Precision GNSS Receiver mPCIe Module

The Swift Navigation Precision GNSS Receiver mPCIe Module (PGM) enables low-cost precision navigation in the toughest environments through Global Navigation Satellite System (GNSS) positioning and inertial sensors fusion technology (INS). Designed with the industry standard "full" Mini PCI Express module form factor, the product is ideally suited as an add-on to embedded computing platforms with mini PCIe expansion slots and applications that require precision position such as automotive, robotics, high precision data collection, video/sensor position, and image time-tagging. The card is designed specifically for the Swift Navigation's Starling Positioning Engine on a host application processor for real-time precision navigation with dual frequency L1/L5 carrier phase differential GNSS RTK and inertial / odometer sensor fusion.

Features:	
GNSS Constellations and Signals	GPS: L1CA, L5; Galileo: E1, E5a; BeiDou: B1, B2a Designed for Swift Navigation Starling Positioning Engine and Skylark corrections service.
GNSS Antenna	U.FL connector for active GNSS L1/L5 antennas. 3.3 V antenna bias. Optional user-provided antenna bias (max. 12 V).
On-Board IMU	Automotive grade 3D accelerometer and gyroscope.
Precision Wheel Odometry Input	Edge-mount pico-blade connector for wheel tick (up to 4 kHz) and reverse indicator vehicle inputs as well as PPS output for synchronization and timing.
Input / Output Interfaces	mPCIe USB 2.0 interface to serial UART adapter (two UARTs up to 460800 baud), one CAN edge-mount interface compatible with high speed CAN and CAN-FD.
Data Protocols	RTCM v3 MSM GNSS Observations (up to 10 Hz) Proprietary RTCM v3 messages for inertial and vehicle data, configuration, control, and firmware updates.
Power	3.3 V supply, 1.3 W (typical), on-board coin battery for GNSS quick starts
Environmental	Operating temperatures: -40°C to +85°C. Vibration (random): 3 g RMS.
Dimensions	50.95 mm x 30 mm x 7.3 mm, Mini PCIe "Full".

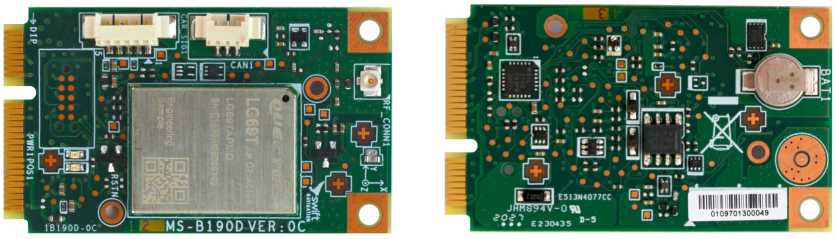

Figure 1: *PGM Top and Bottom Views*

Table of Contents

Functional Block Diagram

The PGM module functional block diagram is presented on Figure 2. The main component is Quectel's LG69T GNSS module that includes ST Microelectronics (STM) TeseoV GNSS measurement engine, STM ASM330 3D IMU sensor and STM32 microcontroller.

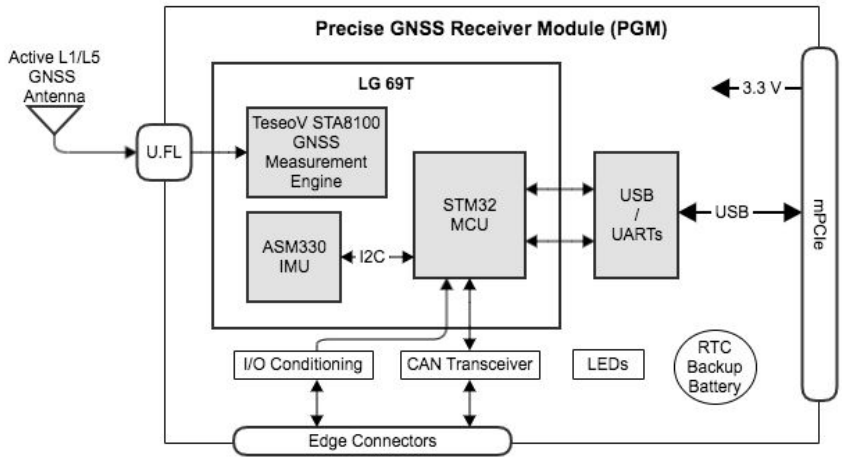

Figure 2: *PGM Functional Block Diagram*

Additionally, the module contains a digital I/O conditioning circuit, CAN transceiver, status LEDs, battery for the RTC and UART/USB adapter for the host communication. Two serial ports (main and AUX) are available for communication.

Mechanical & Environmental

The card conforms to the mini PCIe "Full" form factor according to the "PCI Express® Mini Card Electromechanical Specification Revision 1.2" document (https://s3.amazonaws.com/fit-iot/download/facet-cards/documents/PCI_Express_miniCard_Electromechanical_specs_rev1.2.pdf) available courtesy of PCI Express.

Note: the height of the module is larger than typical modem GNSS cards to accommodate heat sink for operation in warm passively cooled PC enclosures which may have high operating temperatures inside the enclosure. It is important to verify that there is adequate clearance above the mPCIe slot on the motherboard.

Mechanical Specifications	
Length	50.95 mm
Width	30 mm
Height	15 mm (including heat sink), 3.0 mm clearance required under card (designed for mPCIe connectors at least 3 mm high) as measured from top of the PCB surface.
PCB Thickness	1 mm
Weight	7.5 g (22.5 g with heat sink)

MFR part number	MS-B190D (PCB silk screen)
Swift part number	SN-01097 (PCB adhesive label)

Environmental Specifications	
Operating Temperature	-40°C to 85°C
Storage Temperature	-50°C to 85°C
Vibration (random)	3 g RMS

Mechanical Drawings

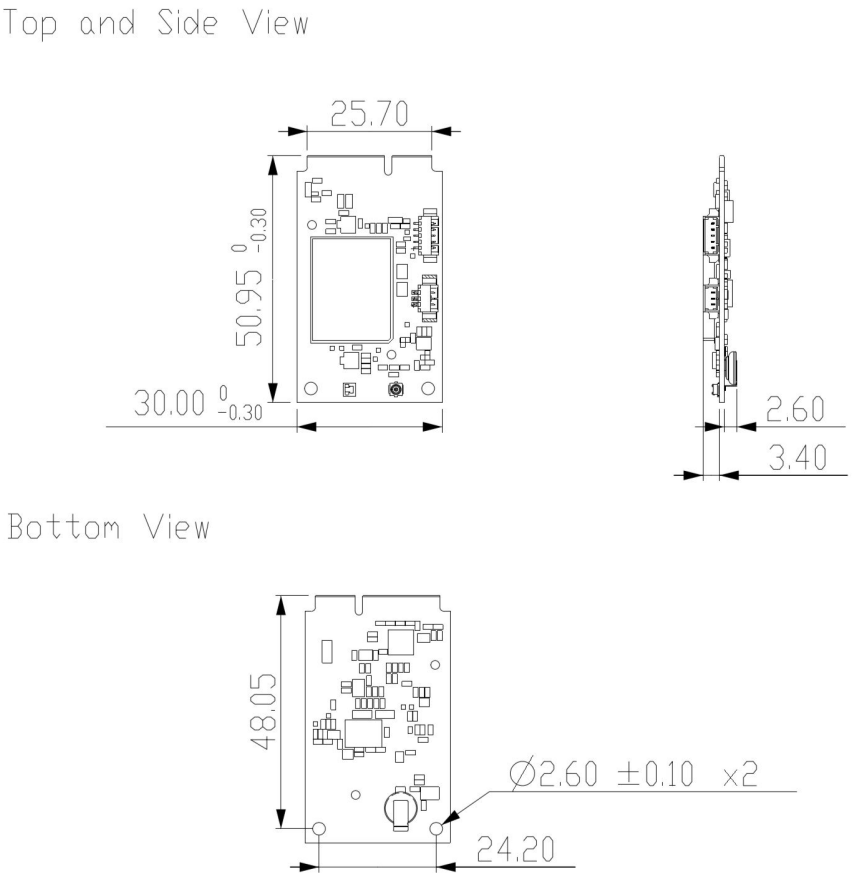

Figure 3: *PGM Dimensioned Drawing Without Heat Sink (Units in mm)*

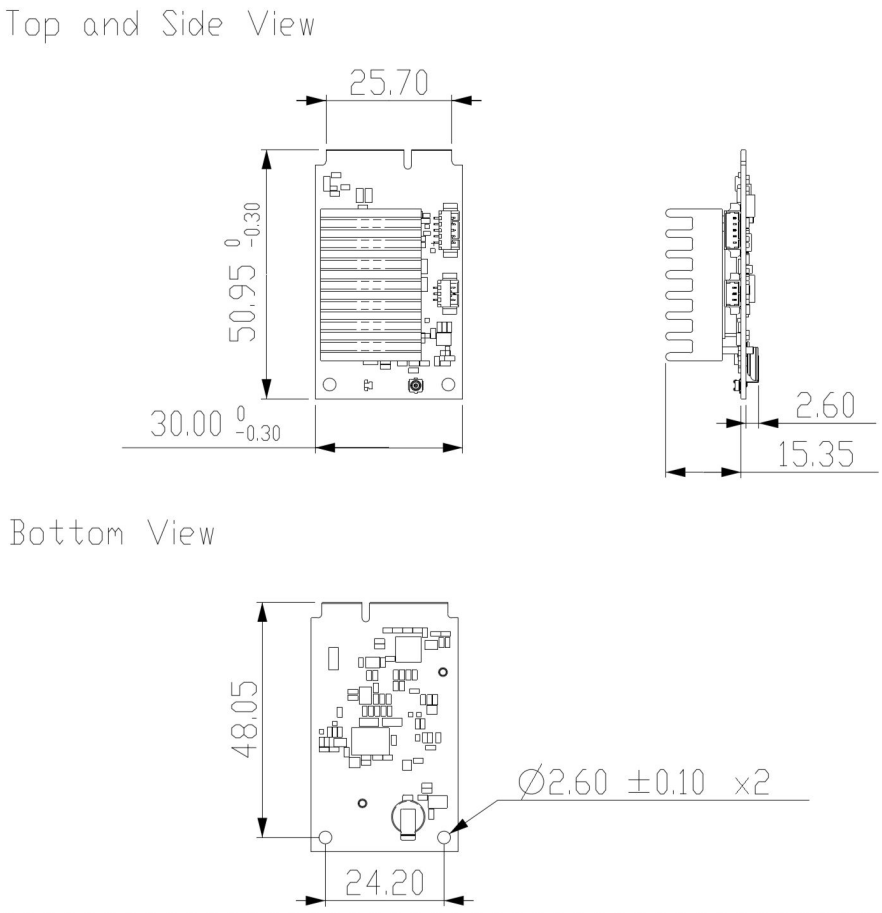

Figure 4: *PGM Dimensioned Drawing With Heat Sink (Units in mm)*

Heat Sink

By default the PGM is provided with an attached heat sink. The heat sink is recommended and can improve module lifespan and navigation performance. Pictured below is the module with the heat sink factory installed.

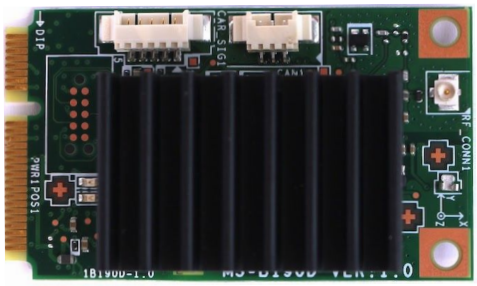

Figure 5: *PGM With Heat Sink Installed*

Electrical

Connectors

The PGM module uses the USB and power pins on the mini-PCIe connector per the industry standard pinout. Additionally, there are 2 optionally used edge mount connectors for additional PGM functionality, as well as an U.FL antenna connector.

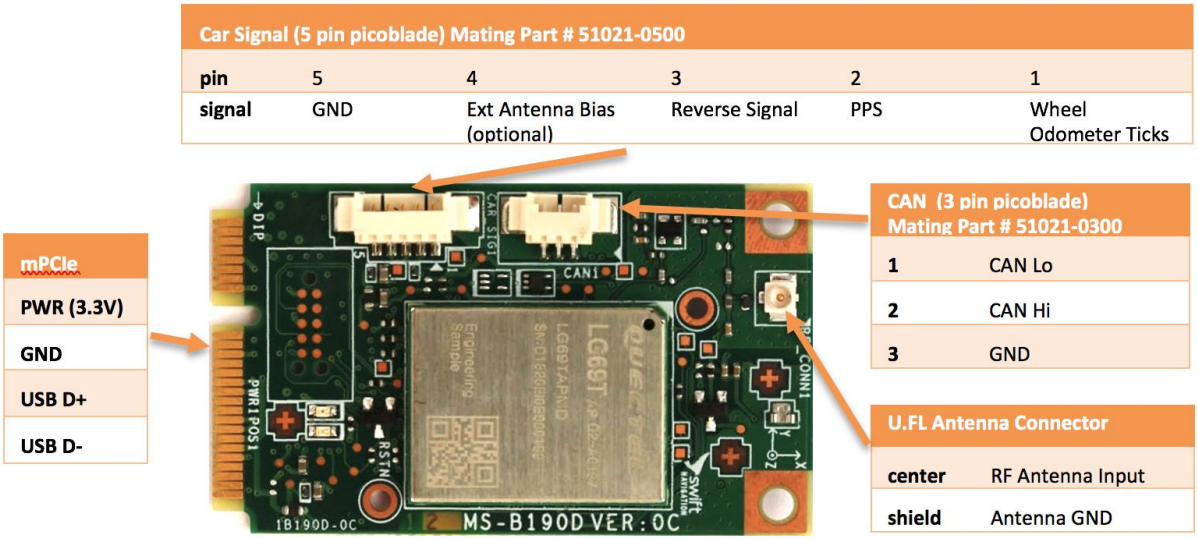

Figure 6: *PGM Connectors Description*

Power	
Supply Voltage	3.3 V, +/- 5%
Current Consumption	600 mA peak, 400 mA typical
Antenna Bias Voltage	3.3 V Optionally 5 to 12 V user supplied voltage (factory load option)
Real Time Clock (RTC) Backup Battery	3 mAh 3 V, Lithium rechargeable coin cell battery (Seiko MS621R II27E) RTC backup enables warm and hot start functionality after a system start, provided the TeseoV has previously received the relevant navigation data.

Digital I/O and CAN	
CAN Bus Termination	Factory load option, unterminated by default
CAN Bus Voltage Range	-14 V to 14 V (differential), ESD protected
GPIO Voltage Range	-1.8 V to 14 V, ESD protected / Surge (> 1.8V logic level high for inputs)
Wheelticks Input	Square wave, up to 4 kHz

Board LEDs	
PWR (Green)	**Power Status** Direct connection to mPCIe 3.3 V power
SAT (Blue)	**GNSS Status** Off: < 4 satellites tracked Blinking: >= 4 satellites tracked, but no GNSS fix On: TeseoV GNSS fix present

UART Configuration

The PGM exposes two USB virtual serial interfaces to a host processor, labelled according to the operating system's driver and any installed udev rules (Linux). Each UART operates at a baudrate of 460800. The "Enhanced" (main) UART is configured to provide raw RTCM information for a hosted Starling process. The "Standard" (AUX) UART is configured for module firmware updates and can send NMEA messages.

UART configuration		
UART Label	**Data**	**Configuration**
SILABS Enhanced UART (Main Port)	Raw RTCM for Starling ingestion.	Baud rate: 460800 Bytesize: 8 Parity: None Stop bit: 1 Flow control: None
SILABS Standard UART (AUX Port)	Firmware update and NMEA output	

Subassembly Specifications

GNSS (Quectel LG69T powered by Teseo V STA8100)	
Sensitivity	Acquisition: -147 dBm Tracking: -163 dBm Reacquisition: -156 dBm
Tracking channels	80
Timing accuracy (PPS)	+/- 20 ns
Time to first fix (TTFF)	Cold Start: < 33 s Warm Start: < 25 s

	Hot Start: < 1.5 s

IMU (STM ASM330LLH)	
Accelerometer range	± 4 g (default), 2 – 16 g (configurable)
Gyro range	± 125 deg/s (default), 125 – 4000 deg/s (configurable)
Gyro angular random walk	0.21 degrees / sqrt(hour) Typical @ 25°C
Gyro bias instability	3 deg/ hour typical @ 25°C
Nonlinearity	0.1% FS
Acceleration noise density	200 µg/sqrt(Hz)
Temperature stability	±100 ppm / Degree Celsius [Acceleration] ±70 ppm / Degree Celsius [Angular rate]

IMU Orientation

Figure 7: *PGM IMU Orientation Marking*

Starling and Skylark System Integration

The PGM module is designed to run together with a Starling positioning engine software on a host application processor. This allows for flexible integration that decouples positioning functionality from embedded hardware. Ultimately the PGM hardware provides GNSS, IMU, and vehicle data in a unified and synchronized way so that data can be consumed by Starling to produce precise navigation solutions. In particular, Starling on a host processor consumes the raw GNSS observations (pseudorange, carrier phase, and doppler), the raw IMU data (acceleration and gyroscope attitude rates), and any raw vehicle data (wheel speed or odometer distance traveled) to produce position, velocity, attitude, and time output. The output can be further improved by connection to Swift Navigation's cloud correction service Skylark which provides GNSS observation corrections to account for GNSS orbit, clock, and atmospheric errors in the GNSS observables. Starling software is delivered as a single binary for the target platform with configuration via YAML formatted text file for any changes to default system configuration. Figure 8 details the overall system architecture.

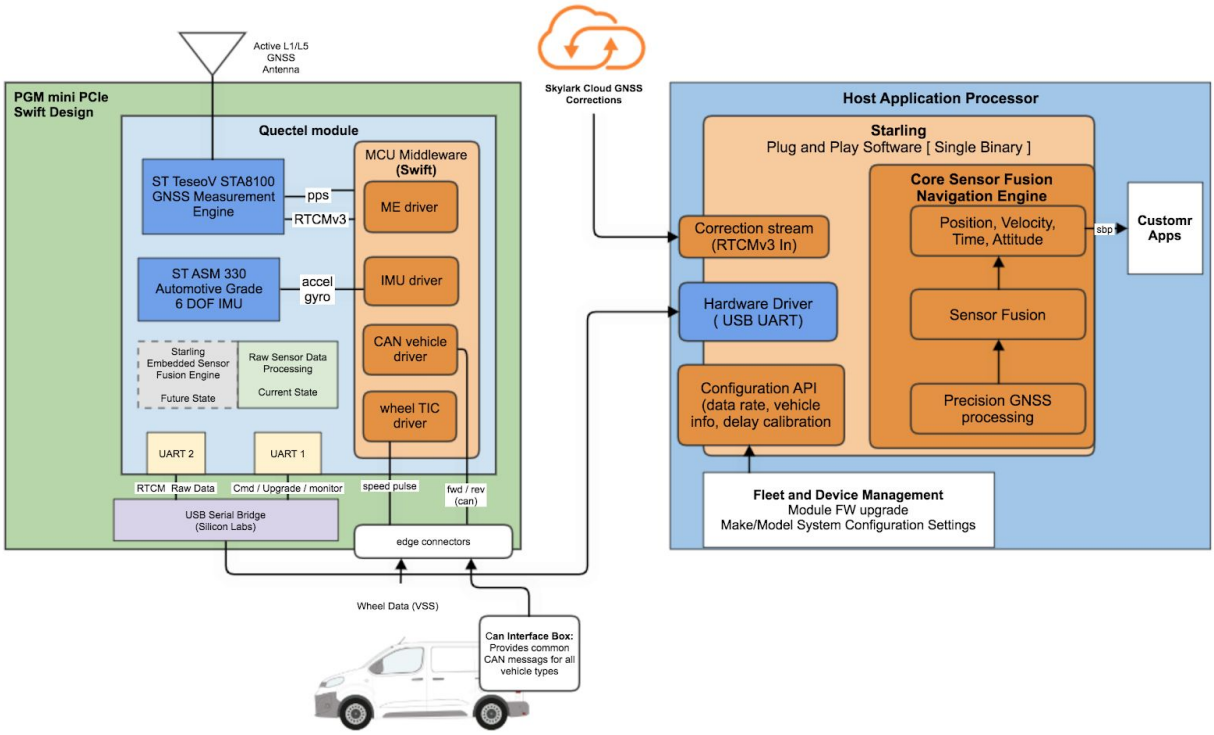

Figure 8: *Starling System Architecture*

The format for the raw data communication to Starling is outlined in the following section (RTCM v3 Communication Protocol from PGM board) and is a published open standard compatible with most commercial sensor fusion and positioning systems.

Starling software can communicate with customer applications in Swift Binary Protocol streaming format, NMEA stream format, or through a custom adapter in the Starling Platform upon customer request. Starling provides a standard set of position, velocity, attitude, quality information, and solution metadata information such as positioning mode, firmware version, configuration, corrections status and

age, and sensor fusion convergence status, detailed in the SBP Communication Protocol in the Starling Software section.

The communication interface between Starling and customer applications can be “pipes” in the operating system, TCP/IP servers/clients, and/or acting directly on file pointers.

RTCM v3 Communication from PGM board

Starling communicates with the PGM board bidirectionally through the RTCM v3 protocol published by the RTCM organization (rtcm.org). This protocol is a lightweight serial frame with a 24 bit CRC and flexible message payload. While designed specifically for communication of GNSS corrections, this product uses the protocol via message IDs reserved by the organization for vendor-specific implementation. This approach was chosen to avoid proprietary protocols in favor of open interoperability, and for best compatibility with the GNSS measurement engine onboard the PGM (TeseoV). The RTCM v3 framing information is shown in Appendix A. Refer to the RTCM v3 specification available from the organization for more detail.

The first 12 bits of the variable length data message are defined as the “message ID” in the protocol. Table below lists the RTCM v3 message IDs produced and possible for receipt by the module, and where appropriate, sub IDs for any vendor message ids supported by the device. This table lists the default supported message ids for integration with Starling for precision navigation.

RTCM ID	Sub ID	<->	Name	Notes
1019	NA	out	GPS eEphemeris	Standard RTCM v3
1042	NA	out	Beidou ephemeris	Standard RTCM v3
1046	NA	out	Galileo ephemeris	Standard RTCM v3
1077	NA	out	MSM7 GPS observations	Standard RTCM v3
1097	NA	out	MSM7 Galileo observations	Standard RTCM v3
1127	NA	out	MSM7 Beidou observations	Standard RTCM v3
4062	2304	out	Swift Navigation raw IMU	See SBP Specification
4062	2305	out	Swift Navigation IMU auxiliary msg	(scaling and IMU type) See SBP Specification
4062	2308	out	Swift wheel tick message	See SBP Specification
4062	160	in	Swift settings write request	Changes Swift MCU firmware configuration. See SBP Specification
4062	175	out	Swift settings write response	Confirms configuration change. See SBP Specification
4062	164	in	Swift settings read request	Queries receiver setting value . See SBP Specification
4062	165	out	Swift settings read response	Returns current receiver configuration value in memory . See SBP Specification
4062	182	in	Swift message reset	Resets receiver and possibly resets receiver settings to default values. See SBP Specification
4075	NA	out	Navigation raw data frame	Not standard, but implemented in many receivers to enable almanac and SBAS

				operation (https://software.rtcm-ntrip.org/wiki/NDF)
999	26	out	Signal quality metrics 2 (SIGQM2)	See TeseoV RTCM v3 document
999	8	out	Inter frequency biases (IFB)	See TeseoV RTCM v3 document
999	4	out	Inter frequency biases (IFB)	Position, velocity, time from on-board ME solution.
999	7	out	GLONASS inter channel biases (ICB)	See TeseoV RTCM v3 document
999	9	out	Ionospheric model parameters	See TeseoV RTCM v3 document
999	24	out	RF status (RFS)	See TeseoV RTCM v3 document
999	25	out	TeseoV firmware version	See TeseoV RTCM v3 document
999	2	in	TeseoV receiver configuration and control (RCC)	See TeseoV RTCM v3 document
999	23	in	Set message transmission interval (SETMTI)	See TeseoV RTCM v3 document

There are two vendor message IDs implemented for receiver communication over RTCM v3.

Vendor message ID 999 (STM)

The RTCM v3 message with ID 999 is implemented by the STMicroelectronics on TeseoV GNSS chipset. This message has an additional 8 bit subtype ID immediately following the message ID, to yield an overall frame structure as defined in the table below. The payload has the contents as described in the STM document titled *AN GNSS RTCM3 Proprietary Interface* available from ST and included as a reference with this document. The messages with ID 999 are used for detailed receiver state information that can improve precision navigation as well as receiver metadata useful for a complete view of receiver state. These messages can also be used to specify GNSS chipset configurations at run time as a passthrough to the GNSS chipset firmware.

RTCM v3 Preamble	Reserved	Msg Len (bytes)	Message ID	Subtype ID	Payload	CRC
8 bits	6 bits	10 bits	12 bits	8 bits	Variable	24 bits

Vendor message ID 4062 (Swift Navigation)

The RTCM v3 message with ID 4062 is implemented by the Swift Navigation. This message has an additional inner frame with SBP information as described in the table below. Message ID 4062 is used to communicate raw IMU and wheel odometer information from the PGM board as well potentially for configuration control of the Swift MCU firmware if it is determined that any runtime configuration is required during system integration for volume customers.

RTCM v3 Preamble	Reserved	Msg Len (bytes)	Msg ID	Reserved	SBP Message ID	SBP Sender	SBP Len	SBP Payload	CRC
8 bits	6 bits	10 bits	12 bits	4 bits	16 bits	16 bits	8 bits	Variable	24 bits

Configuration and Control

The PGM module supports a limited configuration interface over both RTCM v3 message ID 4062 and ID 999 for configuring pertinent settings such as raw measurement output frequency (999), enabled constellations (999), and wheel odometer interface settings (4602). Detailed configuration specification will be provided with firmware documentation and release notes as customer system requirements are clarified and implemented.

Firmware Update

The PGM module is updateable over either UART interface on the board. For test and evaluation, Quectel provides a GNSSFlash tool which is capable of updating both individual firmware images on the board when used in concert with software that puts the board in reset via the Silicon Labs GPIOs (pgm_reset binary or software provided by Silicon Labs), or a serial message to the PGM to reset. Swift provides a unified upgrade Python command line tool ("pgm-update.py") designed for long term fleet management of the product. Custom implementation is possible through integration with software API re-implementation of the serial protocol (documentation upon request). There are two firmware images on the board, one for the TeseoV GNSS measurement engine and one for the System MCU. Swift's unified update software combines those separate images into one ZIP file file. Refer to the *PGM User Manual* for the firmware update details.

Part Numbering

Part Number Revisions

Swift Navigation part number*	**Hardware serial number will begin with**	**Notes**
01097-01 Rev01	0109701	Initial release
01097-02 Rev02**	0109702	Layout updates Updates to IMU interrupt connection to MCU Support for hardware antenna sensing capability
01097-02 Rev03	0109702	Removal of inline LNA to reduce risk of gain saturation in presence of strong interferers

*emitted through NMEA SWFTL / RTCM message 4062.
**will report "PGM HW version: 01097-01 Rev 03" in NMEA SWFTL / RTCM message 4062

As metadata to aid in device tracking, the module reports the following information about its own provenance as either NMEA text or in the RTCM log in message 4062 periodically.

NMEA SWFTL / RTCM message 4062 / SBP MsgLog level "Info" text	**Description**
PGM HW version: 01097-02 Rev03*	Reflects the part number of the device. Refer to table "Part Number Revisions" (above) for details on individual part numbers.
PGM process version: 01109-01 Rev04	Reflects the factory production process version that was used to produce the board.
STA8100 FW version : GNSSLIB_9.3.0.0_RC_ARM	Reflects the STA8100 firmware version
MCU Unique ID: 0042001A-31395119-35333634	Provides a unique ID for the hardware based upon the MCU microcontroller
MCU FW version: v1.0.0	Reflects the Swift Firmware version installed on the PGM module

*01097-02 Rev02 hardware will report "PGM HW version: 01097-01 Rev 03"

Serial Number Coding

PGM serial number is printed on a white sticker at the bottom of the board. It represents Swift Navigation's serial number for the assembly. Additionally, the Quectel module has an individual serial number etched into the RF shield. The coding below refers to the Swift Serial Number.

AAaaa	BB	L	XXXXX
BASE PART NUMBER	REVISION	LOCATION	Serial NUMBER

PGM serial number example:

0109701300088

Appendix A

RTCM v3 Data Protocol

The table below describes the framing structure of the RTCM v3 message.

Preamble	Reserved	Message Length	Variable Length Data Message	CRC
8 bits	6 bits	10 bits	Variable length, integer number of bytes	24 bits
11010011	Not defined – set to 000000	Message length in bytes	0-1023 bytes	QualComm definition CRC-24Q

Refer to the RTCM v3 specifications available from the RTCM organization (rtcm.org) for more information.

SBP Data Protocol

The table below describes the framing structure of the SBP message.

SBP consists of two pieces:

- an over-the-wire message framing format
- structured payload definitions

As of Version 3.2.0, the frame consists of a 6-byte binary header section, a variable-sized payload field, and a 16-bit CRC value. All multibyte values are ordered in **little-endian** format. SBP uses the CCITT CRC16 (XMODEM implementation) for error detection[1].

Offset (bytes)	Size (bytes)	Name	Description
0	1	Preamble	Denotes the start of frame transmission. Always 0x55.
1	2	Message Type	Identifies the payload contents.
3	2	Sender	A unique identifier of the sender. On the Piksi, this is set to the 2 least significant bytes of the device serial number. A stream of SBP messages may also include sender IDs for forwarded messages. By default, clients of 'libsbp' use a sender id value of '0x42'. Sender id '0x42' is used to represent device controllers such as the Piksi Console.
5	1	Length	Length (bytes) of the Payload field.
6	*N*	Payload	Binary message contents.
N + 6	2	CRC	Cyclic Redundancy Check of the frame's binary data from the Message Type up to the end of Payload (does not include the Preamble).
	N + 8		Total Frame Length

The SBP specification is available for download from support.swiftnav.com

NMEA 0183 Data Protocol

NMEA 0183 sentences are delimited by the '$' character and take the form below for the example message with string $GPVTG,089.0,T,,,15.2,N,,,*7F<CR><LF>

	Delimiter	**TalkerID**	**Address**	**Data Fields**	**Checksum Delimiter + Checksum**	**Line feed**
example	$	GP	VTG	089.0,T,,,15.2,N,,,	*7F	<CR><LF>

The checksum is the 8-bit exclusive OR (no start or stop bits) of all characters in the sentence, including "," and "^" delimiters, between but not including the "$" or "!" and the "*" delimiters. The hexadecimal value of the most significant and least significant 4 bits of the result is converted to two ASCII characters (0-9, A-F (upper case)) for transmission. The most significant character is transmitted first.

Refer to the NMEA 0183 specifications available from the NMEA organization (nmea.org) for more information.

Swift has implemented one vendor extension NMEA sentence with talker ID and address "$PSWTL". This encodes the Swift Log message.

