OPENIMU335 USER MANUAL

Document Part Number: 7430-3321-01

ACEINNA, Inc.

email: info@aceinna.com, website: www.aceinna.com

Date	Document Revision	Firmware Applicability	Description	Author
May 07, 2021	-01	V07.02.04	Initial Release	SH

⚠ WARNING

Aceinna has developed this product exclusively for commercial applications. It has not been tested for, and Aceinna makes no representation or warranty as to conformance with, any military specifications or that the product is appropriate for any military application or end-use. Additionally, any use of this product for nuclear, chemical, biological weapons, or weapons research, or for any use in missiles, rockets, and/or UAV's of 300km or greater range, or any other activity prohibited by the Export Administration Regulations, is expressly prohibited without the written consent of Aceinna and without obtaining appropriate, US export license(s) when required by US law. Diversion contrary to U.S. law is prohibited.

Table of Contents

About this Manual

The following annotations have been used to provide additional information.

□ NOTE

Note provides additional information about the topic.

□ EXAMPLE

Examples are given throughout the manual to help the reader understand the terminology.

□ IMPORTANT

This symbol defines items that have significant meaning to the user

⚠ WARNING

The user should pay particular attention to this symbol. It means there is a chance that physical harm could happen to either the person or the equipment.

The following paragraph heading formatting is used in this manual:

1 Heading 1

1.1 Heading 2

1.1.1 Heading 3

1.1.1.1 Heading 4

Normal

1 Introduction

1.1 Manual Overview

This manual provides a comprehensive introduction to ACEINNA's OpenIMU335RI Open-Source IMU with CAN interface This product is a sister part to the turn-key MTLT335. The parts share the same hardware and CAN messaging but differ in their RS-232 messaging. For users wishing to get started quickly, please refer to the two-page Best Practices Guide available online at www.aceinna.com. Table 1 highlights the content in each section and suggests how to use this manual.

Table 1. Manual Content

Manual Section	Who Should Read?
Section 1: Overview	All customers should read section 1.
Section 2: Interface	Customers designing the electrical and mechanical interface to the OpenIMU335RI series products should read Section 2.
Section 3: Theory of Operation	All customers should read Section 3.
Section 5: CAN Port Interface	Customers designing the software interface to the OpenIMU335RI series products CAN Port should review Section 5.
Section 6 - 7: RS-232 Port Interface	Customers designing the software interface to the OpenIMU335RI series products RS-232 Port should review Sections 6 - 7.
Section 8: Bootloader	Customers upgrading firmware should review Sections 8.
Section 9: Warranty and Support	Customers who need the support information should review Section 9.

1.2 Overview of the OpenIMU335RI

This manual introduces the use of ACEINNA's OpenIMU335RI open inertial platform. This manual is intended to be used as a detailed technical reference and operating guide. ACEINNA's OpenIMU335RI combines the latest in high-performance commercial MEMS (Micro-electromechanical Systems) sensors and digital signal processing techniques to provide a small, rugged and cost-effective solution for accurately sensing attitude in dynamic applications.

Table 2. OpenIMU335RI Series Feature Description

Product	Features
OpenIMU335RI	Pitch, Roll, Yaw (with VG/AHRS algorithm), 3D ±8 g acceleration, 3D ±400 deg/s Bias Corrected Rate, 3D ±8 Gauss Magnetic field strength (with optional magnetometer)

The OpenIMU335RI Series is based on ACEINNA's third generation of MEMS-based Inertial Systems, building on over a decade of field experience, and encompassing thousands of deployed units and millions of operational hours in a wide range of land, marine, airborne, and instrumentation applications.

At the core of the OpenIMU335RI Series is a rugged MEMS inertial sensor cluster that is also used in such products as IMU383 and OpenIMU330BI. The OpenIMU335 is available a as 6-DOF (Degrees of Freedom) device or a 9-DOF device (with optional magnetometer). The 6-DOF MEMS inertial sensor cluster includes three axes of MEMS angular rate sensing and three axes of MEMS linear acceleration sensing. The part is populated with a 3-axes magnetometer in the case of the 9-DOF device. Please contact Aceinna to order the 9-DOF device.

These sensors are based on rugged, field proven silicon micromachining technology. Each sensor within the cluster is individually factory calibrated for temperature and non-linearity effects during ACEINNA's manufacturing and test process using automated thermal chambers and precision rate tables.

Coupled to the inertial sensor cluster is a high-performance microprocessor that can be programmed with precompiled IMU, VG/AHRS, INS applications, or user code that utilize the inertial sensor measurements to accurately compute attitude (and for the INS application, position). The attitude algorithm utilizes a multi-state Extended Kalman Filter (EKF) to correct for drift errors and estimate sensor bias values.

The differentiating feature of the OpenIMU335RI Series is the trio of redundant MEMS sensor clusters. This redundancy has two direct benefits:

1) Combining multiple sensors reading improves the noise characteristics of the output signal
2) Using more than one sensor enables the unit to operate through a single sensor-chip failure by detecting and voting out the failed part. Failures include stuck or railed readings as well as sustained inconsistency between the three sensor sets.

Another unique feature of the OpenIMU335RI Series is the extensive field configurability of the units. This field configurability allows the OpenIMU335RI Series of Inertial Systems to satisfy a wide range of applications and performance requirements with a single mass-produced hardware platform. Parameters that can be configured include baud rate, low pass filter settings (acceleration and rate sensors) packet type, update rate, and defining of custom orientation.

The OpenIMU335RI firmware is designed to be field upgradable so units in the field can be upgraded to take advantage of new features or algorithm improvements that may be available in future firmware revision releases.

The OpenIMU335RI is available in a lightweight, rugged, IP69K sealed plastic enclosure with nickel plated brass mounting bushings designed for cost-sensitive commercial applications requiring a robust solution that can be exposed to the elements. The OpenIMU335RI can be configured to output data over a CAN Port and / or a RS-232 serial port. The OpenIMU335RI RS-232 output data port is supported by Aceinna Navigation Studio, a navigation system developer's website and web-platform. It consists of a graphical user interface to control and configure OpenIMU units. The graphical user interface can be customized for user specific messaging and settings without any additional coding and provides field configuration, diagnostics, charting of sensor performance, and data logging with playback.

2 Interfaces

2.1 Electrical Interface

2.1.1 Connector and Mating Connector

The OpenIMU335RI connector is an AMPSEAL16 6-pos defined in Figure 1

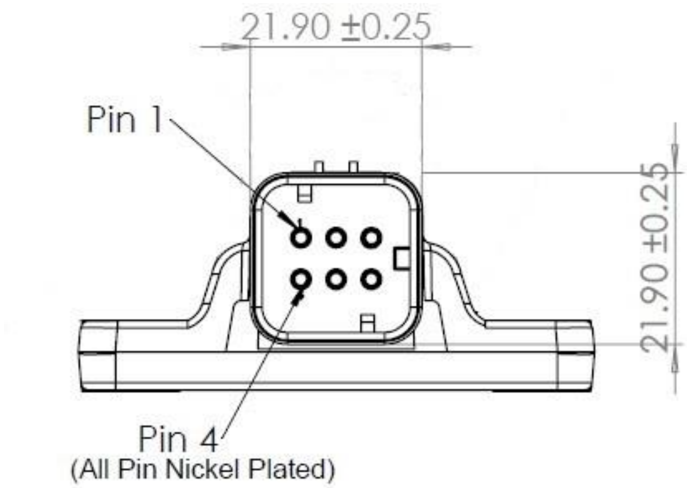


Figure 1 OpenIMU335RI Interface Connector

The mating connector is: TE Connectivity 776531-1 or equivalent, which is currently listed at the following web address: http://www.te.com/usa-en/product-776531-1.html

The definitions of connector pins are in Table 3.

Table 3. OpenIMU335RI Interface Connector Pin Definition

Pin	Signal
1	CAN H
2	CAN L
3	Ground
4	RS-232 RX

5	RS-232 TX
6	Power

2.1.2 Power Input and Power Input Ground

Power is applied to the OpenIMU335RI on pin 6, and ground is Pin 3; The OpenIMU335RI accepts an unregulated 9 to 32 VDC input. It is reverse polarity and ESD protected internally. It is designed to be compatible with 12 V and 24 V power environments of Heavy Equipment and Passenger Vehicle power systems.

2.1.3 CAN Serial Interface

The OpenIMU335RI is equipped with a CAN 2.0 electrical interface and is compliant to the SAE J1939 protocol standard.

Baud Rate: The default CAN baud rate setting is 250kb/s. The OpenIMU335RI can support up to 1000kb/s baud rate. The CAN baud rate can be changed and permanently saved by the user using the RS-232 port and NAV-VIEW 3.x SW running on a PC or unit can be configured in AutoBaud detection mode. See section 5.1.3.

CAN Address: The default CAN address is 0x80. The OpenIMU335RI has address claiming capability. In the event there is a conflict with another module on the bus with the same address, it will attempt to claim a new address and save it in non-volatile memory. The CAN address can also be changed and permanently saved via the CAN bus. See section 5.1.1.

2.1.4 RS-232 Serial Data Interface

The default baud rate of the RS-232 interface is 115200 bps as shipped with the default IMU application. The OpenIMU335RI also supports 38400, 57600, 230400 bps baud rates. Users can choose the desired baud rate and permanently save it in non-volatile memory through Navigation Studio (or alternatively edit the source code).

The RS-232 standard defines the voltage levels that correspond to logical one and logical zero levels for the data transmission and the control signal lines. Valid signals are either in the range of +3 to +15 volts or the range −3 to −15 volts with respect to the "Common Ground" (GND) pin.

A "3-wire" RS-232 connection consisting only of transmit data, receive data, and ground, is commonly used.

2.2 Mechanical Interface

The OpenIMU335RI mechanical interfaces are defined by the outline drawings in Figure 2.

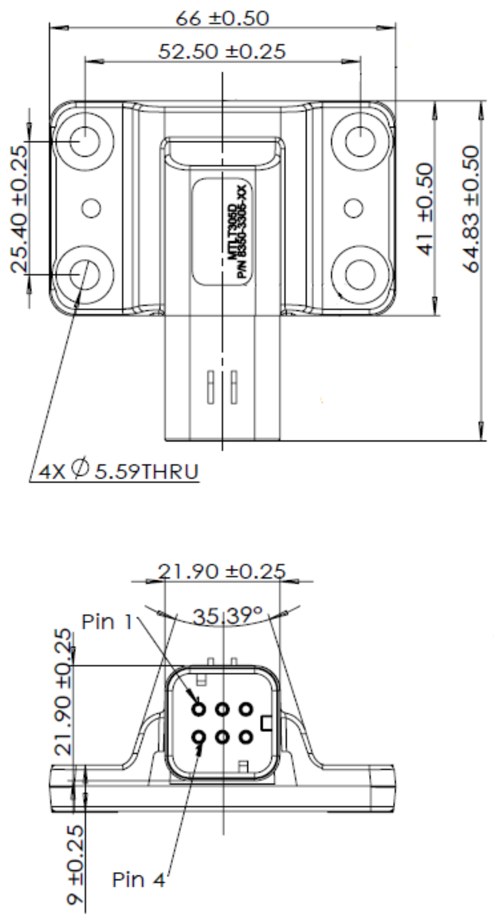


Figure 2. OpenIMU335RI Outline Drawing

Mating Connector: TE Connectivity 776531-1 or equivalent. See Figure 1.

2.2.1 Recommended Mounting Hardware and Torque

Use 4 - M5 Alloy Steel Socket Head Screws to secure the OpenIMU335RI.

It is recommended to use standard M5 washer with outer diameter of 10mm, lock washer and Loctite 242 thread lock.

The washer outer diameter must NOT be larger than the outer diameter of the bushing (11mm).

OpenIMU335RI – Plastic Housing: Torque the screws to 2.37 N-m (21 inch-pounds).

OpenIMU335RI – Metal Housing: Torque the screws to 5.0 N-m (44.25 inch-pounds).

3 Theory of Operation

The OpenIMU335RI can be configured by the user as an IMU, a VG-AHRS or an INS system. At present the VG application is available. When configured as a dynamic inclination (VG) sensor it provides dynamic pitch and roll, 3D linear acceleration, and 3D estimated rate measurement data (Pitch and Roll rates are bias corrected, Yaw is not).

Figure 3 shows the OpenIMU335RI Series hardware block diagram. At the core of the OpenIMU335RI Series is a rugged 6-DOF (Degrees of Freedom) MEMS inertial sensor cluster that is common across all members of the OpenIMU335RI Series. The 6-DOF MEMS inertial sensor cluster includes three axes of MEMS angular rate sensing and three axes of MEMS linear acceleration sensing. These sensors are based on rugged, field proven silicon micromachining technology. Each sensor within the cluster is individually factory calibrated using ACEINNA's automated manufacturing process. Sensor errors compensated for are temperature bias, scale factor, non-linearity and misalignment effects using a proprietary algorithm from data collected during manufacturing. Accelerometer and rate gyro sensor bias shifts over temperature (-40 °C to +85 °C) are compensated and verified using calibrated thermal chambers and precision rate tables.

The acceleration sensors have a full-scale range (FSR) of ±78 ms^{-2}, and the rate sensors have a FSR of 400 degrees/s. The large FSR ensures that the sensors are not over-ranged in the application. Both the acceleration and rate sensors are over sampled at 1000 Hz, and digitally processed through a 3rd order configurable low-pass filter (LPF). The acceleration and rate sensors' LPFs can be set independently. The default setting for the rate data is 25 Hz and the default setting for the acceleration data is 5 Hz. Over sampling and LPF combined together eliminate higher frequency vibration and impulse energy providing greater accuracy in high-vibration environments.

The filtered data is then corrected for temperature related errors and non-linearity by ACEINNA's proprietary compensation algorithm using data collected during ACEINNA's calibration process.

This dataset is used to generate the dynamic roll and pitch estimation and are stabilized by the using the accelerometers as a long-term gravity reference. Internally, the algorithm solves for roll and pitch at a 200 Hz rate, enabling the OpenIMU335RI to continuously maintain the dynamic roll and pitch data as well as the 3D linear acceleration and 3D estimated rate data. The Yaw estimation (RS-232 only) is a free integrating yaw angle measurement that is not stabilized by a magnetometer or compass heading. As shown in the software block diagram Figure 4, after the Sensor Calibration block, the temperature corrected, and filtered data is passed into Integration to Orientation block. The Integration to Orientation block integrates body frame sensed angular rate to orientation at a fixed 200 times per second within the OpenIMU335RI Series products. For improved accuracy and to avoid singularities when dealing with the Euler angles, a quaternion formulation is used in the algorithm to provide attitude propagation.

As also shown in the software block diagram, the Integration to Orientation block receives drift corrections from the Extended Kalman Filter (EKF) or Drift Correction Module. In general, rate sensors and accelerometers suffer from bias drift, misalignment errors, acceleration errors (g-sensitivity), nonlinearity (square terms), and scale factor errors. The largest error in the orientation propagation is associated with the rate sensor bias terms. The EKF module provides an on-the-fly calibration for drift errors, including the rate sensor bias, by providing corrections to the

Integration to Orientation block and a characterization of the gyro bias state. In the OpenIMU335RI, the internally computed gravity reference vector provides a reference measurement for the EKF when the OpenIMU335RI is in quasi-static motion to correct roll and pitch angle drift and to estimate the X and Y gyro rate bias. Because the gravity vector has no horizontal component, the EKF has no ability to estimate either the yaw angle error or the Z gyro rate bias. The OpenIMU335RI adaptively tunes the EKF feedback in order to best balance the bias estimation and attitude correction with distortion free performance during dynamics when the object is accelerating either linearly (speed changes) or centripetally (false gravity forces from turns). Because centripetal and other dynamic accelerations are often associated with yaw rate, the OpenIMU335RI maintains a low-passed filtered yaw rate signal and compares it to the turnSwitch threshold field (user adjustable). When the user platform to which the OpenIMU335RI is attached exceeds the turnSwitch threshold yaw rate, the OpenIMU335RI lowers the feedback gains from the accelerometers to allow the attitude estimate to coast through the dynamic situation with primary reliance on angular rate sensors. This situation is indicated by the softwareStatus turnSwitch status flag. Using the turn switch maintains better attitude accuracy during short-term dynamic situations, but care must be taken to ensure that the duty cycle of the turn switch generally stays below 10% during the vehicle mission. A high turn switch duty cycle does not allow the system to apply enough rate sensor bias correction and could allow the attitude estimate to become unstable.

The OpenIMU335RI algorithm has two major phases of operation. The first phase of operation is the initialization phase. During the initialization phase, the OpenIMU335RI is expected to be stationary or quasi-static in order to get a good initial estimation of the roll and pitch angles, and X, Y rate sensor bias. The initialization phase lasts less than 2 seconds, and the initialization phase can be monitored in the softwareStatus BIT transmitted by default in each RS-232 measurement packet. After the initialization phase, the OpenIMU335RI operates in the dynamic mode to continuously estimate and correct for roll and pitch errors, as well as to estimate X and Y rate sensor bias.

If a user wants to reset the algorithm or re-enter the initialization phase, sending the algorithm reset command, 'AR', will force the algorithm into the reset phase.

The OpenIMU335RI outputs digital measurement data over the CAN or RS-232 port at a selectable fixed rate (100, 50, 25, 20, 10, 5 or 2 Hz) or on as requested basis.

In addition to the configurable baud rate, packet rate, axis orientation, and sensor low-pass filter settings, the OpenIMU335RI provides additional advanced settings, which are selectable for tailoring the OpenIMU335RI to a specific application requirement.

Figure 3 OpenIMU335RI Series Hardware Block Diagram

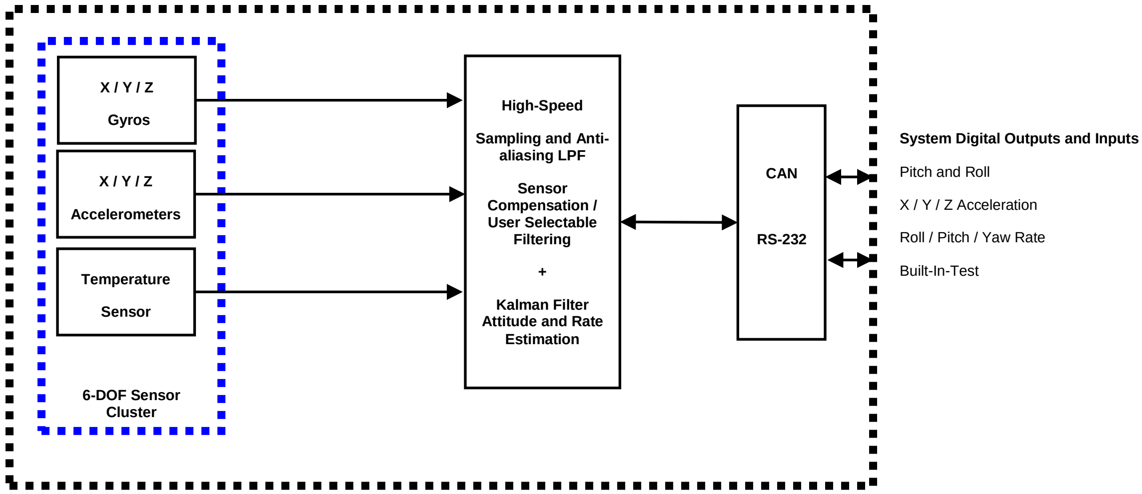


Figure 4 shows the software block diagram. As shown in the software block diagram, the OpenIMU335RI has a unit setting and profile block which configures the algorithm to user and application specific needs. This feature is one of the more powerful features in the OpenIMU335RI architecture as it allows the device to work in a wide range of commercial applications by settings different modes of operation for the OpenIMU335RI.

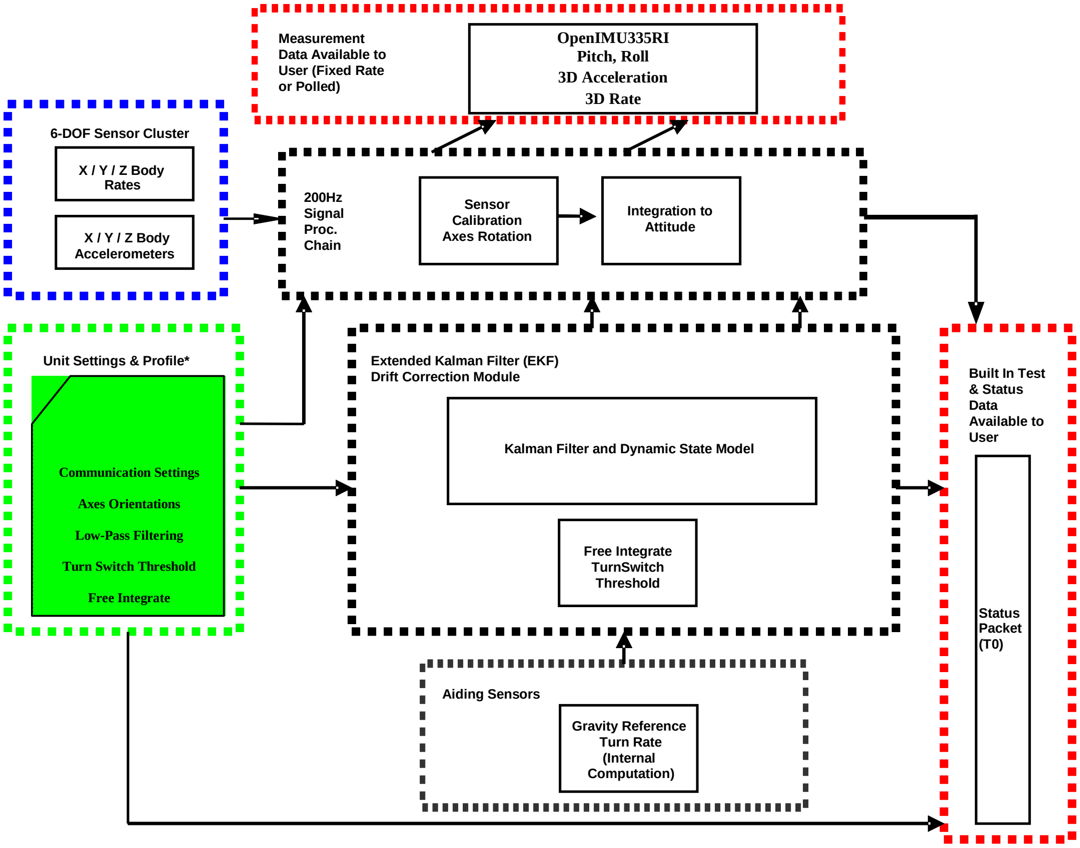


Figure 4 OpenIMU335RI Series Software Block Diagram

The common aiding sensor for the drift correction for the attitude (i.e., roll and pitch only) is a 3-axis accelerometer. This is the default configuration for the OpenIMU335RI products.

3.1 Sensor fault detection

New for the OpenIMU335RI is the incorporation of triple-redundant accelerometer and gyro sensor clusters, which is the basis for a built-in sensor fault detection mechanism. The fault detection routine incorporated into the firmware continually monitors the output of the three sensor chips. If any one of the three sensors disagrees significantly from the other two for a duration greater than the Fault Tolerant Time Interval (FTTI), it is deemed faulty, and removed from the solution. The OpenIMU335RI will continue to work normally using the remaining sensors. The fault detection logic is reset on the next power cycle, including all three sensors in the solution until/unless it is judged faulty again. The Fault Tolerant Time Interval (FTTI) for such a failure is set to 300 ms, however the actual detection time could be changed if needed. Contact the factory for more information.

3.2 OpenIMU335RI Series Default Coordinate System

The OpenIMU335RI Series Inertial System default coordinate systems are shown in Figure 5. As with many elements of the OpenIMU335RI Series, the coordinate system is configurable with either Navigation Studio or by sending the appropriate serial commands over the CAN or RS-232 port. This section of the manual describes the default coordinate system settings of the OpenIMU335RI Series when it leaves the factory.

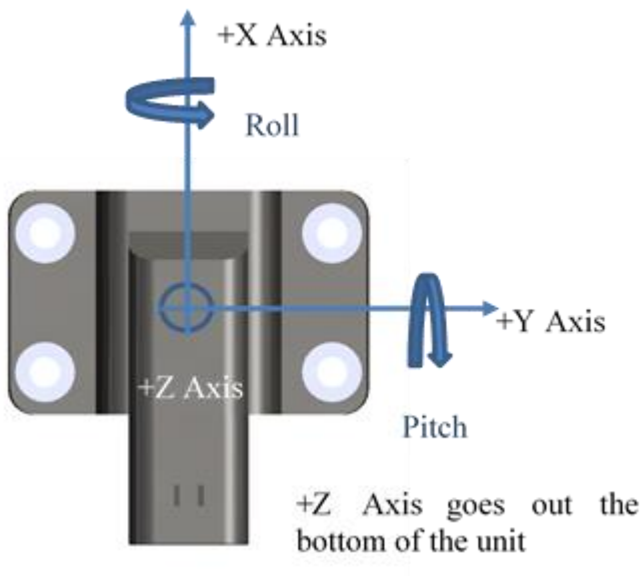


Figure 5. OpenIMU335RI Default Coordinate Frame

The axes form an orthogonal SAE right-handed coordinate system. Acceleration is positive when it is oriented towards the positive side of the coordinate axis. For example, with a OpenIMU335RI Series product sitting on a level table, it will measure zero g along the x and y-axes and -1 g along the z-axis. Normal Force acceleration is directed upward, and thus will be defined as negative for the OpenIMU335RI Series z-axis.

The angular rate sensors are aligned with these same axes. The rate sensors measure angular rotation rate around a given axis. The rate measurements are labeled by the appropriate axis. The direction of a positive rotation is defined by the right-hand rule. With the thumb of your right hand pointing along the axis in a positive direction, your fingers curl around in the positive rotation direction. For example, if the OpenIMU335RI Series product is sitting on a level surface and you rotate it clockwise on that surface, this will be a positive rotation around the z-axis. The x and y-axis rate sensors would measure zero angular rates, and the z-axis sensor would measure a positive angular rate.

Pitch is defined positive for a positive rotation around the y-axis (pitch up). Roll is defined as positive for a positive rotation around the x-axis (roll right). Yaw is defined as positive for a positive rotation around the z-axis (turn right).

"The angles are defined as standard Euler angles using a 3-2-1 system. To rotate from the earth-level frame to the body frame, yaw first, then pitch, and then roll.

3.2.1 Axis Orientation Settings

The OpenIMU335RI gives users the ability to set the axes orientation by selecting which axis aligns with the base axes as well as the sign. The only constraint is the axes must conform to a right-hand definition. The specific selections are provided in Table 45. The default setting is (+Ux, +Uy, +Uz).

Refer to CAN Protocol Section 5.1.5.6 for instructions on changing the orientation for your application over the CAN bus.

Refer to RS-232 Protocol Section 7.2.2 for instructions on changing the orientation for your application over the RS-232 Port.

3.3 Digital Filter

There are two Independent User filters available for filtering the accelerometer and the rate-sensor signals. The Filters are 2nd order Butterworth filters that can be set to 50, 40, 25, 20, 10, and 5 Hz cutoff frequencies. One setting applies to all three of a sensor's axes.

Acceleration sensor Filter default is 5 Hz. Rate Sensor default is 25 Hz.

3.3.1 Acceleration Filter Settings

Decreasing the accelerometer filter cutoff frequency will reduce transmission of the accelerometer noise to the algorithm and enable the system to better estimate roll and pitch angles (as well as rate-sensor bias) under noisy idle-conditions (such as vibration caused by engine noise). The filter will not have a large effect on the estimate of the roll and pitch angles during motion, as the role of the accelerometer is reduced during motion and angles are estimated by integrating the rate-sensor signal.

Refer to CAN Protocol Section 5 for instructions on changing the acceleration sensor filter for your application over the CAN bus.

Refer to RS-232 Protocol Section 6 for instructions on changing the acceleration sensor filter for your application over the RS-232 Port.

3.3.2 Rate Sensor Filter Settings

Decreasing the rate-sensor filter cutoff frequency will reduce transmission of the vibrational noise to the algorithm and enable the system to generate less noisy roll and pitch angle estimates under noisy conditions (such as vibration caused by engine noise). However, very low cutoff frequency settings can increase lag in the signal, which may affect system performance. Settings must be made based on system requirements.

Refer to CAN Protocol Section 5.1.5.5 for instructions on changing the rate sensor filter setting for your application over the CAN bus.

Refer to RS-232 Protocol Section 7.2.2 for instructions on changing the rate sensor filter for your application over the RS-232 Port

□ NOTE on Filter Settings

Why change the filter settings? Generally, there is no reason to change the low-pass filter settings on the OpenIMU335RI Series Inertial Systems. However, when a OpenIMU335RI Series product is installed in an environment with a lot of vibration, it can be helpful to reduce the vibration-based signal energy and noise prior to further processing on the signal. Installing the OpenIMU335RI in the target environment and reviewing the data with NAV-VIEW can be helpful to determine if changing the filter settings would be helpful. Although the filter settings can be helpful in reducing vibration-based noise in the signal, low pass filter settings (e.g.,<10Hz) also reduce the bandwidth of the signal, i.e., can wash out the signals containing the dynamics of a target. Treat the filter settings with caution.

4 Safety Features

OpenIMU335RI has been designed in compliance with the standard ISO13849. The design incorporates the following safety features:

- Upon startup the OpenIMU335RI performs extensive self-diagnostics. If during self-diagnostics there are serious problems detected with the sensors, the unit will cease sending out sensors data and set corresponding bits in the status register. It will also periodically send out a diagnostic message (see PGN 65362). Other diagnostic messages are available on demand (see 5.1.4.7, 5.1.4.8 and 5.1.4.9). In case of severe hardware failures, the unit will disconnect itself from the CAN bus to prevent compromising of system data traffic.

- The OpenIMU335RI constantly monitors its vital configuration/calibration data and will notify master unit of failures by setting corresponding bits in status registers and/or sending diagnostics message PGN 65362. In the event that calibration tables for specific sensors are compromised, the unit will block data from these sensors from participating in the solution.

- The OpenIMU335RI incorporates triple redundant sensors and constantly monitors validity of sensors data using proprietary fault detection mechanism. Upon detection of a fault in one sensor, the OpenIMU335RI will automatically exclude the faulty sensor from the main data processing/presentation stream but will still perform within specifications. If just one of three sensors remains healthy – unit will continue its work but will indicate performance degradation in the data packets. In case when unit detects that data from all sensors is compromised – unit will cease to transmit data and will be sending out periodic diagnostic messages. Contact the factory for more information about fault detection mechanism and its configuration.

- The OpenIMU335RI hardware is designed such that it will be automatically disconnected from CAN bus when its internal core or power supply are compromised.

- The OpenIMU335RI will report degradation of its health – power overconsumption, power supply failure, or unexpected delays in data processing by setting bits in status registers and sending out diagnostics messages.

- Upon restart, the OpenIMU335RI will report the cause of the restart event by setting specific bits in status registers and/or sending out diagnostic message.

5 CAN Port Interface Definition

The CAN interface of MTLT supports the CAN protocol version 2.0B. It has been designed to manage high rates of incoming messages efficiently and meets the priority requirements for transmit message.

OpenIMU335RI supports baud rates of 250kbps, 500Kbps, and 1Mbps, and supports automatic baudrate detection (see section 5.1.3.2).

5.1 SAEJ1939

The OpenIMU335RI supports CAN's higher layer protocol, SAEJ1939, managing the communication within network. J1939 is a set of standards defined by SAE. The physical layer (J1939/11) describes the electrical interface to the bus. The data link layer (J1939/21) describes the rules for constructing a message, accessing the bus, and detecting transmission errors. The application layer (J1939/71 and J1939/73) defines the specific data contained within each message sent across the network.

J1939 uses the 29-bit identifier defined within the CAN 2.0B protocol shown in Table 4.

Table 4. Structure of a 29-bit identifier

Priority	Reserved	Data Page	PDU Format	PDU Specific	Source Address
3 bits	1 bit	1bit	8 bits	8 bits	8 bits

The first three bits of the identifier are used for controlling a message's priority during the arbitration process. A value of zero has the highest priority. Higher priority values are typically given to high-speed control messages.

The next bit of the identifier is reserved for the future use and should be set to zero for transmitted message.

The next bit of the identifier is the data page selector.

The PDU format determines whether the message can be transmitted with a destination address or if the message is always transmitted as a broadcast message.

The interpretation of the PDU specific field changes based on the PF value:

- If the PF is between 0 and 239, the message is addressable, and the PS field contains a destination address
- If the PF is between 240 and 255, the message can only be broadcast. The PS field contains a Group Extension.

The term Parameter Group Number (PGN) is used to refer to the value of the Reserve bit, DP, PF, and PS fields combined into a single 18-bit value.

The last 8 bits of the identifier contain the address of the device transmitting the message.

5.1.1 *ECU's Address*

Each device on the network will be associated with one address. The device address defines a specific communications source or destination for messages. The default CAN address for the OpenIMU335RI is 128.

The allowed range of unit addresses on the CAN bus is from 128 to 247.

Address 255 is reserved as a global address for broadcast and address 254 is reserved as the "null address" used by devices that have not yet claimed an address or failed to claim an address.

It's possible to assign arbitrary unit address from serial interface as well as from CAN bus.

5.1.2 *Address Claim*

In general, most ECU addresses on the CAN bus are pre-assigned and used immediately upon power up. In order to permit J1939 to accommodate future devices and functions, a procedure has been specified for dynamically assigning addresses. This procedure is defined according to J1939. Each device must announce its own address.

In the case of the OpenIMU335RI, the device will announce its address once upon initialization and retain it unless/until another unit with the same address challenges it. The format of the Address Claim message is defined by J1939, and described below:

Priority	**PGN**	**PF**	**PS**	**SA**	**Payload**
6	60928	238	DA	128*	8 bytes

* By default, the OpenIMU335RI will claim the address 128 (0x80), and this address can be re-programmed using the "Unit Behavior" command.

The Address Claim message payload contains 8 bytes as described in the Table below. This payload can be considered as 64-bit unsigned integer representing the ECU Name. In case of a conflict, the device which has the smaller value of this integer wins the conflict, and the device which lost the competition will switch to the next higher available address and repeat the address claim process.

The MTLT 335 will also send out address claim message if another node on the network explicitly requests it.

Table 5. OpenIMU335RI Address Claim Payload

Bits	Contents	Value
63	Arbitrary Address	1
62:60	Industry Group	0
59:56	Vehicle System Instance	0
55:49	System Bits	0
48	Reserved	0

47:40	Function Bits	131
39:35	Function Instance	Variable. Refer to Appendix C
34:32	ECU Bits	0
31:21	Manufacturer code	823
20:0	ID bits	Unit SN – least significant bits

The OpenIMU335RI firmware supports dynamic address conflict resolution. If the device address is already used by another ECU on the same network, the device will attempt to claim another address. The device transmits the address claim message once on startup, and thereafter only in case of an address conflict.

Refer to appendix C for information about using the function instance bits to detect a unit's location on the wiring harness. This can be useful in systems with more than one IMU on the CANbus.

5.1.3 Baud Rate

5.1.3.1 Manual Baud Rate Configuration

The J1939 network is intended to be a single, linear, shielded twisted pair of wires running around the vehicle to each ECU. The default data rate of J1939 is 250Kbps. A typical message containing 8 bytes is 128 bits long, which in time is approximately 500 microseconds. OpenIMU335RI also supports lower speed, 125Kbps and higher speeds, 500Kbps and 1000Kbps.

The baud rate of MTLT is configurable through the RS-232 interface using NAV-VIEW.

5.1.3.2 Auto Baud Detection

Autobaud detection supports 250kb/s, 500kb/s and 1000kb/s. The OpenIMU335RI is configured at the factory for a baud rate of 250kbps. This value is stored in non-volatile memory. When power is applied (or any time the power is cycled), the unit will select a starting baud rate, which will be the last baud rate stored in non-volatile memory and will put itself into Silent Baud Discovery Mode.

In this mode, the unit will listen silently, without transmitting or asserting the Acknowledge bit, and uses an automated error counter mechanism incorporated into its CAN peripheral controller. It checks the error counter at 10ms intervals. At each 10ms interval, the unit checks both the error counter, and the CAN FIFO buffer for a valid 29-bit message identifier. One of three cases can occur:

1. If there is a valid 29-bit message identifier, and the error counter is at zero, the unit will lock to the current baud rate.
2. If the error counter crosses a programmable threshold (default = 20 counts), or if the error counter increases for three sequential 10ms periods, then the unit will switch to the next supported baud rate, clear the message FIFO buffer, and continue monitoring at this new baud rate.
3. Otherwise, the unit will continue monitoring at 10ms intervals, at the current baud rate, until case 1 or case 2 occurs.

Note that an 11-bit standard frame message will not be considered valid while the unit is in Silent Baud Discovery Mode.

The list of supported baud rates is 250kbps, 500kbps, and 1Mbps, and the OpenIMU335RI will cycle indefinitely in that order, beginning with the last value stored in non-volatile memory, until the baud rate is successfully detected.

In normal operation (after Baud Discovery Mode has ended), the unit will maintain a counter which keeps a running count of CAN errors. Each time it detects an RX error, it increments the error counter by +1. Each time it detects a TX error, it increments the error counter by +8; each time it successfully transmits or receives an error free message, it decrements the error counter by 1. If the error counter reaches 255, the unit will go to the "bus off" state and enter silent Baud Discovery Mode (as described above).

5.1.4 Get Commands

Get commands are used for another ECUs on the network to collect the corresponding message from OpenIMU335RI. All of the commands are formed by a Request message of SAEJ1939-21 PGN number 60159. The format of the request message payload provided in Table 6.

Table 6. The format of request message payload

Byte 0	Byte 1	Byte 2
Requested PGN (LSB)	Requested PGN	Requested PGN (MSB)

NOTE: In the previous generation MTLT305, Byte 0 and Byte 2 were reversed. This has been corrected in the OpenIMU335RI, but there is a configuration bit available to make it backward compatible with the MTLT305 (See Unit Behavior Settings).

NOTE: Payload of all responses for Get commands is padded to 8 bytes to make messages look uniform on the bus.

5.1.4.1 Firmware Version

OpenIMU335RI responds to a J1939 request PGN 65242 packet with message, containing 5-byte payload. Table 7 shows the format of firmware version packet.

Table 7. The format of firmware version response packet

Priority	PGN	PF	PS	SA	Payload
6	65242	254	218		5 bytes

The payload contains information about current firmware version in OpenIMU335RI. See Table 8.

Table 8. Firmware version response payload

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

Major	Minor	0	0	Patch	Part Number Low	Part Number Middle	Part Number High

5.1.4.2 Rate of periodic data packets

Upon receiving a request for parameter with PGN 65365 (current rate value of periodic packets) from another ECU on the network, OpenIMU335RI responds with a packet, containing a 2-byte payload. Table 9 shows the format of the response.

Table 9. The format of packet rate divider response packet

Priority	PGN	PF	PS	SA	Payload
6	65365	255	85		2 bytes

The payload consists of the address of requesting unit and current rate value of periodic packets in OpenIMU335RI. See Table 10.

Table 10. Packet rate response payload

Byte 0	Byte 1
Destination Address	Packet rate (see Table 38)

5.1.4.3 Periodic Data Packets Control Settings

Upon receiving request for parameter with PGN 65366 (periodic data packets control) from another ECU on the network the OpenIMU335RI responds with a packet, containing 4-byte payload. Table 11 shows the format of the response.

Table 11. The format of data packet types response packet

Priority	PGN	PF	PS	SA	Payload
6	65366	255	86		4 bytes

The response payload consists of address of requesting unit and bitmasks which represents currently enabled periodic packet types and priorities of these packets See Table 12.

Table 12. Packet types response payload

Byte 0	Byte 1	Byte 2	Byte 3
Destination Address	Packet types (LSB) (see Table 40)	Packet types (MSB) (see Table 40)	Priorities for ACCS/ARI/SSI messages (see Table 41)

It is also possible to request a specific data packet from OpenIMU335. The request payload should contain PGN of the specific data packet (see 5.1.6)

5.1.4.4 Active Digital Filters

Upon receiving a request for parameter with PGN 65367 (currently active digital filters) from another ECU on the network, OpenIMU335RI responds with a packet, containing a 3-byte payload. Table 13 shows the format of the response.

Table 13. The format of digital filters response packet

Priority	PGN	PF	PS	SA	Payload
6	65367	255	87		3 bytes

The response payload consists of address of requesting unit and currently active filters for accelerometers and rate sensors data. See Table 14.

Table 14. Active digital filters response payload

Byte 0	Byte 1	Byte 2
Destination Address	Rate sensor filter cutoff frequency (see section 5.1.5.5)	Accelerometer filter cutoff frequency (see section 5.1.5.5)

5.1.4.5 Current unit orientation

Upon receiving a request for parameter with PGN 65368 (current unit orientation settings) from another ECU on the network, OpenIMU335RI responds with a packet, containing a 3-byte payload. Table 15 shows the format of the response.

Table 15. The format of unit orientation response packet

Priority	PGN	PF	PS	SA	Payload
6	65368	255	88		3 bytes

The response payload consists of address of requesting unit and current unit orientation settings. See Table 16.

Table 16. Unit orientation response payload

Byte 0	Byte 1	Byte 2
Destination Address	Orientation (MSB) (see **5.1.5.6**)	Orientation (LSB)

5.1.4.6 ECU's ID

The ID is a 64-bit long label, which gives every ECU a unique identity, following the definition in SAEJ1939-81.

Table 17 shows the format of ID response packet. Refer to the description of the address claim message (section 5.1.2) for a description of the payload.

Table 17. The format of ID response packet

Priority	PGN	PF	PS	SA	Payload

6	64965	253	197		8 bytes

5.1.4.7 *Master BIT status word*

Upon receiving a request for parameter with PGN 65364 (master BIT status request) from another ECU on the network, the OpenIMU335RI responds with a packet with a 4-bytes payload. Table 18 shows the format of the Master BIT response packet.

Table 18. The format of Master Bit response packet

Priority	PGN	PF	PS	SA	Payload
6	65364	255	84		4 Bytes

Table 19 shows the content and definition of the bits in the Master BIT status response payload.

Table 19. Master BIT response payload

Bit	Description	Value	Solution
0	MasterFail	0=normal; 1=fatal error has occurred.	Check Master BIT payload bits 1-15 for information on failure type.
1	HW Error	0=normal; 1=any of the Failure bit in Hardware BIT status is set (see Table 22) OR Over temperature bits persists for more than 5 minutes.	Reset or power cycle can be tried to recover unit. If not recovered after power cycle the unit should be replaced.
2	SW Error	0=normal; 1=any of the Error bit in Software BIT is set (see Table 20).	Reset or power cycle can be tried to recover unit. If not recovered after power cycle the unit should be replaced.
3	Configuration Error	0=normal; 1=incorrect configuration parameters detected during initialization or periodic self-test.	Unit needs to be reset by command or power cycled. If state persists after power cycle – unit considered to be damaged and needs to be replaced.
4	Calibration Error	0=normal; 1= calibration data for all three sensor chips are corrupted (see Table 20).	Unit needs to be reset by command or power cycled. If state persists after power cycle – unit considered to be damaged and needs to be replaced.
5	Accelerometer Quality Degradation	0= normal; 1 = accel sensor disagreement bit is set (see Table 20), OR Data Processing delay exceeds 5ms for more than 10 consecutive cycles, OR only one active sensor remaining OR Over Range OR Disagreement bit is set in SW BIT Status	Unit may still work normally. To recover from this condition unit needs to be reset by command or power cycled.

6	Rate Sensor Quality Degradation	0= normal; 1 = rate sensor disagreement bit is set (see Table 20), OR Data Processing delay exceeds 5ms for more than 10 consecutive cycles, OR only one active sensor remaining OR Over Range OR Disagreement bit is set in SW BIT Status	Unit may still work normally. To recover from this condition unit needs to be reset by command or power cycled.
7	Forced Restart	0 = normal; 1 = forced restart (restart caused by watchdog, brownout or transmit queue congestions)	
8	Application CRC Error	0 = normal; 1 = Application CRC error detected	Will be sent from Bootloader periodically with interval of 1 second
9	Tx Overflow Error	0 = normal; 1 = Tx Overflow occurred 10 consecutive cycles	This bit resets when overflow condition is cleared OR the unit will restart automatically when this error persists for more than 20 seconds.
10 - 15	Reserved		
16 - 31	Application CRC	Contains application CRC value for reference	

NOTE: Master BIT status word will be also periodically broadcasted on the CAN bus in case if unit triggered fatal internal error (right upon occurrence and every second after that). Bus master then can send request for additional status information.

5.1.4.8 Software BIT status

Upon receiving a request for parameter with PGN 65363 (Software BIT status request) from another ECU on the network, the OpenIMU335RI will respond with a packet with 4-bytes payload. Table 20 shows the format of the Software BIT status response packet.

Table 20. The format of Software BIT status response

Priority	PGN	PF	PS	SA	Payload
6	65363	255	83		4 Bytes

Note: The PS value for this command can be changed using the command "Bank of PS numbers" PS Bank 0.

Table 21 shows the content and definition of the packet.

Table 21. Software BIT status payload fields

Bit	Description	Value	Solution

0	Stack Overflow Error	0=normal; 1= stack is 95% full	
1	Algorithm Error	0=normal, 1=error (initialization took more than 5 seconds).	
2	Initialization Status	0=normal; 1= Algorithm is in the initialization stage and unit is not ready. Bit will be cleared after initialization is complete (~2s).	
3	Reserved		
4:6	Accelerometer Over Range Status	0=normal; 1= Indicates an over range event, (acceleration exceeds 7.95g). It is cleared when condition is removed. This condition will set SW Error flag in Master BIT status if Sensor Over Range persists more than 4 data processing cycles.	Algorithm can be reset using command which can be sent over CAN bus (PGN 65360) or by power cycling unit. If condition persists when system is not moving, after power cycle, the unit is considered faulty and should be replaced.
7:9	Rate Sensor Over Range Status	0=normal; 1= Indicates an over range event, (rotation exceeding 400 dps). It is cleared when condition is removed. EKF will converge to correct solution or Algorithm can be reset. The SW will set SW Error flag in Master BIT status if Sensor Over Range persists more than 4 data processing cycles.	Algorithm can be reset using command which can be sent over CAN bus (PGN 65360) or by power cycling unit. If condition persists when system is not moving, after power cycle, the unit is considered faulty and should be replaced.
10	Configuration Error	0=normal, 1=incorrect configuration parameters detected during initialization or periodic self-test.	Unit needs to be reset or power cycled. If state persists after power cycle – unit considered to be faulty and needs to be replaced.
11	Calibration Data Status - Chip 0	0=normal, 1= non-recoverable error of unit calibration data for sensor chip 0 (bad CRC) detected during initialization or periodic self-test. In this case unit will automatically exclude affected sensor from the solution	Unit needs to be reset or power cycled. If state persists after power cycle, and degradation of unit performance is also detected – unit considered to be damaged and needs to be replaced.
12	Calibration Data Status - Chip 1	0=normal, 1= non-recoverable error of unit calibration data for sensor chip 1 (bad CRC) detected during initialization or periodic self-test. In this case unit will automatically exclude affected sensor from the solution	Unit needs to be reset or power cycled. If state persists after power cycle, and degradation of unit performance is also detected – unit considered to

			be damaged and needs to be replaced.
13	Calibration Data Status - Chip 1	0=normal, 1= non-recoverable error of unit calibration data for sensor chip 2 (bad CRC) detected during initialization or periodic self-test. In this case unit will automatically exclude affected sensor from the solution	Unit needs to be reset or power cycled. If state persists after power cycle, and degradation of unit performance is also detected – unit considered to be damaged and needs to be replaced.
14	Accelerometer Sensor 0 Status	0= normal; 1 = accelerometer sensor 0 is voted out due to detected HW or SW fault.	Unit may still work normally until degradation of performance detected. To recover from this condition unit needs to be reset or power cycled.
15	Accelerometer Sensor 1 Status	0= normal; 1 = accelerometer sensor 1 is voted out due to detected HW or SW fault.	Unit may still work normally until degradation of performance detected. To recover from this condition unit needs to be reset or power cycled.
16	Accelerometer Sensor 2 Status	0= normal; 1 = accelerometer sensor 2 is voted out due to detected HW or SW fault	Unit may still work normally until degradation of performance detected. To recover from this condition unit needs to be reset or power cycled.
17	Rate Sensor 0 Status	0= normal; 1 = rate sensor 0 is voted out due to detected HW or SW fault	Unit may still work normally until degradation of performance detected. To recover from this condition unit needs to be reset or power cycled.
18	Rate Sensor 1 Status	0= normal; 1 = rate sensor 1 is voted out due to detected HW or SW fault	Unit may still work normally until degradation of performance detected. To recover from this condition unit needs to be reset or power cycled.
19	Rate Sensor 2 Status	0= normal; 1 = rate sensor 2 is voted out due to detected HW or SW fault	Unit may still work normally until degradation of performance detected. To recover from this condition unit needs to be reset or power cycled.

20	Accel Sensor Disagreement Error	0= normal; 1 = one of the accelerometer sensors was removed from solution due to detected HW or SW fault AND remaining two accelerometer sensors have disagreement.	The fault detection state is reset by power cycling, and the error state may be cleared if the fault was caused by a temporary environmental condition. If condition persists after power cycling, unit is faulty.
21	Rate Sensor Disagreement Error	0= normal; 1 = one of the rate sensors is voted out due to detected HW or SW fault AND remaining two accelerometer sensors have disagreement.	The fault detection state is reset by power cycling, and the error state may be cleared if the fault was caused by a temporary environmental condition. If condition persists after power cycling, unit is faulty.
22:24	Last Reset Status	000 = Power – On reset 001 = Software (self-reset) 100 = Watchdog reset 101 = Brown-out reset 110 = Transmit queue congestion Reset bit in Master BIT status will be set when cause of last reset is Watchdog or Brown-out reset	
25	Data Processing Over Run Status	0 = no overrun 1 = Data Processing Delay exceeded 5ms. SW Error bit and Accel/Rate Sensor Quality Degradation bits will be set in Master BIT status when Over Run persist for more than 10 data processing cycles.	
26	Turn Switch Status	0=off, 1=yaw rate greater than Turn Switch threshold	
27	Algorithm Mode Status	0 = low gain mode, 1 = high gain mode	
28	Transmit queue overflow Error	0=normal, 1 = Unit transmit queue overflow occurred due do the errors or CAN bus congestion. If this condition persists more than 10 cycles, Tx Overflow Error bit is set in Master BIT Status.	This bit resets when overflow condition is cleared

5.1.4.9 HW BIT status

Upon receiving a request for parameter with PGN 6536 (HW BIT status request) from another ECU on the network, the OpenIMU335RI responds with a packet with 2-bytes status message.

Table 22 shows the format of unit HW status response message. Table 23 shows the contents and definition of unit status response payload.

Table 22. The format of HW BIT status message

Priority	PGN	PF	PS	SA	Payload
6	65362	255	82		2 bytes

Note: The PS value for this command can be changed using the "Bank 0" command.

Table 23. Hardware BIT status payload fields

Bit	Description	Value	Solution
0	Power consumption Error	0=normal, 1=unit consumes excessive amount of power for more than 1 minute.	Check power supply
1	External power supply Error	0=normal, 1=external power supply voltage is not within specified thresholds for more than 1 minute.	Check power supply
2	Internal power supply Error	0=normal, 1=Overvoltage or under voltage on internal power supply for more than 1 minute.	Check power supply
3	Over Temperature Environment	0=normal, 1=MCU temperature exceeds allowed limit Set HW Error in Master BIT Status when temp is 5 deg. over the max temp limit or 5 deg. below min temp limit, for more than 5 minutes.	Cool down the unit
4	Over Temperature Chip 0	0=normal, 1=Chip 0 temperature exceeds allowed limit. Set HW Error in Master BIT Status when temp is 5 deg. over the max temp limit or 5 deg. below min temp limit, for more than 5 minutes, on all three sensors.	

5	Over Temperature Chip 1	0=normal, 1=Chip 1 temperature exceeds allowed limit. Set HW Error in Master BIT Status when temp is 5 deg. over the max temp limit or 5 deg. below min temp limit, for more than 5 minutes, on all three sensors.	
6	Over Temperature Chip 2	0=normal, 1=Chip 2 temperature exceeds allowed limit. Set HW Error in Master BIT Status when temp is 5 deg. over the max temp limit or 5 deg. below min temp limit, for more than 5 minutes, on all three sensors.	
7	Sensor Communication Status – Chip 0	0=normal, 1= communication with sensor chip 0 failed on startup or during runtime self-test. Causes exclusion of the sensor chip from the solution.	Unit needs to be reset by command or power cycled. If state persists after power cycle – unit considered to be damaged and needs to be replaced.
8	Sensor Communication Status – Chip 1	0=normal, 1= communication with sensor chip 1 failed on startup or during runtime self-test. Causes exclusion of the sensor chip from the solution.	Unit needs to be reset by command or power cycled. If state persists after power cycle – unit considered to be damaged and needs to be replaced.
9	Sensor Communication Status – Chip 2	0=normal, 1= communication with sensor chips 2 failed on startup or during runtime self-test. Causes exclusion of the sensor chip from the solution.	Unit needs to be reset by command or power cycled. If state persists after power cycle – unit considered to be damaged and needs to be replaced.
10 : 15	Reserved	Reserved	

5.1.4.10 Unit behavior

Upon receiving a request for parameter with PGN 65369 (current unit behavior configuration) from another ECU on the network, the OpenIMU335RI responds with a packet, containing 2-byte payload. Table 24 shows the format of the response.

Table 24. The format of unit behavior response packet

Priority	PGN	PF	PS	SA	Payload
6	65369	255	89		2 bytes

Note: the PS value for this command can be changed using the "Bank 1" command in case of PGN conflict.

The response payload consists of the address of requesting unit (Byte 0) and the current unit behavior bitmask. When value of specific bit equal to 1 it means that corresponding option is enabled. See Table 25.

Table 25. The format of the unit behavior response payload

Byte Number	Selected Unit Behavior type
0	Destination Address
1	Unit behavior bitmask: Bit 0 – restart on over range (default=0) Bit1 – dynamic motion (default=1) Bit 2 – uncorrected rates in ARI message (default=0) Bit 3 – Change sequence of ARI and ACCS messages from X, Y, Z (like MTLT305) to Y, X, Z (J1939) in all ARI and ACCS messages (default=1) Bit 4 – autobaud detection mode (default=1) Bit 5 – CAN termination resistor control (if available) (default=0) Bit 6 – NWU accelerometer frame (all acceleration messages) (default=1) Bit 7 – Raw (unfiltered) acceleration signal used to detect linear acceleration in EKF algorithm (default=1)
2	Bit 0 – Raw (uncorrected) angular rate used to predict acceleration in EKF (default=0) Bit 1 – Swap Byte 2 and Byte 0 in a GET Message Request payload for backward compatibility with MTLT305, see Section 5.1.4 (default=0) Bits 2:6 – Reserved Bit 7 – VG algorithm is enabled (default=1)

5.1.4.11 Algorithm Control

The OpenIMU335RI responds to a J1939 request PGN 65371 (Algorithm control) packet with a message containing an 8-byte payload. Table 26 shows the format of this Algorithm Control response.

Table 26. The format of the algorithm control response packet

Priority	PGN	PF	PS	SA	Payload
6	65371	255	91		8 bytes

Table 27. The format of the Algorithm control response payload

Byte	Description	Value	Comment
0	Address of Unit	Destination Address	
1	Reserved		
2	Limit for rate integration time	LSB	Value in milliseconds From 10 to 10000
3	Limit for rate integration time	MSB	
4	Limit for accelerometer switch delay	LSB	Value in milliseconds From 10 to 10000
5	Limit for accelerometer switch delay	MSB	
6	Coefficient of reduced Q	LSB	Scale 0.0001 per lsb From 0.0001 to 1
7	Coefficient of reduced Q	MSB	

5.1.4.12 Aiding Signal Configuration

The OpenIMU335RI responds to a J1939 request PGN 65375 (Aiding Signal Configuration) packet with a message containing an 8-byte payload. Table 28 shows the format of this Aiding Signal Configuration response.

Table 28. The format of the Aiding Signal Configuration response packet

Priority	PGN	PF	PS	SA	Payload
6	65375	255	95		8 bytes

Table 29. The format of the Aiding Signal Configuration response payload

Byte	Description	Comment
0	Address of Unit	
1	Signal Source	Configured Aiding Signal Source. 0 = No aiding signal used 1 = using Odometer 2 = using vehicle accelerations signal.
2	PF value of Aiding Message	Configured Odometer or Vehicle Accelerations data message PF value
3	PS value of Aiding Message	Configured Odometer or Vehicle Accelerations data message PS value
4	Message Rate	Configured Aiding Message rate (in Hz)
5	Driving Direction Data Message PF value	Configured Driving Direction message PF Value
6	Driving Direction Data Message PS value	Configured Driving Direction message PS Value
7	Orientation Switches	Bit 0: IMU Mount Location: 0 = IMU does not experience rotation relative to chassis, such as mounted to a truck's chassis 1 = IMU can experience rotation relative to tracks, such as mounted to body of excavator Bit 1: Reserved (set to "0") Bit 2: Aiding Signal Acceleration Direction 0 = +X Axis acceleration point to vehicle forward. 1 = -X Axis acceleration point to vehicle forward.

5.1.4.13 Aiding Signal LeverArm Configuration

Refer to section 5.1.5.10 for a more complete description of the Lever Arm. The OpenIMU335RI responds to a J1939 request PGN 65376 (Aiding Signal LeverArm Configuration) packet with a message containing a 7-byte payload. Table 30 shows the format of this Aiding Signal Configuration response.

Table 30. The format of the Aiding Signal LeverArm Configuration response packet

Priority	PGN	PF	PS	SA	Payload
6	65376	255	96		7 bytes

Table 31. The format of the Aiding Signal Configuration response payload

Byte	Description	Value	Definition	Comment
0	Address of Unit		Destination Address	NA
1	LeverArmX	LSB	Reads back the current LeverArm value in units of mm, in offset binary format (see section 5.1.5.10)	Millimeter Range = [-32000, 32000] Offset value = -32000
2		MSB		
3	LeverArmY	LSB		
4		MSB		
5	LeverArmZ	LSB		
6		MSB		

As an example, for given byte values of "MSB" and "LSB", the Actual Lever Arm (in units of mm) is MSB*256 + LSB – 32,000.

5.1.4.14 DM1 Message Configuration

The OpenIMU335RI responds to a J1939 request PGN 65370 (DM1 message configuration) packet with a message containing 8-byte payload. Table 30 shows the format of the response.

Table 32. DM1 Configuration response message format

Priority	PGN	PF	PS	SA	Payload
6	65370	255	90		8 bytes

Table 33. Manual Content

Byte	Bits	Parameter name	Default Values
1	DA	Destination address	
2	2-1 4-3 6-5 8-7	Protect Lamp Status Amber Warning Lamp Status Red Stop Lamp Status Malfunction Indicator Lamp Status	00 01 00 00
3	2-1 4-3 6-5 8-7	Flash Protect Lamp Flash Amber Warning Lamp Flash Red Stop Lamp Flash Malfunction Indicator Lamp	11 11 11 11
4	8-1	SPN, 8 least significant bits of SPN	0xB3

5	8-1	SPN, second byte of SPN (most significant at bit 8)	0xF4
6	3-1	SPN, 3 most significant bits (most significant at bit 8)	0x07
7		FMI for DTC1	0x0C
8		FMI for DTC2	0x0E

5.1.5 *Set Commands*

Set commands are used for other ECUs to configure the OpenIMU335RI on the network. All the commands are broadcast messages with a destination address. The receivers will decode the payload and decide to ignore the message meant for another ECU.

5.1.5.1 *Configuration Save*

The command is issued by an ECU on the network to the destination unit who is requested to save the current configuration in RAM to EEPROM. The receiver will send a response back and notify the sender that the save command has been executed successfully or not. If Request field equals 2 unit will reset itself after sending back the response.

Table 34 shows the format of the save configuration command. Table 35 is the description of save configuration command payload.

Table 34. The Format of Save Configuration

Priority	PGN	PF	PS	SA	Payload
6	65361	255	81		3 bytes

Table 35. Save Configuration Command Payload

Byte	Description	Value
0	Request or Response	0 =Request, 1=Response, 2 = Reset
1	Address of Unit being saved	Address of destination
2	Success or failure	0=failure, 1=success

5.1.5.2 *Algorithm Reset*

The algorithm reset command is issued by an ECU on the network to the destination unit which is requested to its reset algorithm. The receiver will send a response back and notify the sender that the re-initialization of algorithm has been executed successfully or not. If Request field equals 2 unit will reset itself after sending back the response. Table 36 shows the format of the algorithm reset command.

Table 36. The format of algorithm reset command

Priority	PGN	PF	PS	SA	Payload
6	65360	255	80		3 bytes

The description of the algorithm reset data field is the same as the save configuration shown data field shown in Table 35.

5.1.5.3 Packet Rate Divider

The OpenIMU335RI broadcasts several types of data packet, as defined in Table 40. The default ODR of data packets is 100Hz. The Packet rate divider command is used to change the ODR setting. The 1st byte of the payload is the destination address, and the 2nd byte of the payload sets the new ODR. Table 37 shows the format of packet rate divider command. The values of packet rate divider are defined in

Table 38.

Table 37. The format of packet rate divider command

Priority	PGN	PF	PS	SA	Payload
6	65365	255	85		2 bytes

Table 38. Packet Rate Divider Field Definition

Byte Value	Packet broadcasting rate
0	Quiet mode
1	100Hz (default)
2	50Hz
4	25Hz
5	20Hz
10	10Hz
20	5Hz
25	4Hz
50	2Hz

5.1.5.4 Data Packets Control Settings

The user can configure the combination of output packets and priority of those output packets. The OpenIMU335RI's data packet types and default priorities of these data packets follows SAEJ1939-DA Table SPNs & PGN.

Users can choose any combination of output packets from the following data packets - slope sensor information 2 (PGN 61481), angular rate information (PGN 61482) acceleration sensors (PGN 61485), high resolution acceleration sensors (PGN 65389), high resolution rate sensors (PGN 65387), slope sensor information (PGN 61459).

The 1st byte of the payload is the destination address. 2nd byte of the payload describes the packet type to be set. 3rd payload byte is reserved, any value written to the payload byte 3 will be ignored. 4th byte of the payload contains bitmask of priorities which can be assigned to specific packet type.

5th byte contains bitmask which enables change of the priority of specific packet. Table 39 shows the format of data packet type command.

Table 39. The format of Data Packet Type command

Priority	PGN	PF	PS	SA	Payload
6	65366	255	86		5 bytes

The 2nd byte is a bitmask and used to select which data messages are to be transmitted. To select specific message type set specific bit to 1. Any combination of messages can be selected for transmission. See Table 40.

Table 40. Data Packet Type Field Definition (2nd byte)

Bit Number	Selected Data Packet Type
0	SSI2
1	Angular Rate
2	Acceleration
3	HR Angular Rate
4	HR Acceleration
5	SSI

The 4th byte is a bitmask and used to select priority of specific message types.

The 5th byte is a bitmask and used to enable change of priority of specific message types. Priority can be changed for any or all messages at the same time.

Table 41. Priority of Data Packets (4th byte)

Bit Number	Bit fields assignment
1:0	Angular Rate & HR Angular Rate Packets Priority (0 - 3)[1]
3:2	Acceleration & HR Acceleration Packets Priority (0 - 3)[1]
5:4	SSI & SSI2 Packets Priority (0 - 3)[1]
7:6	**Reserved**

Table 42. Priority Change enable mask (5th byte)

Bit Number	Bit fields assignment
1:0	Enable of changing Angular Rate & HR Angular Rate Packets Priority (11 – enable, other – disable)
3:2	Enable of changing Acceleration & HR Acceleration Packets Priority

[1] Priority of both the data packets will be changed when these bits are configured

	(11 – enable, other – disable)
5:4	Enable of changing SSI & SSI2 Packets Priority (11 – enable, other – disable)
7:6	**Reserved**

5.1.5.5 Digital Filter

The OpenIMU335RI contains configurable 2nd order low pass filters. Users can change the frequencies of low-pass filters applied to rate sensor or accelerometer. The supported frequencies settings are: 5, 10, 20, 25, 40, and 50 Hz. The frequencies are changed by the command issued by another ECU on the network.

The 1st byte of the payload is the destination address, and the 2nd and 3rd bytes are the values of low-pass filter frequency to be set to the rate sensor and accelerometer respectively. Table 30 shows the format of digital filter selection command.

1st byte: destination address

2nd byte: sets low pass cutoff for rate sensor. Cutoff Frequency choices are 0, 5, 10, 20, 25, 40 and 50Hz

3rd byte: sets low pass cutoff for accelerometer. Cutoff Frequency choices are 0, 5, 10, 20, 25, 40 and 50Hz

Table 43. The format of digital filter change command

Priority	PGN	PF	PS	SA	Payload
6	65367	255	87		3 bytes

5.1.5.6 Orientation

Users may change the coordinate system of OpenIMU335RI via the command issued by another ECU on the network. The 1st byte of payload is the destination address and the next 2 bytes (MSB first) define the orientation to be applied. Table 44. shows the format of unit orientation control command.

Table 44. The format of the unit orientation command

Priority	PGN	PF	PS	SA	Payload
6	65368	255	88		3 bytes

Table 45. Possible values of the orientation

Orientation Field Value	*X Axis*	*Y Axis*	*Z Axis*
0x0000	+Ux	+Uy	+Uz
0x0009	-Ux	-Uy	+Uz
0x0023	-Uy	+Ux	+Uz

Orientation Field Value	X Axis	Y Axis	Z Axis
0x002A	+Uy	-Ux	+Uz
0x0041	-Ux	+Uy	-Uz
0x0048	+Ux	-Uy	-Uz
0x0062	+Uy	+Ux	-Uz
0x006B	-Uy	-Ux	-Uz
0x0085	-Uz	+Uy	+Ux
0x008C	+Uz	-Uy	+Ux
0x0092	+Uy	+Uz	+Ux
0x009B	-Uy	-Uz	+Ux
0x00C4	+Uz	+Uy	-Ux
0x00CD	-Uz	-Uy	-Ux
0x00D3	-Uy	+Uz	-Ux
0x00DA	+Uy	-Uz	-Ux
0x0111	-Ux	+Uz	+Uy
0x0118	+Ux	-Uz	+Uy
0x0124	+Uz	+Ux	+Uy
0x012D	-Uz	-Ux	+Uy
0x0150	+Ux	+Uz	-Uy
0x0159	-Ux	-Uz	-Uy
0x0165	-Uz	+Ux	-Uy
0x016C	+Uz	-Ux	-Uy

5.1.5.7 Unit behavior

The user can choose any combination of OpenIMU335RI behaviors. Table 46 shows the format of unit behavior command.

Table 46. The format of Unit Behavior command

Priority	PGN	PF	PS	SA	Payload
6	65369	255	89		6 bytes

The first payload byte is destination address.

The next 3 payload bytes used to enable/disable specific unit behavior settings. To enable a specific behavior, the corresponding bit in byte 1 or 2 should be set to 1. To disable a specific behavior, the corresponding bit in byte 3 or 4 should be set to 1. The "disable" bits will override any "enable" bits if sent in the same message. This method allows individual behaviors to be turned on or off, without the need to "remember" the other settings (a zero in any bit location will leave that setting unchanged). Unit behavior settings can be permanently saved using the "Configuration Save" command (PGN65361). The Unit Behavior payload is described in Table 47.

Table 47. Unit behavior payload

Byte Number	Selected Unit Behavior type
0	Destination Address
1	Enable behavior bitmask: Bit 0 – enable restart on over range (default=0) Bit 1 – enable dynamic motion (default=1) Bit 2 – use uncorrected rates in ARI message (default=0) Bit 3 – Change sequence of ARI and ACCS messages from X, Y, Z (like MTLT305) to Y, X, Z (J1939) (default=1) Bit 4 – enable autobaud detection mode (default=1) Bit 5 – Reserved Bit 6 – enable NWU (north-west-up) frame (accelerometer messages only) (default=1) Bit 7 – use raw (unfiltered) acceleration signal to detect linear acceleration in EKF (default=1). Setting this bit to 0 uses filtered acceleration, which can improve EKF performance in high vibration environments)
2	Bit 0 – Use raw (uncorrected) angular rate to predict acceleration in EKF (default=0). When this bit is 0, bias-corrected angular rate is used. Bit 1 – Swap Byte 2 and Byte 0 in a GET Message Request payload for backward compatibility with MTLT305, see Section 5.1.4 (default=0) Bits 2:6 – Reserved Bit 7 – Enable VG algorithm (default=1)
3	Disable behavior bitmask (turns off the selected behavior from Byte 1 if corresponding bit is set to 1): Bit 0 – disable restart on over range Bit 1 – disable dynamic motion Bit 2 – disable usage of uncorrected rates in ARI message Bit 3 – disable swap X and Y axes in ARI and ACCS messages Bit 4 – disable autobaud detection mode Bit 5 – Reserved Bit 6 – disable NWU accelerometer frame (use NED frame)

	Bit 7 – disable use of raw acceleration signal to detect linear acceleration in EKF (use filtered acceleration signal instead)
4	Disable behavior bitmask (turns off the selected behavior from Byte 2 if corresponding bit is set to 1) Bit 0 – disable use of raw rate (use corrected rate instead) to predict acceleration in EKF Bit 1 – disable "Swap Byte 2 and Byte 0 in a GET Message Request payload" for backward compatibility with MTLT305 Bits 2:6 – Reserved Bit 7 – Disable VG algorithm –
5	New Unit address (valid address from 128 to 247). Will become active after save command issued and unit will go through reset/power cycle.

5.1.5.8 Algorithm Control

The user can adjust certain algorithm parameters which can improve OpenIMU335RI performance. Table 48 shows the format of the algorithm control command.

Table 48. The format of algorithm control command

Priority	PGN	PF	PS	SA	Payload
6	65371	255	91		8 bytes

Table 49. Algorithm control command payload

Byte	Description	Value	Comment
0	Address of Unit	Destination Address	
1	Turn Switch Threshold		Value in deg/s From 1 to 255 (+/-) Setting of 0=disabled
2	Limit for rate integration time	LSB	Value in milliseconds From 10 to 10000 (Default 2000 mS)
3	Limit for rate integration time	MSB	
4	Limit for accelerometer switch delay	LSB	Value in milliseconds From 10 to 10000 (Default 2000 mS)
5	Limit for accelerometer switch delay	MSB	
6	Coefficient of reduced Q	LSB	Scale 0.0001 per lsb From 0.0001 to 1 (Default 0.001)
7	Coefficient of reduced Q	MSB	

5.1.5.9 Aiding Signal Configuration

If a wheel speed or acceleration signal is available on the CAN bus, the user can improve OpenIMU335RI algorithm performance by configuring Aiding Signal information. The aiding signal is configured using two different Set Commands, the first of which is shown in the Table below. The algorithm can accept different CAN messages, including PGN65215, PGN65265, and PGN61445. Contact the factory for availability of alternative messages.

This command indicates the type of aiding signal used (if any), along with the PF and PS values of aiding signal, and the message rate of this aiding signal. If the vehicle acceleration information provided by the aiding signal is unsigned (e.g. PGN65215 and PGN65265), the user also should indicate the PF and PS values of vehicle driving direction signal (e.g. PGN61445), so that unit can recognize the vehicle driving in a forward or reverse direction. If it is signed, the user does not need to indicate PF and PS values of vehicle driving direction signal. Note that the IMU must be mounted with the physical X axis oriented toward the front of the vehicle, Y axis oriented toward the right, and Z axis oriented down, or if mounted differently, should be configured so that the logical X, Y, and Z axes are configured as above. If vehicle accelerations (signal source is 2) is implemented, the IMU is expecting the acceleration data to be defined in a North-West-Up orientation per the J1939 standard, and the data will be converted internally to the IMU NED frame. Note there is a configuration bit (Bit 2 of Byte 7 in the algorithm control payload) to indicate whether the Vehicle X axis acceleration aiding signal is oriented toward the front or rear of the vehicle.

Table 50. The format of Aiding Signal Configuration command

Priority	PGN	PF	PS	SA	Payload
6	65375	255	95		8 bytes

Table 51. Aiding Signal command payload

Byte	Description	Comment
0	Address of Unit	
1	Aiding Signal Source	Indicates if the aiding signal is odometer or vehicle accelerations or neither. 0 = No aiding signal used. 1 = Use Odometer or WSS for aiding 2 = Use vehicle accelerations signal for aiding.
2	PF value of Aiding Message	PF Value of the message that has Odometer or Vehicle Accelerations data
3	PS value of Aiding Message	PS Value of the message that has Odometer or Vehicle Accelerations data
4	Message Rate	Message rate (in Hz) of Odometer or Vehicle Accelerations.
5	Driving Direction Data Message PF value	PF Value of the message that has Driving Direction data (if needed)
6	Driving Direction Data Message PS value	PS Value of the message that has Driving Direction data (if needed)
7	Configuration Switches	Bit 0: IMU Mount Location: 0 = IMU does not experience rotation relative to chassis, such as mounted to a truck's chassis 1 = IMU can experience rotation relative to tracks, such as mounted to body of excavator Bit 1: Reserved (set to "0") Bit 2: Aiding Signal Acceleration Direction 0 = +X Axis acceleration point to vehicle forward. 1 = -X Axis acceleration point to vehicle forward.

5.1.5.10 *Aiding Signal Lever Arm Configuration*

There are two different Lever Arm definitions that may be applied, depending on the application. The first case deals with a vehicle application where the IMU does not experience rotation relative to the vehicle chassis, and where there is an independent measure of linear acceleration available on the CAN bus, which can be used to aid the Kalman Filter's estimates of pitch and roll angles. The second case is intended for an application where there are significant angular rates around an axis with known geometry (for example, the base of an excavator which pivots in the yaw axis, and where the IMU is mounted some known distance from the center of rotation). In both cases the lever arm is configured by PGN65376 shown below, but the definition of the lever arm is different for the two cases.

Case 1: IMU does not experience rotation relative to chassis:

For best performance, if an aiding signal is used, the lever arm of the aiding signal should be configured. This lever arm is defined as the X, Y, Z position of the effective location of the aiding sensor, with respect to the IMU location, as measured in the logical IMU frame. Note the logical IMU frame defines the X, Y, and Z axes as configured by the orientation setting (if different from the physical IMU frame). The purpose of this lever arm is to account for differences in velocity and acceleration at different points in the vehicle, caused by angular rates around the X, Y, and Z axes. The illustration below shows how the lever arm would be measured for an aiding signal at point P (lever arm values would be [x1, y1, -z1], measured in millimeters). Note the values can be either positive or negative. Table 52 below, shows the format of the Aiding Signal Lever Arm Configuration command. This case would be indicated by Byte 1 of the Algorithm Control Command payload = "1" or "2", and Bit 0 of Byte 7 = "0".

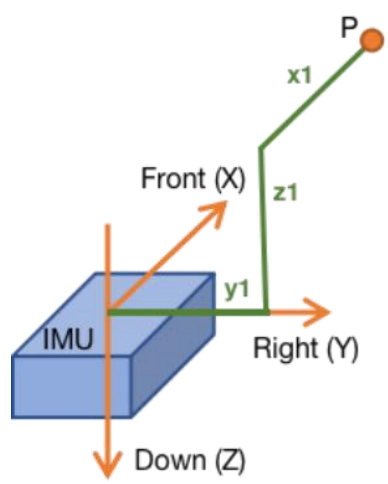


Table 52. The format of Aiding Signal Lever Arm Configuration command

Priority	PGN	PF	PS	SA	Payload
6	65376	255	96		7 bytes

Table 53. Lever Arm command payload (Case 1)

Byte	Description	Value	Definition	Comment
0	Address of Unit		Destination Address	NA
1	LeverArmX	LSB	LeverArm = X, Y, Z position of the odometer or acceleration aiding signal w.r.t. IMU reference point D (see figure at end of this section) in logical IMU frame in units of millimeters, expressed in offset binary format.	Millimeter Range = [-32000, 32000] Offset value = -32000. Default value in mm is [0, 0, 0], which is indicated by LeverArm values of [32000, 32000, 32000]
2		MSB		
3	LeverArmY	LSB		
4		MSB		
5	LeverArmZ	LSB		
6		MSB		

As an example, if the desired value for LeverArmX is 1000mm, then the two-byte value loaded into bytes 1 and 2 should represent the binary number 33,000. The value for LeverArmX (in mm) = 256*MSB + LSB – 32,000.

Case 2: IMU can experience rotation relative to tracks:

In some applications, there may be significant angular rates (which can be internally sensed by the IMU). In these applications, the lever arm indicates the location of the axis of rotation with respect to the IMU. The illustration below shows how the lever arm would be measured for axis of rotation for excavator's body (lever arm values would be [x1, y1, 0], measured in millimeters). Note the values can be either positive or negative. The Table below indicates the definition of the Lever Arm command payload in this case, which would be indicated by Byte 1 of the Algorithm Control Command payload = "1" or "2", and Bit 0 of Byte 7 = "1". Note, in case 2, the CAN aiding signal is only used to determine whether the vehicle is stationary or moving.

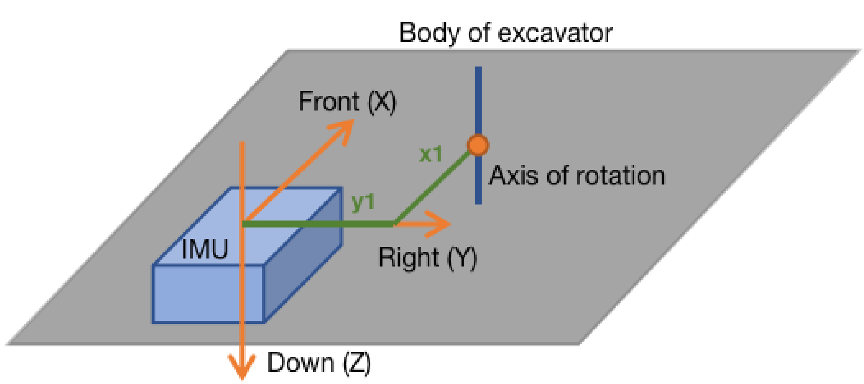
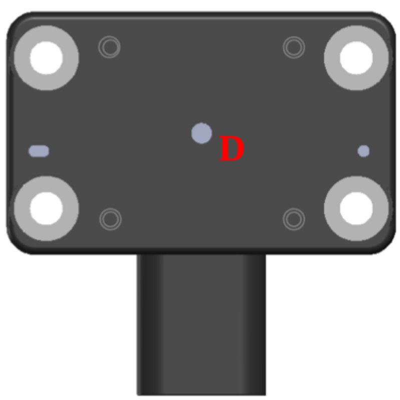


Table 54. Lever Arm command payload (Case 2)

Byte	Description	Value	Definition	Comment
0	Address of Unit		Destination Address	NA
1	LeverArmX	LSB	LeverArm = X, Y, Z position of the axis of rotation with respect to the IMU Reference Point D (see figure below), in units millimeters, expressed in offset binary format.	Millimeter Range = [-32000, 32000] Offset value = -32000 Default in mm is [0, 0, 0], which is indicated by LeverArm values of [32000, 32000, 32000]
2		MSB		
3	LeverArmY	LSB		
4		MSB		
5	LeverArmZ	LSB		
6		MSB		

As an example, if the desired value for LeverArmX is 1000mm, then the two-byte value loaded into bytes 1 and 2 should represent the binary number 33,000. The value for LeverArmX (in mm) = 256*MSB + LSB -32,000.

The following figure shows the location of the Lever Arm reference point (D), which is the center point on the bottom of the unit:



5.1.5.11 DM1 Message Configuration

The IMU can send out DM1 diagnostic messages in case of failures. This message allows to configure specific fields of DM1 messages.

Table 55. DM1 Configuration response message format

Priority	PGN	PF	PS	SA	Payload
6	65370	255	90		8 bytes

Table 56. Manual Content

Byte	Bits	Parameter name	Default Values
1	DA	Destination address	
2	2-1 4-3 6-5 8-7	Protect Lamp Status Amber Warning Lamp Status Red Stop Lamp Status Malfunction Indicator Lamp Status	00 01 00 00
3	2-1 4-3 6-5 8-7	Flash Protect Lamp Flash Amber Warning Lamp Flash Red Stop Lamp Flash Malfunction Indicator Lamp	11 11 11 11
4	8-1	SPN, 8 least significant bits of SPN	0xB3
5	8-1	SPN, second byte of SPN (most significant at bit 8)	0xF4
6	3-1	SPN, 3 most significant bits (most significant at bit 8)	0x07
7		FMI for DTC1	0x0C
8		FMI for DTC2	0x0E

Note: the DM1 messages can be enabled and disabled. To disable message for specific DTC set FMI value to 0xFF.

5.1.5.12 Bank of PS numbers

Users may change the default PS numbers of proprietary OpenIMU335RI PGNs if the pre-configured values have already been used in their system. PS numbers can be re-assigned anywhere in the range from 0x40 to 0x6C (64 to 108). The three commands below are issued by user's ECU and sent to the OpenIMU335RI in order to change the corresponding PS values. Table 57, Table 58 and Table 59 shows the format of these three commands. The OpenIMU335RI will decode these incoming packets and switch the default PS numbers to the values assigned by users. The new PS numbers will take effect immediately and can be permanently saved. Note that PF numbers for all proprietary PGNs have value 255.

Table 57. The format of PS Bank0 command

Priority	PGN	PF	PS	SA	Payload
6	65520	255	240		8 bytes

Table 58. The format of PS Bank1 command

Priority	PGN	PF	PS	SA	Payload
6	65521	255	241		8 bytes

Table 59. The format of PS Bank2 command

Priority	PGN	PF	PS	SA	Payload
6	65522	255	242		8 bytes

The Bank0, Bank1 and Bank2 payloads each contain 8 bytes, where each byte value represents the newly assigned PS number for the corresponding Get or Set command. See **Table 60** and **Table 61** for a description of which byte corresponds with which command. Note if a byte value of zero is detected, that particular field will be ignored (the PS number is not changed).

Table 60. Bank0 Command Payload

Byte	Default PS	Command / Function associated with corresponding Byte
0	N/A	Destination Address
1	80	Algorithm Reset
2	81	Save Configuration
3	82	HW BIT Status
4	83	SW BIT Status
5	84	Master BIT Status
6	107	HR Angular Rate
7	109	HR Acceleration

Table 61. Bank1 Command Payload

Byte	Default PS	Command / Function associated with corresponding Byte
0	N/A	Destination Address
1	85	Packet Rate Divider
2	86	Packet Type(s)
3	87	Digital Filter
4	88	Orientation
5	89	Unit Behavior
6	91	Algorithm Control
7		Reserved

Table 62. Bank2 Command Payload

Byte	Default PS	Command / Function associated with corresponding Byte
0	N/A	Destination Address
1	95	Aiding Signal Configuration
2	96	Aiding Signal LeverArm Configuration
3	90	DM1 Message Configuration
4 – 7		Reserved

5.1.6 Data Packets

The OpenIMU335RI can be configured to periodically output any combination of six different data packets. These packets are slope sensor information and slope sensor information 2, angular rate and acceleration sensor defined by SAEJ1939-DA, and custom high-resolution acceleration and angular rate packets.

5.1.6.1 Slope Sensor Information 2

The payload contains 8 bytes in little-endian mode and follows the definition of SLOT 294 and SAEad11 found in SAEJ1939-DA. The first 24-bits is the pitch value and the next 24-bits is the roll value. See the Tables below for details about the format and payload of the message.

Table 63. The format of SSI2 packet

Priority	PGN	PF	PS	SA	Payload
3	61481	240	41		8 bytes

Table 64. Payload of the SSI2 packet

Bytes	Parameter name	Units	Scaling	Data Range	Offset
1-3	Pitch Angle (Extended Range)	°	1/32768 °/Bit	-250 to 252	-250
4-6	Roll Angle (Extended Range)	°	1/32768 °/Bit	-250 to 252	-250
7.1, 2	Pitch Angle Compensation	-	00=On, 01=Off, 10=Error, 11=N/A		
7.3, 4	Pitch Angle FOM	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
7.5, 6	Roll Angle Compensation	-	00=On, 01=Off, 10=Error, 11=N/A		
7.7, 8	Roll Angle Figure of FOM	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
8	Roll and Pitch Latency	ms	0.5 ms/Bit	0 to 125	0

5.1.6.2 Slope Sensor Information

The payload contains 8 bytes in little-endian mode and follows up the definition of message with PGN 61459 found in SAEJ1939-DA. The first 16-bit is the pitch angle value, next 16 bit is roll angle value and following 16 bit is pitch rate value. See the Tables below for a description of the format and payload of the SSI message.

Table 65. The format of SSI packet

Priority	PGN	PF	PS	SA	Payload
3	61459	240	19		8 bytes

Table 66. Payload of the SSI Packet

Bytes	Parameter name	Units	Scaling	Data Range	Offset
1 – 2	Pitch Angle	°	0.002	-64 to 64.51	-64
3 – 4	Roll Angle	°	0.002	-64 to 64.51	-64
5 - 6	Pitch Rate	°/s	0.002	-64 to 64.51	-64
7.1,2	Pitch Angle FOM	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
7.3,4	Roll Angle FOM	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
7.5,6	Pitch Rate FOM	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
7.7,8	Pitch and Roll Comp	-	00=On, 01=Off, 10=Error, 11=N/A		
8	Roll and Pitch Latency	ms/bit	0.5ms/bit	0 to 125	0

5.1.6.3 Angular Rate

The payload contains 8 bytes in little-endian mode and follows the definition of SLOT 288 and SAEva03 found in SAEJ1939-DA. Each of 16 bits is sequentially allocated to the angular velocity in Y, X, and Z per the J1939 standard. See the Tables below for a description of the message format and payload.

Table 67. The format of angular rate packet

Priority	PGN	PF	PS	SA	Payload
3	61482	240	42		8 bytes

Table 68. ARI Payload Description

Bytes	Parameter name	Units	Scaling	Data Range	Offset
1 – 2	Pitch Angular Rate	°/s	1/128 °/s/Bit	-250 to 250.99	-250
3 – 4	Roll Angular Rate	°/s	1/128 °/s/Bit	-250 to 250.99	-250
5 - 6	Yaw Angular Rate	°/s	1/128 °/s/Bit	-250 to 250.99	-250
7.1, 2	Pitch Rate FOM (Extended Range)	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		

7.3, 4	Roll Rate FOM (Extended Range)	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
7.5, 6	Yaw Rate FOM (Extended Range)	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
8	Angular Rate Measurement Latency	ms	0.5 ms/Bit	0 to 125	0

□ NOTE

By default, the Angular rate packet is arranged in the sequence Pitch, Roll, and Yaw rates, as specified by the J1939 standard. This is different from the previous generation OpenIMU335RI, which defines the message sequence as Roll, Pitch, and Yaw rate. It's possible to switch the sequence of X and Y in the message for backward compatibility (See Unit Behavior Settings).

□ NOTE

It's possible to configure the Angular rate packet to provide either raw or corrected (for bias) angular rates. See Unit Behavior settings for more information.

5.1.6.4 High Resolution Angular Rate

The payload contains 8 bytes in little-endian mode. Rate Sensors data transmitted in 19-bit words sequentially allocated to the angular velocity, Y, X and Z. For each, the range is within -250 to 250.992 deg/s and the resolution is 1/1024 deg/s/bit, with an offset of -250 degrees/second. Table 69 shows the format of angular rate message.

Table 69. The format of High-Resolution angular rate packet

Priority	PGN	PF	PS	SA	Payload
3	65387	255	107		8 bytes

Table 70. HR Angular Rate Payload Description

Bits	Parameter name	Units	Scaling	Data Range	Offset
0 - 18	Pitch Angular Rate	°/s	1/1024 °/s/Bit	-250 to 250.99	-250
19 - 37	Roll Angular Rate	°/s	1/1024 °/s/Bit	-250 to 250.99	-250
38 - 56	Yaw Angular Rate	°/s	1/1024 °/s/Bit	-250 to 250.99	-250
57 - 58	Pitch Rate FOM (Extended Range)	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
59 -60	Roll Rate FOM (Extended Range)	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
61 - 62	Yaw Rate FOM (Extended Range)	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
63	Reserved				

□ NOTE

By default, the Angular rate packet is arranged in the sequence Pitch, Roll, and Yaw rates, as specified by the J1939 standard. This is different from the previous generation OpenIMU335RI, which defines the message sequence as Roll, Pitch, and Yaw rate. It's possible to switch the sequence of X and Y in the message for backward compatibility (See Unit Behavior Settings).

NOTE

It's possible to configure the Angular rate packet to provide either raw or corrected (for bias) angular rates. See Unit Behavior settings for more information.

5.1.6.5 Acceleration Sensor

The payload contains 8 bytes in little-endian mode, and follows the definition of SLOT 303 and SAEad11 found in SAEJ1939-DA. Each of 16 bits is sequentially allocated to the acceleration in Y, X and Z. See the Tables below for details of the message format and payload.

Table 71. The format of acceleration sensor packet

Priority	PGN	PF	PS	SA	Payload
2	61485	240	45		8 bytes

Table 72. ACCS Payload Description

Bytes	Parameter name	Units	Scaling	Data Range	Offset
1 – 2	Acceleration Y axis	m/s^2	0.01 m/s^2/Bit	-320 to 322.55	-320
3 – 4	Acceleration X axis	m/s^2	0.01 m/s^2/Bit	-320 to 322.55	-320
5 - 6	Acceleration Z axis	m/s^2	0.01 m/s^2/Bit	-320 to 322.55	-320
7.1, 2	Lateral Acceleration FOM	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
7.3, 4	Longitudinal Acceleration FOM	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
7.5, 6	Vertical Acceleration FOM	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
7.7, 8	Support Var Tx rep Accel	-	10=20ms Transmission Rate Supported, 11=only 10ms Transmission Rate Supported.		

NOTE

By default, the Acceleration packet is arranged in the sequence Y, X, and Z, and in the North-West-Up orientation, as specified by the J1939 standard. This is different from the previous generation OpenIMU335RI, which defines the message sequence as X, Y, and Z, and North-East-Down orientation. For backward compatibility, it's possible to switch the sequence of X and Y in the message, and also to switch to NED orientation (See Unit Behavior Settings).

5.1.6.6 *High Resolution Acceleration Sensor data*

The payload contains 8 bytes in little-endian mode. Each of 16 bits is sequentially allocated to the acceleration in Y, X, and Z, and in the NWU orientation. As with the ACCS message, it's possible to change both the sequence of X and Y, and also change to NED orientation (See Unit Behavior Settings). See the Tables below for more details on the format and payload for this message.

Table 73. The format of HR acceleration sensor packet

Priority	PGN	PF	PS	SA	Payload
2	65389	255	109		8 bytes

Bits	Parameter name	Units	Scaling	Data Range	Offset
0 –18	Acceleration Y axis	m/s^2	0.00125 m/s^2/Bit	-320 to 322.55	-320
19 –37	Acceleration X axis	m/s^2	0.00125 m/s^2/Bit	-320 to 322.55	-320
38 -56	Acceleration Z axis	m/s^2	0.00125 m/s^2/Bit	-320 to 322.55	-320
57, 58	Lateral Acceleration FOM	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
59, 60	Longitudinal Acceleration FOM	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
61,62	Vertical Acceleration FOM	-	00=Fully Functional, 01=Degraded, 10=Error, 11=N/A		
63	Support Var Tx rep Accel	-	1=20ms Transmission Rate Supported, 0=only 10ms Transmission Rate Supported.		

5.1.7 *Diagnostic Messages DM1*

In case of failures OpenIMU335RI will periodically send out diagnostic messages DM1 with 10 seconds interval. The format of messages is defined in SAEJ1939-73. Two DTC are defined:

DTC 1: Failure indication which requires restart of unit:

This message will be sent out if unit detects the following critical errors:

Critical HW Error:

Supply current over limit (~2x nominal) for > 1 min
Internal power supply voltage out of range for > 1 min
MCU temperature 5C over limit for > 5 mins
All three sensors over temperature limit by 5C for > 5mins

Critical SW Error:

Stack overflow (>95% full)
Algorithm error - when Algorithm is not initialized five seconds after startup.
Data processing delay exceeds 5ms for more than 10 cycles (50ms)
TX queue overflow for more than 10 cycles (50ms)

Fatal Error:

Calibration CRC Error (for all three sensors)
Internal SPI Communication Error (with all three sensors) for > 10 cycles (50ms)

DTC 2: Failure indication which requires reprograming unit:

This message will be sent out if unit detected the following critical errors:

Configuration CRC Error
Application CRC Error

5.1.7.1 DM1 Message Format

This message has default SPN 521395 use FMI values 12 and 14. But SPN and FMI values can be configured. The format of DM1 messages provided below:

Table 74. The format of DM1 packet

Priority	PGN	PF	PS	SA	Payload
6	65226	254	202		8 bytes

Table 75. DM1 messages payload (default)

Byte	Bits	Parameter name	Values DTC 1	Values DTC2
1	2-1 4-3 6-5 8-7	Protect Lamp Status Amber Warning Lamp Status Red Stop Lamp Status Malfunction Indicator Lamp Status	00 01 00 00	00 01 00 00
2	2-1 4-3 6-5 8-7	Flash Protect Lamp Flash Amber Warning Lamp Flash Red Stop Lamp Flash Malfunction Indicator Lamp	11 11 11 11	11 11 11 11
3	8-1	SPN, 8 least significant bits of SPN	0xB3	0xB3
4	8-1	SPN, second byte of SPN (most significant at bit 8)	0xF4	0xF4
5	8-6	SPN, 3 most significant bits (most significant at bit 8)	0x07	0x07

	5-1	FMI	0x0C	0x0E
6	7-1 8	Occurrence Count SPN Conversion Method	 0	 0
7		Reserved	0xFF	0xFF
8		Reserved	0xFF	0xFF

□ **NOTE**

The DM1 messages can be enabled and disabled. To disable the message for specific DTC set the corresponding FMI value to 0xFF.

5.1.7.2 DTC Reset

The user can decide to reset/clear an active DTC condition. It can be done by sending the DM11 message to unit. DM11 message has the following format:

Table 76. The format of DTC Reset Packet

Priority	PGN	PF	PS	SA	Payload
6	65235	254	211		0 bytes

Upon reception of this message unit will clear active DTC and send back positive acknowledgement with the following format:

Table 77. The format of DTC Reset Response

Priority	PGN	PF	PS	SA	Payload
6	59392	232	255		8 bytes

Table 78 Acknowledgement message payload

Byte	Parameter name	Value DTC 1
1	Control byte	0
2	Group function value	0
3	Reserved	0xFF
4	Reserved	0xFF
5	Reserved	0xFF
6	DM11 PGN (LSB)	0xD3
7	DM11 PGN (2nd byte)	0xFE
8	DM11 PGN (MSB)	0x00

5.2 DBC File

A DBC file is a proprietary but common descriptor database file that is used to describes the CAN network and message decoding. DBC files are created and administered with either the CANdb editor or the CANdb++ editor available from Vector.

The format of DBC file follows the definition of Vector DBC Standard. The link is https://vector.com/vi_candb_en.html

A simplified description of DBC file is available on https://github.com/stefanhoelzl/CANpy/tree/master/docs. However, users must contact Vector if the full syntax of DBC file is desired.

A DBC file supporting ACEINNA OpenIMU335RI is available for download from ACEINNA website from the OpenIMU335RI Product page: https://www.aceinna.com/inertial-systems/OpenIMU335RI

6 RS-232 Port Interface Definition

The OpenIMU335RI supports a common packet structure that includes both command or input data packets (data sent to the OpenIMU335RI) and measurement output or response packet formats (data sent from the OpenIMU335RI). This section of the manual explains these packet formats as well as the supported commands. Navigation studio also features a number of tools that can help a user understand the packet types available and the information contained within the packets. This section of the manual assumes that the user is familiar with ANSI C programming language and data type conventions.

6.1 General Settings

The serial port settings are RS-232 with 1 start bit, 8 data bits, no parity bit, 1 stop bit, and no flow control. Standard baud rates supported are 38400, 57600, 115200, 230400

Common definitions include:

- A word is defined to be 2 bytes or 16 bits.
- All communications to and from the unit are packets that start with a single word alternating bit preamble 0x5555. This is the ASCII string "UU".
- All multiple byte values are transmitted Big Endian (Most Significant Byte First).
- All communication packets end with a single word CRC (2 bytes). CRC's are calculated on all packet bytes excluding the preamble and CRC itself. Input packets with incorrect CRC's will be ignored.
- Each complete communication packet must be transmitted to the OpenIMU335RI Series inertial system within a 4 second period.

6.2 Number Formats

Number Format Conventions include:

- 0x as a prefix to hexadecimal values
- single quotes ('') to delimit ASCII characters
- no prefix or delimiters to specify decimal values.

Table 79. Number Formats

Descriptor	Description	Size (bytes)	Comment	Range
U1	Unsigned Char	1		0 to 255
U2	Unsigned Short	2		0 to 65535
U4	Unsigned Int	4		0 to 2^32-1

I2	Signed Short	2	2' s Complement	-2^15 to 2^15-1
I2*	Signed Short	2	Shifted 2' s Complement	Shifted to specified range
I4	Signed Int	4	2' s Complement	-2^31 to 2^31-1
F4	Floating Point	4	IEEE754 Single Precision	-1*2^127 to 2^127
D	Double	8	IEEE-754 Double Precision	2.23^-308 to 1.80^308
SN	String	N	ASCII	

6.3 Packet Format

All of the Input and Output packets, except the Ping command, conform to the following structure:

0x5555	<2-byte packet type (U2)>	<payload byte-length (U1)>	<variable length payload>	<2-byte CRC (U2)>

To Ping a OpenIMU335RI Series unit, type the ASCII string 'UUPK'. If properly connected, the OpenIMU335RI Series unit will respond with 'PK'. All other communications with the OpenIMU335RI Series unit require the 2-byte CRC. (Note: A OpenIMU335RI Series unit will also respond to a ping command using the full packet formation with payload 0 and correctly calculated CRC. Example: 0x5555504B009ef4).

6.3.1 Packet Header

The packet header is always the bit pattern 0x5555.

6.3.2 Packet Type

The packet type is always two bytes long in unsigned short integer format. Most input and output packet types can be interpreted as a pair of ASCII characters. As a semantic aid consider the following single character acronyms:

P = packet

F = fields

Refers to Fields which are settings or data contained in the unit

E = EEPROM

Refers to factory data stored in EEPROM

R = read

Reads default non-volatile fields

G = get

Gets current volatile fields or settings

W = write

Writes default non-volatile fields. These fields are stored in non-volatile memory and determine the unit's behavior on power up. Modifying default fields take effect on the next power up and thereafter.

S = set

Sets current volatile fields or settings. Modifying current fields will take effect immediately by modifying internal RAM and are lost on a power cycle

6.3.3 Payload Length

The payload length is always a one-byte unsigned character with a range of 0-255. The payload length byte is the length (in bytes) of the <*variable length payload*> portion of the packet only and does not include the CRC.

6.3.4 Payload

The payload is of variable length based on the packet type.

6.3.5 16-bit CRC-CCITT

Packets end with a 16-bit CRC-CCITT calculated on the entire packet excluding the 0x5555 header and the CRC field itself. A discussion of the 16-bit CRC-CCITT and sample code for implementing the computation of the CRC is included at the end of this document. This 16-bit CRC standard is maintained by the International Telecommunication Union (ITU). The highlights are:

Width = 16 bits

Polynomial 0x1021

Initial value = 0xFFFF

No XOR performed on the final value.

See Appendix A for sample code that implements the 16-bit CRC algorithm.

6.3.6 Messaging Overview

Table 80 summarizes the messages available by OpenIMU335RI Series model. Packet types are assigned mostly using the ASCII mnemonics defined above and are indicated in the summary Table 80 and in the detailed sections for each command. The payload byte-length is often related to other data elements in the packet as defined in the table below. The referenced variables are defined in the detailed sections following. Output messages are sent from the OpenIMU335RI Series inertial system to the user system as a result of a poll request or a continuous packet output setting. Input messages are sent from the user system to the OpenIMU335RI Series inertial system and will result in an associated Reply Message or NAK message. Not that both Input message and its reply message typically have the same ***<2-byte packet type (U2)>.***

Table 80. Message Table

ASCII Mnemonic	***<2-byte packet type (U2)>***	***<payload byte-length (U1)>***	**Description**	**Type**
Link Test				
pG	0x7047	0 / N	Ping Command and Response	Input/Reply Message
Interactive Commands				
uC	0x7543	8+8*N / 4	Update Config	Input/Reply Message
uP	0x7550	12 / 4	Update Parameter	Input/Reply Message
uA	0x7541	8*N / 4	Update All	Input/Reply Message
sC	0x7343	0	Save Configuration	Input/Reply Message
rD	0x7244	0	Restore Defaults	Input/Reply Message
gC	0x6743	8 / 8+8*N	Get Config	Input/Reply Message
gA	0x6741	0 / 8*N	Get All	Input/Reply Message
gP	0x6750	4 /12	Get Parameter	Input/Reply Message
gV	0x6756	0 / N	Get User Version	Input/Reply Message
NAK	0x1515	2	Error Response	Reply Message
Output Messages				
z1	0x7A31	40	Scaled IMU Data	Output Message
a1	0x6131	47	VG Data Packet	Output Message
a2	0x6132	48	VG_AHRS Data Packet	Output Message
s1	0x7331	52	Scaled-Sensor Data	Output Message

7 OpenIMU335RI RS-232 Port Commands and Messages

7.1 Link Test

7.1.1 Ping Command

Ping ('pG' = 0x7047)			
Preamble	Packet Type	Length	Termination
0x5555	0x7047	0x00	<CRC (U2)>

The ping command has no payload. Sending the ping command will cause the unit to send a ping response.

7.1.2 Ping Response

Ping ('pG' = 0x7047)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7047	N	Unit Model and Serial Number <S> (string)	<CRC (U2)>

The unit will send this packet in response to a ping command.

7.2 Interactive Commands

Interactive commands are used to interactively request data from the OpenIMU335RI, and to calibrate or reset the OpenIMU335RI Series.

7.2.1 Update Config

Update Config ('uC' = 0x7543)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7543	8+8*N	N Parameters	<CRC (U2)>

The Update Config command is used to update and apply N consecutive user-defined configuration parameters at a time to a unit. Each parameter value is 64 bit (8 bytes) and can have arbitrary type.

uC Payload Contents

Byte Offset	Name	Format	Notes
0	Number of consecutive parameters to update	U4	LSB First
4	Offset of first parameter in unit config structure	U4	LSB First
8	Parameter Value	U8, I8, F8, Double, S8, or A8	LSB First
:	:	:	:
8+N*8	Parameter Value	U8, I8, F8, Double, S8, or A8	LSB First

Upon reception, each parameter is validated (if desired) and if validation passes the parameter gets written into the gUserConfiguration structure and also applied to the system on-the-fly (if desired). If the value of one parameter is invalid – all parameters ignored. Updated configuration parameters will be active until the next unit power cycle or reset.

Update Config command will have the following response:

uC Response				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7543	0x04	Error Code (I4)	<CRC (U2)>

Where error code can be: (0) – "Success", (-3) – "Invalid Payload Size", (-1) – "Invalid parameter number", (-2) – "Invalid parameter value".

7.2.2 Update Parameter

Update Parameter ('uP' = 0x7550)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7550	0x0C	<uP Payload>	<CRC (U2)>

The Update Parameter command is used to update and apply single a user-defined configuration parameter to the unit. The parameter value is 64 bits (8 bytes) and can have arbitrary type.

uP Payload Contents			
Byte Offset	Name	Format	Notes

0	Offset of parameter in unit config structure	U4	LSB First
4	Parameter Value	U8, I8, F8, Double, S8, or A8	LSB First

Upon reception, the parameter value is validated (if desired) and if validation passes the parameter gets written into the gUserConfiguration structure and also applied to the system on-the-fly (if desired). If the value of the parameter is invalid – it is ignored. The updated configuration parameter will be active until next unit power cycle or reset.

The update Parameter command will have the following response:

uP Response				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7550	0x04	Error Code (I4)	<CRC (U2)>

Where error code can be: (0) – "Success", (-3) – "Invalid Payload Size", (-1) – "Invalid parameter number", (-2) – "Invalid parameter value"

7.2.3 Update All Command

Update All ('uA' = 0x7541)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7541	8N	N (up to 30) Parameters	<CRC (U2)>

The Update All command is used to update/apply up to 30 consecutive user-defined configuration parameters at a time to a unit, starting from the first parameter in the user configuration structure. Each parameter has a size of 8 bytes (64 bit) and can have arbitrary type.

Update All Payload Format:

uA Payload Contents			
Byte Offset	Name	Format	Notes
0	First parameter value	U8, I8, F8, Double, S8, or A8	LSB First
:	:	:	:
N*8	Last parameter value	U8, I8, F8, Double, S8, or A8	LSB First

Upon reception, each parameter is validated (if desired) and if the validation passes the parameter gets written into the gUserConfiguration structure, starting from first parameter (offset 0) and is also applied to the system on-the-fly (if desired). If the value of one parameter is invalid, all parameters are ignored. The first two parameters are ignored. Updated configuration parameters will be active until the next unit power cycle or reset.

The Update All command will have following response:

uA Response				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7541	0x04	Error Code (I4)	<CRC (U2)>

Where the error code can be: (0) – "Success", (-3) – "Invalid Payload Size", (-1) – "Invalid parameter number", (-2) – "Invalid parameter value"

7.2.4 Save Config Command

Save Config ('sC' = 0x7343)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7343	0		<CRC (U2)>

The Save Config command has no payload. Upon reception of the "Save Config" command, the unit will save the current gUnitConfiguration structure into EEPROM and updated parameters will be applied to the unit upon all future startups (until new changes are made).

The Save Config command will have following response if saving completed successfully:

sC Response				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7343	0		<CRC (U2)>

7.2.5 Restore Defaults

Restore Defaults ('rD' = 0x7244)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7244	0		<CRC (U2)>

The Restore defaults command has no payload. Upon reception of the "Restore Defaults" command, the unit will save the default configuration structure gDefaultUserConfig into EEPROM and updated parameters will be applied to the unit upon all future startups (until new changes are made).

The Restore Defaults command will have following response if completed successfully:

rD Response				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7244	0		<CRC (U2)>

7.2.6 Get Config Command

Get Config ('gC' = 0x6743)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6743	8	<Payload>	<CRC (U2)>

The Get Config command is used to retrieve N consecutive user-defined configuration parameters at a time from a unit. Parameter value is 64 bit (8 bytes) and can have arbitrary type.

Get Config Payload Format:

gC Payload Contents			
Byte Offset	Name	Format	Notes
0	Number of consecutive parameters to read	U4	LSB First
4	Offset of first parameter in unit config structure	U4	LSB First

A successful Get Config command will have the following response:

Successful gC Response				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6743	8+8*N	N parameters	<CRC (U2)>

Get Config Response Payload Format in the event of a successful completion:

Successful gC Payload Contents			
Byte Offset	Name	Format	Notes
0	Number of consecutive parameters to read	U4	LSB First
4	Offset of first parameter in unit config structure	U4	LSB First
8	Parameter Value	U8, I8, F8,	LSB First

		Double, S8, or A8	
:	:	:	:
8+N*8	Parameter Value	U8, I8, F8, Double, S8, or A8	LSB First

Get Config Response in the event of an unsuccessful completion:

Unsuccessful gC Response				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6743	0x04	Error Code (I4)	<CRC (U2)>

Where error code can be: (0) – "Success", (-3) – "Invalid Payload Size", (-1) – "Invalid parameter number", (-2) – "Invalid parameter value"

7.2.7 Get All Command

Get All ('gA' = 0x6741)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6741	0		<CRC (U2)>

The Get All command used to retrieve N (up to 30) consecutive user-defined configuration parameters at a time from unit, starting from first parameter in gUserConfiguration structure. Parameter value is 64 bit (8 bytes) and can have arbitrary type.

Get All command will have next response:

gA response				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6741	8*N	<Payload>	<CRC (U2)>

Get All Response Payload Format in the event of a successful completion:

Successful gC Payload Contents			
Byte Offset	Name	Format	Notes
0	Number of consecutive parameters to read	U4	LSB First

4	Offset of first parameter in unit config structure	U4	LSB First
8	Parameter Value	U8, I8, F8, Double, S8, or A8	LSB First
:	:	:	:
8+N*8	Parameter Value	U8, I8, F8, Double, S8, or A8	LSB First

Get All Response in the event of an unsuccessful completion:

Unsuccessful gA Response				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6741	0x04	Error Code (I4)	<CRC (U2)>

Where error code can be: (0) – "Success", (-3) – "Invalid Payload Size", (-1) – "Invalid parameter number", (-2) – "Invalid parameter value"

7.2.8 Get Parameter Command

Get Parameter ('gP' = 0x6750)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6750	4	<Payload>	<CRC (U2)>

The Get Parameter command is used to retrieve one user-defined configuration parameter from the gUserConfiguration structure. The parameter value is 64 bit (8 bytes) long and can have arbitrary type.

Get Parameter command payload format:

gP Payload Contents			
Byte Offset	Name	Format	Notes
0	Offset of parameter in unit config structure	U4	LSB First

Get Parameter command will have following response if successful:

Successful gP Response				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6750	12	<Payload>	<CRC (U2)>

Get Parameter response rayload format in case of success:

Successful gP Payload Contents			
Byte Offset	Name	Format	Notes
0	Number of consecutive parameters to read	U4	LSB First
4	Parameter Value	U8, I8, F8, Double, S8, or A8	LSB First

Get Parameter response payload format in case of error:

Unsuccessful gP Response				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6750	0x04	Error Code (I4)	<CRC (U2)>

Where error code can be: (0) – "Success", (-3) – "Invalid Payload Size", (-1) – "Invalid parameter number", (-2) – "Invalid parameter value"

7.2.9 Get User Version Command

Get User Version ('gV' = 0x6756)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6756	0	<Payload>	<CRC (U2)>

The Get Version command has no payload. Sending the Get Version command will cause the unit to send a response with next format:

gV Response				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6756	N	User Version String	<CRC (U2)>

The Get Version response will return null-terminated string, user version. User version string defined in the UserMessaging.c file.

7.2.10 Error Response

Error Response (ASCII NAK, NAK = 0x1515)

Preamble	Packet Type	Length	Payload	Termination
0x5555	0x1515	0x02	<NAK payload>	<CRC (U2)>

The unit will send this packet in place of a normal response to a *failedInputPacketType* request if it could not be completed successfully.

NAK Payload Contents			
Byte Offset	Name	Format	Description
0	failedInputPacketType	U2	the failed request

7.3 RS-232 Output Packets

7.3.1 Scaled IMU Data Packet

The 'z1' packet has the following format:

Scaled IMU Data ('z1' = 0x7A31)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7A31	0x28	<z1 payload>	<CRC (U2)>

The payload of the 'z1'packet is defined as follows:

z1 Payload Contents				
Byte Offset	Name	Format	Units	Description
0	Time	U4	us	System Timer at the moment of sensors sampling
4	xAccel	F4	m/s^2	Acceleration value for axis X
8	yAccel	F4	m/s^2	Acceleration value for axis Y
12	zAccel	F4	m/s^2	Acceleration value for axis Z
16	xRate	F4	°/sec	Rotation speed for axis X
20	yRate	F4	°/sec	Rotation speed for axis Y
24	zRate	F4	°/sec	Rotation speed for axis Z
38	xMag	F4	G	Magnetic field for axis X
32	yMag	F4	G	Magnetic field for axis Y
36	zMag	F4	G	Magnetic field for axis Z

7.3.2 Vertical Gyro Data Packet

The 'a1' packet has the following format:

VG Data ('a1' = 0x6131)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6131	0x2F	<a1 payload>	<CRC (U2)>

The payload of the 'a1'packet is defined as follows:

a1 Payload Contents				
Byte Offset	Name	Format	Units	Description
0	Time_ms	U4	ms	System Timer at the moment of sensors sampling
4	Time_s	D	s	System Timer at the moment of sensors sampling
12	Roll	F4	°	Roll angle
16	Pitch	F4	°	Pitch angle
20	xRate	F4	°/sec	Rotation speed for axis X
24	yRate	F4	°/sec	Rotation speed for axis Y
28	zRate	F4	°/sec	Rotation speed for axis Z
32	xAccel	F4	m/s^2	Acceleration value for axis X
36	yAccel	F4	m/s^2	Acceleration value for axis Y
40	zAccel	F4	m/s^2	Acceleration value for axis Z
44	opMode	U1	n/a	Operation Mode
45	linAccSw	U1	n/a	Linear Acceleration Switch
46	turnSw	U1	n/a	Turn Switch

7.3.3 Vertical Gyro / Attitude Heading Reference System Data Packet

The 'a2' packet has the following format:

VG_AHRS Data ('a2' = 0x6132)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x6132	0x30	<a2 payload>	<CRC (U2)>

The payload of the 'a2'packet is defined as follows:

a2 Payload Contents				
Byte Offset	Name	Format	Units	Description
0	Time_ms	U4	ms	System Timer at the moment of sensors sampling
4	Time_s	D	s	System Timer at the moment of sensors sampling
12	Roll	F4	°	Roll angle
16	Pitch	F4	°	Pitch angle
20	Heading	F4	°	Yaw/Heading angle
24	xRate	F4	°/sec	Rotation speed for axis X
28	yRate	F4	°/sec	Rotation speed for axis Y
32	zRate	F4	°/sec	Rotation speed for axis Z
36	xAccel	F4	m/s^2	Acceleration value for axis X
40	yAccel	F4	m/s^2	Acceleration value for axis Y
44	zAccel	F4	m/s^2	Acceleration value for axis Z

7.3.4 Scaled Sensor Data Packet

The 's1' packet has the following format:

Scaled Sensor Data ('s1' = 0x7331)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x7331	0x28	<s1 payload>	<CRC (U2)>

The payload of the 's1'packet is defined as follows:

s1 Payload Contents				
Byte Offset	Name	Format	Units	Description
0	Time_ms	U4	ms	System Timer at the moment of sensors sampling
4	Time_s	D	s	System Timer at the moment of sensors sampling
12	xAccel	F4	m/s^2	Acceleration value for axis X

16	yAccel	F4	m/s^2	Acceleration value for axis Y
20	zAccel	F4	m/s^2	Acceleration value for axis Z
24	xRate	F4	°/sec	Rotation speed for axis X
28	yRate	F4	°/sec	Rotation speed for axis Y
32	zRate	F4	°/sec	Rotation speed for axis Z
36	xMag	F4	G	Magnetic field for axis X
40	yMag	F4	G	Magnetic field for axis Y
44	zMag	F4	G	Magnetic field for axis Z
48	Temp	F4	°C	Temperature

8 Bootloader

8.1 Firmware Upgrade

OpenIMU335RI/OpenIMU335 units have a built-in bootloader. Firmware can be upgraded over the RS-232 interface or over CAN bus interface.

Firmware upgrade over the RS-232 interface is described in section 9.2.

Firmware upgrade over CAN bus interface described in the document "Upgrading FW on Aceinna IMUs over CAN bus"

8.2 Firmware Upgrade Over the RS-232 Interface

A user can initiate firmware upgrade procedure on RS-232 interface at any time by sending 'JI' or "JB" command (see below command's format) to application program. This command forces the unit enter Bootloader mode. The unit will communicate at 57.6Kbps baud rate regardless of the original baud rate the unit is configured to. The Bootloader always communicates at 57.6Kbps until the firmware upgrade is complete.

As an additional device recovery option immediately after powering up, every MTLT3xD will enter a recovery window of 200ms prior to application start. During this 200mS window, the user can send 'JI' command at 57.6Kbs to the Bootloader in order to force the unit to remain in Bootloader mode.

Once the device enters Bootloader mode via the 'JI' command either during recovery window or from normal operation, a user can send a sequence 'WA' commands to write a complete application image into the device's FLASH.

After loading the entire firmware image with successive 'WA' commands, a 'JA' command is sent to instruct the unit to exit Bootloader mode and begin application execution. At this point, the device will return to its original baud rate.

Optionally, the system can be reboot by cycling power to restart the system.

The commands detailed in Sections 8.2.1 are used for upgrading a new firmware version.

8.2.1 Commands used for FW upgrade via RS-232 Interface

Firmware upgrade is performed by a Write APP command through the RS-232 port, through Navigation Studio, NAV-View, or a python program.

The following commands allow users to install a pre-built binary into flash memory and force system to enter either Bootloader or application mode.

8.2.1.1 Enforce Bootloader Mode

Jump To Bootloader ('JI' = 0x4A49)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4A49	0x00		CRC (U2)

The command permanently enforces unit into Bootloader mode until unit receives "JA" command.

8.2.1.2 Jump to Bootloader Command

Jump To Bootloader ('JB' = 0x4A49)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4A49	0x00		CRC (U2)

The command allows system to enter Bootloader mode. Unit can return back to application mode after reset command or power cycle command if no changes to application image were done.

8.2.1.3 Write APP Command

Write APP ("WA" = 0x5741)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x5741	length+5		CRC (U2)

The command allows users to write binary sequentially to flash memory in Bootloader mode. The total length is the sum of payload's length and 4-byte address followed by 1-byte data length. See the following table of the payload's format.

WA Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	startingAddress	U4	-	bytes	The FLASH word offset to begin writing data
4	byteLength	U1	-	bytes	The word length of the data to write
5	dataByte0	U1	-	-	FLASH data
6	dataByte1	U1	-	-	FLASH data
…	…				
4+byteLength	dataByte	U1	-	-	FLASH data

Payload starts from 4-byte address of flash memory where the binary is located. The fifth byte is the number of bytes of *dataBytess*, but less than 240 bytes. User must truncate the binary to less than 240-byte blocks and fill each of blocks into payload starting from the sixth byte.

8.2.1.4 Jump to Application Command

Jump to Application ('JA' = 0x4A41)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4A41	0x00		CRC (U2)

The command allows system directly to enter application mode.

9 Warranty and Support Information

9.1 Customer Service

As a ACEINNA customer you have access to product support services, which include:

- Single-point return service
- Web-based support service
- Same day troubleshooting assistance
- Worldwide ACEINNA representation
- Onsite and factory training available
- Preventative maintenance and repair programs
- Installation assistance available

9.2 Contact Directory

Email: techsupport@aceinna.com

http://www.aceinna.com/support/index.cfm

9.3 Return Procedure

9.3.1 Authorization

Before returning any equipment, please contact ACEINNA to obtain a Returned Material Authorization number (RMA).

Be ready to provide the following information when requesting a RMA:

- Name
- Address
- Telephone, Fax, Email
- Equipment Model Number
- Equipment Serial Number
- Installation Date
- Failure Date
- Fault Description
- Will it connect to NAV-VIEW 3.X?

9.3.2 Identification and Protection

If the equipment is to be shipped to ACEINNA for service or repair, please attach a tag TO THE EQUIPMENT, as well as the shipping container(s), identifying the owner. Also indicate the service or repair required, the problems encountered, and other information considered valuable to the service facility such as the list of information provided to request the RMA number.

Place the equipment in the original shipping container(s), making sure there is adequate packing around all sides of the equipment. If the original shipping containers were discarded, use heavy boxes with adequate padding and protection.

9.3.3 Sealing the Container

Seal the shipping container(s) with heavy tape or metal bands strong enough to handle the weight of the equipment and the container.

9.3.4 Marking

Please write the words, "FRAGILE, DELICATE INSTRUMENT" in several places on the outside of the shipping container(s). In all correspondence, please refer to the equipment by the model number, the serial number, and the RMA number.

9.4 Warranty

The ACEINNA product warranty is one year from date of shipment.

Appendix A: 16-bit CRC Implementation Sample Code

```
uint16_t CalculateCRC (uint8_t *buf, uint16_t length)
{
    uint16_t crc = 0x1D0F;

    for (int i=0; i < length; i++) {
    crc ^= buf[i] << 8;
    for (int j=0; j<8; j++) {
        if (crc & 0x8000) {
            crc = (crc << 1) ^ 0x1021;
        }
        else {
        crc = crc << 1;
        }
    }
  }
  return ((crc << 8) & 0xFF00) | ((crc >> 8) & 0xFF);
}
```

Appendix B: Function Instance (Detecting location on wire harness)

The OpenIMU335RI includes a feature for detecting the installation location on a bus which includes more than one IMU module. The feature uses the state of RS-232 pins to populate the "Function Instance" Field (bits 24:28) in the address claim payload, according to the definition in the Table below:

Table 1: Definition of Function Instance Field based on RS-232 pin states

Function Instance Field (Bits 35:39 of "Name")	State of RS-232 RX pin (pin 4)
00000	Pin 4 (RX) = open or GND*
00001	Pin 4 = VDD or above 2 Volts
00010	Pin 4 (RX) shorted to Pin 5 (TX)
00011	1kHz clock
00100	2kHz clock
00101	3kHz clock
00110	4kHz clock
00111	5kHz clock
11111	Invalid frequency detected

* Pin 4 has an internal pull-down resistor

In order for a valid frequency signal to be detected, the unit must receive at least 2 cycles of the given frequency during a window in time starting 50ms after valid power supply and ending 200ms after valid power supply. The accuracy of the frequency must be within 5% to be considered valid.

By reading the "Function Instance" field (bits 24:28 of the address claim payload), the main ECU can associate an installation location with a specific device SA. This allows uploading onto IMU a specific configuration, or even a different firmware version, based on the installation location.

When power is first applied, the unit will enter boot mode, and (if automatic baudrate detection is enabled) silent baud discovery mode. In parallel, the unit is detecting the state of the RS-232 pins. During the configuration detection period (which spans the time from 50ms to 200ms after POR), the unit is looking for a valid frequency. As soon as a valid frequency is detected, the detection process is terminated. If no frequency is detected by the time the 200ms period has expired, the unit will attempt to send a message from the TX pin to the RX pin; if it is unsuccessful, the last state of the RX pin (low or high) will be used to determine the configuration. If a frequency is detected that is not valid, the Function Instance bits will be populated with "11111", indicating an invalid configuration. After 200ms (from valid power supply) has passed, and provided a valid baudrate has been detected, the unit will issue an address claim message, with the Function Instance field populated based on the detected state of the RS-232 pins. If a valid application has been detected, the unit will exit from boot mode and control will be passed to main application. If no valid application has been detected, the unit will remain in boot mode until a valid application has been uploaded. If an INVALID application is detected (application not recognized, or has invalid CRC detected), the unit will remain in boot mode, and will also indicate this error by periodically transmitting the Master BIT (PGN65364), with the appropriate bits set (see description of PGN65364).

Note that using the function instance to detect wire harness location is a dual use of the RS-232 pins. This typically does not present any issues in the end application, where CAN J1939 is the primary communication method. However, there is also an RS-232 bootloader available. If 1) an RS-232 Boot command is received during the 200ms power up sequence or 2) application received command to enter BOOT mode from RS-232 port, control will be passed to the RS-232 Bootloader, and the location detection function will be aborted (case 1) or not performed (case 2).

Appendix C: OpenIMU335RI Evaluation Kit

The evaluation kit for the OpenIMU335RI consists of an evaluation board and ST-Link/V2 programmer / debugger. There is also an interface cable terminated with a TE Connectivity 776531-1 connector on one end that connects to the OpenIMU335RI. The other end of the cable is terminated by female DB9 connectors, for both the CAN and RS-232 interfaces, and power (red) and ground (black) supply wires.

The evaluation board differs from the standard OpenIMU335RI in that a JTAG connector is brought out for programming and debugging user code. While the firmware of the standard part can be updated via the bootloader, it is recommended that anyone writing their own code use the evaluation kit as they can create a full recovery image of the part prior to any development work.

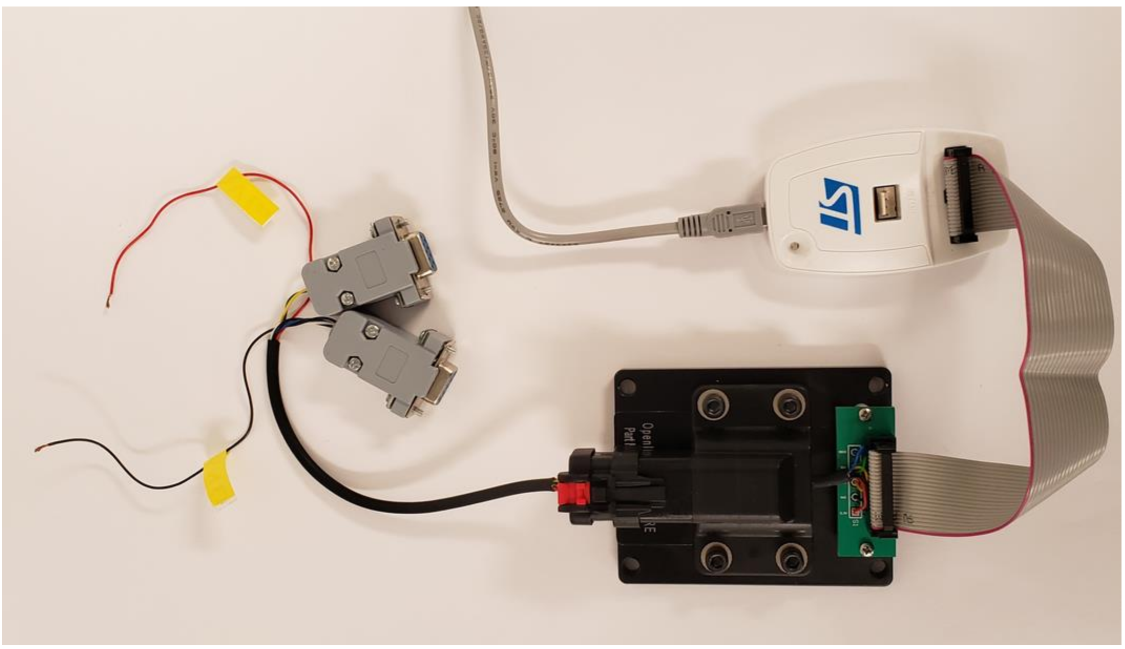


Figure 6. OpenIMU335RI Evaluation Kit

To get started with the OpenIMU335RI evaluation kit refer to the Aceinna Navigation Studio website where there is documentation on how to:

- Download a pc server application that will allow you to evaluate the part over the RS-232 interface using the Chrome® web browser (link)
- Update the firmware on the OpenIMU335RI using one of Aceinna's pre-compiled applications (link)
- Install the OpenIMU programming environment for user code development (link)

To evaluate the part using the CAN interface simply connect to either a CAN analyzer, or network, and refer to the CAN Port Interface Definition section of this user manual.