2-way operation)	
$MT	Messages to Transmit Management	
$PO	Power Off	
$PW	Power Status	Yes
$RD	Receive Data Unsolicited Message	Yes
$RS	Restart Device	
$RT	Receive Test	Yes
$SL	Sleep Mode	
$M138	Modem Status Unsolicited Message	Yes
$TD	Transmit Data	Yes

Table 15: Table of all Modem commands

$CS - Configuration Settings

Retrieve and display the configuration settings for the Swarm device ID. These settings are determined by Swarm for identifying and communicating with each individual device. Since there are no variable parameters, the correct checksum has been provided below.

`$CS*10`

Returns:

`$CS DI=<dev_ID>,DN=<dev_name>*xx`

Value	Description
`dev_ID`	Device ID that identifies this device on the Swarm network
`dev_name`	Device type name

Notes:

See the section *Command responses* for a description of <error_type>.

Example:

```
$CS*10
```

```
$CS DI=0x00e57,DN=M138*43
```

The Device ID is **0x00e57**, and the device is a **M138 Modem**

$DT - Date/Time Status

Set or query the rate for **$DT** unsolicited report messages for date and time. Also can retrieve the most current **$DT** message. See unsolicited message description for **$DT** message format.

`$DT <@|?|<rate>>*xx`

Parameter	Description
`@`	Repeat most recent **$DT** message
`?`	Query current **$DT** rate
`rate`	Disable or set rate of **$DT** messages

Returns one of:

Value	Description
`$DT <YYYY><MM><DD><hh><mm><ss>,<flag>*xx`	The most recent **$DT** message.
`$DT <rate>*xx`	The current **$DT** rate
`$DT OK*xx`	**`rate`** updated successfully
`$DT ERR,<error_type>*xx`	Command input error

Parameter	Description
YYYY	Year (1970..2099)
MM	Month (01..12)
DD	Day (01..31)
hh	Hour (00..23)
mm	Minutes (00..59)
ss	Seconds (00..59)
flag	**I** Date/time is invalid **V** Date/time is valid

Notes:

See the section *Command responses* for a description of **<error_type>**.

<rate> is a value between 1 and 2147483647 (2^{31}-1) or 0. It will be the number of seconds in between each message. If **<rate>** is 0, no messages will be sent.

If **<rate>** is valid, no **$DT** messages will be sent by the device until the GPS has obtained a valid time reference to set its internal date and time, as indicated by the **$M138 DATETIME*56** message. The valid flag will show **V** if the modem has acquired a valid time reference at least once since powering on.

Examples:

Calling the most recent date/time message:

```
$DT @*70
```

```
$DT 20190408195123,V*41
```

Returns a date/time of **`April 8th, 2019 7:51:23 PM GMT`**. The date/time is **`valid`**. **`<rate>`** does not have to be valid to call the most recent date/time message.

Setting the rate of date/time messages:

```
$DT 300*03
```

```
$DT OK*34
```

Sets the rate of date/time messages to one message every **`300`** seconds.

Querying the rate of date/time messages:

```
$DT ?*0f
```

```
$DT 60*36
```

Returns a rate of one message every **`60`** seconds.

$FV - Firmware Version Read

Returns the current device firmware version. Since there are no parameters, the correct checksum has been added.

`$FV*10`

Returns one of:

Value	Description
`$FV <version_string>*xx`	The current firmware version
`$FV ERR,<error_type>*xx`	Command input error

Notes:

See the section *Command responses* for a description of **`<error_type>`**.

Example:

```
$FV*10
```

```
$FV 2021-07-16T00:10:21,v1.1.0*74
```

The firmware version on the device is **1.1.0**

$GJ - GPS Jamming/Spoofing Indication

Set or query the rate for **$GJ** unsolicited report messages for jamming and spoofing indicators. Also can retrieve the most current **$GJ** message.

`$GJ <@|?|<rate>>*xx`

Parameter	Description
`@`	Repeat most recent **$GJ** message
`?`	Query current **$GJ** rate
`rate`	Disable or set rate of **$GJ** messages

Returns one of:

Parameter	Description
`$GJ <spoof_state>,<jamming_level>*xx`	The most recent **$GJ** message.
`$GJ <rate>*xx`	The current **$GJ** rate
`$GJ OK*xx`	Parameters updated successfully
`$GJ ERR,<error_type>*xx`	Command input error

Parameter	Description
`spoof_state`	**0** Spoofing unknown or detection deactivated **1** No spoofing indicated **2** Spoofing indicated **3** Multiple spoofing indications
`jamming_level`	Relative value ranging from 0 to 255 indicating how much carrier wave (CW) jamming is detected. 0 = no CW jamming, 255 = strong CW jamming

Notes:

See the section *Command responses* for a description of **<error_type>**.

<rate> is a value between 1 and 2147483647 (2^{31}-1). It will be the number of seconds in between each message. If **<rate>** is 0, no messages will be sent.

Examples:

Calling the most recent GPS jamming/spoofing message:

```
$GJ @*6d
```

```
$GJ 1,23*31
```

Returns a spoof state of **No spoofing indicated**. The carrier wave jamming level is **23.**

Setting the rate of GPS jamming/spoofing messages:

```
$GJ 3600*28
```

```
$GJ OK*29
```

Sets the rate of GPS jamming/spoofing messages to one message every **3600** seconds.

Querying the rate ofGPS jamming/spoofing messages:

```
$GJ ?*12
```

```
$GJ 10*2c
```

Returns a rate of one message every **10** seconds.

$GN - Geospatial Information

Set or query the rate for **$GN** unsolicited report messages for geospatial information. Also can retrieve the most current **$GN** message.

`$GN <@|?|<rate>>*xx`

Parameter	Description
`@`	Repeat most recent **$GN** message
`?`	Query current **$GN** rate
`rate`	Disable or set rate of **$GN** messages

Returns one of:

Value	Description
`$GN <latitude>,<longitude>,<altitude>, <course>,<speed>*xx`	The most recent **$GN** message
`$GN <rate>*xx`	The current **$GN** rate
`$GN OK*xx`	Parameters updated successfully
`$GN ERR*xx`	Command input error

Parameter	Description
latitude	Latitude in d.dddd format (float). The latitude is presented in the N basis (negative latitudes are in the southern hemisphere)
longitude	Longitude in d.dddd format (float). The longitude is presented in the E basis (negative longitudes are in the western hemisphere)
altitude	Altitude in meters (integer)
course	Course in degrees (0..359) (integer). Course proceeds clockwise, with 0=north, 90=east, 180=south, and 270=west
speed	Speed in kilometers per hour (0..999) (integer)

Notes:

See the section *Command responses* for a description of **<error_type>**.

<rate> is a value between 1 and 2147483647 (2^{31}-1). It will be the number of seconds in between each message. If **<rate>** is 0, no messages will be sent.

If **<rate>** is valid, no **$GN** messages will be sent by the device until the GPS has obtained a valid position reference as indicated by the **$M138 POSITION*4e** message .

Examples:

Calling the most recent GPS message:

```
$GN @*69
```

```
$GN 37.8921,-122.0155,77,89,2*01
```

Returns a location of **37.8921N, 122.0155W**. The device's altitude is **77m**, its course is **89 degrees**, and it is moving at **2 kilometers per hour**.

Setting the rate of GPS messages:

```
$GN 30*2a
```

```
$GN OK*2d
```

Sets the rate of GPS messages to one message every **30** seconds.

Querying the rate of GPS messages:

```
$GN ?*16
```

```
$GN 15*2d
```

Returns a rate of one message every **15** seconds.

$GP - GPIO1 Control/Status

This command allows control of the GPIO1 pin to allow indications or control the operation of the Modem.

$GP <@|?|<mode>>*xx

Parameter	Description
@	Read pin state (digital input and ADC modes ONLY)
?	Display current GPIO1 mode
mode	Set GPIO1 pin mode

Returns one of:

Value	Description
$GP <volts>V*xx	GPIO1 ADC reading in volts (mode 1)
$GP <H|L>*xx	GPIO1 digital input status (mode 2)
$GP <mode>*xx	The current **$GP** mode
$GP OK*xx	Parameters updated successfully
$GP ERR,<error_type>*xx	Command input error

Notes:

See the section *Command responses* for a description of **<error_type>**.

The **?** option allows reading back the current setting. The **mode** parameter allows specifying how the GPIO1 pin will operate. The available modes are:

Mode	Description	Note
0	Analog, pin is internally disconnected and not used (default)	
1	Analog ADC, pin can be read to measure input voltage (0-3.3V)	
2	Input, pin can be read as a general purpose digital input (High or Low)	
3	Input, low-to-high transition exits sleep mode	
4	Input, high-to-low transition exits sleep mode	
5	Output, set low as a general purpose digital output	
6	Output, set high as a general purpose digital output	
7	Output, low indicates unread messages pending for user	1
8	Output, high indicates unread messages pending for user	1
9	Output, low indicates unsent messages pending for transmit	2
10	Output, high indicates unsent messages pending for transmit	2
11	Output, low indicates unread or unsent messages	3
12	Output, high indicates unread or unsent messages	3
13	Output, low indicates sleep mode is active. Otherwise output is high	4
14	Output, high indicates sleep mode is active. Otherwise output is low	4

It is the responsibility of the user to provide pull-up resistors to the 3.3V supply rail. See the reference design for resistor sizing. All output modes are open drain.

(1) - If either of these modes are selected, the pin will indicate if the Modem has received one or more unread messages and is holding them for the user. If multiple messages are pending for the user, the pin will maintain the state until all messages have been read.

(2) - If either of these modes are selected, the pin will indicate if the Modem is holding one or more unsent messages until they can be transmitted. If multiple messages are pending for the user, the pin will maintain the state until all messages have been sent.

(3) - If either of these modes are selected, the pin will indicate if the Modem is holding at least one unread incoming message or at least one unsent outgoing message.

(4) - If either of these modes are selected, the pin will be set to the selected state after the user has issued the **$SL** command. The pin will return to the awake state only if the sleep mode is terminated by the wakeup time parameter being reached, or alternatively if activity is detected on the serial RX line.

Examples:

Querying the GPIO1 pin mode:

```
$GP ?*08
```

```
$GP 7*00
```

The GPIO1 pin mode is set to **Output, low while in sleep mode. Otherwise output is high**.

Reading the GPIO1 pin in ADC mode (must be in mode 1):

```
$GP @*77
```

```
$GP 3.282V*44
```

The GPIO1 pin mode has an ADC reading of 3.282V.

Reading the GPIO1 pin in digital input mode:

```
$GP @*77
```

```
$GP L*7b
```

The GPIO1 pin mode is a logic low.

Setting the GPIO1 pin mode to input and wake on a high-to-low transition:

```
$GP 2*05
```

```
$GP OK*33
```

Sets the GPIO1 pin mode to **input and wake on a high-to-low transition**.

$GS - GPS Fix Quality

Set or query the rate for **$GS** unsolicited report messages for GPS fix quality. Also can retrieve the most current **$GS** message..

`$GS <@|?|<rate>>*xx`

Parameter	Description
`@`	Repeat most recent **$GS** message
`?`	Query current **$GS** rate
`rate`	Disable or set rate of **$GS** messages

Returns one of:

Parameter	Description
`$GS <hdop>,<vdop>,<gnss_sats>,<unused>,<fix>*xx`	The most recent **$GS** message
`$GS <rate>*xx`	The current **$GS** rate
`$GS OK*xx`	Parameters updated successfully
`$GS ERR,<error_type>*xx`	Command input error

Parameter	Description
hdop	Horizontal dilution of precision (0..9999) (integer = actual hdop * 100)
vdop	Vertical dilution of precision (0..9999) (integer = actual vdop * 100)
gnss_sats	Number of GNSS satellites used in solution (integer)
unused	Always reads as 0, unused
fix_type	**NF** No fix **DR** Dead reckoning only solution **G2** Standalone 2D solution **G3** Standalone 3D solution **D2** Differential 2D solution **D3** Differential 3D solution **RK** Combined GNSS + dead reckoning solution **TT** Time only solution

Notes:

See the section *Command responses* for a description of **<error_type>**.

<rate> is a value between 1 and 2147483647 (2^{31}-1). It will be the number of seconds in between each message. If **<rate>** is 0, no messages will be sent.

If **<rate>** is valid, no **$GS** messages will be sent by the device until the GPS has obtained a valid position reference as indicated by the **$M138 POSITION*4e** message .

Examples:

Setting the rate for geospatial information messages to 1 per second:

```
$GS 1*05
```

```
$GS OK*30
```

If **<rate>** is valid, no **$GN** messages will be sent by the device until the GPS has obtained a valid position reference as indicated by the **$M138 POSITION*4e** message .

Calling the most recent geospatial information message:

```
$GS @*74
```

```
$GS 109,214,9,0,G3*46
```

Returns an HDOP of **1.09,** VDOP of **2.14,** the device is using **9** GNSS satellites for this solution**,** and it is a **Standalone 3D solution**.

Querying the rate of geospatial information messages:

```
$GS ?*0b
```

```
$GS 120*07
```

Returns a rate of one message every **120** seconds.

$MM - Messages Received Management (2-way operation)

Manage received messages in the device incoming message queue. Has ability to return the count, mark, and delete messages.

$MM <C=<U|*>|D=<msgID|R|*>|L=<msgID>|M=<msgID|*>|N=<D|E|?>|R=<msgID|O|N>>*xx

Parameter	Description
C=<U|*>	Return count of unread (**U**) or all (*****) messages
D=<msgID|R|*>	Delete message ID (**msgID**), all read (**R**), or all messages (*****)
L=<msgID>	List message ID (**msgID**), does not change message state
M=<msgID|*>	Mark message ID (**msgID**) or all messages (*****) as read
N=<D|E|?>	Message notifications disabled (**D**), enabled (**E**), or query (**?**)
R=<msgID|O|N>	Read message ID (**msgID**), oldest (**O**), or newest (**N**) message

Returns one of:

Parameter	Description
$MM <msg_count>*xx	Number of messages read/all/deleted
$MM <appID>,<data>,<msg_id>,<es>*xx	Response to reading a message
$MM MARKED,<msg_id>*xx	<**msg_id**> marked as read successfully
$MM DELETED,<msg_id>*xx	<**msg_id**> deleted successfully
$MM OK*xx	Message command succeeded
$MM ERR,<error_type>*xx	Command input error
$MM ERR,DBX_INVMSGID*xx	Invalid message ID in **D, L** or **R** command
$MM ERR,DBX_NOMORE*xx	No messages found when using **R=<O|N>**

Notes:

See the section *Command responses* for a description of **<error_type>**.

Messages have three states: Unread, read, and deleted. Once an unread message is read, it's state changes to read. It can subsequently be read again when called by its **msgID**. If a message is deleted, it can no longer be counted, read, marked, or deleted.

If a message is marked read using the **M=<msg_id>** or **M=*** command, marking it as read again is not an error.

<msg_count> is a number indicating the number of messages that are unread in response to the **C=U** command, total number of read and unread messages in response to the **C=*** command, or the number of messages deleted in response to the **D=*** command.

<appID> is the application ID tag of the message.

<**data>** is in the same format as the original **$RD** response returns.

<**msg_id>** is assigned by the device, and is an unsigned 64-bit value comprised of the device ID, a day of year counter, and a message of day counter. Responses that have a 0 as the message ID indicate the message has not been placed in the queue and therefore has no ID. The value should be treated as a simple arbitrary number.

<es> is the epoch seconds time when the message was received by the Modem.

Examples:

Counting all unread messages:

```
$MM C=U*0b
```

```
$MM 3*13
```

Three messages are unread.

Deleting all read messages:

```
$MM D=R*0b
```

```
$MM 4*14
```

Four read messages were deleted.

Marking a specific message as read:

```
$MM M=21990235111426*56
```

```
$MM MARKED,21990235111426*1e
```

Message with msg_id **21990235111426** has been marked as read.

Reading the newest received message:

```
$MM R=N*01
```

```
$MM
AI=10650,6578616d706c65206d657373616765,21990235111426,1584494275*52
```

The message (msg_id = **21990235111426)** is returned. The message was sent with the application ID **10650**. The hexadecimal data returned is **6578616d706c65206d657373616765** (in ascii **= "example message").** The epoch seconds at which the Modem received the message is **1584494275** (Date/Time = **Wednesday, March 18, 2020 1:17:55 PM)**. This message is now marked as read.

Reading the newest received message again:

```
$MM R=N*01
```

```
$MM ERR,DBX_NOMORE*03
```

There are no existing unread messages.

Enable message notifications:

```
$MM N=E*16
```

```
$MM OK*24
```

Message notifications are now enabled.

Query message notifications:

```
$MM N=?*6c
```

```
$MM N=E*16
```

Message notifications are enabled.

$MT - Messages to Transmit Management

Manage messages to be transmitted in the device outgoing message queue. Has ability to return the count, list, and delete unsent messages.

$MT <C=<U>|D=<msgID|U>|<L=msgID>>*xx

Parameter	Description
C=<U>	Return count of unsent (**U**) messages
D=<msgID|U>	Delete message ID (**msgID**) or all unsent (**U**) messages
L=<msgID>	List message ID (**msgID**) messages

Returns one of:

Parameter	Description
$MT <msg_count>	Number of unsent messages
$MT <data>,<msg_id>,<es>*xx	Response to listing a message
$MT DELETED*xx	<**msg_id**> deleted successfully
$MT OK*xx	Message command succeeded
$MT ERR,<error_type>*xx	Command input error
$MT ERR,DBX_INVMSGID*xx	Invalid message ID in **D** or **L** command
$MT ERR,DBX_NOMORE*xx	No messages found

Notes:

See the section *Command responses* for a description of **<error_type>**.

<msg_count> is a number indicating the number of messages that are unsent in response to the **C=U** command or the number of messages deleted in response to the **D=U** command

<data> is in the same format as the original **$TD** command accepts for message content

<msg_id> is assigned by the device, and is an unsigned 64-bit value comprised of the device ID, a day of year counter, and a message of day counter. Responses that have a 0 as the message ID indicate that the message has not been placed in the queue and therefore has no ID. The value should be treated as a simple arbitrary number.

Examples:

Counting all unsent messages:

```
$MT C=U*12
```

```
$MT 12*3a
```

Twelve messages are unsent.

Listing a specific unsent message:

```
$MT L=4428826476689*7c
```

```
$MT AI=0,068692066726f6d20737761726d,4428826476689,1605639598*55
```

The unsent message (msg_id = **4428826476689)** is returned. The hexadecimal data returned is **68692066726f6d20737761726d** (in ascii **= "hi from swarm").** The epoch seconds at which the Modem received the message is **1605639598** (Date/Time = **Tuesday, November 17, 2020 6:59:58 PM)**.

Deleting all unsent messages:

```
$MT D=U*15
```

```
$MT 1*08
```

One message was deleted.

$PO - Power Off

Power off the device. If fully supported, after issuing **$PO*1f** the customer should command any Modem power supplies to disconnect. If power is not disconnected, the Modem enters a low power mode until power is completely removed and restored. Since there are no variable parameters, the correct checksum has been added.

`$PO*1f`

Returns one of:

Value	Description
`$PO OK*xx`	Command has been accepted and the Modem will immediately attempt to power off
`$PO ERR,<error_type>*xx`	Command input error

Notes:

The **`$PO*1f`** command should be given anytime power is removed from the modem.

See the section *Command responses* for a description of **`<error_type>`**.

An **`OK`** response confirms that the Modem will shut down. The user should disconnect power from the Modem at this point. If left connected, the Modem will draw approximately 3mA on its 3.3V input. The Modem will not boot again until power has been completely removed and then restored.

Example:

`$PO*1f`

`$PO OK*3b`

`$M138 BOOT,SHUTDOWN*65`

$PW - Power Status

Set or query the rate for **$PW** unsolicited report messages for device power state. Also can retrieve the most current **$PW** message.

`$PW <@|?|<rate>>*xx`

Parameter	Description
`@`	Repeat most recent **$PW** message
`?`	Query current **$PW** rate
`rate`	Disable or set rate of **$PW** messages

Returns one of:

Parameter	Description
$PW <cpu_volts>,<unused>,<unused>,<unused>,<temp> *xx	The most recent **$PW** message
`$PW <rate>*xx`	The current **$PW** rate
`$PW OK*xx`	Parameters updated successfully
`$PW ERR,<error_type>*xx`	Command input error

Parameter	Description
`cpu_volts`	Voltage measured at input to the CPU
`unused`	Will always show as 0.00000
`temp`	CPU Temperature in degrees C to one decimal point (float)

Notes:

See the section *Command responses* for a description of **<error_type>**.

<rate> is a value between 1 and 2147483647 (2^{31}-1). It will be the number of seconds in between each message. If **<rate>** is 0, no messages will be sent.

Examples:

Calling the most recent power status message:

```
$PW @*67
```

```
$PW 3.28700,0.00000,0.00000,0.00000,28.0*3d
```

Returns a CPU voltage of **3.287**V and device temperature of **25.0**℃.

Setting the rate of power status messages:

```
$PW 30*24
```

```
$PW OK*23
```

Sets the rate of power status messages to one message every **30** seconds.

Querying the rate of power status messages:

```
$PW ?*18
```

```
$PW 900*1e
```

Returns a rate of one message every **900** seconds.

$RD - Receive Data Unsolicited Message

This unsolicited message provides an ASCII-encoded hexadecimal string with the user data received from the Swarm network. Some fields also include signal quality information for the received message. Received data unsolicited messages can be enabled/disabled using the **$MM** command with the message notification option.

```
$RD <appID>,<rssi>,<snr>,<fdev>,<data>*xx
```

Parameters	Description
appID	Application ID tag of message
rssi	Received signal strength in dBm for packet (integer)
snr	Signal to noise ratio in dB for packet (integer)
fdev	Frequency deviation in Hz for packet (integer)
data	ASCII encoded data packet

Example:

```
$RD AI=6078,RSSI=-84,SNR=12,FDEV=9,596f7527766520676f74206d61696c21*6e
```

The received packet has an **application ID** of 6078**,** a **rssi** of -84 dBm, a **snr** of -12 dB, and a **fdev** of 9 Hz. The returned encoded hexadecimal data from the packet is **596f7527766520676f74206d61696c21** (in ascii **= “You’ve got mail!”).**

$RS - Restart Device

Perform a software cold restart of the device. This command has an optional parameter that can also be used to erase the incoming and outgoing message storage before performing the restart.

`$RS [deletedb]*xx`

Parameter	Description
`deletedb`	Delete incoming and outgoing message storage prior to restart

Returns one of:

Value	Description
`$RS OK*xx`	Command has been accepted and the device will immediately perform a hardware restart
`$RS ERR,<error_type>*xx`	Command input error

Notes:

See the section *Command responses* for a description of **`<error_type>`**.

An **`OK`** response confirms that the device will successfully restart. No external power cycling is required.

Example:

```
$RS*01
```

```
$RS OK*25
```

```
$M138 BOOT,RESTART*3a
```

$RT - Receive Test

Set or query the rate for **$RT** unsolicited report messages for device power state. Also can retrieve the most current **$RT** message.

`$RT <@|?|<rate>>*xx`

Parameter	Description
`@`	Repeat most recent **$RT** message
`?`	Query current **$RT** rate
`rate`	Disable or set rate of **$RT** messages

Returns one of:

Parameter	Description
$RT RSSI=<rssi_sat>,SNR=<snr>,FDEV =<fdev>,TS=<time>,DI=<sat_id>*xx	The most recent **$RT** message from a satellite. Appears independently of the **rssi_background** noise messages
`$RT RSSI=<rssi_bkgnd>*xx`	The most recent **$RT** message. This is a measure of the background noise environment (and is not related to **rssi_sat**)
`$RT <rate>*xx`	The current **$RT** rate
`$RT OK*xx`	Parameters updated successfully
`$RT ERR,<error_type>*xx`	Command input error

Parameter	Description
rssi_bkgnd	Received background noise signal strength in dBm for open channel (integer). For reliable operation, this value should consistently be less than (more negative) than **-93** dBm: **rssi_bkgnd** >= -90 dBm Bad (not likely to work) **rssi_bkgnd** <= -93 dBm Marginal **rssi_bkgnd** = -97 dBm OK **rssi_bkgnd** <= -100 dBm Good **rssi_bkgnd** <= -105 dBm Great
rssi_sat	Received signal strength in dBm for packet from satellite (integer)
snr	Signal to noise ratio in dB for packet (integer)
fdev	Frequency deviation in Hz for packet (integer)
time	Date and time (UTC) of received packet
sat_id	Device ID of satellite heard (hexadecimal)

Notes:

See the section *Command responses* for a description of **<error_type>**.

<rate> is a value between 1 and 2147483647 (2^{31}-1). It will be the number of seconds in between each message. If **<rate>** is 0, no messages will be sent, and no packets received from a satellite will be displayed.

For reliable performance, the noise floor can be measured with an antenna connected to the device in the final built-up configuration. The **rssi_bkgnd** value reported by **$RT** should be less than (a more negative number) than **-93** dBm. If the **rssi_bkgnd** noise floor is more than (a less negative number, e.g. -88 dBm) **-93** dBm, then the customer should relocate the device to a different location, reconfigure the device ground plane, and also check that their DC-DC power supplies on their own PCB are not injecting RF background noise into the device.

Examples:

Calling the most recent receive test message. This will return the last packet received from a satellite:

```
$RT @*66
```

```
$RT RSSI=-102,SNR=-1,FDEV=426,TS=2020-10-02T13:56:21,DI=0x000568*70
```

Returns a satellite packet **rssi_sat** of **-102** dBm, SNR of **-1** dB, frequency deviation of **426** hz, a received time of **October 2nd, 2020 at 1:56:21 PM**, and the satellite's device ID is **0x000568**.

Setting the rate of receive test messages:

```
$RT 1*17
```

```
$RT OK*22
```

Sets the rate of receive test messages to one message every **1** second. The format of these scheduled messages will be the simple noise floor message. Any packets received from a satellite will be displayed as they are received, independently of the **$RT** rate.

Querying the rate of receive test messages:

```
$RT ?*19
```

```
$RT 5*13
```

Returns a rate of one message every **5** seconds.

$SL - Sleep mode

This command puts the device into a low-power sleep mode.

`$SL [S=<seconds>|U=<[YYYY-MM-DD ]T[hh:mm:ss]>]*xx`

Parameter	Description
`S=<seconds>`	Sleep for this many seconds
`U=<[YYYY-MM-DD ]T[hh:mm:ss]>`	Sleep until date (optional) and time

Returns one of:

Value	Description
`$SL OK*xx`	Sleep period accepted, device is now non-responsive
`$SL WAKE,<cause>*xx`	Device has woken from selected sleep mode
`$SL ERR,GPS_TIMENOTSET*xx`	Time not yet set from GPS
`$SL ERR,<error_type>*xx`	Command input error

The **S** parameter is the number of seconds to sleep. This value may range from 5 to 31536000 (approximately 1 year) seconds. If the command is accepted, the device will emit **`$SL OK`** and enter sleep mode for the requested duration.

The **U** parameter is a time and optional date the device should sleep until and then wake. If the date is not specified and the time to sleep until is less than the current time, the time is presumed to be the next day. For example, if the current time is 11:00:00 and **`$SL U=09:00:00`** is issued, the device will wake 22 hours from now. If a date and time are specified, and that date/time is before the current date/time, **`$SL WAKE`** will be immediately issued.

The **`$SL WAKE,<cause>`** message is emitted after the Modem wakes from a user-commanded sleep mode. The value of **`cause`** will be one of the following:

Cause	Description

GPIO	GPIO input changed from inactive to active state
SERIAL	Activity was detected on the RX pin of the Modem's UART
TIME	The **S** or **U** parameter time has been reached

If UART activity wakes the Modem, the **TIMEOUT** message will not be emitted as the Modem is now awake.

In sleep mode, the real-time clock is not GPS disciplined, and is therefore subject to some degree of drift. The longer the device is asleep, the more the drift will accumulate. The drift is estimated to be 20ppm, so if the Modem is commanded to sleep for 100 hours, the drift may result in the Modem actually sleeping in the range of 99.998 to 100.002 hours. The user should be aware of this when selecting a sleep with a long duration.

If the GPIO1 pin is configured as an input to wake the Modem, the sleep mode will be terminated if activity occurs on GPIO1. If the GPIO1 pin is configured as an output that indicates the Modem's sleep mode, GPIO1 will transition to the appropriate state if the **$SL OK** message is emitted.

Notes:

See the section *Command responses* for a description of **<error_type>**.

Examples:

Commanding the Modem to sleep for 3600 seconds (1 hour):

```
$SL S=3600*54
```

```
$SL OK*3b
```

```
$SL WAKE,TIME @ 2019-04-11T18:58:03*03
```

If the Modem receives any serial input before the planned wake time, then the Modem will wake with a message such as:

```
$SL WAKE,SERIAL @ 2019-04-11T18:57:45*1b
```

Similarly, if the GPIO1 pin is configured to wake on a high-to-low (or low-to-high) transition, then on transition on the GPIO1 pin, the Modem will also wake with a message such as:

```
$SL WAKE,GPIO @ 2019-04-11T18:57:55*0b
```

Commanding the Modem to sleep until October 1st, 2021 at 4:30:00 PM:

```
$SL U=2021-10-01T16:30:00*06
```

```
$SL OK*3b
```

$M138 - Modem Status Unsolicited Message

These unsolicited status messages indicate the readiness of the Modem for normal operation. This includes the conditions at power up, GPS acquisition, and certain error conditions. Modem status messages cannot be disabled.

`$M138 <msg>,[<data>]*xx`

Parameter	Description
`msg`	**`BOOT`** - Boot process progress with the following **`data`** reason: • **`ABORT`** - A firmware crash occurred that caused a restart • **`DEVICEID`** - Displays the device ID of the Modem • **`POWERON`** - Power has been applied • **`RUNNING`** - Boot has completed and ready to accept commands • **`UPDATED`** - A firmware update was performed • **`VERSION`** - Current firmware version information **`DATETIME`** - The first time GPS has acquired a valid date/time reference **`POSITION`** - The first time GPS has acquired a valid position 3D fix **`DEBUG`** - Debug message (**`data`** - debug text) **`ERROR`** - Error message (**`data`** - error text)

Notes:

A **`data`** message follows the **`BOOT`** message to indicate the reason for the startup.

The **`POSITION`** message and **`DATETIME`** message may occur in any order. Depending on the GPS signal quality, it may take several minutes before either message is emitted.

The customer application should wait until the boot process is complete and it has received the **$M138 BOOT,RUNNING*2a** message before executing any commands.

$TD - Transmit data

This command transmits data to the Swarm network.

`$TD [AI=<appID>|HD=<hold_dur>|ET=<expire_time>,]<[string|data]>*xx`

Parameter	Description
`AI=<appID>`	Application ID tag for message (optional, default = 0, maximum is 64999)
`HD=<hold_dur>`	Hold duration of message in seconds (optional, default = 172800 seconds, minimum = 60 seconds)
`ET=<expire_time>`	Expiration time of message in epoch seconds (optional, if omitted, same as **`hold_dur)`**
`<string|data>`	1 to 192 bytes of data (ASCII string) 2 to 384 bytes (hexadecimal written as ascii)

Notes:

See the section *Command responses* for a description of **`<error_type>`**.

In order to send a **$TD** command, you must first wait for a **$M138 DATETIME*56** response after power up. The Modem must wait for a valid time in order to accept a transmit command. All other commands are functional prior to receiving the **$M138 DATETIME*56** response.

Returns one of:

Value	Description
$TD OK,<msg_id>*xx	Message accepted for sending
$TD SENT RSSI=<rssi_sat>,SNR=<snr>,FDEV=<fdev>,<msg_id>*xx	This is an unsolicited response that occurs at the time when the message is sent to and acknowledged by the satellite. It includes the RF information for the satellite reply.
$TD ERR,TD_APPIDNOTANINT*xx	Application ID not recognized as integer
$TD ERR,TD_APPIDTOOLARGE*xx	Application ID greater than maximum allowed
$TD ERR,CMD_TD_HOLDDURNOTANINT*xx	Hold duration not recognized as integer
$TD ERR,CMD_TD_HOLDDURTOOLONG*xx	Hold duration OR expiration time too far in future (maximum 13 months)
$TD ERR,CMD_TD_HOLDDURTOOSHORT*xx	Hold duration OR expiration time less than 60 seconds in future
$TD ERR,CMD_TD_EXPIRATIONNOTANINT*xx	Expiration time not recognized as an integer (epoch seconds format)
$TD ERR,CMD_TD_DATATOOLONG*xx	Message is too large to send
$TD ERR,DBX_OUTGOINGFULL*xx	Outgoing message queue is full. Maximum of 1024 messages may be stored in the queue.
$TD ERR,GPS_TIMENOTSET*xx	Command received before time set by GPS
$TD ERR,CMD_BADPARAMLENGTH*xx	For data field: Message has odd number or non-hex characters when sending data as hexadecimal
$TD ERR,<error_type>*1c	Command input error

Notes:

Messages may be delivered to the Hive out of sequence. It's suggested if ordering is vital, for customers to add their own timestamp or sequence number.

The **AI**, **HD** and **ET** parameters are optional but must occur before the **<data>** portion of the command.

<appID> is the application ID tag for the message. If this option is not specified the message will be sent with application ID 0.

HD and **ET** affect the time when a message has expired and is no longer considered available to transmit. If neither **HD** or **ET** option is present, the default hold time of **172800** seconds (48 hours) will be used. If both options are specified, the **<hold_dur>** will be ignored. The outgoing message queue is scanned periodically and messages may be removed from the outgoing queue at any time after the expiration time has passed without notice.

Application ID Value	Description
0 to **64999**	The application ID tag for the message. Swarm reserves the use of application IDs between 65000 to 65535.

<hold_dur> is the number of seconds to hold the message if it has not been sent before expiring it from the outgoing message queue..

Hold Duration Value	Description
60 to **34819200** (13 months)	The message will be considered expired if not sent within the specified number of seconds. The maximum duration is 13 months from the current time.

<expire_time> is an epoch second date after which the message will be expired if it has not been sent.

Expiration Time Value	Description
<cur_time> + 60 secs to **<cur_time> + 34819200 secs**	The message will be considered expired if not sent before the specified time. The epoch seconds must lie between 60 seconds in the future and 13 months in the future. If not within this range an error will indicate it is too short or too long.

<string|data> may be expressed one of two different ways. If all the data to be sent is in the ASCII character range from 0x20 (space) to 0x7e (tilde), then the data may be sent as a string. A string is specified by enclosing the data in double quotes, e.g., "Hello, world". It is permissible for the string to contain double quotes within the string, e.g., "Today is a "new" day". If the data to be sent includes one or more character outside the 0x20 to 0x7e range, then it must be specified as pairs of hex characters ('0'..'9', 'A'..'F', 'a'..'f'), and must be a multiple of 2. Sending 'Hello' as hex would be 48656C6C6F. Illegal characters or an odd number of characters will cause a **CMD_BADPARAMVALUE** message to be returned.

<msg_id> is assigned by the device, and is an unsigned 64-bit value comprised of the device ID, a day of year counter, and a message of day counter. Responses that have a 0 as the message ID indicate the message has not been placed in the queue and therefore has no ID. The value should be treated as a simple arbitrary number.

Typical messaging flow:

1. User queues a message to be sent with application ID **40**, with a hold duration time of **7200** seconds (2 hours). The use of indexing parameters in the body of the message is recommended to aid in message identification and sequencing:

$TD AI=40,HD=7200,"{"d":"Demo message","t":"2021-02-26 14:28:56","seq":"00015"}"*09

2. Device acknowledges receipt of message, and message enters into device's internal outgoing message queue. Message is given the message ID of **5354468575855**. The outgoing message queue provides for a maximum storage of 1024 unsent messages. The outgoing message queue is non-volatile, and messages persist through sleep mode and power cycling:

$TD OK,5354468575855*2a

3. If the message has not expired due to time (it was previously set to be held until 2 hours from when it was entered), when the device hears a satellite pass overhead it attempts to send any messages in its outgoing queue. Messages are scheduled to be sent in the order they were entered into the device (e.g. first in outgoing queue, first transmitted). Each message heard by the satellite is individually acknowledged with a **$TD SENT** response - the acknowledged message is referred to by its message ID (**5354468575855**)

$TD SENT,RSSI=-99,SNR=5,FDEV=32,5354468575855*6f

4. The acknowledged message is marked in the outgoing message queue for deletion by the device. Any unsent messages will be attempted to be sent by the device at a later time.

Examples:

Sending a message from the device in ASCII:

$TD "{"d":"Demo message","t":"2021-02-26 14:48:56","seq":"00017"}"*08

$TD OK,5354468575916*2c

$TD SENT,RSSI=-104,SNR=-3,FDEV=345,5354468575916*44

Sending a message from the device in ASCII that will be held for 1 hour:

$TD HD=3600,"{"d":"Demo message","t":"2021-02-26 14:51:21","seq":"00018"}"*17

$TD OK,5354468575917*2d

$TD SENT,RSSI=-103,SNR=2,FDEV=-67,5354468575917*70

Sending a message from the device in ASCII that will expire on 2022-01-01T12:34:56:

$TD ET=1641040496,"{"d":"Demo","t":"2021-02-26 15:01:22","seq":"00020"}"*45

$TD OK,5354468575919*23

$TD SENT,RSSI=-100,SNR=-3,FDEV=437,5354468575919*4d

Sending a message from the device in ASCII with application ID 3000:

$TD AI=3000,"{"d":"Demo","t":"2021-02-26 15:33:22","seq":"00021"}"*52

$TD OK,5354468575919*23

$TD SENT,RSSI=-100,SNR=-3,FDEV=437,5354468575919*4d

Sending a message from the device in HEXASCII:

$TD 49206C6F766520537761726D*4c

$TD OK,5354468575916*2c

$TD SENT,RSSI=-107,SNR=3,FDEV=199,5354468575916*69