DMU281ZA, DMU381ZA, and DMU481ZA SERIES USER MANUAL

Document Part Number: 7430-3881-01

ACEINNA, Inc.

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

WARNING

This product has been developed by ACEINNA 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 US law is prohibited.

Revision History

Date	Document Revision	Firmware Applicability	Description	Author
Mar 12, 2018	Rev. 1.0	v19.1.x	Baseline release of IMUx81ZA manual	Feng Liu

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

Normal

1 Introduction

1.1 Manual Overview

This manual provides a comprehensive introduction to ACEINNA's DMU281ZA, DMU381ZA, and DMU481ZA Series Inertial System products (**D**ynamic **M**easurement **U**nit 281ZA, 381ZA, or 481ZA).

As the functionality of the different series are identical (only the performance specs are different), for simplicity all references will be to the DMUx81ZA, where x is 2, 3, or 4. For users wishing to get started quickly, please refer to the two-page quick start guide included with each evaluation kit shipment. 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: Manual Overview	All customers should read sections 1.1 and 1.2.
Section 2: Interface	Customers designing the electrical and mechanical interface to the DMUx81ZA series products should read Section 2.
Section 3: Theory of Operation	All customers should read Section 3. As the DMUx81ZA Series products are inter-related, use the chart at the beginning of Section 3 to ensure that you get an overview of all of the functions and features of your DMUx81ZA Series system. For example, if you have purchased an INSx81ZA, you should read not only the section on the INSx81ZA, but also familiarize yourself with the theory of operation for the IMUx81ZA, VGx81ZA, and AHRSx81ZA. The INSx81ZA builds on the capabilities of the IMUx81ZA, VGx81ZA and AHRSx81ZA.
Section 4: Application Guide	Customers who want product configuration tips for operating the DMUx81ZA Series Inertial Systems in a wide range of applications – fixed wing, rotary wing, unmanned vehicles, land vehicles, marine vessels, and more, should review the part of Section 4 that is relevant to your application. Note: INS and AHRS DMUx81ZA Series units are preconfigured for airborne applications with "normal" dynamics. VGx81ZA Series units are preconfigured for land applications with "automotive testing" dynamics. All DMUx81ZA Series products allow for complete flexibility in configuration by the user.
Section 5: SPI Port Interface	Customers designing the software interface to the DMUx81ZA series products SPI Port should review Section 5.
Section 6-9: UART Port Interface	Customers designing the software interface to the DMUx81ZA series products UART Port should review Sections 6-9.

1.2 Overview of the DMUx81ZA Series Inertial Systems

This manual provides a comprehensive introduction to the use of ACEINNA's DMUx81ZA Series Inertial System products listed in Table 2. This manual is intended to be used as a detailed technical reference and operating guide. ACEINNA's DMUx81ZA Series products combine the latest in high-performance commercial MEMS (Micro-electromechanical Systems) sensors and digital signal processing techniques to provide a small, cost-effective alternative to existing IMU systems.

Table 2 DMUx81ZA Series Feature Description

Product	Features
IMUx81ZA (-200,-209,-409)	6-DOF Digital IMU, 9-DOF Digital IMU Standard Range, 9-DOF Digital IMU High Range
VGx81ZA (-200,-400)	6-DOF IMU plus Roll and Pitch Standard Range, High Range
AHRSx81ZA (-200, -400)	9-DOF IMU (3-Axis Internal Magnetometer) plus Roll, Pitch, and Heading Standard Range, High Range
INSx81ZA (-200, -400)	9-DOF IMU (3-Axis Internal Magnetometer) with interface for External GPS Receiver plus Position, Velocity, Roll, Pitch, and Heading Standard Range, High Range

The DMUx81ZA Series is ACEINNA's fourth 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. It is designed for OEM applications.

At the core of the DMUx81ZA Series is a rugged 6-DOF (Degrees of Freedom) MEMS inertial sensor cluster that is common across all members of the DMUx81ZA 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 bulk 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 rate tables.

Coupled to the 6-DOF MEMS inertial sensor cluster is a high performance microprocessor that utilizes the inertial sensor measurements to accurately compute navigation information including attitude, heading, and linear velocity through dynamic maneuvers (actual measurements are a function of the DMUx81ZA Series product as shown in Table 2). In addition, the processor makes use of internal magnetic sensor and external GPS data to aid the performance of the inertial algorithms and help correct long term drift and estimate errors from the inertial sensors and computations. The navigation algorithm utilizes a multi-state configurable Extended Kalman Filter (EKF) to correct for drift errors and estimate sensor bias values.

Another unique feature of the DMUx81ZA Series is the extensive field configurability of the units. This field configurability allows the DMUx81ZA Series of Inertial Systems to satisfy a wide range of applications and performance requirements with a single mass produced hardware platform. The basic configurability includes parameters such as baud rate (UART), clock speed (SPI), packet type, and update rate, and the advanced configurability includes the defining of custom axes and how the sensor feedback is utilized in the Kalman filter during the navigation process.

The DMUx81ZA Series is packaged in a light-weight, rugged, unsealed metal enclosure that is designed for cost-sensitive commercial and OEM applications. The DMUx81 can be configured

to output data over a SPI Port or a low level UART serial port. The port choice is user controlled by grounding the appropriate pin on the connector. The DMUx81 low level UART output data port is supported by ACEINNA's NAV-VIEW 3.X, a powerful PC-based operating tool that provides complete field configuration, diagnostics, charting of sensor performance, and data logging with playback.

2 Interface

2.1 Electrical Interface

2.1.1 Connector and Mating Connector

The DMUx81ZA main connector is a SAMTEC FTM-110-02-F-DV-P defined in Figure 1. The mating connector that can be used is the SAMTEC CLM-110-02.

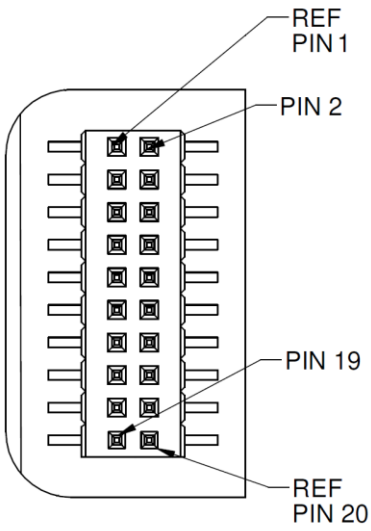


Figure 1 DMUx81ZA Interface Connector

Table 3 Interface Connector Pin Definition DMUx80 vs. DMUx81 (Previous Generation)

Pin	Pin Description- DMUx81 (For Reference)	Pin Description (DMUx81)
1	Reserved – factory use only	Inertial-Sensor Sampling Indicator (sampling upon falling edge)
2	Synchronization Input (1KHz Pulse used to synchronize SPI Comm) / 1PPS Input (External GPS)	Synchronization Input (1KHz Pulse used to synchronize SPI Comm) / 1PPS Input (External GPS)
3	SPI Clock (SCLK) / UART TX	SPI Clock (SCLK) / UART TX
4	SPI Data Output (MISO) / UART RX	SPI Data Output (MISO) / UART RX
5	SPI Data Input (MOSI)	SPI Data Input (MOSI)
6	SPI Chip Select (SS)	SPI Chip Select (SS)
7	Data Ready (SPI Communication Data Ready) / SPI-UART Port Select	Data Ready (SPI Communication Data Ready) / SPI-UART Port Select

8	External Reset (NRST)	External Reset (NRST)
9	Reserved – factory use only	Reserved – factory use only
10	Power VIN (3-5 VDC)	Power VIN (3-5 VDC)
11	Power VIN (3-5 VDC)	Power VIN (3-5 VDC)
12	Power VIN (3-5 VDC)	Power VIN (3-5 VDC)
13	Power GND	Power GND
14	Power GND	Power GND
15	Power GND	Power GND
16	Reserved – factory use only	Reserved – factory use only
17	External GPS UART TX	External GPS UART TX
18	Reserved – factory use only	Reserved – factory use only
19	External GPS UART RX	External GPS UART RX
20	Reserved – factory use only	Reserved – factory use only

2.1.2 Power Input and Power Input Ground

Power is applied to the DMUx81ZA on pins 10 through 15. Pins 13-15 are ground; Pins 10-12 accepts 3 to 5 VDC unregulated input. Note that these are redundant power ground input pairs.

⚠ WARNING

Do not reverse the power leads or damage may occur. Do not add greater than 5.5 volts on the power pins or damage may occur. This system has no reverse voltage or over-voltage protection.

2.1.3 Serial Data Interface

The user can select the serial interface used with the DMUx81ZA by controlling the logic level on connector pin 7 at system startup. If pin 7 is left floating then the DMUx81ZA is configured for SPI communications on pins 3-6. Pin 7 is set as an output and used as the DATA READY discrete for SPI communications. Additionally, the user can synchronize the output data on the SPI port by providing a 1 KHz input pulse on Pin 2. For the complete SPI interface definition, please refer to Section 5.

If the connector pin 7 is grounded then the DMUx81ZA is configured for low-level UART output on pins 3 and 4. This is a standard UART asynchronous output data port. For the complete UART interface definition, please refer to Sections 6-8. Note that the two output port interface methods are controlled independently from each other. The UART port output controls apply only to data being output over the UART port, and the SPI output controls apply only to data being output over the SPI port.

2.1.4 External GPS Aiding (VGx81ZA, AHRSx81ZA and INSx81ZA)

The VGx81ZA/AHRSx81ZA/INSx81ZA allows the use of an external GPS receiver to be connected the external GPS UART port (pins 17, 19). The user is required to configure the GPS receiver to output the GPS messages that the DMUx81ZA Series expects. **Table 4** shows the supported GPS protocols and guidelines for configuration. Note that the details of the GPS messages can be found in the respective GPS protocol documents. The user must configure the VG/AHRS/INSx81ZA to accept external GPS information using NAV-VIEW as described in Appendix A. If the VG/AHRS/INSx81ZA is parsing valid external GPS data and the GPS receiver has 3D lock, then the comStatus → noExternalGPS flag will be zero, otherwise it will be one. See Section 9 for a complete description of system status indications.

Since NMEA protocol does not provide vertical velocity, the vertical velocity that the DMUx81ZA Series estimates (based upon GPS altitude changes) may not be sufficient for airborne applications (see **Table 4**). Therefore, the NMEA protocol is not recommended for airborne applications.

Table 4 External GPS Receiver for VG/AHRS/INSx81ZA

Protocols	Required Messages	Required Message Rate	Baud rate
SIRF Binary		1 Hz	115,200
NovAtel Binary	BestPosB, BestVelB	1Hz	115,200
NMEA*	GPGGA, GPVTG	1Hz	115,200

*Not recommended for airborne applications.

The external GPS UART port should be configured to 8 data bits, 1 start bit, 1 stop bit, no parity bit, and no flow control.

2.1.4.1 1 PPS Input Interface

When using external GPS aiding for a VG/AHRS/INSx81ZA system, Pin 2 should be used for the 1PPS input signal to force synchronization of sensor data collection to a 1Hz rising-edge signal. The signal must maintain 0.0-0.2 V zero logic and 3.0-5.0 volts high logic and stay within 100ms of the internal system 1 second timing. Sending this signal to the system will align the sensor data collection and algorithm processing to its rising edge and 10ms boundaries thereafter.

2.1.4.2 SPI Com Synchronization Input

If the user does not have 1PPS available from an external GPS receiver, then Pin 2 can be used as a sync pulse to force synchronization of sensor data collection to a 1 kHz rising-edge signal for output over the SPI port. See Section 5.9 for a more complete description.

2.1.4.3 External GPS Receiver Antenna Connection

The external GPS receiver needs to receive signals from as many satellites as possible. A GPS receiver doesn't work properly in narrow streets and underground parking lots or if objects or human beings cover the antenna. Poor visibility may result in position drift or a prolonged Time-To-First-Fix (TTFF). A good sky visibility is therefore a prerequisite. Even the best receiver can't make up for signal loss due to a poor antenna, in-band jamming or a poor RF cable. Placing the antenna on a 4 inch or larger ground plane is highly recommended.

IMPORTANT

Place the antenna with optimal sky visibility and use a ground plane. Route the GPS Antenna RF cable away from sources of radiated energy (i.e. switching power supplies).

2.1.5 Reserved – Factory Use Only

During normal operation of the DMUx81ZA, no connection is made to the Reserved – factory use only pins. These pins have internal pull-up mechanisms and must have no connections for the DMUx81ZA to operate properly.

2.2 Mechanical Interface

The DMUx81ZA mechanical interface is defined by the outline drawing in

Figure 2.

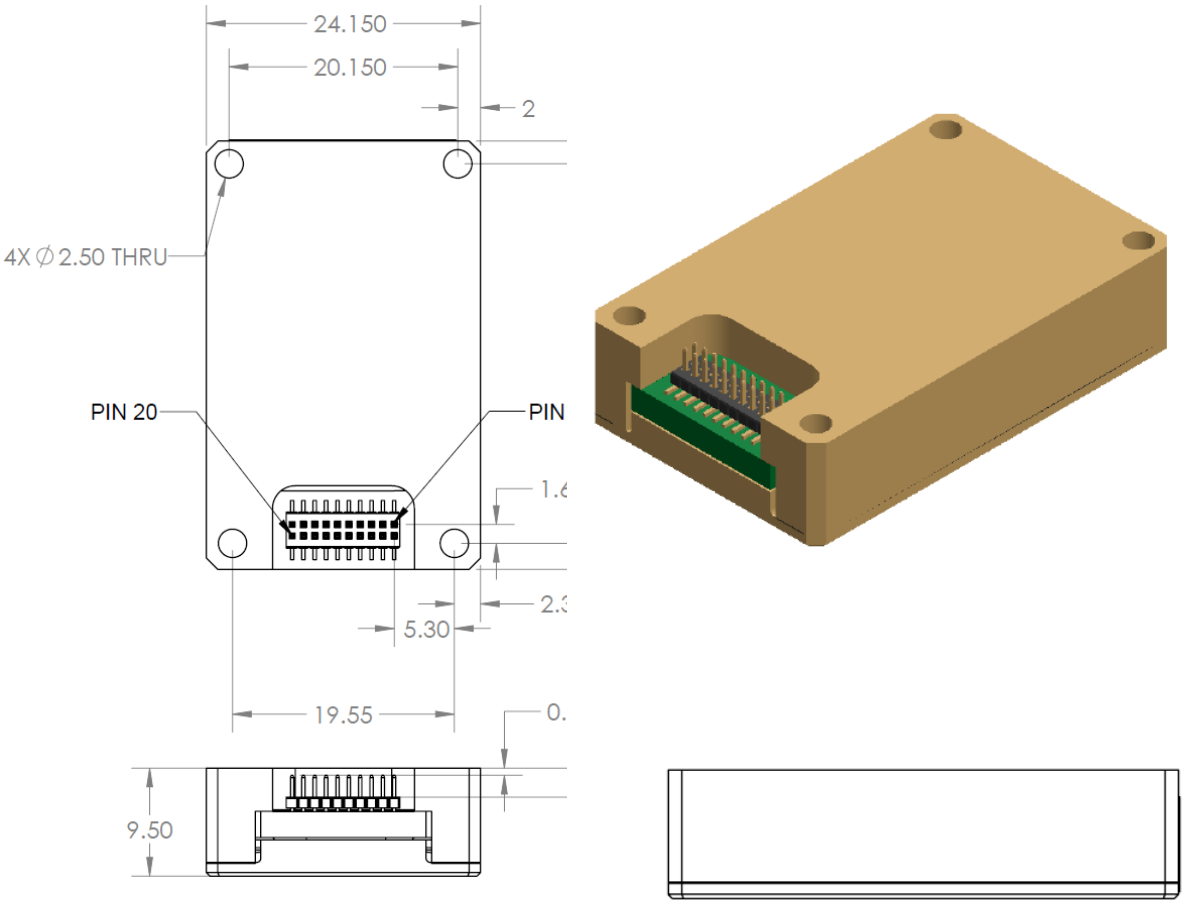


Figure 2 DMUx81ZA Outline Drawing

NOTES UNLESS OTHERWISE STATED:

1) MATING CONNECTOR SAMTEC CLM-110-02
2) DIMENSION TO CENTROID OF PIN ONE

3 Theory of Operation

This section of the manual covers detailed theory of operation for each member of the DMUx81ZA Series starting with the basic IMUx81ZA and then reviewing each major variant (VG, AHRS and INS) with their associated additional features, outputs, and performance. Table 5 shows the basic features of each member of the DMUx81ZA Series with cross references to important sections for review.

Table 5 DMUx81ZA Series Overview

Product	Features	Learning More
IMUx81ZA	6-DOF IMU, 9-DOF IMU	Read 3.1 and 3.2
VGx81ZA	6-DOF IMU Roll, Pitch	Read 3.1, 3.2, and 3.3
AHRSx81ZA	9-DOF IMU (3-Axis Internal Magnetometer) Roll, Pitch, and Heading	Read 3.1, 3.2, 3.3, and 3.4
INSx81ZA	9-DOF IMU (3-Axis Internal Magnetometer and external GPS Receiver) Position, Dynamic Velocity, Roll, Pitch, and Heading	Read 3.1, 3.2, 3.3, 3.4, and 3.5

Figure 3 shows the DMUx81ZA Series hardware block diagram. At the core of the DMUx81ZA Series is a rugged 6-DOF (Degrees of Freedom) MEMS inertial sensor cluster that is common across all members of the DMUx81ZA 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 bulk micromachining technology. Each sensor within the cluster is individually factory calibrated using ACEINNA's automated manufacturing process. Sensor errors are compensated for 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 ^{0}C to +71 ^{0}C) are compensated and verified using calibrated thermal chambers and rate tables.

The 6-DOF inertial sensor cluster data is fed into a high speed signal processing chain, which provides the sensor compensation and digital filtering. The processor also calculates attitude and navigation data for the appropriate models (VG, AHRS and INS). Measurement data packets are available at fixed continuous output rates or on a polled basis from the SPI port or the UART port. The SPI port outputs data via registers, and the user can perform polled reads of each register, or a block burst read of a set of predefined registers. Output data over the SPI port can be synchronized to an external 1 KHz pulse. Alternatively, users can input a 1 PPS signal from an external GPS receiver when providing external GPS data over the secondary UART2 port. The complete SPI interface is defined in Section 4. The UART port outputs data packets are asynchronous and defined in Sections 5-7. As shown in the block diagram (Figure 3), the INSx81ZA and AHRSx81ZA include an internal 3-axis magnetometer.

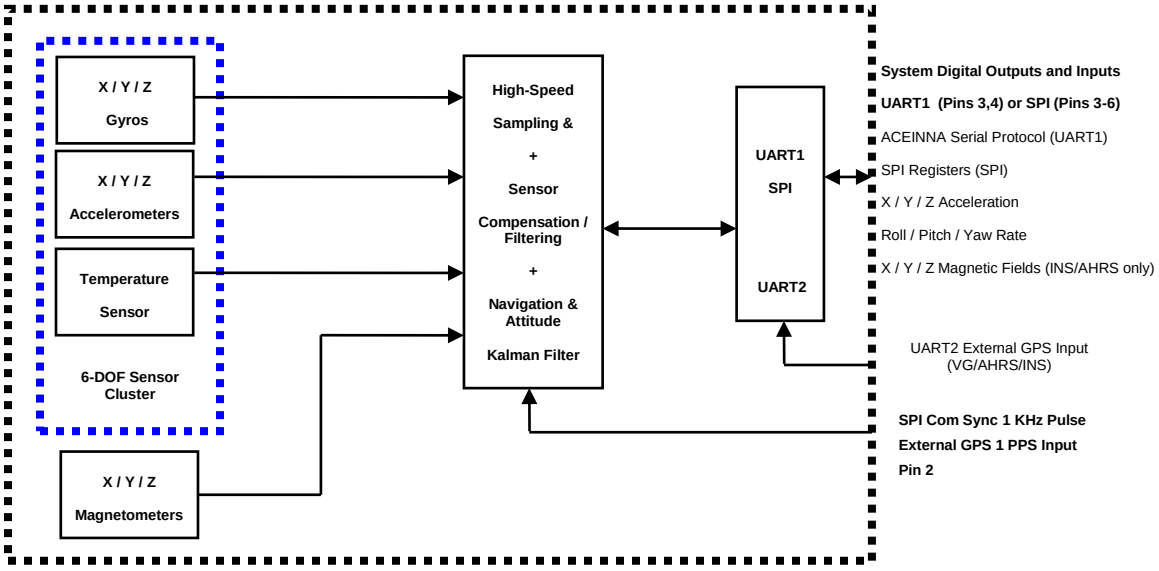


Figure 3 DMUx81ZA Series Hardware Block Diagram

Figure 4 shows the software block diagram. The 6-DOF inertial sensor cluster data is fed into a high speed 200Hz signal processing chain. These 6-DOF signals pass through one or more of the processing blocks and these signals are converted into output measurement data as shown. Measurement data packets are available at fixed continuous output rates or on a polled basis. The type of measurement data packets available depends on the unit type according to the software block diagram and Table . Aiding sensor data is used by an Extended Kalman Filter (EKF) for drift correction in the INS, AHRS and VG Series products. Built-In-Test and Status data is available in the measurement packet or via the special Status Packet T0.

As shown in the software block diagram, the DMUx81ZA Series 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 DMUx81ZA Series architecture as it allows the DMUx81ZA Series to work in a wide range of commercial applications by settings different modes of operation for the DMUx81ZA Series.

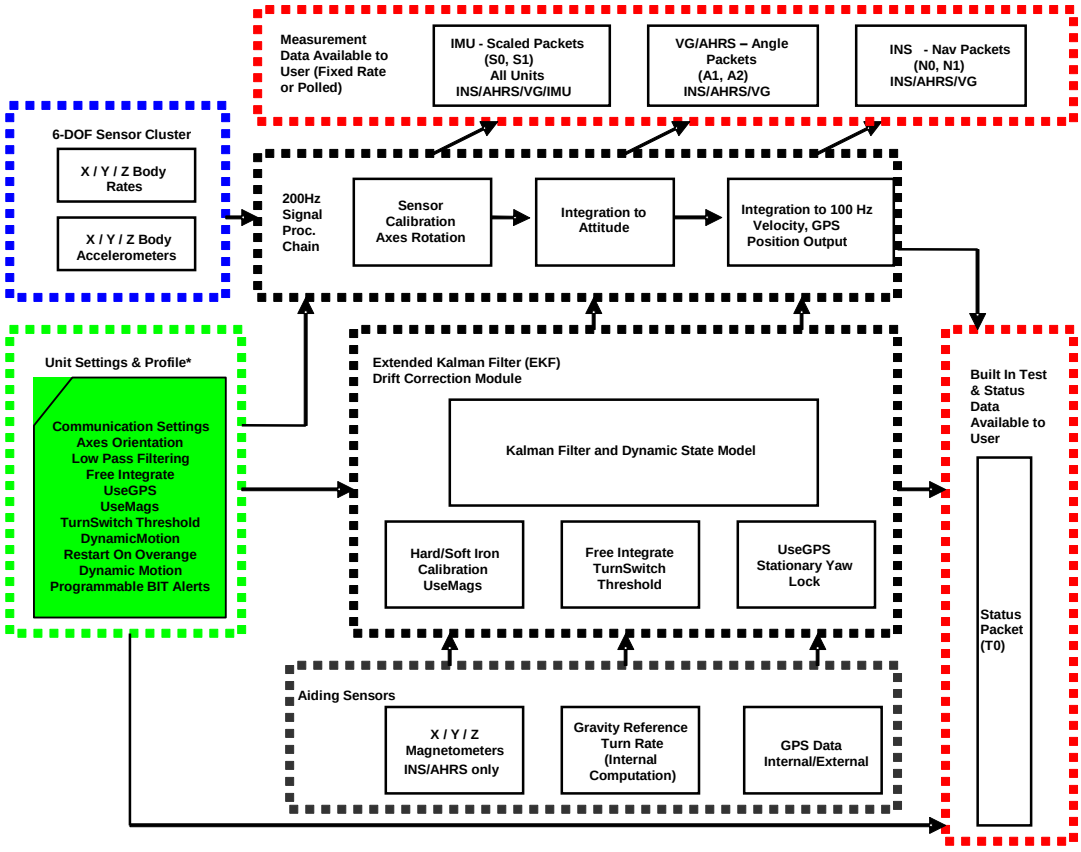


Figure 4 DMUx81ZA Series Software Block Diagram

Simplified functional block diagrams for INS, AHRS and VG series products derived from Figure 4 are shown in Figure 5 to highlight key features of each product. The DMUx81ZA Series products are mainly differentiated by types of aiding sensors used in the EKF for the drift correction of the 6-DOF inertial sensor cluster.

For the AHRS product, a 3-axis magnetometer is used for correcting the drift on yaw/heading angle. For the INS product, a 3-axis magnetometer and a GPS receiver are used for correcting the drift on yaw/heading angle, increasing the accuracy of the attitude estimation by incorporating these sensor signals into the EKF, and providing a navigation solution. 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 VG product.

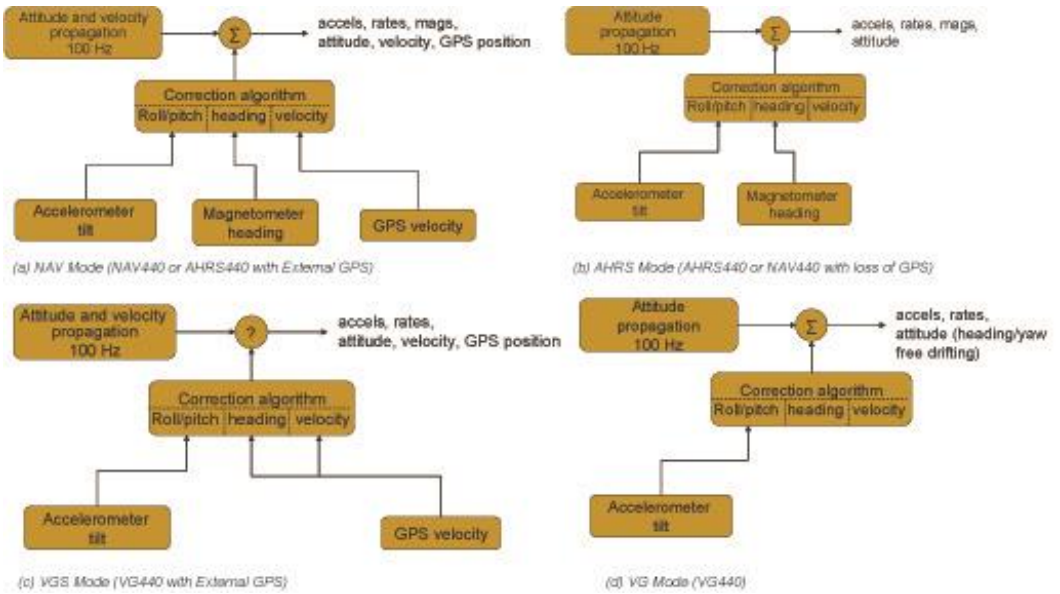
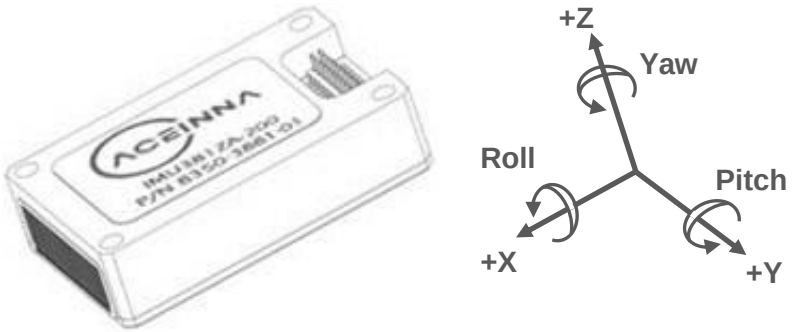


Figure 5 Functional Block Diagram of INS, AHRS and VG Default Operating Mode

3.1 DMUx81ZA Series Default Coordinate System

The DMUx81ZA Series Inertial System default coordinate systems are shown in Figure 6 and Figure 7. As with many elements of the DMUx81ZA Series, the coordinate system is configurable with either NAV-VIEW or by sending the appropriate serial commands over the UART port. These configurable elements are known as ***Advanced Settings***. This section of the manual describes the default coordinate system settings of the DMUx81ZA Series when it leaves



the factory.

Figure 6 IMU381ZA-200 Default Coordinate Frame

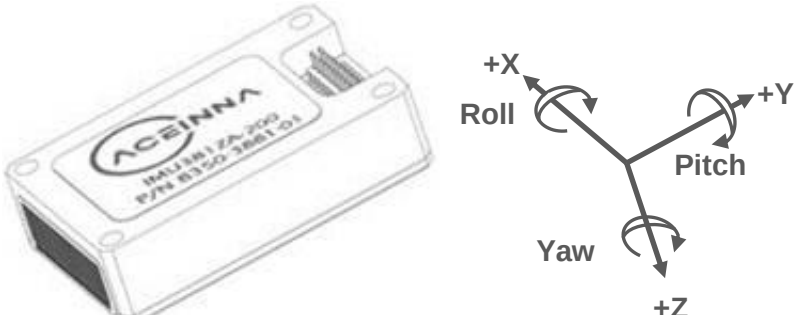


Figure 7 IMU381ZA (-209, -409) VG/AHRS/INS381ZA (-200, -400) Default Coordinate Frame

it is oriented towards the positive side of the coordinate axis. For example, with a DMUx81ZA 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 DMUx81ZA 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 DMUx81ZA 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.

The magnetic sensors are aligned with the same axes definitions and sign as the linear accelerometers. For example, when oriented towards magnetic North, you will read approximately +0.25 Gauss along X, 0.0 Gauss along Y, and +0.35 Gauss along Z direction (North America). Magnetic values at other geographic locations can be found at https://ngdc.noaa.gov/geomag/WMM/DoDWMM.shtml.

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 body frame to an earth-level frame, roll first, then pitch, and then yaw.

The position output from GPS is represented in Latitude, Longitude, and Altitude (LLA) convention on the WGS84 Ellipsoid. This is the most commonly used spherical coordinate system. The GPS velocity is defined in North, East and Down reference frame. The users can convert this into Cartesian coordinate system, called Earth-Centered, Earth-Fixed (ECEF). ECEF uses three-dimensional XYZ coordinates (in meters) to describe the location of a GPS user or satellite. See Appendix E.

3.1.1 Advanced Settings

The DMUx81ZA Series Inertial Systems have a number of advanced settings that can be changed. The specific settings available vary from unit to unit, and a detailed description of each unit (IMU, VG, AHRS, and INS) is found in the subsequent sections of this manual. All units

support baud rate[1], power-up output packet type, output rate, sensor low pass filtering, and custom axes configuration. The units can be configured using NAV-VIEW, as described in Appendix A, and also directly with serial commands as described in Sections 6-9.

3.2 IMUx81ZA Theory of Operation

The product name, IMUx81ZA, stands for Inertial Measurement Unit x81, and the name is indicative of the inertial measurement unit functionality that the IMUx81ZA provides by providing inertial rate and acceleration data in 6-DOF (six degrees of freedom). The IMUx81ZA signal processing chain consists of the 6-DOF sensor cluster, programmable low-pass filters, and the signal processor for sensor error compensation. The IMUx81ZA, as with other DMUx81ZA Series variants, has a UART input/output port and a SPI input/output port.

After passing through a digitally controlled programmable low-pass filter, the rate and acceleration sensor signals are obtained at 200Hz. The sensor data is filtered by the processor using FIR filters. The factory calibration data, stored in EEPROM, is used by the processor to remove temperature bias, misalignment, scale factor errors, and non-linearities from the sensor data. Additionally any advanced user settings such as axes rotation are applied to the IMU data. The 200Hz IMU data is continuously being maintained inside the IMUx81ZA, and is available at 200Hz on the SPI output port registers. Digital IMU data is output over the UART port at a selectable fixed rate (100, 50, 25, 20, 10, 5 or 2 Hz) or on as requested basis using the GP, 'Get Packet' command. The digital IMU data is available in one of several measurement packet formats including Scaled Sensor Data ('S1' Packet). In the Scaled Sensor Data ('S1' Packet) data is output in scaled engineering units. See Section 7 of the manual for full packet descriptions.

3.2.1 IMUx81ZA Advanced Settings

The IMUx81ZA advanced settings are described in Table 6. All of the advanced settings are accessible through NAV-VIEW under the Configuration Menu, Unit Configuration settings. For a full definition of the SPI port please see section 5.

Table 6 IMUx81ZA Advanced Settings

Setting	Default	Comments
Baud Rate	38,400 baud	57600, 115200, and 230400 also available
Packet Type	S0	S1 also available
Packet Rate	100Hz	This setting sets the rate at which selected Packet Type, packets are output. If polled mode is desired, then select Quiet. If Quiet is selected, the IMUx81ZA will only send measurement packets in response to GP commands.
Orientation	See Figure 6 and Figure 7.	To configure the axis orientation, select the desired measurement for each axis: NAV-VIEW will show the corresponding image of the IMUx81ZA, so it easy to visualize the mode of operation. Refer to Section 8.4 Orientation Field settings for the twenty four possible orientation settings.
Filter Settings (unfiltered, 2, 5, 10, 20, 25, 40 50	20 Hz	The low pass filters are set to a default of 20 Hz for the accelerometers, and 20 Hz for the angular rate sensors. There is one filter setting for all three angular rate sensors. There is one filter setting for all three accelerometers. Setting

[1] Note: certain combinations of baud-rate, packet-type, and output data rate are invalid because the time to transmit the data exceeds a limit on the permissible message length. The DMU381 limits the output packet width to 80% of the time between data packets. For instance, if the packet is output every 10 milliseconds (100 Hz) then the packet width must be less than 8 milliseconds or the combination is not allowed. This prevents messages from overlapping and causing communication problems. For this reason, 57.6 kbps and higher baud-rates are suggested.

Hz).		either to zero disables the low-pass filter.
BIT		See 9.1

NOTE on Filter Settings

Why change the filter settings? Generally there is no reason to change the low-pass filter settings on the IMUx81ZA or other DMUx81ZA Series Inertial Systems. However, when a DMUx81ZA 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 IMUx81ZA 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 filter settings (e.g., 5Hz) 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.

3.2.2 IMUx81ZA Built-In Test

The IMUx81ZA Built-In Test capability allows users of the IMUx81ZA to monitor health, diagnostic, and system status information of the unit in real-time. The Built-In Test information consists of a BIT word (2 bytes) transmitted in every measurement packet. In addition, there is a diagnostic packet 'T0' that can be requested via the Get Packet 'GP' command which contains a complete set of status for each hardware and software subsystem in the IMUx81ZA. See Sections 6-8 for details on the 'T0' packet.

The BIT word, which is contained within each measurement packet, is detailed below. The LSB (Least Significant Bit) is the Error byte, and the MSB (Most Significant Bit) is a Status byte with programmable alerts. Internal health and status are monitored and communicated in both hardware and software. The ultimate indication of a fatal problem is the masterFail flag.

The masterStatus flag is a configurable indication that can be modified by the user. This flag is asserted as a result of any asserted alert signals which have been enabled. See Advanced BIT (Section 9) for details regarding the configuration of the masterStatus flags. Table 7 shows the BIT definition and default settings for BIT programmable alerts in the IMUx81ZA.

Table 7 IMUx81ZA Default BIT Status Definition

BITstatus Field	*Bits*	*Meaning*	*Category*
masterFail	0	0 = normal, 1 = fatal error has occurred	BIT
HardwareError	1	0 = normal, 1= internal hardware error	BIT
comError	2	0 = normal, 1 = communication error	BIT
softwareError	3	0 = normal, 1 = internal software error	BIT
Reserved	4:7	N/A	
masterStatus	8	0 = nominal, 1 = Alert, Sensor Over Range	Status
hardwareStatus	9	Disabled	Status
comStatus	10	Disabled	Status
softwareStatus	11	Disabled	Status
sensorStatus	12	0 = nominal, 1 = Sensor Over Range	Status
Reserved	13:15	N/A	

The IMUx81ZA also allows a user to configure the Status byte within the BIT message. To configure the word, select the BIT Configuration tab from the Unit Configuration menu. The dialog box allows selection of which status types to enable (hardware, software, sensor, and comm). In the case of the IMUx81ZA which has fewer features and options than other DMUx81ZA Series products, the only meaningful parameter is sensor over-range. It is recommended that users leave the default configuration, which is sensorStatus enabled and flag on sensor over-range. The over-range only applies to the rotational rate sensors. Because instantaneous acceleration levels due to vibration can exceed the accelerometer sensor range in many applications, none of the DMUx81ZA Series products trigger over-range on accelerometer readings.

3.3 VGx81ZA Theory of Operation

The VGx81ZA supports all of the features and operating modes of the IMUx81ZA, and it includes additional internal software, running on the processor, for the computation of dynamic roll and pitch. The product name, VGx81ZA, stands for Vertical Gyro x81, and it is indicative of the vertical gyro functionality that the VGx81ZA replicates by providing dynamic roll and pitch measurements, in addition to the IMU data. The dynamic roll and pitch measurements are stabilized by the using the accelerometers as a long-term gravity reference. Unlike the VG400 and earlier ACEINNA VG Series products, the VGx81ZA can also output a free integrating yaw angle measurement that is not stabilized by a magnetometer or compass heading (see AHRSx81ZA or INSx81ZA for stabilized heading). At a fixed 200Hz rate, the VGx81ZA continuously maintains both the digital IMU data as well as the dynamic roll and pitch data. As shown in the software block diagram Figure 4, after the Sensor Calibration block, the IMU data is passed into an Integration to Orientation block (Please refer to the Figure 5 if external GPS aiding will be used). The Integration to Orientation block integrates body frame sensed angular rate to orientation at a fixed 200 times per second within all of the DMUx81ZA Series products. For improved accuracy and to avoid singularities when dealing with the cosine rotation matrix, 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 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 Extended Kalman Filter (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 VGx81ZA, the internally computed gravity reference vector provides a reference measurement for the EKF when the VGx81ZA 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 VGx81ZA 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 VGx81ZA 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 VGx81ZA is attached exceeds the turnSwitch threshold yaw rate, the VGx81ZA 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 VGx81ZA algorithm has two major phases of operation. The first phase of operation is the initialization phase. During the initialization phase, the VGx81ZA is expected to be stationary or quasi-static so the EKF weights the accelerometer gravity reference heavily in order to rapidly estimate the roll and pitch angles, and X, Y rate sensor bias. The initialization phase lasts approximately 60 seconds, and the initialization phase can be monitored in the softwareStatus BIT transmitted by default in each measurement packet. After the initialization phase, the VGx81ZA operates with lower levels of feedback (also referred to as EKF gain) from the accelerometers 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 VGx81ZA outputs digital measurement data over the UART port at a selectable fixed rate (100, 50, 25, 20, 10, 5 or 2 Hz) or on as requested basis using the GP, 'Get Packet' command. In addition to the scaled sensor packets described in the IMUx81ZA section, the VGx81ZA has additional measurement output packets including the default 'A2' Angle Packet which outputs the roll angle, pitch angle, and digital IMU data. 'N0' and 'N1' packets are also available for use with an external GPS receiver. See Section 6 and 7 of the manual for full packet descriptions. All data is also available on the SPI output port registers. Please refer to section 5 for a complete description of the SPI port functionality.

3.3.1 VGx81ZA Advanced Settings

In addition to the configurable baud rate, packet rate, axis orientation, and sensor low-pass filter settings, the VGx81ZA provides additional advanced settings which are selectable for tailoring the VGx81ZA to a specific application requirements. These VGx81ZA advanced settings are shown in Table 8 below:

Table 8 VGx81ZA Series Advanced Settings

Setting	*Default*	*Comments*
Baud Rate	38,400 baud	57600, 115200, and 230400 also available
Packet Type	A2	S1, N0, N1 also available
Packet Rate	25Hz	This setting sets the rate at which selected Packet Type, packets are output. If polled mode is desired, then select Quiet. If Quiet is selected, the VGx81ZA will only send measurement packets in response to GP commands.
Orientation	See Figure 6 and Figure 7.	To configure the axis orientation, select the desired measurement for each axes: NAV-VIEW will show the corresponding image of the VGx81ZA, so it easy to visualize the mode of operation. See Section 8.4 Orientation Field settings for the twenty four possible orientation settings. The default setting points the connector AFT.
Filter Settings (unfiltered, 2, 5, 10, 20, 25, 40, 50 Hz)	20 Hz	The low pass filters are set to a default of 5Hz for the accelerometers, and 20 Hz for the angular rate sensors. There is one filter setting for all three angular rate sensors. There are two settings for the accelerometers, one for the X and Y axes, and a separate setting for the Z axis. The reason for a separate setting in the Z-axis is that in many installations, the Z-axis vibration level is much higher than in the X and Y axes, and it can prove helpful to filter the Z-axis at a lower cutoff than the X and Y axes. Setting either to zero disables

		the low-pass filter.
Freely Integrate	OFF	The Freely Integrate setting allows a user to turn the VGx81ZA into a 'free gyro'. In free gyro mode, the roll, pitch and yaw are computed exclusively from angular rate with no Kalman filter based corrections of roll, pitch, or yaw. When turned on, there is no coupling of acceleration based signals into the roll and pitch. As a result, the roll, pitch, and yaw outputs will drift roughly linearly with time due to sensor bias. For best performance, the Freely Integrate mode should be used after the algorithm has initialized. This allows the Kalman Filter to estimate the roll and pitch rate sensor bias prior to entering the free gyro mode. Upon exiting the 'free gyro' mode (OFF), one of two behaviors will occur (1) If the VGx81ZA has been in freely integrate mode for less than sixty seconds, the algorithm will resume operation at normal gain settings (2) If the VGx81ZA has been in freely integrate mode for greater than sixty seconds, the algorithm will force a reset and reinitialize with high gains automatically.
Restart On Over Range	OFF	This setting forces an algorithm reset when a sensor over range occurs i.e., a rotational rate on any of the three axes exceeds the maximum range. The default setting is OFF for the VGx81ZA. Algorithm reset returns the VGx81ZA to a high gain state, where the VGx81ZA rapidly estimates the gyro bias and uses the accelerometer feedback heavily. This setting is recommended when the source of over-range is likely to be sustained and potentially much greater than the rate sensor operating limit. Large and sustained angular rate over-ranges result in unrecoverable errors in roll and pitch outputs. An unrecoverable error is one where the EKF can not stabilize the resulting roll and pitch reading. If the over-ranges are expected to be of short duration (<1 sec) and a modest percentage over the maximum operating range, it is recommended that the restart on over range setting be turned off. Handling of an inertial rate sensor over-range is controlled using the restartOnOverRange switch. If this switch is off, the system will flag the overRange status flag and continue to operate through it. If this switch is on, the system will flag a masterFail error during an over-range condition and continue to operate with this flag until a quasi-static condition is met to allow for an algorithm restart. The quasi-static condition required is that the absolute value of each low-passed rate sensor fall below 3 deg/sec to begin initialization. The system will then attempt a normal algorithm start.
Dynamic Motion	ON	The default setting is ON for the VGx81ZA. Turning off the dynamic motion setting results in a higher gain state that uses the accelerometer feedback heavily. During periods of time when there is known low dynamic acceleration, this switch can be turned off to allow the attitude estimate to quickly stabilize.
Turn Switch threshold	10.0 deg/sec	With respect to centripetal or false gravity forces from turning dynamics (or coordinated turn), the VGx81ZA monitors the yaw-rate. If the yaw rate exceeds a given Turnswitch threshold, the feedback gains from the accelerometer signals for attitude correction are reduced because they are likely corrupted.
BIT		See 4.3.2

3.3.2 VGx81ZA Built-In Test

As with the IMUx81ZA, the VGx81ZA Built-In Test capability allows users of the VGx81ZA to monitor health, diagnostic, and system status information of the unit in real-time. The Built-In Test information consists of a BIT word (2 bytes) transmitted in every measurement packet. In addition, there is a diagnostic packet 'T0' that can be requested via the Get Packet 'GP' command which contains a complete set of status for each hardware and software subsystem in the VGx81ZA. See Sections 6-8 for details on the 'T0' packet.

The BIT word contained within each measurement packet is detailed below. The LSB (Least Significant Bit) is the Error byte, and the MSB (Most Significant Bit) is a Status byte with programmable alerts. Internal health and status are monitored and communicated in both hardware and software. The ultimate indication of a fatal problem is the masterFail flag.

The masterStatus flag is a configurable indication that can be modified by the user. This flag is asserted as a result of any asserted alert signals which have been enabled. See Advanced BIT

(Section 9) for details on configuring the masterStatus flags. Table 9 shows the BIT definition and default settings for BIT programmable alerts in the VGx81ZA.

Table 9 VGx81ZA Default BIT Status Definition

BITstatus Field	*Bits*	*Meaning*	*Category*
masterFail	0	0 = normal, 1 = fatal error has occurred	BIT
HardwareError	1	0 = normal, 1= internal hardware error	BIT
comError	2	0 = normal, 1 = communication error	BIT
softwareError	3	0 = normal, 1 = internal software error	BIT
Reserved	4:7	N/A	
masterStatus	8	0 = nominal, 1 = one or more status alerts	Status
hardwareStatus	9	Disabled	Status
comStatus	10	0 = nominal, 1 = No External GPS Comm	Status
softwareStatus	11	0 = nominal, 1 = Algorithm Initialization or High Gain	Status
sensorStatus	12	0 = nominal, 1 = Sensor Over Range	Status
Reserved	13:15	N/A	

The VGx81ZA also allows a user to configure the Status byte within the BIT message. To configure the word, select the BIT Configuration tab from the Unit Configuration menu. The dialog box allows selection of which status types to enable (hardware, software, sensor, and comm). Like the IMUx81ZA, ACEINNA recommends for the vast majority of users, that the default Status byte for the VGx81ZA is sufficient. For users, who wish to have additional visibility to when the VGx81ZA EFK algorithm estimates that the VGx81ZA is turning about its Z or Yaw axis, the softwareStatus bit can be configured to go high during a turn. In other words, the turnSwitch will turn on the softwareStatus bit. In the VGx81ZA, the turnSwitch is by default set at 10.0 deg/sec about the z-axis.

3.4 AHRSx81ZA Theory of Operation

The AHRSx81ZA supports all of the features and operating modes of the IMUx81ZA and VGx81ZA, and it includes an additional internal 3-axis magnetometer and associated software running on the processor, for the computation of dynamic heading, as well as dynamic roll and pitch. The product name, AHRSx81ZA, stands for Attitude Heading Reference System x81, and it is indicative of the attitude and heading reference functionality that the AHRSx81ZA replicates by providing dynamic heading, roll, and pitch measurements, in addition to the VG and IMU data. The dynamic heading measurement is stabilized using the 3-axis magnetometer as a magnetic north reference. As in the VGx81ZA, the dynamic roll and pitch measurements are stabilized using the accelerometers as a long-term gravity reference. Unlike the AHRS400 and earlier ACEINNA AHRS Series products, the AHRSx81ZA can be configured to turn on and off the magnetic reference for user defined periods of time (see Section 8 Advanced Commands). In addition, the AHRSx81ZA can accept external GPS data (refer to the INSx81ZA section for details) for improved performance.

At a fixed 200Hz rate, the AHRSx81ZA continuously maintains the digital IMU data as well as the dynamic roll, pitch, and heading. As shown in Figure 4, after the Sensor Calibration Block, the IMU data is passed to the 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 all of the DMUx81ZA Series products. For improved accuracy and to avoid singularities when dealing with the cosine rotation matrix, 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 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 Extended Kalman Filter (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 AHRSx81ZA, the internally computed gravity reference vector and the distortion corrected magnetic field vector provide an attitude and a heading reference measurement for the EKF when the AHRSx81ZA is in quasi-static motion to correct roll, pitch, and heading angle drift and to estimate the X, Y and Z gyro rate bias. The AHRSx81ZA adaptively tunes the EKF feedback gains 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 AHRSx81ZA maintains a low-passed filtered yaw rate signal and compares it to the turnSwitch threshold field (user adjustable). When the user platform (with the AHRSx81ZA attached) exceeds the turnSwitch threshold yaw rate, the AHRSx81ZA 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.

As described in 3.3 VGx81ZA theory of operation, the AHRSx81ZA algorithm also has two major phases of operation. The first phase of operation is the high-gain initialization phase. During the initialization phase, the AHRSx81ZA is expected to be stationary or quasi-static so the EKF weights the accelerometer gravity reference and Earth's magnetic field reference heavily in order to rapidly estimate the X, Y, and Z rate sensor bias, and the initial attitude and heading of the AHRSx81ZA. The initialization phase lasts approximately 60 seconds, and the initialization phase can be monitored in the softwareStatus BIT transmitted by default in each measurement packet. After the initialization phase, the AHRSx81ZA operates with lower levels of feedback (also referred to as EKF gain) from the accelerometers and magnetometers to continuously estimate and correct for roll, pitch, and heading (yaw) errors, as well as to estimate X, Y, and Z rate sensor bias.

The AHRSx81ZA digital data is output over the UART port at a selectable fixed rate (100, 50, 25, 20, 10, 5 or 2 Hz) or on as requested basis using the GP, 'Get Packet' command. The AHRS400 supports the same scaled sensor and angle mode packet format of the VGx81ZA. The AHRSx81ZA defaults to the 'A1' Angle Packet which outputs the roll angle, pitch angle, yaw angle, and digital IMU data. In the AHRSx81ZA, the 'A1' packet contains accurate magnetometer readings. See Sections 6 and 7 of the manual for full packet descriptions. All data

is also available on the SPI output port registers. Please refer to section 5 for a complete description of the SPI port functionality.

IMPORTANT

For proper operation, the AHRSx81ZA relies on magnetic field readings from its internal 3-axis magnetometer. The AHRSx81ZA must be installed correctly and calibrated for hard-iron and soft iron effects to avoid any system performance degradation. See section 3.4.1 for information and tips regarding installation and calibration.

3.4.1 AHRSx81ZA Magnetometer Calibration and Alignment

The AHRSx81ZA uses magnetic sensors to compute heading. Ideally, the magnetic sensors would measure only the earth's magnetic field to compute the heading angle. In the real world, however, residual magnetism in your system will add to the magnetic field measured by the AHRSx81ZA. This extra magnetic field will create errors in the heading measurement if they are not accounted for. These extra magnetic fields are called hard iron magnetic fields. In addition, magnetic material can change the direction of the magnetic field as a function of the input magnetic field. This dependence of the local magnetic field on input direction is called the soft iron effect. The AHRSx81ZA can actually measure any constant magnetic field that is associated with your system and correct for it. The AHRSx81ZA can also make a correction for some soft iron effects. The process of measuring these non-ideal effects and correcting for them is called hard iron and soft iron calibration. This calibration will help correct for magnetic fields that are fixed with respect to the AHRSx81ZA. It cannot help for time varying fields, or fields created by parts that move with respect to the AHRSx81ZA. Because time varying fields cannot be compensated, selection of a proper installation location is important.

During the calibration procedure, the AHRSx81ZA makes a series of measurements while the user system is being turned through a complete 360 degree circle. A 360 degree rotation gives the AHRSx81ZA visibility to hard and soft iron distortion in the horizontal plane. Using NAV-VIEW, a user can see the hard and soft iron effects by selecting the Misalignment option on the Configuration Menu, and viewing the magnetic circle during the calibration.

The AHRSx81ZA uses these measurements to model the hard iron and soft iron environment in your system, and store these as calibration constants in the EEPROM. The status of the AHRSx81ZA magnetometer calibration is indicated by the softwareError→dataError→magAlignOutOfBounds error flag available in the 'T0' packet. The current release of this software does not currently implement this feature however. In future releases, this functionality will be restored. The user can access the hardIron and softIronScaleRatio calibration data as configuration fields in NAV-VIEW, or by using the communication protocol over UART or SPI. Also, the softwareError bit of the masterFail byte within the BIT word is transmitted in every measurement packet. When the AHRSx81ZA has not been properly calibrated, this softwareError bit will be set to fail (high). The current release of this software does not currently implement this feature however. In future releases, this functionality will be restored.

In order for the AHRSx81ZA calibration to work properly, the AHRSx81ZA must be installed in your system prior to calibration. If you perform the calibration process with the AHRSx81ZA by itself, you will only correct for the magnetism in the AHRSx81ZA itself. If you then install the AHRSx81ZA in a vehicle (for instance), and the vehicle is magnetic, you will still see errors arising from the magnetism of the vehicle. The AHRSx81ZA must be calibrated after installation and prior to use of the system

The AHRSx81ZA also provides a command interface for initiating the hard iron / soft iron calibration without the using NAV-VIEW. The user can send a 'WC' command to initiate the calibration, and then rotate the user system through 360 degrees. The 'WC' command has two options – auto-termination and manual termination. With, auto-termination, the AHRSx81ZA tracks the yaw movement and after x81 degrees of rotation returns the calibration complete response, 'CD'. The auto-termination sequence can falsely terminate if the 360 degree rotation is not completed within 2 minutes of the 'WC' command initiation. Manual termination requires the user to send a second 'WC' command with the termination code in the payload. Manual termination is a good option when the user system moves very slowly (e.g., large marine vessel) and completing the 360 degree rotation may require more than two minutes.

The calibration complete, 'CD', command response message contains the X and Y hard iron bias, as well as the soft iron ratio and soft iron phase angle. This information can be interpreted to give an indication of the quality of the calibration. See the section *Hard Iron/Soft Iron Overview* in Appendix A: Installation and Operation of NAV-VIEW for more information on the hard iron bias, soft iron ratio and soft iron phase angle. Section 7 has programming details for the 'WC' and 'CD' commands, as well as the "GF" commands that allow the user to request the parameters committed to EEPROM memory.

⚠ WARNING

The AHRSx81ZA and INSx81ZA units must be mounted at least 24" away from large ferrous objects and fluctuating magnetic fields. Failure to locate the unit in a clean magnetic environment will affect the attitude solution.

3.4.2 AHRSx81ZA Advanced Settings

In addition to the configurable baud rate, packet rate, axis orientation, and sensor low-pass filter settings, the AHRSx81ZA provides additional advanced settings which are selectable for tailoring the AHRSx81ZA to a specific application requirements. The AHRSx81ZA advanced settings are shown in Table 10:

Table 10 AHRSx81ZA Series Advanced Settings

Setting	Default	Comments
Baud Rate	38,400 baud	57600, 115200, and 230400 also available
Packet Type	A1	S0, S1, A2, N0, N1 also available
Packet Rate	25 Hz	This setting sets the rate at which selected Packet Type, packets are output. If polled mode is desired, then select Quiet. If Quiet is selected, the VGx81ZA will only send measurement packets in response to GP commands.
Orientation	See Figure 6 and Figure 7.	To configure the axis orientation, select the desired measurement for each axes: NAV-VIEW will show the corresponding image of the AHRSx81ZA, so it easy to visualize the mode of operation. See section 8.4 Orientation Field settings for the twenty four possible orientation settings. The default setting points the connector AFT.
Filter Settings (unfiltered, 2, 5, 10, 20,	20 Hz accels 20 Hz	The low pass filters are set to a default of 20Hz for the accelerometers, and 20Hz for the angular rate sensors. There is one filter setting for all three angular rate sensors. There is one filter setting for all three accelerometer sensors. The reason for filtering the accelerometers is that in many installations, the vibration level can be high, and it can

25, 40, 50 Hz)	rates	prove helpful to filter accelerometers. Setting either to zero disables the low-pass filter.
Freely Integrate	OFF	The Freely Integrate setting allows a user to turn the AHRSx81ZA into a 'free gyro'. In free gyro mode, the roll, pitch and yaw are computed exclusively from angular rate with no kalman filter based corrections of roll, pitch, or yaw. When turned on, there is no coupling of acceleration based signals into the roll and pitch or magnetometer based signals to the yaw. As a result, the roll, pitch, and yaw outputs will drift roughly linearly with time due to sensor bias. For best performance, the Freely Integrate mode should be used after the algorithm has initialized. This allows the Kalman Filter to estimate the roll and pitch rate sensor bias prior to entering the free gyro mode. Upon exiting the 'free gyro' mode (OFF), one of two behaviors will occur (1) If the AHRSx81ZA has been in freely integrate mode for less than sixty seconds, the algorithm will resume operation at normal gain settings (2) If the AHRSx81ZA has been in freely integrate mode for greater than sixty seconds, the algorithm will force a reset and reinitialize with high gains automatically.
Use Mags	ON	The Use Mags setting allows users to turn on and off the magnetometer feedback for yaw/heading stabilization. The default setting is ON for the AHRSx81ZA. When Use Mags is turned ON, the AHRSx81ZA uses the magnetic field sensor readings to stabilize the drift in yaw, and it slaves the yaw to the compass reading provided from the magnetic field sensor readings. When Use Mags is turned OFF, the heading (yaw) angle measurement of the AHRSx81ZA will drift and freely integrate. In effect, this setting converts an AHRSx81ZA into the functionality of the VGx81ZA. However, unlike a VGx81ZA this can be done on a selectable basis and changed in real time during a mission. The reason for this setting is to give the user an ability to turn off the magnetometer stabilization when severe magnetic distortion may be occurring. This setting is desirable when the user system temporarily moves in close proximity to a large ferrous object. When the Use Mags switch is turned from OFF to ON, the AHRSx81ZA will reinitialize the yaw/heading angle with the compass reading provided from the magnetic field sensor readings.
Restart On Over Range	OFF	This setting forces an algorithm reset when a sensor over range occurs i.e., a rotational rate on any of the three axes exceeds the maximum range. The default setting is OFF for the AHRSx81ZA. Algorithm reset returns the AHRSx81ZA to a high gain state, where the AHRSx81ZA rapidly estimates the gyro bias and uses the accelerometer feedback heavily. This setting is recommended when the source of over-range is likely to be sustained and potentially much greater than the rate sensor operating limit. Large and sustained angular rate over-ranges result in unrecoverable errors in roll and pitch outputs. An unrecoverable error is one where the EKF can not stabilize the resulting roll and pitch reading. If the over-ranges are expected to be of short duration (<1 sec) and a modest percentage over the maximum operating range, it is recommended that the restart on over range setting be turned off. Handling of an inertial rate sensor over-range is controlled using the restartOnOverRange switch. If this switch is off, the system will flag the overRange status flag and continue to operate through it. If this switch is on, the system will flag a masterFail error during an over-range condition and continue to operate with this flag until a quasi-static condition is met to allow for an algorithm restart. The quasi-static condition required is that the absolute value of each low-passed rate sensor fall below 3 deg/sec to begin initialization. The system will then attempt a normal algorithm start.
Dynamic Motion	ON	The default setting is ON for the AHRSx81ZA. Turning off the dynamic motion setting results in a higher gain state that uses the accelerometer feedback heavily. During periods of time when there is known low dynamic acceleration, this switch can be turned off to allow the attitude estimate to quickly stabilize.
Turn Switch threshold	0.5 deg/sec	With respect to centripetal or false gravity forces from turning dynamics (or coordinated turn), the AHRSx81ZA monitors the yaw-rate. If the yaw rate exceeds a given Turnswitch threshold, the feedback gains from the accelerometer signals for attitude correction are

		reduced because they are likely corrupted.
BIT		See 4.4.3

3.4.3 AHRSx81ZA Built-In Test

As with the IMUx81ZA and VGx81ZA, the Built-In Test capability allows users of the AHRSx81ZA to monitor health, diagnostic, and system status information of the unit in real-time. The Built-In Test information consists of a BIT word (2 bytes) transmitted in every measurement packet. In addition, there is a diagnostic packet 'T0' that can be requested via the Get Packet 'GP' command which contains a complete set of status for each hardware and software subsystem in the AHRSx81ZA. See Sections 6 and 7 of the Programming Guide, for details on the 'T0' packet.

The BIT word contained within each measurement packet is detailed below. The LSB (Least Significant Bit)) is the Error byte, and the MSB (Most Significant Bit) is a Status byte with programmable alerts. Internal health and status are monitored and communicated in both hardware and software. The ultimate indication of a fatal problem is the masterFail flag. The softwareError bit also provides useful information regarding the status and quality of the AHRSx81ZA magnetic alignment. If the AHRSx81ZA has not been properly magnetically calibrated, the AHRSx81ZA shall indicate a softwareError.

The masterStatus flag is a configurable indication that can be modified by the user. This flag is asserted as a result of any asserted alert signals which has been enabled. See Section 9 Advanced BIT for details on configuring the masterStatus flags. Table 11 shows the BIT definition and default settings for BIT programmable alerts in the AHRSx81ZA.

Table 11 AHRSx81ZA Default BIT Status Definitions

BITstatus Field	***Bits***	***Meaning***	***Category***
masterFail	0	0 = normal, 1 = fatal error has occurred	BIT
HardwareError	1	0 = normal, 1= internal hardware error	BIT
comError	2	0 = normal, 1 = communication error	BIT
softwareError	3	0 = normal, 1 = internal software error or magAlignOutofBounds	BIT
Reserved	4:7	N/A	
masterStatus	8	0 = nominal, 1 = one or more status alerts	Status
hardwareStatus	9	Disabled	Status
comStatus	10	0 = nominal, 1 = No External GPS Comm	Status
softwareStatus	11	0 = nominal, 1 = Algorithm Initialization, or High Gain	Status
sensorStatus	12	0 = nominal, 1 = Sensor Over Range	Status
Reserved	13:15	N/A	

The AHRSx81ZA also allows a user to configure the Status byte within the BIT message. To configure the word, select the BIT Configuration tab from the Unit Configuration menu. The dialog box allows selection of which status types to enable (hardware, software, sensor, and comm). Like the VGx81ZA and IMUx81ZA, ACEINNA recommends for the vast majority of users, that the default Status byte for the AHRSx81ZA is sufficient. For users, who wish to have additional visibility to when the AHRSx81ZA EFK algorithm estimates that the AHRSx81ZA is turning about its Z or Yaw axis, the softwareStatus bit can be configured to go high during a turn. In other words, the turnSwitch will turn on the softwareStatus bit. In the AHRSx81ZA, the turnSwitch is by default set at 0.5 deg/sec about the Z-axis.

3.5 INSx81ZA Theory of Operation

The INSx81ZA supports all of the features and operating modes of the IMU/VG/AHRSx81ZA, and it includes additional capability of interfacing with an external GPS receiver and associated software running on the processor, for the computation of navigation information as well as orientation information. The product name, INSx81ZA, stands for Inertial Navigation System x81, and it is indicative of the navigation reference functionality that the INSx81ZA provides by outputting inertially-aided navigation information (Latitude, Longitude, and Altitude), inertially-aided 3-axis velocity information, as well as heading, roll, and pitch measurements, in addition to digital IMU data.

At a fixed 100Hz rate, the INSx81ZA continuously maintains the digital IMU data; the dynamic roll, pitch, and heading data; as well as the navigation data. As shown in the software block diagram in Figure 4, after the Sensor Calibration block, the IMU data is passed into an "Integration to Orientation" block. The "Integration to Orientation" block integrates body frame sensed angular rate to orientation at a fixed 100 times per second within all of the DMUx81ZA Series products (except IMUx81ZA). For improved accuracy and to avoid singularities when dealing with the cosine rotation matrix, a quaternion formulation is used in the algorithm to provide attitude propagation. Following the integration to orientation block, the body frame accelerometer signals are rotated into the NED level frame and are integrated to velocity. At this point, the data is blended with GPS position data, and output as a complete navigation solution.

As shown in Figure 4, the Integration to Orientation and the Integration to Velocity signal processing blocks receive drift corrections from the Extended Kalman Filter (EKF) drift correction module. The drift correction module uses data from the aiding sensors, when they are available, to correct the errors in the velocity, attitude, and heading outputs. Additionally, when aiding sensors are available corrections to the rate gyro and accelerometers are performed.

The INSx81ZA blends GPS derived heading and accelerometer measurements into the EKF update depending on the health and status of the associated sensors. If the GPS link is lost or poor, the Kalman Filter solution stops tracking accelerometer bias, but the algorithm continues to apply gyro bias correction and provides stabilized angle outputs. The EKF tracking states are reduced to angles and gyro bias only. The accelerometers will continue to integrate velocity, however, accelerometer noise, bias, and attitude error will cause the velocity estimates to start drifting within a few seconds. The attitude tracking performance will degrade, the heading will freely drift, and the filter will revert to the VG only EKF formulation. The UTC packet synchronization will drift due to internal clock drift.

The status of GPS signal acquisition can be monitored from the hardwareStatus BIT as discussed in Section 3.5.3 INSx81ZA Built in Test. From a cold start, it typically takes 40 seconds for GPS to lock. The actual lock time depends on the antenna's view of the sky and the number of satellites in view.

The processor performs time-triggered trajectory propagation at 100Hz and will synchronize the sensor sampling with the GPS UTC (Universal Coordinated Time) second boundary when available.

As with the AHRSx81ZA and VGx81ZA, the algorithm has two major phases of operation. Immediately after power-up, the INSx81ZA uses the accelerometers and magnetometers to compute the initial roll, pitch and yaw angles. The roll and pitch attitude will be initialized using the accelerometer's reference of gravity, and yaw will be initialized using the leveled magnetometers X and Y axis reference of the earth's magnetic field. During the first 60 seconds of startup, the INSx81ZA should remain approximately motionless in order to properly initialize the rate sensor bias. The initialization phase lasts approximately 60 seconds, and the initialization phase can be monitored in the softwareStatus BIT transmitted by default in each measurement packet. After the initialization phase, the INSx81ZA operates with lower levels of feedback (also referred to as EKF gain) from the GPS, accelerometers, and magnetometers.

Digital data is output over the UART port at a selectable fixed rate (100, 50, 25, 20, 10, 5 or 2 Hz) or on as requested basis using the GP, 'Get Packet' command. In addition to the angle mode packets of the AHRSx81ZA and scaled sensor packets of the IMUx81ZA, the INSx81ZA has additional output measurement packets including the default 'N1' Navigation Packet which outputs the Latitude, Longitude, Altitude, X,Y,Z velocities, accelerations, and roll angle, pitch angle, yaw angle, and digital IMU data. See Sections 6 and 7 of the manual for full packet descriptions. All data is also available on the SPI output port registers. Please refer to section 5 for a complete description of the SPI port functionality.

IMPORTANT

For proper operation, the INSx81ZA relies on magnetic field readings from its internal 3-axis magnetometer. The INSx81ZA must be installed correctly and calibrated for hard-iron and soft iron effects to avoid any system performance degradation. See section 3.4.1 for information and tips regarding installation and calibration and why magnetic calibration is necessary. Please review this section of the manual before proceeding to use the INSx81ZA.

IMPORTANT

For optimal performance the INSx81ZA utilizes GPS readings from an external GPS receiver. The GPS receiver requires proper antennae installation for operation. See section 2.1.4 for information and tips regarding antenna installation.

3.5.1 INSx81ZA Magnetometer Calibration and Alignment

The INSx81ZA requires the three axis magnetic field sensor to be calibrated while installed in its operating environment. See section 3.4.1 for information and tips regarding installation and calibration and why magnetic calibration is necessary. Please review this section of the manual before proceeding to use the INSx81ZA.

3.5.2 INSx81ZA Advanced Settings

In addition to the configurable baud rate, packet rate, axis orientation, and sensor low-pass filter settings, the INSx81ZA provides additional advanced settings which are selectable for tailoring the INSx81ZA to a specific application requirements. The INSx81ZA advanced settings are shown in Table 12 below:

Table 12 INSx81ZA Series Advanced Settings

Setting	Default	Comments
Baud Rate	38,400 baud	57600, 115200, and 230400 also available
Packet Type	N1	S0, S1, A1, A2, N0 also available
Packet Rate	25 Hz	This setting sets the rate at which selected Packet Type, packets are output. If polled mode is desired, then select Quiet. If Quiet is selected, the INSx81ZA will only send measurement packets in response to GP commands.
Orientation	See Figure 6 and Figure 7.	To configure the axis orientation, select the desired measurement for each axes; NAV-VIEW will show the corresponding image of the INSx81ZA, so it easy to visualize the mode of operation. See section 8.4 Orientation Field settings for the twenty four possible orientation settings. The default setting points the connector AFT.
Filter Settings (unfiltered, 2, 5, 10, 20, 25, 40, 50 Hz)	20 Hz accels 20 Hz rates	The low pass filters are set to a default of 20Hz for the accelerometers, and 20Hz for the angular rate sensors. There is one filter setting for all three angular rate sensors. There is one filter setting for all three accelerometer sensors. The reason for filtering the accelerometers is that in many installations, the vibration level can be high, and it can prove helpful to filter accelerometers. Setting either to zero disables the low-pass filter.
Freely Integrate	OFF	The Freely Integrate setting allows a user to turn the INSx81ZA into a 'free gyro'. In free gyro mode, the roll, pitch and yaw are computed exclusively from angular rate with no kalman filter based corrections of roll, pitch, and yaw. When turned on, there is no coupling of acceleration based signals into the roll and pitch or magnetometer based signal to the yaw. As a result, the roll, pitch, and yaw outputs will drift roughly linearly with time due to sensor bias. For best performance, the Freely Integrate mode should be used after the algorithm has initialized. This allows the Kalman Filter to estimate the roll and pitch rate sensor bias prior to entering the free gyro mode. Upon exiting the 'free gyro' mode (OFF), one of two behaviors will occur (1) If the INSx81ZA has been in freely integrate mode for less than sixty seconds, the algorithm will resume operation at normal gain settings (2) If the INSx81ZA has been in freely integrate mode for greater than sixty seconds, the algorithm will force a reset and reinitialize with high gains automatically.
Use GPS	ON	The Use GPS setting allows users to turn on and off the GPS feedback. The default setting is ON for the INSx81ZA. When Use GPS is turned OFF, the INSx81ZA's behavior will revert to that of an AHRSx81ZA. See the AHRSx81ZA Theory of Operation for detailed description.
Stationary Yaw Lock	OFF	This setting defaults to OFF on the INSx81ZA, and it is recommended to be OFF for the INSx81ZA. The stationary yaw lock setting is only recommended for consideration when the INSx81ZA is operating with GPS (Use GPS = ON) and WITHOUT magnetometer feedback (Use Mags = OFF). Stationary yaw lock may be appropriate if the user platform is a wheeled land vehicle.
Use Mags	ON	The Use Mags setting allows users to turn on and off the magnetometer feedback for yaw/heading stabilization. The default setting is ON for the INSx81ZA. When Use Mags is turned ON, the INSx81ZA uses the magnetic field sensor readings to stabilize the drift in yaw, and it slaves the yaw to the compass reading provided from the magnetic field sensor readings. When UseMags is turned OFF, the heading (yaw) angle measurement of the INSx81ZA will be slaved to the GPS heading if GPS is available, otherwise the

		heading will drift feely. The reason for this setting is to give the user an ability to turn off the magnetometer stabilization when severe magnetic distortion may be occurring. This setting is desirable when the user vehicle temporarily moves in close proximity to a large ferrous object. When the Use Mags switch is turned from OFF to ON, the INSx81ZA will reinitialize the yaw/heading angle with the compass reading provided from the magnetic field sensor readings.
Restart On Over Range	OFF	This setting forces an algorithm reset when a sensor over range occurs i.e., a rotational rate on any of the three axes exceeds the maximum range. The default setting is OFF for the INSx81ZA. Algorithm reset returns the INSx81ZA to a high gain state, where the INSx81ZA rapidly estimates the gyro bias and uses the accelerometer feedback heavily. This setting is recommended when the source of over-range is likely to be sustained and potentially much greater than the rate sensor operating limit. Large and sustained angular rate over-ranges result in unrecoverable errors in roll and pitch outputs. An unrecoverable error is one where the EKF can not stabilize the resulting roll and pitch reading. If the over-ranges are expected to be of short duration (<1 sec) and a modest percentage over the maximum operating range, it is recommended that the restart on over range setting be turned off. Handling of an inertial rate sensor over-range is controlled using the restartOnOverRange switch. If this switch is off, the system will flag the overRange status flag and continue to operate through it. If this switch is on, the system will flag a masterFail error during an over-range condition and continue to operate with this flag until a quasi-static condition is met to allow for an algorithm restart. The quasi-static condition required is that the absolute value of each low-passed rate sensor fall below 3 deg/sec to begin initialization. The system will then attempt a normal algorithm start.
Dynamic Motion	ON	The default setting is ON for the INSx81ZA. Turning off the dynamic motion setting results in a higher gain state that uses the accelerometer feedback heavily. During periods of time when there is known low dynamic acceleration, this switch can be turned off to allow the attitude estimate to quickly stabilize.
Turn Switch threshold	0.5 deg/sec	With respect to centripetal or false gravity forces from turning dynamics (or coordinated turn), the INSx81ZA monitors the yaw-rate. If the yaw rate exceeds a given Turnswitch threshold, the feedback gains from the accelerometer signals for attitude correction are reduced because they are likely corrupted.
BIT		See 4.5.3

3.5.3 INSx81ZA Built-In Test

As with the IMU, VG and AHRSx81ZA, the Built-In Test capability allows users of the INSx81ZA to monitor health, diagnostic, and system status information of the unit in real-time. The Built-In Test information consists of a BIT word (2 bytes) transmitted in every measurement packet. In addition, there is a diagnostic packet 'T0' that can be requested via the Get Packet 'GP' command which contains a complete set of status for each hardware and software subsystem in the INSx81ZA. See Sections 6 and 7 of the manual for details on the 'T0' packet.

The BIT word contained within each measurement packet is detailed below. The LSB (Least Significant Bit) is the Error byte, and the MSB (Most Significant Bit) is a Status byte with programmable alerts. Internal health and status are monitored and communicated in both hardware and software. The ultimate indication of a fatal problem is the masterFail flag. The softwareError bit also provides useful information regarding the status and quality of the INSx81ZA magnetic alignment. If the INSx81ZA has not been properly magnetically calibrated, the INSx81ZA shall indicate a softwareError.

The masterStatus flag is a configurable indication that can be modified by the user. This flag is asserted as a result of any asserted alert signals which have been enabled. See Advanced Settings

for details for configuring the masterStatus flags. Table 13 shows the BIT definition and default settings for BIT programmable alerts in the INSx81ZA.

Table 13 INSx81ZA Default BIT Status Definitions

BITstatus Field	*Bits*	*Meaning*	*Category*
masterFail	0	0 = normal, 1 = fatal error has occurred	BIT
HardwareError	1	0 = normal, 1= internal hardware error	BIT
comError	2	0 = normal, 1 = communication error	BIT
softwareError	3	0 = normal, 1 = internal software error or magAlignOutofBounds	BIT
Reserved	4:7	N/A	
masterStatus	8	0 = nominal, 1 = one or more status alert	Status
hardwareStatus	9	0 = nominal, 1 = Internal GPS unlocked or 1PPS invalid	Status
comStatus	10	Disabled	Status
softwareStatus	11	0 = nominal, 1 = Algorithm Initialization or high gain	Status
sensorStatus	12	0 = nominal, 1 = Sensor Over Range	Status
Reserved	13:15	N/A	

The INSx81ZA also allows a user to configure the Status byte within the BIT message. To configure the word, select the BIT Configuration tab from the Unit Configuration menu. The dialog box allows selection of which status types to enable (hardware, software, sensor, and comm). Like the IMU, VG and AHRSx81ZA, ACEINNA recommends for the vast majority of users, that the default Status byte for the INSx81ZA is sufficient. For users, who wish to have additional visibility or alerts relative to the GPS sensor status or algorithm status, they can configure additional triggers for both the softwareStatus and hardwareStatus (See Section 9 of the user's manual for a description of all the BIT fields).

4 Application Guide

4.1 Introduction

This section provides recommended advanced settings for tailoring the DMUx81ZA Series of inertial systems to different types of application and platform requirements.

4.2 Fixed Wing Aircraft

A fixed-wing aircraft is a heavier-than-air craft where movement of the wings in relation to the aircraft is not used to generate lift. The term is used to distinguish from rotary-wing aircraft, where the movement of the wing surfaces relative to the aircraft generates lift. The fixed wing aircraft can range in size from the smallest experimental plane to the largest commercial jet. The dynamic characteristics of the fixed wing aircraft depends upon types of aircraft (i.e., glider, propeller aircraft, and jet aircraft) and mission phases (i.e., launch, landing, and maneuver). In order to meet application requirements, users must dial in proper advanced settings so that the DMUx81ZA Series can provide the best possible solution under given dynamic conditions. For example, Table provides the recommended advanced settings for four different dynamic conditions.

Table 14 Recommended Advanced Settings for Fixed Wing Aircraft

Recommended Product	***AHRSx81ZA or INSx81ZA***			
Recommended Settings	***Dynamic Condition***			
	Pre-launch or known straight and level un-accelerated flight	***Launch***	***Normal Dynamics (Default)***	***High Dynamics***
UseMags	ON	ON	ON	ON
UseGPS	ON	ON (< 4g)	ON	ON (< 4g)
FreelyIntegrate	OFF	OFF**	OFF	OFF (< 2g)
Stationary Yaw Lock	OFF	OFF	OFF	OFF
Restart Over Range	ON	OFF	OFF	OFF
Dynamic Motion	OFF	ON	ON	ON
Turn Switch Threshold	0.5 deg/s	0.5 deg/s	0.5 deg/s	0.5 deg/s
XY Filter Accel	5 Hz	5 Hz	5 Hz*	15 Hz
Z Filter Accel	5 Hz	5 Hz	5 Hz*	15 Hz
Filter Rate Sensor	20 Hz	20 Hz	20 Hz*	20 Hz

*A cutoff frequency of filters may be varied depending on the fastest dynamic mode of the aircraft. For example, the conventional aircraft has five dynamic modes, short-period, phugoid, spiral, dutch-roll, and roll, and the fastest one is the roll mode. The natural frequency of this mode is around 6~8 radian/sec or (about 2 Hz) in most cases. Therefore, the recommended filter setting would not reject desired frequency components (or dynamic modes) that one wants to capture. However, the larger the bandwidth (or cutoff frequency) is, the noisier the corresponding signal is, which may result in the performance degradation. If

the aircraft is operated under severe vibrations, also, the recommended filter setting may need to be further reduced in order to reject the frequency components caused by the vibration.

**FreelyIntegrate should only be set to "ON" for severe launch conditions. Normal takeoff dynamics that a standard aircraft would experience will see the best performance with this setting in the "OFF" position.

4.3 Rotorcraft

Rotorcraft is a category of heavier-than-air flying machines that use lift generated by rotors. They may also include the use of static lifting surfaces, but the primary distinguishing feature being lift provided by rotating lift structures. Rotorcraft includes helicopters, autogyros, gyrodynes and tiltrotors.

The rotor blade dynamics itself is much faster than that of the fixed wing aircraft and contains high frequency components. At the same time, however, it may cause severe vibrations on the airframe. Also, the overall dynamics (translational and rotational motion) of the rotor craft is much slower than the fixed wing aircraft due to a mechanical mechanism of rotors generating the aerodynamic forces and moments. Table 15 provides the recommended advanced settings for two different dynamic conditions.

Table 15 Recommended Advanced Settings for Rotorcraft

Recommended Product	***AHRSx81ZA or INSx81ZA***	
Recommended Settings	***Dynamic Condition***	
	Normal Dynamics	***High Dynamics (with uncoordinated tail motion)***
UseMags	ON	ON
UseGPS	ON	ON (< 4g)
FreelyIntegrate	OFF	OFF (< 2g)
Stationary Yaw Lock	OFF	OFF
Restart Over Range	OFF	ON
Dynamic Motion	ON	ON
Turn Switch Threshold	1.0 deg/s**	30.0 deg/s**
XY Filter Accel	5 Hz*	5 Hz
Z Filter Accel	5 Hz*	5 Hz
Filter Rate Sensor	20 Hz*	20 Hz

**The helicopter can change its heading angle rapidly unlike the aircraft which requires banking. A turn switch threshold that is too low may cause turn switch activation with high duty cycle causing random walk in roll and pitch angles due to low feedback gains.

*A cutoff frequency must be far away from major frequency components caused by the rotor vibration.

4.4 Land Vehicle

Some examples of land vehicles are: Automobiles, trucks, heavy equipment, trains, snowmobiles, and other tracked vehicles. Table 16 provides the recommended advanced settings for two different types of application.

Table 16 Recommended Advanced Settings for Land Vehicle

Recommended Product	VGx81ZA or INSx81ZA	
Recommended Settings	Dynamic Condition	
	Heavy Equipment Application	Automotive Testing (IMU and VG default)
UseMags	ON*	ON*
UseGPS	ON	ON (< 4g)
FreelyIntegrate	OFF	OFF
Stationary Yaw Lock	OFF	OFF
Restart Over Range	ON	OFF
Dynamic Motion	ON	ON
Turn Switch Threshold	5.0 deg/s	10.0 deg/s
XY Filter Accel	5 Hz	5 Hz
Z Filter Accel	5 Hz	5 Hz
Filter Rate Sensor	20 Hz	20 Hz

*When not in distorted magnetic environment.

4.5 Water Vehicle

Water vehicle is a craft or vessel designed to float on or submerge and provide transport over and under water. Table 17 provides the recommended advanced settings for two different types of application.

Table 17 Recommended Advanced Settings for Water Vehicle

Recommended Product	*INSx81ZA*	
Recommended Settings	*Application*	
	Surfaced	*Submerged*
UseMags	ON*	ON*
UseGPS	ON	OFF
FreeIntegrate	OFF	OFF

Stationary Yaw Lock	OFF	OFF
Restart Over Range	OFF	OFF
Dynamic Motion	ON	ON
Turn Switch Threshold	10 deg/s	5 deg/s
XY Filter Accel	5 Hz	2 Hz
Z Filter Accel	5 Hz	2 Hz
Filter Rate Sensor	15 Hz	10 Hz

*When not in distorted magnetic environment.

☑ EXAMPLE

Figure 8 shows a typical flight profile of the fixed wing aircraft and the corresponding advanced settings that one can configure adaptively depending on a flight phase:

- **Prelaunch** is the phase of flight in which an aircraft goes through a series of checkups (hardware and software) on the ground before takeoff. The aircraft is a static condition,
- **Takeoff** is the phase of flight in which an aircraft goes through a transition from moving along the ground (taxiing) to flying in the air, usually along a runway. The aircraft is under horizontal acceleration and may suffer from vibrations coming from an engine and ground contact forces transmitted from its landing gear.
- **Climb** is the phase of a flight, after take-off, consisting of getting the aircraft to the desired flight level altitude. More generally, the term 'climb' means increasing the altitude. The aircraft is under vertical acceleration until it reaches the steady-state climb rate.
- **Straight and level flight** is the phase of flight in which an aircraft reaches its nominal flight altitude and maintains its speed and altitude. The aircraft is under equilibrium (See Figure 8).
- **Maneuver** is the phase of flight in which an aircraft accelerates, decelerates, and turns. The aircraft is under non-gravitational acceleration and/or deceleration (See Figure 8).
- **Descent** is the phase of flight in which an aircraft decreases altitude for an approach to landing. The aircraft is under vertical deceleration until it captures a glide slope (See Figure 8).
- **Landing** is the last part of a flight, where the aircraft returns to the ground (See Figure 8).

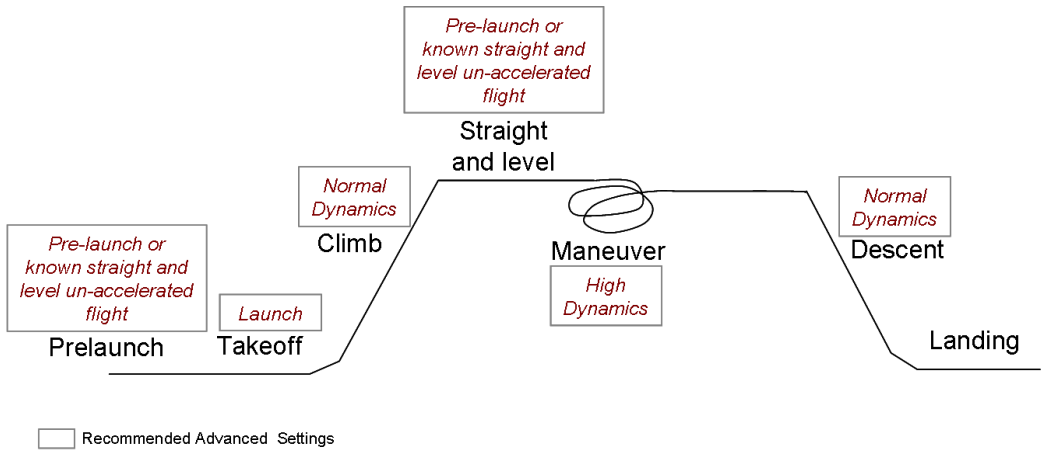


Figure 8 Typical flight profiles of fixed wing aircraft and the corresponding advanced settings

5 DMUx81ZA SPI Port Interface Definition

The DMUx81ZA provides a SPI interface for data communications. This section of the user's manual defines the DMUx81ZA register map, register control capabilities, and the data register reading and writing methodologies.

The DMUx81ZA operates as a slave device. The master device must be configured to communicate with the DMUx81ZA using the following settings:

- Data transferred in 16-bit word-length and MSB-first
- $f_{CLK} \leq 2.0$ MHz
- CPOL = 1 (clock polarity) and CPHA = 1 (clock phase)

Additional operational requirements are described in Section 5.8.

5.1 DMUx81ZA Register Map

Table 18 describes the DMUx81ZA register map.

Table 4 DMUx81ZA Register Map[2]

Name	Read/Write	Address	Default	Function
Reserved	N/A	0x00 to 0x03	N/A	
X_RATE	R	0x04	N/A	X-Axis Rate-Sensor Output
Y_RATE	R	0x06		Y-Axis Rate-Sensor Output
Z_RATE	R	0x08		Z-Axis Rate-Sensor Output
X_ACCEL	R	0x0A	N/A	X-Axis Accelerometer Output
Y_ACCEL	R	0x0C		Y-Axis Accelerometer Output
Z_ACCEL	R	0x0E		Z-Axis Accelerometer Output
X_MAG	R	0x10	N/A	X-Axis Magnetometer Output
Y_MAG	R	0x12		Y-Axis Magnetometer Output
Z_MAG	R	0x14		Z-Axis Magnetometer Output
RATE_TEMP	R	0x16	N/A	Rate-sensor temperature
BOARD_TEMP	R	0x18	N/A	Board temperature
Reserved	R	0x1A to 0x33	N/A	
SELF_TEST[3]	R/W	0x34/0x35	0x00	See Table 25: Initiate Self-Test / Configure Data-Ready output signal
DATA_READY	R/W	0x35/0x34	0x04	
OUTPUT_DATA_RATE	R/W	0x36/0x37	0x01	See Table 26 Table 12: Set Output Data Rate (ODR)
Reserved	N/A	0x37/0x36	0x01	
RS_DYNAMIC_RANGE	R/W	0x38/0x39	0x02	See Table 13: Set rate-sensor dynamic range (SPI only) /

[2] Register and data-packet availability is based on the features of the DMU381ZA (see Table 2).

[3] Register reads are performed 2-bytes at a time while writes are a single byte in length. In operation, the SELF_TEST/DATA_READY register should be read together starting at register 0x34. This applies to other shared registers as well.

Name	Read/Write	Address	Default	Function
LOW_PASS_FILTER	R/W	0x39/0x38	0x06	Select digital filter
Reserved	N/A	0x3A to 0x3B	N/A	
STATUS	R	0x3C	N/A	See Table 9: Diagnostic register
STNDRD_BURST	R	0x3E	N/A	Command to perform a burst-read of the standard data-packet
Reserved	R	0x3F to 0x40		
S0_BURST	R	0x41	N/A	Burst-Mode Command for UCB scaled-sensor 0 data-packet (see Section 7.4.1)
S1_BURST	R	0x42	N/A	Burst-Mode Command for UCB scaled-sensor 1 data-packet (see Section 7.4.2)
A1_BURST	R	0x43	N/A	Burst-Mode Command for UCB angle 1 data-packet (see Section 7.4.3)
A2_BURST	R	0x44	N/A	Burst-Mode Command for UCB angle 2 data-packet (see Section 7.4.4)
N0_BURST	R	0x45	N/A	Burst-Mode Command for UCB nav 0 data-packet (see Section 7.4.5)
Reserved	N/A	0x46 to 0x47	N/A	
X_HARD_IRON	R	0x48	0x0000	Hard-iron bias (X-Axis)
Y_HARD_IRON	R	0x4A	0x0000	Hard-iron bias (Y-Axis)
SF_SOFT_IRON	R	0x4C	0x8000	Soft-iron scale factor
ANG_SOFT_IRON	R	0x4E	0x0000	Soft-iron angle
MAG_ALIGN[4]	R/W	0x50/0x51	N/A	See Table 20: Magnetic-alignment control and status
MANUF_CODE	R	0x52	0x1310	Manufacturing code indicating year and location
UNIT_CODE	R	0x54	0x0000	Unit information code
PRODUCT_ID	R	0x56	0x3810	Product identification code
SERIAL_NUMBER	R	0x58	Varies	Serial number
MASTER_STATUS	R	0x5A	N/A	See Section 9.2: Master BIT and Status Field
HW_STATUS	R	0x5C	N/A	See Section 9.3: Hardware BIT Field
SW_MASTER	R	0x5E	N/A	See Section 9.9: Software BIT Field
SW_STATUS	R	0x60	N/A	See Section 9.14: Software Status Field
SW_ALGO	R	0x62	N/A	See Section 9.10: Software Algorithm BIT Field
SW_DATA	R	0x64	N/A	See Section 9.11: Software Data BIT Field
COMM_MASTER	R	0x66	N/A	See Section 9.6: Com BIT Field
COMM_DATA_STATUS	R	0x68	N/A	See Section 9.13: Com Status Field
COMM_BUS_A	R	0x6A	N/A	See Section 9.7: Com Serial A BIT Field
COMM_BUS_B	R	0x6C	N/A	See Section 9.8: Com Serial B BIT Field
SENSOR_STATUS	R	0x6E	N/A	See Section 9.15: Sensor Status Field

[4] This command only applies to AHRS and INS variants and will not work with IMU or VG units

Name	Read/Write	Address	Default	Function
RS_SCALE ACCEL_SCALE	R/W R/W	0x70/0x71 0x71/0x70	0x1F 0x3F	See Section 5.7.4
MAG_SCALE Reserved	R/W R/W	0x72/0x73 0x73/0x72	0x10 0x03	See Section 5.7.4
ORIENTATION_MSB ORIENTATION_LSB	R/W R/W	0x74 0x75	0x00 0x00	See Table 29 for valid orientation settings. The orientation register must be written in order (MSB followed by LSB) for write to take effect.
EEPROM_WRITE	W	0x76	N/A	See Section 5.7.6
Reserved	N/A	0x78 to 0x7D	N/A	
HW_SW_VERSION	R	0x7E	0x00	See Section 5.7.8

5.2 DMUx81ZA SPI Register Read Methodology

The DMUx81ZA SPI port uses registers to store information such as:

- Sensor data
- Algorithm output data
- Configuration/Status information

A SPI master accesses information via the SPI bus in one of two ways:

- Polled-Mode
- Burst-Mode

In polled-mode, the DMUx81ZA transfers information from any register back to the master in two (or more) SPI cycles[5]. In Burst-Mode, the DMUx81ZA transfers predefined blocks of data in one contiguous group of nine to twenty SPI cycles.

5.2.1 DMUx81ZA SPI Port Polled-Mode Read

In polled-mode, data transfer begins when the SPI master sets the chip-select line (nSS) low and clocks a 16-bit word, comprised of the register-address byte and a zero-byte, across the MOSI line. For example, to request the unit's serial number, stored in register 0x58, the master sends the command 0x5800. The DMUx81ZA returns information from this address across the MISO line during the following 16 clock-cycles.

Subsequent SPI-master commands sent to the DMUx81ZA consist of either:

- Sixteen zero-bits (0x0000) to complete the read of a single register.
- The address of another register followed by a zero-byte. This permits back-to-back reads of data-registers.

Single-Register Polled-Read

Figure 9 illustrates a polled-mode read of a single register (x-axis rate-sensor data), which is composed of two bytes, starting at register address 0x04.

In this example, the SPI-master initiates a register read by clocking in the address followed by 0x00, i.e. 0x0400, via MOSI; this combination is referred to as a read-command[6]. This is followed by 16 zero-bits to complete the SPI data-transfer cycle.

[5] A SPI cycle consists of 16 clock cycles.

As the master transmits the read command over MOSI, the DMUx81ZA transmits information back over MISO. In this transmission, the first data-word sent by the DMUx81ZA (as the read-command is sent) consists of 16-bits of non-applicable data. The subsequent 16-bit message contains the x-axis rate-sensor information (most significant byte followed by least-significant byte).

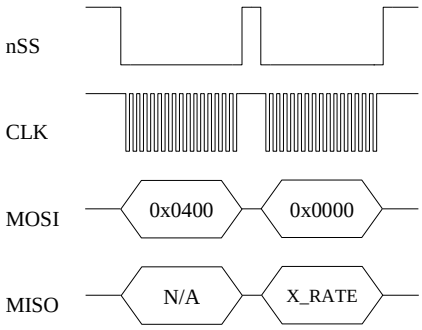


Figure 9 Single Register Read via Polled-Mode

Multiple-Register Polled-Read

Figure 10 illustrates a polled-mode read of multiple registers. In this case, the SPI-master transmits an initial read-command (the desired register-address appended by 0x00) across MOSI followed by any number of additional read-commands (one for each register of interest). The DMUx81ZA transfers the requested information concurrently across MISO to the master. To complete the data transfer, the final read-command must be followed by an additional 16 clock cycles to transfer the last 16-bits of data.

In this example, the master requests data from four separate registers: x-axis rate (0x0400), y-axis rate (0x0600), z-axis acceleration (0x0E00), and system status (0x3C00). The transfer of 0x0000 across MOSI completes the read by returning the status data via the MISO line.

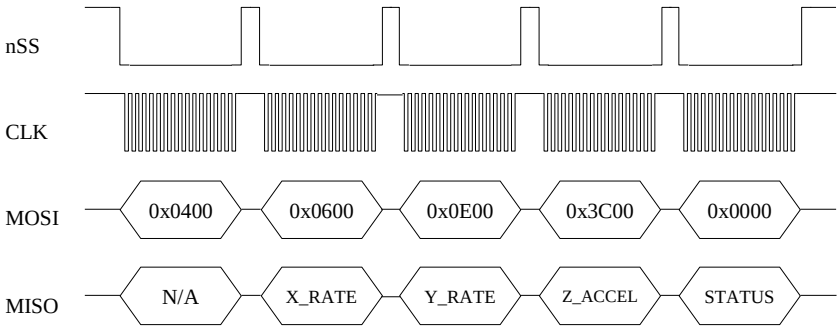


Figure 10 Multiple Register Read via Polled-Mode

[6] A read-command consists of an 8-bit register address and a zero byte (0x00).

5.2.2 DMUx81ZA SPI Port Burst-Mode Read

In burst-mode, the DMUx81ZA returns predefined blocks of data in single groups, referred to as data-packets, without the need to send multiple read commands. These groups vary from eight to nineteen words in length, depending on the packet selected. Table 5 lists the data-packets available for the DMUx81ZA. The data packets are described in more detail, including data-ordering and conversion factor information, in Section 7.4.

Table 5 DMUx81ZA Burst-Mode Data-Packets

Data-Packet	Register Address	Number of 16-bit Words	Pertinent Section	Availability
Standard	0x3E	8	5.2.2	All systems
Scaled Sensor 0	0x41	15	7.4.1	All systems except IMUx81ZA-200 and VGx81ZA
Scaled Sensor 1	0x42	12	7.4.2	All systems
Angle Data 1	0x43	16	7.4.3	All systems except IMUx81ZA and VGx81ZA
Angle Data 2	0x44	15	7.4.4	All systems except IMUx81ZA
Nav 0	0x45	16	7.4.5	INSx81ZA

Burst-Read of Standard Data-Packet

The standard data-packet comprises data from eight predefined registers. Table 6 lists the data contained in a standard packet along with the corresponding registers. The registers are listed in the order in which they are sent during a burst-mode read.

Table 6 DMUx81ZA Burst-Mode Output Registers

Register Name	Register Address	Description
STATUS	0x3C	System Status
X_RATE	0x04	Rate Sensor Output (X-Axis)
Y_RATE	0x06	Rate Sensor Output (Y-Axis)
Z_RATE	0x08	Rate Sensor Output (Z-Axis)
X_ACCEL	0x0A	Accelerometer Output (X-Axis)
Y_ACCEL	0x0C	Accelerometer Output (Y-Axis)
Z_ACCEL	0x0E	Accelerometer Output (Z-Axis)
BOARD_TEMP	0x18	System Temperature

Burst-mode begins when the master requests a read from a burst-mode data-packet (i.e. 0x3E). Eight additional SPI cycles complete the read (one for each word in the standard data-packet). Figure 11 illustrates the burst-mode sequence. Note: if the incorrect number of SPI cycles follow the burst-mode command, the SPI transfer will either complete early or remain in burst-mode; subsequent reads/writes will be out of sync with the SPI transfer cycle of the DMUx81ZA.

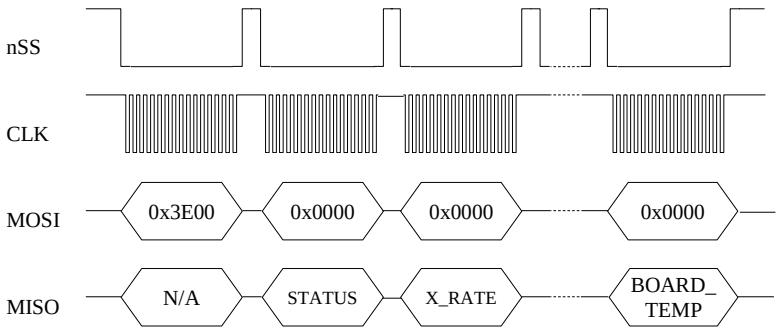


Figure 11 Multiple Register Read via Burst-Mode

Operational notes:

1. When combining polled and burst reads, use only single-register polled-reads.
2. Burst-mode reads for other data-packets are performed in a manner similar to the standard packet. The only deviation from the method described above is the register address and the subsequent number of data words, listed in Table 6.
3. Care must be taken when switching between data-packets as values returned during the first burst-read of a new packet are invalid. A single read-cycle is needed to populate the internal burst-mode register; subsequent reads from the same packet contain valid information.
4. During a burst read, the chip-select line (nSS) can be controlled in one of two ways:
   - Toggle nSS in between each of the 16-bit words (as shown in Figure 11).
   - Set and hold nSS low during the entire read. After the transfer is complete, set chip-select high.

5.3 Output Data Registers

Output data registers hold the sensor information as it is measured; they are overwritten only when new data is available. Table 7 lists each register, its memory address, and its conversion factor. Note: the scale-factor described below only applies to the values in the data registers and standard burst-mode. Scale-factors for the other output data packets follow the values listed in Section 7.4.

Table 7 DMUx81ZA Data Output Registers

Name	Read Address	Function
X_RATE	0x04	X, Y, Z-axis rate-sensor information, twos complement format, conversion factor: 200 LSB/[°/sec] (default); changes with selected dynamic range (Table 14)
Y_RATE	0x06	
Z_RATE	0x08	
X_ ACCEL	0x0A	X, Y, Z-axis accelerometer information, twos complement format, conversion factor: 4000 LSB/g (default) ; changes with selected dynamic range (Table 17)
Y_ ACCEL	0x0C	
Z_ACCEL	0x0E	
X_MAG	0x10	X, Y, Z-axis magnetometer information, twos complement format, conversion factor: 16000 LSB/G (default) ; changes with selected dynamic range (Table 19)
Y_MAG	0x12	
Z_MAG	0x14	

RATE_TEMP	0x16	Rate-sensor temperature information, twos complement format, conversion: Tout [°C] = Vout · 0.07311 [°C/LSB]
BOARD_TEMP	0x18	System temperature information, twos complement format, conversion: Tout [°C] = Vout · 0.07311 [°C/LSB] + 31.0 [°C]

5.4 System Registers

In addition to the output data registers, there are further read-only registers that provide DMUx81ZA system information to the SPI master. Table 8 provides a description of each along with their read-addresses.

Table 8 DMUx81ZA System Registers

Name	Read Address	Function
DIAGNOSTIC_STATUS	0x3C	Sensor self-test and over-range information (See Section 5.5)
X_HARD_IRON	0x48	Results of the magnetic-alignment procedure (see Section 5.7.7)
Y_HARD_IRON	0x4A	
SF_SOFT_IRON	0x4C	
ANG_SOFT_IRON	0x4E	
MANUF_CODE	0x52	Product manufacturing code
UNIT_CODE	0x54	Additional product manufacturing information
PRODUCT_ID	0x56	Product ID (0x3810)
SERIAL_NUMBER	0x58	Unique product identification number
MASTER_STATUS	0x5A	See Section 9.2: Master BIT and Status Field
HW_STATUS	0x5C	See Section 9.3: Hardware BIT Field
SW_MASTER	0x5E	See Section 9.9: Software BIT Field
SW_STATUS	0x60	See Section 9.14: Software Status Field
SW_ALGO	0x62	See Section 9.10: Software Algorithm BIT Field
SW_DATA	0x64	See Section 9.11: Software Data BIT Field
COMM_MASTER	0x66	See Section 9.6: Com BIT Field
COMM_DATA_STATUS	0x68	See Section 9.13: Com Status Field
COMM_BUS_A	0x6A	See Section 9.7: Com Serial A BIT Field
COMM_BUS_B	0x6C	See Section 9.8: Com Serial B BIT Field
SENSOR_STATUS	0x6E	See Section 9.15: Sensor Status Field
HW_SW_VERSION	0x7E	Hardware and Software Versions (See Section 5.7.8)

5.5 Diagnostic Status Register

The diagnostic status register contains information describing the results of the self-test as well as sensor over-range information. It is defined in Table 9.

Table 9 Diagnostic Status Register

(Base Address: 0x3C), Read-Only	
Bits	**Description (Default: 0x0000)**
15	Accelerometer Z-Axis self-test bit 0: Pass, 1: Fail
14	Accelerometer Y-Axis self-test bit 0: Pass, 1: Fail
13	Accelerometer X-Axis self-test bit 0: Pass, 1: Fail
12	Rate-Sensor Z-Axis self-test bit 0: Pass, 1: Fail
11	Rate-Sensor Y-Axis self-test bit 0: Pass, 1: Fail
10	Rate-Sensor X-Axis self-test bit 0: Pass, 1: Fail
[9:6]	Unused
5	Self-Test Success/Failure bit 0: Success, 1: Failure
4	Sensor over-range bit (a 1 indicates one or more sensors have over-ranged)
[3:0]	Unused

5.6 DMUx81ZA SPI Register Write Methodology

The SPI master configures the DMUx81ZA by writing to specific registers. However, unlike reads, writes are performed *one byte at a time*. The specific registers that affect system configuration are listed in Table 10 along with their write-addresses.

Table 10 DMUx81ZA Configuration Registers

Name	Write Address	Function
SELF_TEST DATA_READY	0x35 0x34	See Table 11: Initiate self-test and Configure Data-Ready output signal
OUTPUT_DATA_RATE	0x37	See Table 12: Sets Output Data Rate (ODR) of the unit
RS_DYNAMIC_RANGE LOW_PASS_FILTER	0x39 0x38	See Table 13: Set the rate-sensor dynamic range and the digital filter
MAG_ALIGN	0x50	See Section 5.7.7: Command to initiate a magnetic-alignment on AHRS and INS variants
RS_SCALE ACCEL_SCALE MAG_SCALE	0x71 0x70 0x73	See Section 5.7.4: Set the dynamic range of the sensors
ORIENTATION_MSB ORIENTATION_LSB	0x74 0x75	See Sections 5.7.5 and 8.4: Sets the orientation (x, y, and z-axes) of the unit
EEPROM_WRITE	0x76	Save the settings to the EEPROM

The following example highlights how write-commands are formed in order to initiate a sensor self-test:

- Select the write address of the desired register, e.g. 0x35 for self-test
- Change the most-significant bit of the address to 1 (the write-bit), e.g. 0x35 becomes 0xB5
- Create the write command by appending the write-bit/address combination with the value to be written to the register, e.g. 0xB504 (see Table 11 for a description of the self-test register)

Figure 12 illustrates the sensor self-test command sent over SPI.

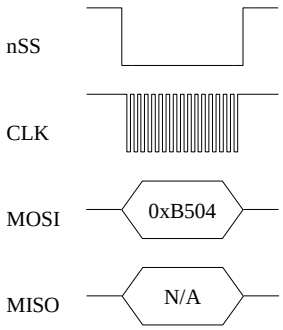


Figure 12 Single Register Write to Initiate Self-Test

As described in Section 5.7.1, the self-test command bit remains set until the test completes. The master must read from register 0x34 to assess if the test is complete (Figure 13). Note: as described in the Register Reads section, a register read returns two bytes, in this case a read from register 0x34 returns data from registers 0x34 (self-test information) and 0x35 (data-ready settings). The value read from the DMUx81ZA must be parsed according to Table 11 to determine self-test completion status.

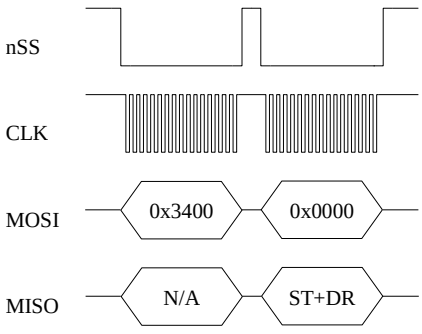


Figure 13 Polled-Read of the Self-Test/Data-Ready Register

5.7 Configuration Registers

5.7.1 Self-Test/Data-Ready

Self-test and data-ready registers are combined into a single 16-bit register at memory location 0x34; individual bits are assigned according to Table 11.

Table 11 Self-Test/Data-Ready Register

(Base Address: 0x34), Read/Write	
Bits	**Description (Default: 0x0004)**
[15:11]	Unused
10	Unit self-test bit (bit reset upon completion of self-test) 0: Disabled (default) 1: Enabled
[9:8]	Unused
[7:3]	Unused
2	Data-ready enable bit 0: Disabled 1: Enabled (default)
1	Data-ready line polarity 0: Low upon data-ready (default) 1: High upon data-ready
0	Unused

The self-test enables the system to test individual sensors by applying a temporary bias to determine if they are responding correctly. Once self-test completes, the self-test bit (bit 10) is reset to indicate that the test is finished. Results of the self-test are store in the status register, 0x3C. To initiate self-test, the master sends 0xB504 across the SPI bus.

The data-ready bits enable the master to enable or disable the data-ready signal provided on pin 7 of the DMUx81ZA and to set the data-ready signal polarity (high or low). To enable data-ready with a high signal, the master sends 0xB406.

5.7.2 Output Data Rate

Output data rate (ODR) is contained in register 0x36; individual bits are assigned according to Table 12. Note: these settings apply only to data output via the DMUx81ZA SPI port and do not affect the low-level UART output port.

Table 12 Output Data Rate/Clock Configuration Register

(Base Address: 0x36), Read/Write	
Bits	**Description (Default: 0x0101)**
[15:12]	Unused

[8:11]	System Output Data Rate 0x00 (0): Data output suppressed 0x01 (1): 200 Hz (default) 0x02 (2): 100 Hz 0x03 (3): 50 Hz 0x04 (4): 25 Hz 0x05 (5): 20 Hz 0x06 (6): 10 Hz 0x07 (7): 5 Hz 0x08 (8): 4 Hz 0x09 (9): 2 Hz 0x10 (10): 1 Hz
[7:0]	Reserved

The ODR enables the master to specify the output rate of data provided by the DMUx81ZA. Setting this register directly affects the data-ready signal. The default ODR is 200 Hz; to change the ODR to 100 Hz, the master sends 0xB702.

5.7.3 Rate-Sensor Scaling/Low-Pass Filter

The rate-sensor scaling and digital low-pass filter configuration are combined into a single 16-bit register at memory location 0x38; individual bits are assigned according to Table 13. Note: these settings apply only to data output via the DMUx81ZA SPI port and do not affect the low-level UART output port.

Table 13 Sensor Scaling/Digital Low-Pass Filter Register

(Base Address: 0x38), Read/Write	
Bits	**Description (Default: 0x0206)**
[15:8]	Rate-Sensor Scaling/Dynamic Range Selector 0x01 (1): +/-62.5°/sec 0x02 (2): +/-125.0°/sec (default) 0x04 (4): +/-250.0°/sec 0x08 (8): +/-500.0°/sec 0x10 (16): +/-1000.0°/sec
[7:0]	Digital Low-Pass Filter 0x00 (0): Unfiltered 0x03 (3): 40 Hz Bartlett 0x04 (4): 20 Hz Bartlett 0x05 (5): 10 Hz Bartlett 0x06 (6): 5 Hz Bartlett (default) 0x30 (48): 50 Hz Butterworth 0x40 (64): 20 Hz Butterworth 0x50 (80): 10 Hz Butterworth 0x60 (96): 5 Hz Butterworth

The rate-sensor scaling selector adjusts the output scaling applied to the rate-sensor values[7] in registers 0x04 through 0x08 as well as the values in the standard data-packet (scaling in the other data-packets are not affected). Additionally, this setting affects the limits that control the sensor over-range bit in the diagnostic status register (Table 9); if the system undergoes motion that exceeds this limit, the over-range bit is set. The default scaling is 125.0°/sec; to change the scaling to 62.5°/sec, the master sends 0xB901.

The rate sensor dynamic range selection maps to a bit-weight scale factor as defined in Table 14.

Table 14 Rate-Sensor Scaling Factor

Dynamic Range	Scale Factor	Signal Limit	Over-Range Limit
+/-62.5°/sec	400 LSB/(°/sec)	+/-80.0 °/sec	+/-62.5 °/sec
+/-125.0°/sec	200 LSB/(°/sec)	+/-160.0 °/sec	+/-125.0 °/sec
+/-250.0°/sec	100 LSB/(°/sec)	+/-200.0 °/sec	+/-220.0 °/sec
+/-500.0°/sec	50 LSB/(°/sec)	+/-400.0 °/sec	+/-440.0 °/sec
+/-1000.0°/sec	25 LSB/(°/sec)	+/-600.0 °/sec	+/-660.0 °/sec

The digital low-pass filter register sets the type and cutoff frequency of the filter applied to the scaled sensor data. The default setting is a 5 Hz Bartlett filter; to switch to a 20 Hz Butterworth filter, the master sends 0xB840. Figure 14 describes the output response of the different Bartlett filter settings.

[7] Limits will affect the signal output only if the system is capable of generating a signal of that level. For instance, for an IMU381ZA-200, a 220 °/sec limit will apply however the 440 °/sec limit will not, as the sensor is incapable of outputting signals greater than 220 °/sec.

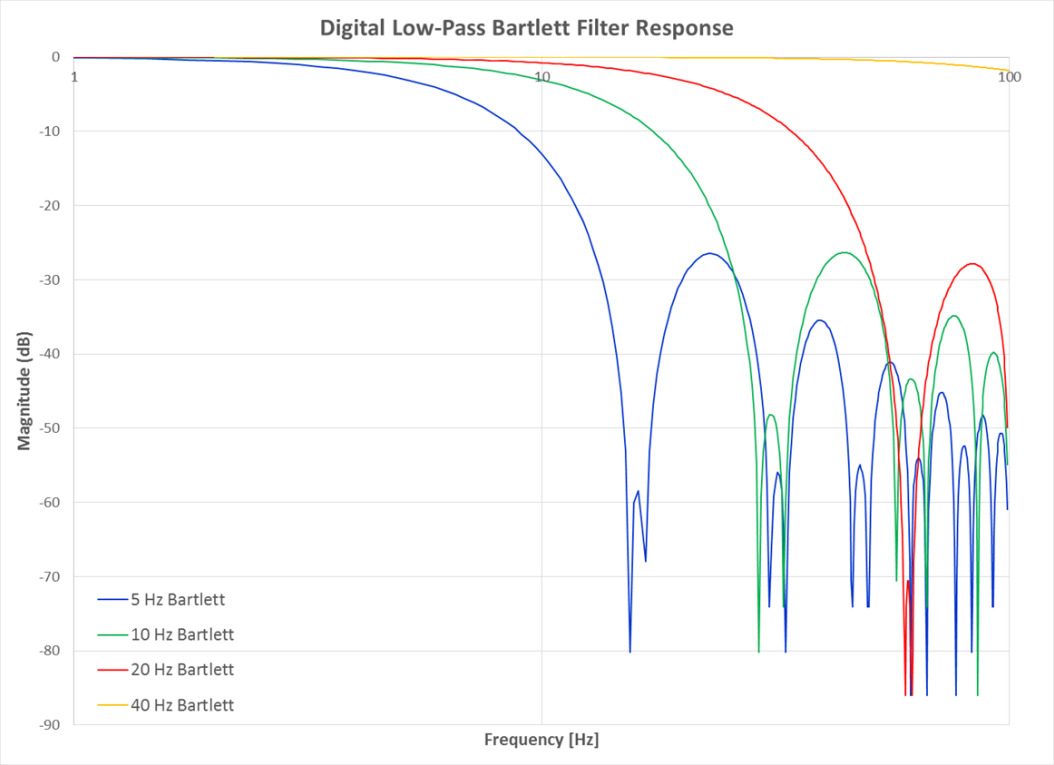


Figure 14 DMUx81ZA Bartlett Filter Response

5.7.4 Accelerometer, Magnetometer, and Alternate Rate-Sensor Scaling

The scaling and limits of the accelerometer and magnetometer output can be configured in a manner similar to the method described in Section 5.7.3. Additionally, the rate-sensor output can be configured via an alternate register. Changes in register 0x38 will be reflected in this alternate register (0x71) and vice-versa.

Rate-sensor and accelerometer scaling and limits are combined into a single 16-bit register at memory location 0x70; individual bits are assigned according to Table 15. Note: these settings apply only to data output via the DMUx81ZA SPI port and do not affect the low-level UART output port.

Table 15 Rate-Sensor and Accelerometer Output Scaling

(Base Address: 0x70), Read/Write	
Bits	**Description (Default: 0x1F3F)**
[15:12]	Rate-Sensor Scaling/Dynamic Range Selector 0x0: +/-62.5°/sec 0x1: +/-125.0°/sec (default) 0x2: +/-250.0°/sec 0x3: +/-500.0°/sec 0x4: +/-1000.0°/sec
[11:8]	Reserved for future use
[7:4]	Accelerometer Scaling/Dynamic Range Selector 0x0: +/-1.0 [g] 0x1: +/-2.0 [g] 0x2: +/-4.0 [g] 0x3: +/-5.0 [g] (default) 0x4: +/-8.0 [g]
[3:0]	Reserved for future use

As described in the previous section, the rate-sensor scaling selector adjusts the output scaling applied to the rate-sensor values as well as the limits that control the sensor over-range bit in the diagnostic status register (Table 9). The accelerometer scaling and limits work in the same fashion.

The rate sensor dynamic range selection maps to a bit-weight scale factor as defined in Table 16. The accelerometer dynamic range mapping is defined in Table 17.

Table 16 Rate-Sensor Scaling Factor

Dynamic Range	Scale Factor	Signal Limit	Over-Range Limit
+/-62.5°/sec	400 LSB/(°/sec)	+/-80.0 °/sec	+/-62.5 °/sec
+/-125.0°/sec	200 LSB/(°/sec)	+/-160.0 °/sec	+/-125.0 °/sec
+/-250.0°/sec	100 LSB/(°/sec)	+/-200.0 °/sec	+/-220.0 °/sec
+/-500.0°/sec	50 LSB/(°/sec)	+/-400.0 °/sec	+/-440.0 °/sec
+/-1000.0°/sec	25 LSB/(°/sec)	+/-600.0 °/sec	+/-660.0 °/sec

Table 17 Accelerometer Scaling Factor

Dynamic Range	Scale Factor	Signal Limit	Over-Range Limit
+/-1.0 [g]	32768 LSB/[g]	+/-1.0 [g]	+/-0.9 [g]
+/-2.0 [g]	16384 LSB/[g]	+/-1.96 [g]	+/-1.8 [g]
+/-4.0 [g]	8192 LSB/[g]	+/-3.92 [g]	+/-3.6 [g]
+/-5.0 [g]	4000 LSB/[g]	+/-4.5 [g]	+/-4.5 [g]
+/-8.0 [g]	4096 LSB/[g]	+/-7.84 [g]	+/-7.2 [g]

Magnetometer scaling and limits are in the register at memory location 0x72; individual bits are assigned according to Table 18.

Table 18 Magnetometer Output Scaling

(Base Address: 0x72), Read/Write	
Bits	**Description (Default: 0x1003)**
[15:12]	Magnetometer Scaling/Dynamic Range Selector 0x0: +/-1.0 [G] 0x1: +/-2.0 [G] (default) 0x2: +/-4.0 [G] 0x3: +/-8.0 [G]
[11:0]	Reserved for future use

Just like the rate-sensor and accelerometer scaling, the magnetometer scaling selector adjusts the output scaling applied to the magnetometer values. However, the limit only affects the sensor output, it does not affect the over-range bit.

The magnetometer dynamic range selection maps to a bit-weight scale factor as defined in Table 19.

Table 19 Magnetometer Scaling Factor

Dynamic Range	Scale Factor	Signal Limit
+/-1.0 [G]	32768 LSB/[G]	+/-0.98 [G]
+/-2.0 [G]	16384 LSB/[G]	+/-1.96 [G]
+/-4.0 [G]	8192 LSB/[G]	+/-3.92 [G]
+/-8.0 [G]	4096 LSB/[G]	+/-7.84 [G]

5.7.5 Axis Orientation Settings

The DMUx81 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 available settings are described in Section 8.5. The specific selections are provided in Table 29. The default setting is (+Ux, +Uy, +Uz).

To specify the orientation over SPI requires the user to write to two SPI registers (0x74 and 0x75) in succession. Writing to register 0x75 prior to 0x74 will have no effect. Additionally, reading the current orientation from register 0x74 will reset the write and require the user to rewrite the two bytes again (if done before both bytes are written).

To write the orientation, the user must first select the orientation and corresponding value from Table 29. Then the value must be split into most-significant and least-significant bytes. The most-significant byte is then written to register 0x74. This is followed by writing the least-significant byte to 0x75. Only by writing the two bytes back-to-back will the selection take effect.

For example, to select an orientation of (-Ux, +Uz, +Uy) the user must write 0x01 to 0x74 followed by 0x11 to 0x75. Note: this register does not require the user to swap bytes for the write to load the bytes properly, unlike other registers.

5.7.6 Saving the Configuration to EEPROM

The DMUx81 enables the user to save certain settings to the EEPROM so they are set automatically the next time the system is started. At this time, only the Orientation field can be saved. To save the value either write the address of the register to 0x76 (to save an individual configuration setting) or a zero to save all settings.

5.7.7 Magnetic-Alignment

On models with magnetometers and AHRS or INS algorithms (INSx81ZA and AHRSx81ZA), the system is capable of compensating for the hard-iron bias and soft-iron scaling of the mounting environment. Once found, the values are used by the Kalman filter algorithm to compensate the heading for the magnetic environment. A complete discussion of the process is discussed in the section *Mag Alignment Procedure* found in Appendix A: Installation and Operation of NAV-VIEW.

To initiate a magnetic alignment over the SPI bus, perform a write to register 0x50 by appending the write-bit/address combination with 0x01, e.g. 0xD001. Table 20 provides a description of the mag-alignment register.

Table 20 Magnetic-Alignment Register

(Base Address: 0x50), Read/Write	
Bits	**Description (Default: 0x0000)**
[15:8]	Mag-Align Initiation byte
[7:0]	Mag-Align Status byte 0x00: Disabled (default) 0x0B: Alignment process complete 0x0C: Alignment process in-progress

Once the mag-align procedure has begun, the Mag-Align Status byte will be set to 0x0C. The master must monitor the least-significant byte of register 0x50 to assess test status. Once the byte changes to 0x0B the alignment procedure is complete. At this point, the hard-iron and soft-iron estimates are written to registers 0x48 through 0x4F and saved to the EEPROM. The Kalman filter algorithm is reset to stabilization mode. It remains in this state for five seconds to allow the user to bring the system to rest while the initialization process completes.

Conversion factors from values in the hard and soft-iron registers (0x48 through 0x4E) to decimal equivalents are provided in Table 21.

Table 21 DMUx81 Magnetic Alignment Parameters

Name	*Register Address*	*Format*	*Scaling*	*Range*	*Units*
X-Axis Hard-Iron Bias	0x48	Signed-Integer	20/2^16	[-10,10]	Gauss
Y-Axis Hard-Iron Bias	0x4A	Signed-Integer	20/2^16	[-10,10]	Gauss
Soft Iron Scale Ratio	0x4C	Unsigned-Integer	2/2^16	[0,2]	N.D.

Soft-Iron Angle	0x4E	Signed-Integer	2*pi/(2^15 – 1)	[-pi,pi]	Radians

5.7.8 Hardware and Software Version

SPI register 0x7E contains information about the hardware and software of the DMUx81, as listed in Table 22. The software version is contains both the major and minor version numbers concatenated. For example, a value of 0x7F = 127, refers to a major version of 12 and a minor version of 7.

Table 22 Hardware and Software Version

(Base Address: 0x7E), Read Only	
Bits	**Description (Default: 0x0000)**
[15:8]	Hardware Version
[7:0]	Software Version (Major and Minor versions concatenated)

5.8 Suggested Operation

The following operational procedure and timing specifications should be adhered to while communicating with the DMUx81 via SPI to ensure proper system operation. These points are further highlighted later in this section.

Startup Timing

The following timing applies at system startup (Figure 15):

- During system setup, the DMUx81 should be held in reset (nRST line held low) until the SPI master is configured and the system is ready to begin communications with the DMUx81
- After releasing the reset line, the DMUx81 requires 550 msec ($t_{System\ Delay}$) before the system is ready for use
- Data should be read from the DMUx81ZA when the data-ready line is set (see Section 5.7.1)

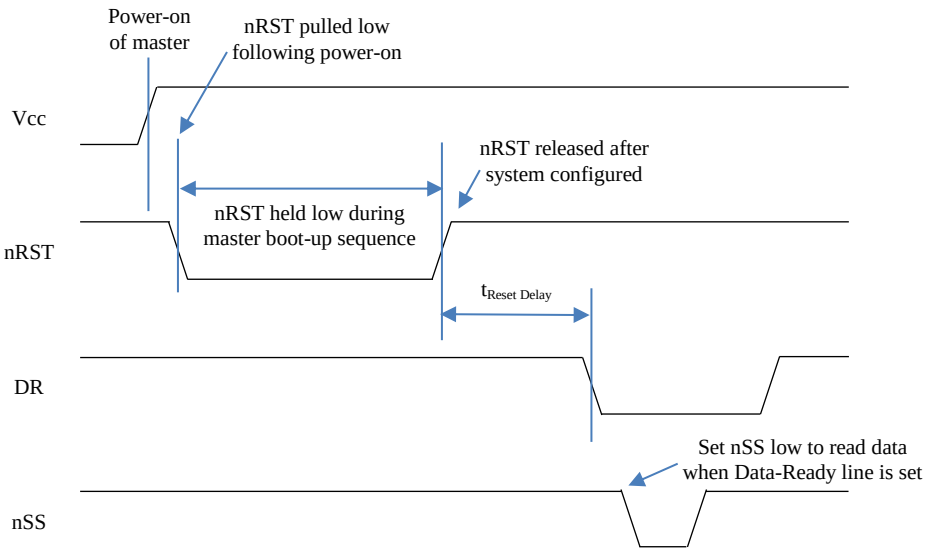


Figure 15 Startup Timing

SPI Timing

The timing requirements for the SPI are listed in Table 23 and illustrated in Figure 16 and Figure 17. In addition, the following operational constraints apply to the SPI communications:

- The unit operates with CPOL = 1 (polarity) and CPHA = 1 (phase)
- Data is transmitted 16-bits words, Most Significant Bit (MSB) first

Table 23 SPI Timing Requirements

Parameter	*Description*	*Value*	*Units*
f_{CL}	SPI clock frequency	2 (max)	MHz
t_{DELAY}	Time between successive clock cycles (Figure 16)	9 (min)	usec
$t_{SU,NSS}$	nSS setup time prior to clocking data (Figure 17)	133	nsec
$t_{h,NSS}$	nSS hold time following clock signal (Figure 17)	67	nsec
$t_{V,MISO}$	Time after falling edge of previous clock-edge that MISO data-bit is invalid (Figure 17)	25	nsec
$t_{SU,MOSI}$	Data input setup time prior to rising edge of clock (Figure 17)	5	nsec
$t_{h,MOSI}$	Data input hold time following rising edge of clock (Figure 17)	4	nsec

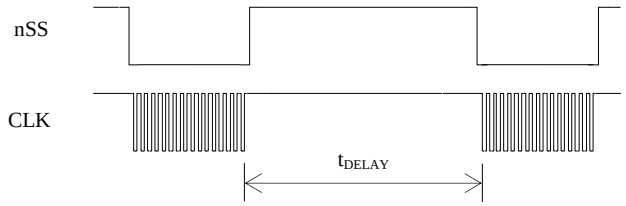


Figure 16 Delay Time

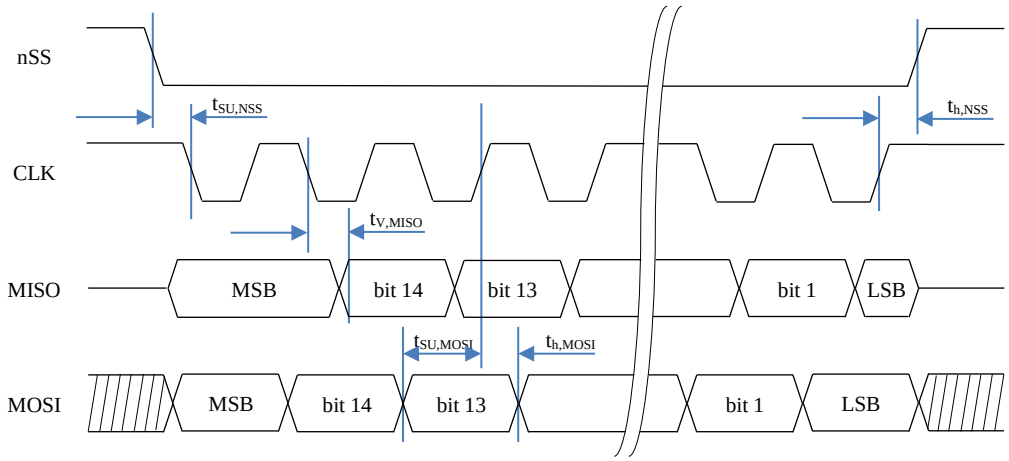


Figure 17 SPI Timing Diagram

5.9 Signal Synchronization (in the DMUX81 product only)

The IMUX81 is capable of synchronizing its output with a 1 kHz external clock signal, in the form of a square wave, applied to Pin 2. When detected, the DMUX81 ignores its internal timer, replacing it with the external clock. Care must be taken to ensure the signal is a true 1 kHz clock, as the firmware will assume all signals on the line have a 1 kHz frequency. Also, once an external sync pulse is applied, the signal must remain or the unit will cease its sampling and processing functions; the system cannot return to internal timing without resetting the system and removing the sync signal.

While providing a 1 kHz clock locks the system's output to the external signal, there still remains ambiguity as to which clock-edge corresponds to the sampling and data-processing task. This is due to the decimation by five of the input clock to the maximum output data rate (200 Hz). The following method will enable the user to create a deterministic lock between the clock and the sampling task of the DMUx81.

Locking Data-Processing to the Input Clock Signal

The following steps describe the process to lock the data-processing to the input clock signal:

1. Hold the external reset line (pin 8) low while applying power to the unit. When ready to configure the unit and receive data, set the reset line high to release the unit from hold.
2. Wait 550 milliseconds to allow initialization of the x81to complete. At this point the data-ready signal will toggle, indicating when the SPI data-buffer is populated with processed data.
3. Configure the unit as needed.
4. At any point following this, begin the synchronization process by providing a 1 kHz square-wave signal to the synchronization input of the DMUX81 (pin 2).
5. The first rising edge of the input clock signal triggers the synchronization process but actual lock does not occur until the 66th rising edge of the input signal. This is due to initialization of the external sync and handoff of control from the internal timer.
6. An additional 5 clock cycles are required before data, synchronized to the external clock, is available. This occurs on the 71st rising edge of the 1 kHz signal (70 milliseconds after the first rising edge).

The lock forces the data-processing task in the firmware to begin on the rising edge of the clock signal. Data-ready (pin 7) is set approximately 400 micro-seconds after the task begins (the time it takes to process the data), indicating that the latest data has been placed into the SPI register. Subsequent data, locked to the external clock signal, is placed into the SPI register depending upon the ODR (set via SPI register 0x37). For instance, if the ODR is set to 100 Hz (corresponding to a value 0x02 in register 0x36), then the process is repeated at every tenth rising edge of the external clock. For a 200 Hz ODR, the process repeats at every fifth rising edge.

Inertial-Sensor Sampling Indicator

When the user requires finer knowledge of the instant that data is sampled, the DMUX81 provides the ability to determine when the sensor read is performed. A falling edge of the signal provided on pin 1 indicates when the inertial sensors are sampled. By combining this information

with the synchronization process described above, the user can account for the sensor latency due to the data-sampling and processing tasks.

5.10 Bootloader

DMUx81 possesses the capability to upgrade the firmware via an on-board bootloader. See Section 10 for a description of the process.

6 DMUx81 UART Port Interface Definition

The DMUx81ZA Series contains a number of different products which have different measurement capabilities. Depending on the model you purchased, various commands and output modes are supported. However, all models support a common packet structure that includes both command or input data packets (data sent to the DMUx81ZA Series) and measurement output or response packet formats (data sent from the DMUx81ZA Series). This section of the manual explains these packet formats as well as the supported commands. NAV-VIEW 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.

For an example of the code required to parse input data packets, please see refer to Appendix C.

For qualified commercial OEM users, a source code license of NAV-VIEW can be made available under certain conditions. Please contact your ACEINNA representative for more information.

6.1 General Settings

The serial port settings are RS232 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, and 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 DMUx81ZA 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 24 defines number formats:

Table 24 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
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)>*

The Ping Command does not require a CRC, so a DMUx81ZA Series unit can be pinged from a terminal emulator. To Ping a DMUx81ZA Series unit, type the ASCII string 'UUPK'. If properly connected, the DMUx81ZA Series unit will respond with 'PK'. All other communications with the DMUx81ZA Series unit require the 2-byte CRC. {Note: A DMUx81ZA 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 C for sample code that implements the 16-bit CRC algorithm.

6.3.6 Messaging Overview

Table 25 summarizes the messages available by DMUx81ZA Series model. Packet types are assigned mostly using the ASCII mnemonics defined above and are indicated in the summary table below 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 DMUx81ZA 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 DMUx81ZA Series inertial system and will result in an associated Reply Message or NAK message. Note that reply messages typically have the same ***<2-byte packet type (U2)>*** as the input message that evoked it but with a different payload.

Table 25 Message Table

ASCII Mnemonic	*<2-byte packet type (U2)>*	*<payload byte-length (U1)>*	Description	Type	Products Available
Link Test					
PK	0x504B	0	Ping Command and Response	Input/Reply Message	ALL
CH	0x4348	N	Echo Command and Response	Input/Reply Message	ALL
Interactive Commands					
GP	0x4750	2	Get Packet Request	Input Message	ALL
AR	0x4152	0	Algorithm Reset	Input/Reply Message	VG,AHRS, INS
NAK	0x1515	2	Error Response	Reply Message	ALL
WC	0x5743	2	Calibrate Command and Response	Input/Reply Message	AHRS, INS
CD	0x4344	10	Calibration Completed	Reply Message	AHRS, INS
Output Messages: Status & Other, (Polled Only)					
ID	0x4944	5+N	Identification Data	Output Message	ALL
VR	0x5652	5	Version Data	Output Message	ALL
T0	0x5430	28	Test 0 (Detailed BIT and Status)	Output Message	ALL
Output Messages: Measurement Data (Continuous or Polled)					
S0	0x5330	30	Scaled Sensor 0 Data	Output Message	IMUx81ZA (-209, -409), AHRS, INS
S1	0x5331	24	Scaled Sensor 1 Data	Output Message	ALL
A1	0x4131	32	Angle 1 Data	Output Message	AHRS, INS
A2	0x4132	30	Angle 2 Data	Output Message	VG, AHRS, INS
A3	0x4133	30	Angle 3 Data	Output Message	VG, AHRS, INS
N0	0x4E30	32	Nav 0 Data	Output Message	VG, AHRS, INS
N1	0x4E31	42	Nav 1 Data	Output Message	VG, AHRS, INS

Advanced Commands					
WF	0x5746	numFields*4+1	Write Fields Request	Input Message	ALL
WF	0x5746	numFields*2+1	Write Fields Response	Reply Message	ALL
SF	0x5346	numFields*4+1	Set Fields Request	Input Message	ALL
SF	0x5346	numFields*2+1	Set Fields Response	Reply Message	ALL
RF	0x5246	numFields*2+1	Read Fields Request	Input Message	ALL
RF	0x5246	numFields*4+1	Read Fields Response	Reply Message	ALL
GF	0x4746	numFields*2+1	Get Fields Request	Input Message	ALL
GF	0x4746	numFields*4+1	Get Fields Response	Reply Message	ALL

7 DMUx81 Standard UART Port Commands and Messages

7.1 Link Test.

7.1.1 Ping Command

Ping ('PK' = 0x504B)			
Preamble	Packet Type	Length	Termination
0x5555	0x504B	-	-

The ping command has no payload. Sending the ping command will cause the unit to send a ping response. To facilitate human input from a terminal, the length and CRC fields are not required. (Example: 0x5555504B009ef4 or 0x5555504B))

7.1.2 Ping Response

Ping ('PK' = 0x504B)			
Preamble	Packet Type	Length	Termination
0x5555	0x504B	0x00	<CRC (U2)>

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

7.1.3 Echo Command

Echo ('CH' = 0x4348)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4348	N	<echo payload>	<CRC (U2)>

The echo command allows testing and verification of the communication link. The unit will respond with an echo response containing the *echo data*. The *echo data* is N bytes long.

7.1.4 Echo Response

Echo Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	echoData0	U1	-	-	first byte of echo data
1	echoData1	U1	-	-	Second byte of echo data
…	…	U1	-	-	Echo data
N-2	echoData...	U1	-	-	Second to last byte of echo data
N-1	echoData…	U1	-	-	Last byte of echo data

7.2 Interactive Commands

Interactive commands are used to interactively request data from the DMUx81ZA Series, and to calibrate or reset the DMUx81ZA Series.

7.2.1 Get Packet Request

Get Packet ('GP' = 0x4750)				
Preamble	Packet Type	Length	Payload	Termination

0x5555	0x4750	0x02	<GP payload>	<CRC (U2)>

This command allows the user to poll for both measurement packets and special purpose output packets including 'T0', 'VR', and 'ID'.

GP Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	requestedPacketType	U2	-	-	The requested packet type

Refer to the sections below for Packet Definitions sent in response to the 'GP' command

7.2.2 Algorithm Reset Command

Algorithm Reset ('AR' = 0x4152)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4152	0x00	-	<CRC (U2)>

This command resets the state estimation algorithm without reloading fields from EEPROM. All current field values will remain in affect. The unit will respond with an algorithm reset response.

7.2.3 Algorithm Reset Response

Algorithm Reset ('AR' = 0x4152)			
Preamble	Packet Type	Length	Termination
0x5555	0x4152	0x00	<CRC (U2)>

The unit will send this packet in response to an algorithm reset command.

7.2.4 Calibrate Command

Calibrate ('WC' = 0x5743)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x5743	0x02	<WC payload>	<CRC (U2)>

This command allows the user to perform various calibration tasks with the DMUx81ZA Series. See the calibration command table below for details. The unit will respond immediately with a calibrate response containing the *calibrationRequest* received or an error response if the command cannot be performed.

WC Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	calibrationRequest	U2	-	-	The requested calibration task

Currently, magnetic alignment is the only function supported by the calibrate command. There are two magnetic alignment procedures supported; (1) magnetic alignment with automatic yaw tracking termination, and magnetic alignment without automatic termination.

calibrationRequest	*Description*
0x0009	**Begin magnetic alignment** without automatic termination. Rotate vehicle through >360 degrees yaw and then send 0x000B calibration request to terminate.
0x000B	Terminate magnetic alignment. The unit will send a CC response containing the hard-iron and soft-iron values. To accept the parameters, store them using the write magnetic calibration command.
0x000C	Begin magnetic calibration with automatic termination. Rotate the unit through x81 degrees in yaw. The unit will send a CC response containing the hard-iron and soft-iron values upon completion of the turn. To accept the parameters, store them using the write magnetic calibration command.
0x000E	Write magnetic calibration. The unit will write the parameters to EEPROM and then send a calibration response.

7.2.5 Calibrate Acknowledgement Response

Calibrate ('WC' = 0x5743)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x5743	0x02	<WC payload>	<CRC (U2)>

The unit will send this packet in response to a calibrate request if the procedure can be performed or initiated.

WC Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	calibrationRequest	U2	-	-	The requested calibration task

7.2.6 Calibration Completed Parameters Response

Calibrate Completed ('CD' = 0x4344)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4344	0x0A	<CD payload>	<CRC (U2)>

The unit will send this packet after a calibration has been completed. Currently, there is only one message of this type sent after a magnetic calibration has been completed (with or without automatic termination) and the parameters have been calculated. Thus, the calibrationRequest field will be 0x000B or 0x000C.

CD Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	calibrationRequest	U2	-	-	The requested calibration task
2	xHardIron	I2	20/2^16	G	The x hard iron bias
4	yHardIron	I2	20/2^16	G	The y hard iron bias

6	softIronScaleRatio	U2	2/2^16	-	The scaling ratio between the x and y axis
8	softIronAngle	I2	2*pi/2^16 (360/2^16)	Rad Deg	The soft iron phase angle between x and y axis

7.2.9 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 *faiiledInputPacketType* request if it could not be completed successfully.

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

7.3 Output Packets (Polled)

The following packet formats are special informational packets which can be requested using the 'GP' command.

7.3.1 Identification Data Packet

Identification Data ('ID' = 0x4944)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4944	5+N	<ID payload>	<CRC (U2)>

This packet contains the unit *serialNumber* and *modelString*. The model string is terminated with 0x00. The model string contains the programmed versionString (8-bit Ascii values) followed by the firmware part number string delimited by a whitespace.

ID Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	serialNumber	U4	-	-	Unit serial number
4	modelString	SN	-	-	Unit Version String
4+N	0x00	U1	-	-	Zero Delimiter

7.3.2 Version Data Packet

Version Data ('VR' = 0x5652)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x5652	5	<VR payload>	<CRC (U2)>

This packet contains firmware version information. *majorVersion* changes may introduce serious incompatibilities. *minorVersion* changes may add or modify functionality, but maintain backward

compatibility with previous minor versions. *patch* level changes reflect bug fixes and internal modifications with little effect on the user. The build *stage* is one of the following: 0=release candidate, 1=development, 2=alpha, 3=beta. The *buildNumber* is incremented with each engineering firmware build. The *buildNumber* and *stage* for released firmware are both zero. The final beta candidate is v.w.x.3.y, which is then changed to v.w.x.0.1 to create the first release candidate. The last release candidate is v.w.x.0.z, which is then changed to v.w.x.0.0 for release.

VR Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	majorVersion	U1	-	-	Major firmware version
1	minorVersion	U1	-	-	Minor firmware version
2	patch	U1	-	-	Patch level
3	stage	-	-	-	Development Stage (0=release candidate, 1=development, 2=alpha, 3=beta)
4	buildNumber	U1	-	-	Build number

7.3.3 Test 0 (Detailed BIT and Status) Packet

Test ('T0' = 0x5430)				
Preamble	Packet Type	Length	Payload	Termination
03.3x5555	0x5430	0x1C	<T0 payload>	<CRC (U2)>

This packet contains detailed BIT and status information. The full BIT Status details are described in Section 9 of this manual.

T0 Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	BITstatus	U2	-	-	Master BIT and Status Field
2	hardwareBIT	U2	-	-	Hardware BIT Field
4	hardwarePowerBIT	U2	-	-	Hardware Power BIT Field
6	hardwareEnvironmentalBIT	U2	-	-	Hardware Environmental BIT Field
8	comBIT	U2	-	-	communication BIT Field
10	comSerialABIT	U2	-	-	Communication Serial A BIT Field
12	comSerialBBIT	U2	-	-	Communication Serial B BIT Field
14	softwareBIT	U2	-	-	Software BIT Field
16	softwareAlgorithmBIT	U2	-	-	Software Algorithm BIT Field
18	softwareDataBIT	U2	-	-	Software Data BIT Field
20	hardwareStatus	U2	-	-	Hardware Status Field
22	comStatus	U2	-	-	Communication Status Field
24	softwareStatus	U2	-	-	Software Status Field
26	sensorStatus	U2	-	-	Sensor Status Field

7.4 Output Packets (Polled or Continuous)

7.4.1 Scaled Sensor Data Packet 0

Scaled Sensor Data ('S0' = 0x5330)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x5330	0x1E	<S0 payload>	<CRC (U2)>

This packet contains scaled sensor data. The scaled sensor data is fixed point, 2 bytes per sensor, MSB first, for 13 sensors in the following order: accels(x,y,z); gyros(x,y,z); mags(x,y,z); temps(x,y,z,board). Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees. Note the timer value can be used for synchronization and computation of DeltaT. It may appear in NAV-VIEW log files under another column heading.

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10) g

Magnetometers: scaled to a range of [-1,+1) Gauss

Temperature: scaled to a range of [-100, +100)°C

S0 Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	xAccel	I2	20/2^16	g	X accelerometer
2	yAccel	I2	20/2^16	g	Y accelerometer
4	zAccel	I2	20/2^16	g	Z accelerometer
6	xRate	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate
8	yRate	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate
10	zRate	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate
12	xMag	I2	20/2^16	Gauss	X magnetometer
14	yMag	I2	20/2^16	Gauss	Y magnetometer
16	zMag	I2	20/2^16	Gauss	Z magnetometer
18	xRateTemp	I2	200/2^16	deg. C	X rate temperature
20	yRateTemp	I2	200/2^16	deg. C	Y rate temperature
22	zRateTemp	I2	200/2^16	deg. C	Z rate temperature
24	boardTemp	I2	200/2^16	deg. C	CPU board temperature
26	timer	U2	15.259022	uS	Free running fast counter 1s= 65535, captured at sampling
28	BITstatus	U2	-	-	Master BIT and Status

7.4.2 Scaled Sensor Data Packet 1 (Default IMU Data)

Scaled Sensor Data ('S1' = 0x5331)				
Preamble	Packet Type	Length	Payload	Termination

0x5555	0x5331	0x18	<S1 payload>	<CRC (U2)>

This packet contains scaled sensor data. Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees. Note the timer value can be used for synchronization and computation of DeltaT. It may appear in NAV-VIEW log files under another column heading.

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10)g

Temperature: scaled to a range of [-100, +100)°C

S1 Payload Contents

Byte Offset	Name	Format	Scaling	Units	Description
0	xAccel	I2	20/2^16	g	X accelerometer
2	yAccel	I2	20/2^16	g	Y accelerometer
4	zAccel	I2	20/2^16	g	Z accelerometer
6	xRate	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate
8	yRate	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate
10	zRate	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate
12	xRateTemp	I2	200/2^16	deg. C	X rate temperature
14	yRateTemp	I2	200/2^16	deg. C	Y rate temperature
16	zRateTemp	I2	200/2^16	deg. C	Z rate temperature
18	boardTemp	I2	200/2^16	deg. C	CPU board temperature
20	timer	U2	15.259022	uS	Free running fast counter 1s= 65535, captured at sampling
22	BITstatus	U2	-	-	Master BIT and Status

7.4.3 Angle Data Packet 1 (Default AHRS Data)

Angle Data ('A1' = 0x4131)

Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4131	0x20	<A1 payload>	<CRC (U2)>

This packet contains angle data and selected sensor data scaled in most cases to a signed 2^16 2's complement number. Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees. Counter may appear in NAV-VIEW log files under another column heading.

Angles: scaled to a range of [-pi,+pi) or [-180 deg to +180 deg).

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10) g

Magnetometers: scaled to a range of [-10,+10) Gauss

Temperature: scaled to a range of [-100, +100) °C

A1 Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	rollAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Roll angle
2	pitchAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Pitch angle
4	yawAngleMag	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Yaw angle (magnetic north)
6	xRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate Corrected
8	yRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate Corrected
10	zRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate Corrected
12	xAccel	I2	20/2^16	g	X accelerometer
14	yAccel	I2	20/2^16	g	Y accelerometer
16	zAccel	I2	20/2^16	g	Z accelerometer
18	xMag	I2	20/2^16	Gauss	X magnetometer
20	yMag	I2	20/2^16	Gauss	Y magnetometer
22	zMag	I2	20/2^16	Gauss	Z magnetometer
24	xRateTemp	I2	200/2^16	Deg C	X rate temperature
26	counter	U4	1	ms	Free running counter, captured at sampling
30	BITstatus	U2	-	-	Master BIT and Status

7.4.4 Angle Data Packet 2 (Default VG Data)

Angle Data ('A2' = 0x4132)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4132	0x1E	<A2 payload>	<CRC (U2)>

This packet contains angle data and selected sensor data scaled in most cases to a signed 2^16 2's complement number. Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees. Counter may appear in NAV-VIEW log files under another column heading.

Angles: scaled to a range of [-pi,+pi) or [-180 deg to +180 deg).

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10) g

Temperature: scaled to a range of [-100, +100) °C

A2 Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	rollAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Roll angle
2	pitchAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Pitch angle
4	yawAngleTrue	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Yaw angle (free)
6	xRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate corrected
8	yRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate corrected
10	zRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate corrected
12	xAccel	I2	20/2^16	g	X accelerometer
14	yAccel	I2	20/2^16	g	Y accelerometer
16	zAccel	I2	20/2^16	g	Z accelerometer
18	xRateTemp	I2	200/2^16	deg. C	X rate temperature
20	yRateTemp	I2	200/2^16	deg. C	Y rate temperature
22	zRateTemp	I2	200/2^16	deg. C	Z rate temperature
24	counter	U4	1	ms	Free running counter, captured at sampling
28	BITstatus	U2	-	-	Master BIT and Status

7.4.5 Angle Data Packet 3

Angle Data ('A3' = 0x4133)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4133	0x1E	<A3 payload>	<CRC (U2)>

This packet contains angle data and selected sensor data scaled in most cases to a signed 2^16 2's complement number. Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees. Counter may appear in NAV-VIEW log files under another column heading.

Angles: scaled to a range of [-pi,+pi) or [-180 deg to +180 deg).

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10) g

Temperature: scaled to a range of [-100, +100) °C

A3 Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	rollAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Roll angle
2	pitchAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Pitch angle
4	yawAngleTrue	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Yaw angle (free)
6	xRateScaled	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate scaled
8	yRateScaled	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate scaled
10	zRateScaled	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate scaled
12	xAccel	I2	20/2^16	g	X accelerometer
14	yAccel	I2	20/2^16	g	Y accelerometer
16	zAccel	I2	20/2^16	g	Z accelerometer
18	xRateTemp	I2	200/2^16	deg. C	X rate temperature
20	yRateTemp	I2	200/2^16	deg. C	Y rate temperature
22	zRateTemp	I2	200/2^16	deg. C	Z rate temperature
24	counter	U4	1	ms	Free running counter, captured at sampling
28	BITstatus	U2	-	-	Master BIT and Status

7.4.6 Nav Data Packet 0

Nav Data ('N0' = 0x4E30)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4E30	0x20	<N0 payload>	<CRC (U2)>

This packet contains navigation data and selected sensor data scaled in most cases to a signed 2^16 2's complement number. Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees.

Angles: scaled to a range of [-pi,+pi) or [-180 deg to +180 deg).

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10) g

Temperature: scaled to a range of [-100, +100) °C

Velocities are scaled to a range of [-256,256) m/s

Altitude is scaled to a range of [-100,16284) m using a shifted 2's complement representation.

Longitude and latitude are scaled to a range of [-pi,pi) or [-180 deg to +180 deg).

N0 Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	rollAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Roll angle
2	pitchAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Pitch angle
4	yawAngleTrue	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Yaw angle (true north)
6	xRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate corrected
8	yRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate corrected
10	zRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate corrected
12	nVel	I2	512/2^16	m/s	North velocity
14	eVel	I2	512/2^16	m/s	East velocity
16	dVel	I2	512/2^16	m/s	Down velocity
18	longitude	I4	2*pi/2^32 [360°/2^32]	Radians [°]	Longitude
22	latitude	I4	2*pi/2^32 [360°/2^32]	Radians [°]	Latitude
26	altitude	I2*	2^14/2^16	m	GPS altitude [-100,16284)
28	ITOW	U2	truncated	ms	ITOW (lower 2 bytes)
30	BITstatus	U2	-	-	Master BIT and Status

7.4.7 Nav Data Packet 1 (Default INS Data)

Nav Data ('N1' = 0x4E31)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4E31	0x2A	<N1 payload>	<CRC (U2)>

This packet contains navigation data and selected sensor data scaled in most cases to a signed 2^16 2's complement number. Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees.

Angles: scaled to a range of [-pi,+pi) or [-180 deg to +180 deg).

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10) g

Temperature: scaled to a range of [-100, +100) °C

Velocities are scaled to a range of [-256,256) m/s

Altitude is scaled to a range of [-100,16284) m using a shifted 2's complement representation.

Longitude and latitude are scaled to a range of [-pi,pi) or [-180 deg to +180 deg).

N1 Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	rollAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Roll angle
2	pitchAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Pitch angle
4	yawAngleTrue	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Yaw angle (true north)
6	xRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate corrected
8	yRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate corrected
10	zRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate corrected
12	xAccel	I2	20/2^16	g	X accelerometer
14	yAccel	I2	20/2^16	g	Y accelerometer
16	zAccel	I2	20/2^16	g	Z accelerometer
18	nVel	I2	512/2^16	m/s	North velocity
20	eVel	I2	512/2^16	m/s	East velocity
22	dVel	I2	512/2^16	m/s	Down velocity
24	longitude	I4	2*pi/2^32 [360°/2^32]	Radians [°]	Longitude
28	latitude	I4	2*pi/2^32 [360°/2^32]	Radians [°]	Latitude
32	altitude	I2*	2^14/2^16	m	Altitude [-100,16284)
34	xRateTemp	I2	200/2^16	deg C	X rate sensor temperature
36	ITOW	U4	1	ms	ITOW (sync to GPS)
40	BITstatus	U2	-	-	Master BIT and Status

8 DMUx81ZA Advanced UART Port Commands

The advanced commands allow users to programmatically change the DMUx81ZA Series settings. This section of the manual documents all of the settings and options contained under the Unit Configuration tab within NAV-VIEW. Using these advanced commands, a user's system can change or modify the settings without the need for NAV-VIEW.

8.1 Configuration Fields

Configuration fields determine various behaviors of the unit that can be modified by the user. These include settings like baud rate, packet output rate and type, algorithm type, etc. These fields are stored in EEPROM and loaded on power up. These fields can be read from the EEPROM using the 'RF' command. These fields can be written to the EEPROM affecting the default power up behavior using the 'WF' command. The current value of these fields (which may be different from the value stored in the EEPROM) can also be accessed using the 'GF' command. All of these fields can also be modified immediately for the duration of the current power cycle using the 'SF' command. The unit will always power up in the configuration stored in the EEPROM. Configuration fields can only be set or written with valid data from Table 26 below.

Table 26 Configuration Fields

configuration fields	*field ID*	*Valid Values*	*description*
Packet rate divider	0x0001	0,1,2,4,5,10, 20, 25, 50	quiet, 100Hz, 50Hz, 25Hz, 20Hz, 10Hz, 5Hz, 4Hz, 2Hz
Unit BAUD rate	0x0002	2,3,5,6	38400, 57600, 115200, 230400
Continuous packet type	0x0003	Any output packet type	Not all output packets available for all products. See detailed product descriptions.
Unused	0x0004		
Gyro Filter Setting	0x0005	18750-65535 [2Hz] 8035-18749 [5Hz] 4018-8034 [10Hz] 2411-4017 [20Hz] 1741-2410 [25Hz] 1205-1740 [40Hz] 1-1204 [50 Hz] 0 [Unfiltered]	Sets low pass cutoff for rate sensors. Cutoff Frequency choices are 2, 5, 10, 20, 25, 40, and 50Hz
Accelerometer Filter Setting	0x0006	18750-65535 [2Hz] 8035-18749 [5Hz] 4018-8034 [10Hz] 2411-4017 [20Hz] 1741-2410 [25Hz] 1205-1740 [40Hz] 1-1204 [50 Hz] 0 [Unfiltered]	Sets low pass cutoff for accelerometers. Cutoff Frequency choices are 2, 5, 10, 20, 25, 40, and 50Hz
Orientation	0x0007	See below	Determine forward, rightward, and downward facing sides
User Behavior Switches	0x0008	Any	Free Integrate (60 sec), Use Mags, Use GPS, Stationary Yaw Lock, …
X Hard Iron Bias	0x0009	Any	I2 scaled from [-1,1)

Y Hard Iron Bias	0x000A	Any	I2 scaled from [-1,1)
Soft Iron Scale Ratio	0x000B	Any	U2 scaled from [0,2)
Soft Iron Phase Angle	0x000E	Any	I2 scaled from [-pi,pi]

Note: BAUD rate SF has immediate effect. Some output data may be lost. Response will be received at new BAUD rate.

8.2 Continuous Packet Type Field

This is the packet type that is being continually output. The supported packet depends on the model number. Please refer to Section 7.4 for a complete list of the available packet types.

8.3 Digital Filter Settings

These two fields set the digital low pass filter cutoff frequencies (See Table 27). Each sensor listed is defined in the default factory orientation. Users must consider any additional rotation to their intended orientation.

Table 27 Digital Filter Settings

Filter Setting	Sensor
FilterGyro	Ux,Uy,Uz Rate
FilterAccel	Ux,Uy,Uz Accel

8.4 Orientation Field

This field defines the rotation from the factory to user axis sets. This rotation is relative to the default factory orientation for the appropriate DMUx81 family model. The default factory axis setting for the IMUx81ZA-200 orientation field is (-Uy, -Ux, -Uz) which defines the connector pointing in the +Z direction and the +X direction going from the connector through the serial number label at the end of the DMUx81. The user axis set (X, Y, Z) as defined by this field setting is depicted in Figure 18 below:

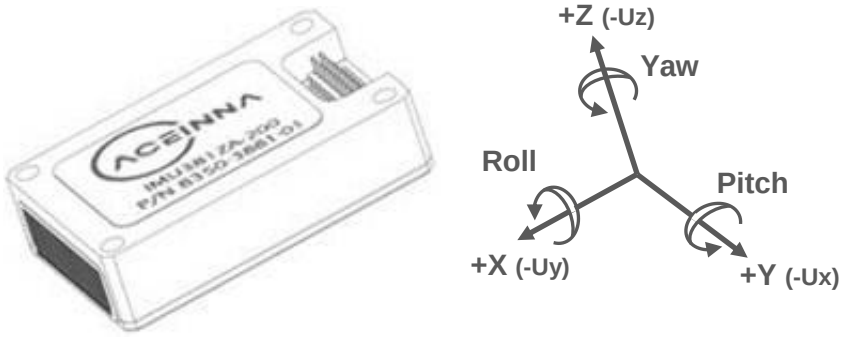


Figure 18 IMUx81ZA-200 Default Orientation Field (0x006B)

Table 28 DMUx81 Orientation Definitions

Description	*Bits*	*Meaning*
X Axis Sign	0	0 = positive, 1 = negative
X Axis	1:2	0 = Ux, 1 = Uy, 2 = Uz, 3 = N/A

Y Axis Sign	3	0 = positive, 1 = negative
Y Axis	4:5	0 = Uy, 1 = Uz, 2 = Ux, 3 = N/A
Z Axis Sign	6	0 = positive, 1 = negative
Z Axis	7:8	0 = Uz, 1 = Ux, 2 = Uy, 3 = N/A
Reserved	9:15	N/A

There are 24 possible orientation configurations (See Table 29). Setting/Writing the field to anything else generates a NAK and has no effect.

Table 29 DMUx81 Orientation Fields

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

The default factory axis setting for all other DMUx81 family model's orientation field is (+Ux, +Uy, +Uz) which defines the base of the DMUx81 pointing in the +Z direction and the +Y direction going from the serial number label at the end through the connector of the DMUx81. The user axis set (X, Y, Z) as defined by this field setting is depicted in Figure 19 below:

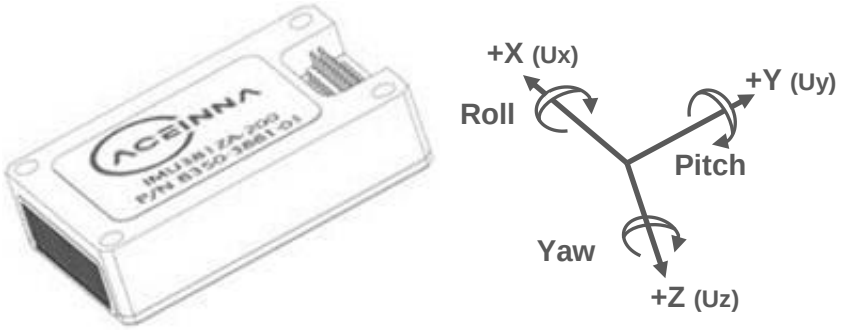


Figure 19 IMU381ZA (-209, -409) VG/AHRS/INS381ZA (-200, -400) Default Orientation Field (0x0000)

8.5 User Behavior Switches

This field allows on the fly user interaction with behavioral aspects of the algorithm (See Table 30).

Table 30 DMUx81 Behavior Switches

Description	*Bits*	*Meaning*
Free Integrate	0	0 = use feedback to stabilize the algorithm, 1 = 6DOF inertial integration without stabilized feedback for 60 seconds
Use Mags	1	0 = Do not use mags to stabilize heading (heading will run open loop or be stabilized by GPS track), 1 = Use mags to stabilize heading
Use GPS	2	0 = Do not use GPS to stabilize the system, 1 = Use GPS when available
Stationary Yaw Lock	3	0 = Do not lock yaw when GPS speed is near zero (<0.75 m/s), 1 = Lock yaw when GPS speed is near zero
Restart on Over-range	4	0 = Do not restart the system after a sensor over-range, 1 = restart the system after a sensor over-range
Dynamic Motion	5	0=vehicle is static, force high gain corrections, 1= vehicle is dynamic, use nominal corrections
Reserved	6:15	N/A

8.6 Hard and Soft Iron Values

These fields allow access to hard iron bias and soft iron scale ratio values for magnetometer alignment (See Table 31):

Table 31 DMUx81 Magnetic Alignment Parameters

Field Name	*Field ID*	*Format*	*Scaling*	*Units*
X Hard Iron Bias	0x0009	I2	2/2^16	Gauss
Y Hard Iron Bias	0x000A	I2	2/2^16	Gauss
Soft Iron Scale Ratio	0x000B	U2	2/2^16	-
Soft Iron Phase Angle	0x000E	I2	2*pi/2^16	Radians

The hard iron bias values are scaled from [-1,1] Gauss. These values are subtracted from the tangent plane magnetometer vector before the heading reference is calculated for the filter. The soft iron scale ratio is scaled from [0,2] and is multiplied by the tangent plane x magnetometer value before the heading reference is calculated for the filter. The soft iron phase angle is scaled from [-pi,pi] and is applied to the tangent plane x magnetometer value before the heading reference is calculated for the filter. This compensates for elliptical soft iron errors that generate an ellipse at an angle away from the major or minor axis following the full rotation of a magnetometer alignment. Note that none of these parameters are applied to the output magnetometer vector data in message A1. They are only applied internally to the data for use in the heading reference for the Kalman filter.

8.7 Heading Track Offset

This field is used to set the offset between vehicle heading and vehicle track to be used by the navigation mode filter when no magnetometer heading measurements are available (See Table 32).

Table 32 DMUx81 Heading Track Offset

Field Name	*Field ID*	*Format*	*Scaling*	*Units*
Heading Track Offset	0x000C	I2	2*pi/2^16 [360°/2^16]	Radians (heading-track) [°]

8.8 Commands to Program Configuration

8.8.1 Write Fields Command

Write Fields ('WF' = 0x5746)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x5746	1+numFields*4	<WF payload>	<CRC (U2)>

This command allows the user to write default power-up configuration fields to the EEPROM. Writing the default configuration will not take affect until the unit is power cycled. *NumFields* is the number of words to be written. The *field0, field1, etc.* are the field

IDs that will be written with the *field0Data, field1Data, etc.*, respectively. The unit will not write to calibration or algorithm fields. If at least one field is successfully written, the unit will respond with a write fields response containing the field IDs of the successfully written fields. If any field is unable to be written, the unit will respond with an error response. Note that both a write fields and an error response may be received as a result of a write fields command. Attempts to write a field with an invalid value is one way to generate an error response. A table of field IDs and valid field values is available in Section 8.1.

WF Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	numFields	U1	-	-	The number of fields to write
1	field0	U2	-	-	The first field ID to write
3	field0Data	U2	-	-	The first field ID's data to write
5	field1	U2	-	-	The second field ID to write
7	field1Data	U2	-	-	The second field ID's data

…	…	U2	-	-	…
numFields*4 -3	field…	U2	-	-	The last field ID to write
numFields*4 -1	field…Data	U2	-	-	The last field ID's data to write

Write Fields Response

Write Fields ('WF' = 0x5746)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x5746	1+numFields*2	<WF payload>	<CRC (U2)>

The unit will send this packet in response to a write fields command if the command has completed without errors.

WF Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	numFields	U1	-	-	The number of fields written
1	field0	U2	-	-	The first field ID written
3	field1	U2	-	-	The second field ID written
…	…	U2	-	-	More field IDs written
numFields*2 – 1	Field…	U2	-	-	The last field ID written

8.8.2 Set Fields Command

Set Fields ('SF' = 0x5346)				
Preamble	Packet Type	Length	Payload	Termination
0x5555	0x5346	1+numFields*4	<SF payload>	<CRC (U2)>

This command allows the user to set the unit's current configuration (SF) fields immediately which will then be lost on power down. *NumFields* is the number of words to be set. The *field0, field1, etc.* are the field IDs that will be written with the *field0Data, field1Data, etc.*, respectively. This command can be used to set configuration fields. The unit will not set calibration or algorithm fields. If at least one field is successfully set, the unit will respond with a set fields response containing the field IDs of the successfully set fields. If any field is unable to be set, the unit will respond with an error response. Note that both a set fields and an error response may be received as a result of one set fields command. Attempts to set a field with an invalid value is one way to generate an error response. A table of field IDs and valid field values is available in Section 8.1.

SF Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	numFields	U1	-	-	The number of fields to set
1	field0	U2	-	-	The first field ID to set
3	field0Data	U2	-	-	The first field ID's data to set
5	field1	U2	-	-	The second field ID to set
7	field1Data	U2	-	-	The second field ID's data to set
…	…	U2	-	-	…

numFields*4 -3	field…	U2	-	-	The last field ID to set
numFields*4 -1	field…Data	U2	-	-	The last field ID's data to set

Set Fields Response

Set Fields ('SF' = 0x5346)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5346	1+numFields*2	<SF payload>	<CRC (U2)>

The unit will send this packet in response to a set fields command if the command has completed without errors.

SF Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	numFields	U1	-	-	The number of fields set
1	field0	U2	-	-	The first field ID set
3	field1	U2	-	-	The second field ID set
…	…	U2	-	-	More field IDs set
numFields*2 - 1	Field…	U2	-	-	The last field ID set

8.9 Read Fields Command

Read Fields ('RF' = 0x5246)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5246	1+numFields*2	<RF payload>	<CRC (U2)>

This command allows the user to read the default power-up configuration fields from the EEPROM. *NumFields* is the number of fields to read. The *field0, field1, etc.* are the field IDs to read. RF may be used to read configuration and calibration fields from the EEPROM. If at least one field is successfully read, the unit will respond with a read fields response containing the field IDs and data from the successfully read fields. If any field is unable to be read, the unit will respond with an error response. Note that both a read fields and an error response may be received as a result of a read fields command.

RF Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	numFields	U1	-	-	The number of fields to read
1	field0	U2	-	-	The first field ID to read
3	field1	U2	-	-	The second field ID to read
…	…	U2	-	-	More field IDs to read
numFields*2 - 1	Field…	U2	-	-	The last field ID to read

8.10 Read Fields Response

Read Fields ('RF' = 0x5246)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5246	1+numFields*4	<RF payload>	<CRC (U2)>

The unit will send this packet in response to a read fields request if the command has completed without errors.

RF Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	numFields	U1	-	-	The number of fields read
1	field0	U2	-	-	The first field ID read
3	field0Data	U2	-	-	The first field ID's data read
5	field1	U2	-	-	The second field ID read
7	field1Data	U2	-	-	The second field ID's data read
…	…	U2	-	-	…
numFields*4 -3	field…	U2	-	-	The last field ID read
numFields*4 -1	field…Data	U2	-	-	The last field ID's data read

8.11 Get Fields Command

Get Fields ('GF' = 0x4746)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4746	1+numFields*2	<GF Data>	<CRC (U2)>

This command allows the user to get the unit's current configuration fields. *NumFields* is the number of fields to get. The *field0, field1, etc.* are the field IDs to get. GF may be used to get configuration, calibration, and algorithm fields from RAM. Multiple algorithm fields will not necessarily be from the same algorithm iteration. If at least one field is successfully collected, the unit will respond with a get fields response with data containing the field IDs of the successfully received fields. If any field is unable to be received, the unit will respond with an error response. Note that both a get fields and an error response may be received as the result of a get fields command.

GF Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	numFields	U1	-	-	The number of fields to get
1	field0	U2	-	-	The first field ID to get
3	field1	U2	-	-	The second field ID to get
…	…	U2	-	-	More field IDs to get
numFields*2 – 1	Field…	U2	-	-	The last field ID to get

8.12 Get Fields Response

Get Fields ('GF' = 0x4746)

Preamble	Packet Type	Length	Payload	Termination
0x5555	0x4746	1+numFields*4	<GF Data>	<CRC (U2)>

The unit will send this packet in response to a get fields request if the command has completed without errors.

GF Payload Contents					
Byte Offset	Name	Format	Scaling	Units	Description
0	numFields	U1	-	-	The number of fields retrieved
1	field0	U2	-	-	The first field ID retrieved
3	field0Data	U2	-	-	The first field ID's data retrieved
5	field1	U2	-	-	The second field ID retrieved
7	field1Data	U2	-	-	The second field ID's data
…	…	U2	-	-	…
numFields*4 -3	field…	U2	-	-	The last field ID retrieved
numFields*4 -1	field…Data	U2	-	-	The last field ID's data retrieved

9 DMUx81ZA Advanced UART Port BIT

9.1 Built In Test (BIT) and Status Fields

Internal health and status are monitored and communicated in both hardware and software. The ultimate indication of a fatal problem is a hardware BIT signal on the user connector which is mirrored in the software BIT field as the masterFail flag. This flag is thrown as a result of a number of instantly fatal conditions (known as a "hard" failure) or a persistent serious problem (known as a "soft" failure). Soft errors are those which must be triggered multiple times within a specified time window to be considered fatal. Soft errors are managed using a digital high-pass error counter with a trigger threshold.

The masterStatus flag is a configurable indication as determined by the user. This flag is asserted as a result of any asserted alert signals which the user has enabled.

The hierarchy of BIT and Status *fields* and signals is depicted here:

- *BITstatus Field*
  - masterFail
    - hardwareError
      - *hardwareBIT Field*
        - powerError
          - *hardwarePowerBIT Field*
            - inpPower
            - inpCurrent
            - inpVoltage
            - fiveVolt
            - threeVolt
            - twoVolt
            - twoFiveRef
            - sixVolt
            - grdRef
        - environmentalError
          - *hardwareEnvironmentalBIT Field*
            - pcbTemp
    - comError
      - *comBIT Field*
        - serialAError
          - *comSerialABIT Field*
            - transmitBufferOverflow

- receiveBufferOverflow
- framingError
- breakDetect
- parityError

- serialBError
  - *comSerialBBIT Field*
    - transmitBufferOverflow
    - receiveBufferOverflow
    - framingError
    - breakDetect
    - parityError

- softwareError
  - *softwareBIT Field*
    - algorithmError
      - *softwareAlgorithmBIT Field*
        - initialization
        - overRange
        - missedIntegrationStep
    - dataError
      - *softwareDataBIT Field*
        - calibrationCRCError
        - magAlignOutOfBounds

- masterStatus
  - hardwareStatus
    - *hardwareStatus Field*
      - unlocked1PPS (enabled by default on INS)
      - unlockedInternalGPS (enabled by default on INS)
      - noDGPS
      - unlockedEEPROM
  - comStatus
    - *comStatus Field*
      - noExternalGPS (enabled by default on VG and AHRS)
  - softwareStatus
    - *softwareStatus Field*

- algorithmInitialization (enabled by default)
- highGain (enabled by default)
- attitudeOnlyAlgorithm
- turnSwitch

- sensorStatus
  - *sensorStatus Field*
    - overRange (enabled by default)

9.2 Master BIT and Status (BITstatus) Field

The BITstatus field is the global indication of health and status of the DMUx81ZA Series product (See Table 33). The LSB contains BIT information and the MSB contains status information.

There are four intermediate signals that are used to determine when masterFail and the hardware BIT signal are asserted. These signals are controlled by various systems checks in software that are classified into three categories: hardware, communication, and software. Instantaneous soft failures in each of these four categories will trigger these intermediate signals, but will not trigger the masterFail until the persistency conditions are met.

There are four intermediate signals that are used to determine when the masterStatus flag is asserted: hardwareStatus, sensorStatus, comStatus, and softwareStatus. masterStatus is the logical OR of these intermediate signals. Each of these intermediate signals has a separate field with individual indication flags. Each of these indication flags can be enabled or disabled by the user. Any enabled indication flag will trigger the associated intermediate signal and masterStatus flag.

Table 33 DMUx81 BIT Status Field

BITstatus Field	*Bits*	*Meaning*	*Category*
masterFail	0	0 = normal, 1 = fatal error has occurred	BIT
HardwareError	1	0 = normal, 1= internal hardware error	BIT
comError	2	0 = normal, 1 = communication error	BIT
softwareError	3	0 = normal, 1 = internal software error	BIT
Reserved	4:7	N/A	
masterStatus	8	0 = nominal, 1 = hardware, sensor, com, or software alert	Status
hardwareStatus	9	0 = nominal, 1 = programmable alert	Status
comStatus	10	0 = nominal, 1 = programmable alert	Status
softwareStatus	11	0 = nominal, 1 = programmable alert	Status
sensorStatus	12	0 = nominal, 1 = programmable alert	Status
Reserved	13:15	N/A	

9.3 hardwareBIT Field

The hardwareBIT field contains flags that indicate various types of internal hardware errors (See Table 34). Each of these types has an associated message with low level error signals. The hardwareError flag in the BITstatus field is the bit-wise OR of this hardwareBIT field.

Table 34 DMUx81 Hardware BIT Field

hardwareBIT Field	*Bits*	*Meaning*	*Category*
powerError	0	0 = normal, 1 = error	Soft
environmentalError	1	0 = normal, 1 = error	Soft
reserved	2:15	N/A	

9.4 hardwarePowerBIT Field

The hardwarePowerBIT field contains flags that indicate low level power system errors (See Table 35). The powerError flag in the hardwareBIT field is the bit-wise OR of this hardwarePowerBIT field.

Table 35 DMUx81 Hardware Power BIT Field

hardwarePowerBIT Field	*Bits*	*Meaning*	*Category*
inpPower	0	0 = normal, 1 = out of bounds	Soft
inpCurrent	1	0 = normal, 1 = out of bounds	Soft
inpVoltage	2	0 = normal, 1 = out of bounds	Soft
fiveVolt	3	0 = normal, 1 = out of bounds	Soft
threeVolt	4	0 = normal, 1 = out of bounds	Soft
twoVolt	5	0 = normal, 1 = out of bounds	Soft
twoFiveRef	6	0 = normal, 1 = out of bounds	Soft
sixVolt	7	0 = normal, 1 = out of bounds	Soft
grdRef	8	0 = normal, 1 = out of bounds	Soft
Reserved	9:15	N/A	

9.5 hardwareEnvironmentalBIT Field

The hardwareEnvironmentalBIT field contains flags that indicate low level hardware environmental errors (See Table 36). The environmentalError flag in the hardwareBIT field is the bit-wise OR of this hardwareEnvironmentalBIT field.

Table 36 DMUx81 Hardware Environment BIT Field

hardwareEnvironmentalBIT Field	*Bits*	*Meaning*	*Category*
pcbTemp	0	0 = normal, 1 = out of bounds	Soft
Reserved	9:15	N/A	

9.6 comBIT Field

The comBIT field contains flags that indicate communication errors with external devices (See Table 37). Each external device has an associated message with low level error signals. The comError flag in the BITstatus field is the bit-wise OR of this comBIT field.

Table 37 DMUx81 COM BIT Field

comBIT Field	*Bits*	*Meaning*	*Category*
serialAError	0	0 = normal, 1 = error	Soft
serialBError	1	0 = normal, 1 = error	Soft
Reserved	2:15	N/A	

9.7 comSerialABIT Field

The comSerialABIT field (See Table 38) contains flags that indicate low level errors with external serial port A (the user serial port). The serialAError flag in the comBIT field is the bit-wise OR of this comSerialABIT field.

Table 38 DMUx81 Serial Port A BIT Field

comSerialABIT Field	*Bits*	*Meaning*	*Category*
transmitBufferOverflow	0	0 = normal, 1 = overflow	Soft
receiveBufferOverflow	1	0 = normal, 1 = overflow	Soft
framingError	2	0 = normal, 1 = error	Soft
breakDetect	3	0 = normal, 1 = error	Soft
parityError	4	0 = normal, 1 = error	Soft
Reserved	5:15	N/A	

9.8 comSerialBBIT Field

The comSerialBBIT field (See Table 39) contains flags that indicate low level errors with external serial port B (the aiding serial port). The serialBError flag in the comBIT field is the bit-wise OR of this comSerialBBIT field.

Table 39 DMUx81 Serial Port B BIT Field

comSerialBBIT Field	*Bits*	*Meaning*	*Category*
transmitBufferOverflow	0	0 = normal, 1 = overflow	Soft
receiveBufferOverflow	1	0 = normal, 1 = overflow	Soft
framingError	2	0 = normal, 1 = error	Soft
breakDetect	3	0 = normal, 1 = error	Soft
parityError	4	0 = normal, 1 = error	Soft
Reserved	5:15	N/A	

9.9 softwareBIT Field

The softwareBIT field contains flags that indicate various types of software errors (See Table 40). Each type has an associated message with low level error signals. The softwareError flag in the BITstatus field is the bit-wise OR of this softwareBIT field.

Table 40 DMUx81 Softrware BIT Field

softwareBIT Field	*Bits*	*Meaning*	*Category*
algorithmError	0	0 = normal, 1 = error	Soft
dataError	1	0 = normal, 1 = error	Soft
Reserved	2:15	N/A	

9.10 softwareAlgorithmBIT Field

The softwareAlgorithmBIT field contains flags that indicate low level software algorithm errors (See Table 41). The algorithmError flag in the softwareBIT field is the bit-wise OR of this softwareAlgorithmBIT field.

Table 41 DMUx81 Software Algorithm BIT Field

SoftwareAlgorithmBIT Field	*Bits*	*Meaning*	*Category*
initialization	0	0 = normal, 1 = error during algorithm initialization	Hard
overRange	1	0 = normal, 1 = fatal sensor over-range	Hard
missedNavigationStep	2	0 = normal, 1 = fatal hard deadline missed for navigation	Hard
Reserved	3:15	N/A	

9.11 softwareDataBIT Field

The softwareDataBIT field contains flags that indicate low level software data errors (See Table 42). The dataError flag in the softwareBIT field is the bit-wise OR of this softwareDataBIT field.

Table 42 DMUx81 Software Data BIT Field

SoftwareDataBIT Field	*Bits*	*Meaning*	*Category*
calibrationCRCError	0	0 = normal, 1 = incorrect CRC on calibration EEPROM data or data has been compromised by a WE command.	Hard
magAlignOutOfBounds	1	0 = normal, 1 = hard and soft iron parameters are out of bounds	Hard
Reserved	2:15	N/A	

9.12 hardwareStatus Field

The hardwareStatus field contains flags that indicate various internal hardware conditions and alerts that are not errors or problems (See Table 43). The hardwareStatus flag in the BITstatus field is the bit-wise OR of the logical AND of the hardwareStatus field and the hardwareStatusEnable field. The hardwareStatusEnable field is a bit mask that allows the user to select items of interest that will logically flow up to the masterStatus flag.

Table 43 DMUx81 Hardware Status BIT Field

hardwareStatus Field	*Bits*	*Meaning*
unlocked1PPS	0	0 = not asserted, 1 = asserted
unlockedInternalGPS	1	0 = not asserted, 1 = asserted
noDGPS	2	0 = DGPS lock, 1 = no DGPS
unlockedEEPROM	3	0=locked, WE disabled, 1=unlocked, WE enabled
Reserved	4:15	N/A

9.13 comStatus Field

The comStatus field contains flags that indicate various external communication conditions and alerts that are not errors or problems (See Table 44). The comStatus flag in the BITstatus field is the bit-wise OR of the logical AND of the comStatus field and the comStatusEnable field. The comStatusEnable field is a bit mask that allows the user to select items of interest that will logically flow up to the masterStatus flag.

Table 44 DMUx81 COM Status BIT Field

comStatus Field	*Bits*	*Meaning*
noExternalGPS	0	0 = external GPS data is being received, 1 = no external GPS data is available
Reserved	1:15	N/A

9.14 softwareStatus Field

The softwareStatus field contains flags that indicate various software conditions and alerts that are not errors or problems (See Table 45). The softwareStatus flag in the BITstatus field is the bit-wise OR of the logical AND of the softwareStatus field and the softwareStatusEnable field. The softwareStatusEnable field is a bit mask that allows the user to select items of interest that will logically flow up to the masterStatus flag.

Table 45 DMUx81 Software Status Field

softwareStatus Field	*Bits*	*Meaning*
algorithmInit	0	0 = normal, 1 = the algorithm is in initialization mode
highGain	1	0 = low gain mode, 1 high gain mode
attitudeOnlyAlgorithm	2	0 = navigation state tracking, 1 = attitude only state tracking
turnSwitch	3	0 = off, 1 = yaw rate greater than turnSwitch threshold
Reserved	4:15	N/A

9.15 sensorStatus Field

The sensorStatus field contains flags that indicate various internal sensor conditions and alerts that are not errors or problems (See Table 46). The sensorStatus flag in the BITstatus field is the bit-wise OR of the logical AND of the sensorStatus field and the sensorStatusEnable field. The sensorStatusEnable field is a bit mask that allows the user to select items of interest that will logically flow up to the masterStatus flag.

Table 46 DMUx81 Sensor Status Field

sensorStatus Field	*Bits*	*Meaning*
overRange	0	0 = not asserted, 1 = asserted
Reserved	1:15	N/A

9.16 Configuring the Master Status

The masterStatus byte and its associated programmable alerts are configured using the Read Field and Write Field command as described in Section 8, Advanced Commands. Table 47 shows the definition of the bit mask for configuring the status signals.

Table 47 DMUx81 Master Status Byte Configuration Fields

configuration fields	*field ID*	*Valid Values*	*Description*
hardwareStatusEnable	0x0010	Any	Bit mask of enabled hardware status signals
comStatusEnable	0x0011	Any	Bit mask of enabled communication status signals
softwareStatusEnable	0x0012	Any	Bit mask of enabled software status signals
sensorStatusEnable	0x0013	Any	Bit mask of enabled sensor status signals

9.16.1 hardwareStatusEnable Field

This field is a bit mask of the hardwareStatus field (see BIT and status definitions). This field allows the user to determine which low level hardwareStatus field signals will flag the hardwareStatus and masterStatus flags in the BITstatus field. Any asserted bits in this field imply that the corresponding hardwareStatus field signal, if asserted, will cause the hardwareStatus and masterStatus flags to be asserted in the BITstatus field.

9.16.2 comStatusEnable Field

This field is a bit mask of the comStatus field (see BIT and status definitions). This field allows the user to determine which low level comStatus field signals will flag the comStatus and masterStatus flags in the BITstatus field. Any asserted bits in this field imply that the corresponding comStatus field signal, if asserted, will cause the comStatus and masterStatus flags to be asserted in the BITstatus field.

9.16.3 softwareStatusEnable Field

This field is a bit mask of the softwareStatus field (see BIT and status definitions). This field allows the user to determine which low level softwareStatus field signals will flag the softwareStatus and masterStatus flags in the BITstatus field. Any asserted bits in this field imply that the corresponding softwareStatus field signal, if asserted, will cause the softwareStatus and masterStatus flags to be asserted in the BITstatus field.

9.16.4 sensorStatusEnable Field

This field is a bit mask of the sensorStatus field (see BIT and status definitions). This field allows the user to determine which low level sensorStatus field signals will flag the sensorStatus and masterStatus flags in the BITstatus field. Any asserted bits in this field imply that the

corresponding sensorStatus field signal, if asserted, will cause the sensorStatus and masterStatus flags to be asserted in the BITstatus field.

10 DMUx81ZA BOOTLOADER

10.1 Bootloader Initialization

A user can initiate bootloader at any time by sending 'JI' 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 IMU381ZA will enter a recovery window of 100ms prior to application start. During this 100mS 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 toggling power or toggling nRst (pull low and release) to restart the system.

10.2 Firmware Update Commands

The commands detailed in Sections 10.2.1 and 10.2.2 is used for upgrading a new firmware version.

10.2.1 UART Interface

Firmware upgrade is performed by a Write APP command through UART port, through Windows GUI, NAV-View, or a python program. See Appendix A and F.

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

10.2.1.1 Jump to BootLoader Command

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

The command allows system to enter bootloader mode.

10.2.1.2 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.

10.2.1.3 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.

10.2.2 SPI Interface

TBD

11 Warranty and Support Information

11.1 Customer Service

As an 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

11.2 Contact Directory

United States: Email: techsupport@aceinna.com

Non-U.S.: Refer to website www.aceinna.com

11.3 Return Procedure

11.3.1 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?

11.3.2 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.

11.3.3 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.

11.3.4 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.

11.4 Warranty

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

Appendix A: Installation and Operation of NAV-VIEW

NAV-VIEW has been designed to allow users to control all aspects of the DMUx81ZA Series operation including data recording, configuration and data transfer. For the first time, you will be able to control the orientation of the unit, sampling rate, packet type, hard iron calibration and filter settings through NAV-VIEW. For proper use with the DMUx81ZA family version 3.5.2 or higher of NAV-VIEW is required.

NAV-VIEW Computer Requirements

The following are minimum requirements for the installation of the NAV-VIEW Software:

- CPU: Pentium-class (1.5GHz minimum)
- RAM Memory: 500MB minimum, 1GB+ recommended
- Hard Drive Free Memory: 20MB
- Operating System: Windows 7 and 10
- Properly installed Microsoft .NET 2.0 or higher

Install NAV-VIEW

To install NAV-VIEW onto your computer:

1. Insert the CD "Inertial Systems Product Support" (Part No. 8160-0063) in the CD-ROM drive.
2. Locate the "NAV-VIEW" folder. Double click on the "setup.exe" file.
3. Follow the setup wizard instructions. You will install NAV-VIEW and .NET 2.0.

Connections

DMUx81ZA is shipped with a 6-pin TTL-to-232R cable used for connection between unit and a PC's USB port.

1. Hook up this cable between UART port on DMUx81ZA's demo board and PC's USB port.
2. The input voltage is 3.3VDC and maximum current is draw of 350 mA.
3. Wait around 60 seconds during the initialization of DMUx81ZA.

⚠ WARNING

Do not reverse the power leads! Reversing the power leads to the DMUx81ZA Series can damage the unit; although there is reverse power protection, ACEINNA is not responsible for resulting damage to the unit should the reverse voltage protection electronics fail.

Setting up NAV-VIEW

With the DMUx81ZA Series product powered up and connected to your PC serial port, open the NAV-VIEW software application.

1. NAV-VIEW should automatically detect the DMUx81ZA Series product and display the serial number and firmware version if it is connected.
2. If NAV-VIEW does not connect, check that you have the correct COM port selected. You will find this under the "Setup" menu. Select the appropriate COM port and allow the unit to automatically match the baud rate by leaving the "Auto: match baud rate" selection marked.
3. If the status indicator at the bottom is green and states, Unit Connected, you're ready to go. If the status indicator doesn't say connected and is red, check the connections between the DMUx81ZA Series product and the computer, check the power supply, and verify that the COM port is not occupied by another device.
4. Under the "View" menu you have several choices of data presentation. Graph display is the default setting and will provide a real time graph of all the DMUx81ZA Series data. The remaining choices will be discussed in the following pages.

Data Recording

NAV-VIEW allows the user to log data to a text file (.txt) using the simple interface at the top of the screen. Customers can now tailor the type of data, rate of logging and can even establish predetermined recording lengths.

To begin logging data follow the steps below (See Figure 20):

1. Locate the  icon at the top of the page or select "Log to File" from the "File" drop down menu.
2. The following menu will appear.

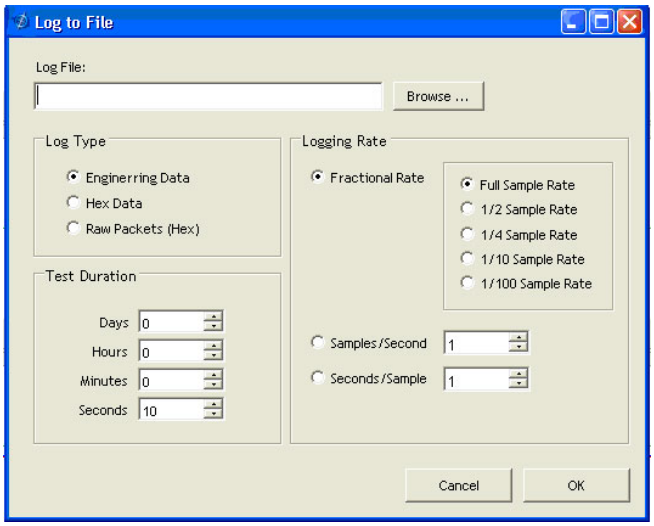


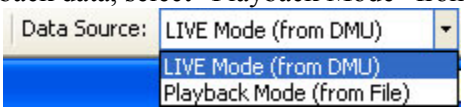
Figure 20. Log to File Dialog Screen

3. Select the "Browse" box to enter the file name and location that you wish to save your data to.
4. Select the type of data you wish to record. "Engineering Data" records the converted values provided from the system in engineering units, "Hex Data" provides the raw hex values separated into columns displaying the value, and the "Raw Packets" will simply record the raw hex strings as they are sent from the unit.

5. Users can also select a predetermined "Test Duration" from the menu. Using the arrows, simply select the duration of your data recording.
6. Logging Rate can also be adjusted using the features on the right side of the menu.
7. Once you have completed the customization of your data recording, you will be returned to the main screen where you can start the recording process using the button at the top of the page or select "Start Logging" from the "File" menu. Stopping the data recording can be accomplished using the button and the recording can also be paused using the button.

Data Playback

In addition to data recording, NAV-VIEW allows the user to replay saved data that has been stored in a log file.

1. To playback data, select "Playback Mode" from the "Data Source" drop down menu at the top. 

2. Selecting Playback mode will open a text prompt which will allow users to specify the location of the file they wish to play back. All three file formats are supported (Engineering, Hex, and Raw) for playback. In addition, each time recording is stopped/started a new section is created. These sections can be individually played back by using the drop down menu and associated VCR controls.
3. Once the file is selected, users can utilize the VCR style controls at the top of the page to start, stop, and pause the playback of the data.
4. NAV-VIEW also provides users with the ability to alter the start time for data playback. Using the  slide bar at the top of the page users can adjust the starting time.

Raw Data Console

NAV-VIEW offers some unique debugging tools that may assist programmers in the development process. One such tool is the Raw Data Console. From the "View" drop down menu, simply select the "Raw Data Console". This console provides users with a simple display of the packets that have been transmitted to the unit (Tx) and the messages received (Rx). An example is provided in Figure 21.

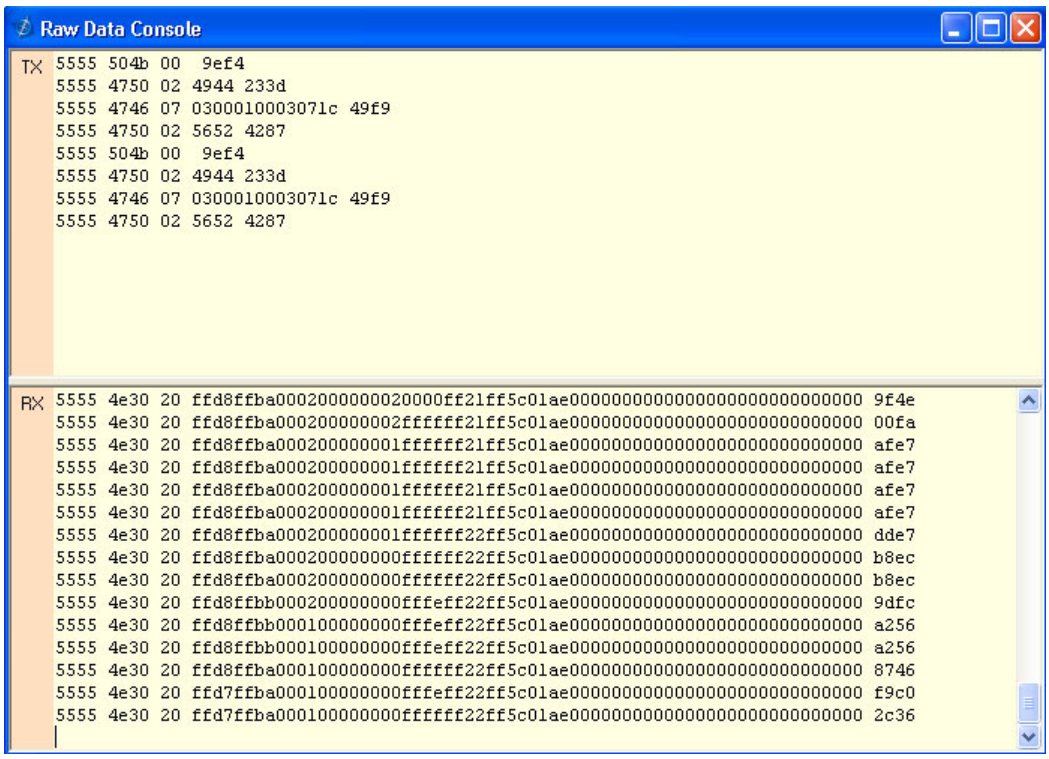


Figure 21 Raw Data Console

Horizon and Compass View

If the DMUx81ZA Series product you have connected is capable of providing heading and angle information (see Table 2), NAV-VIEW can provide a compass and a simulated artificial horizon view. To activate these views, simply select "Horizon View" and/or "Compass View" from the "View" drop down menu at the top of the page (See Figure 22).

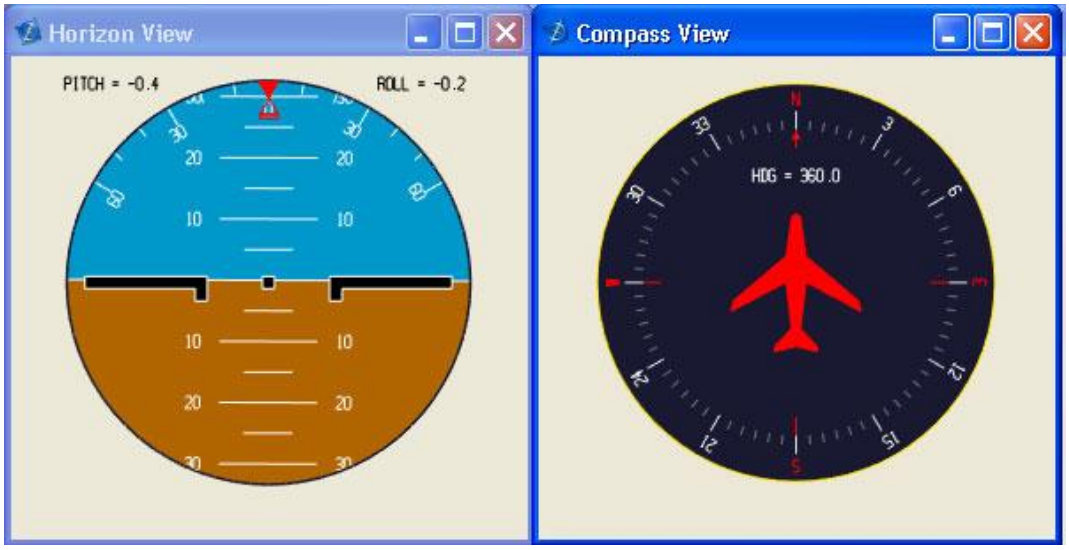


Figure 22 Horizon and Compass View

Packet Statistics View

Packet statistics can be obtained from the "View" menu by selecting the "Packet Statistics" option (See Figure 23). This view simply provides the user with a short list of vital statistics (including Packet Rate, CRC Failures, and overall Elapsed Time) that are calculated over a one second window. This tool should be used to gather information regarding the overall health of the user configuration. Incorrectly configured communication settings can result in a large number of CRC Failures and poor data transfer.

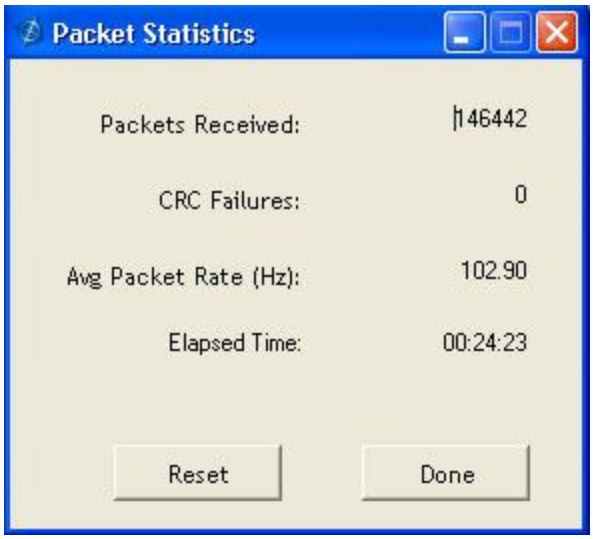


Figure 23 Packet Statistics

Unit Configuration

The Unit Configuration window (See Figure 24) gives the user the ability to view and alter the system settings. This window is accessed through the "Unit Configuration" menu item under the configuration menu. Under the "General" tab, users have the ability to verify the current configuration by selecting the "Get All Values" button. This button simply provides users with the currently set configuration of the unit and displays the values in the left column of boxes.

There are three tabs within the "Unit Configuration" menu; General, Advanced and BIT Configuration. The General tab displays some of the most commonly used settings. The Advanced and BIT Configuration menus provide users with more detailed setting information that they can tailor to meet their specific needs.

To alter a setting, simply select the check box on the left of the value that you wish to modify and then select the value using the drop down menu on the right side. Once you have selected the appropriate value, these settings can be set temporarily or permanently (a software reset or power cycle is required for the changes to take affect) by selecting from the choices at the bottom of the dialog box. Once the settings have been altered a "Success" box will appear at the bottom of the page.

IMPORTANT

Caution must be taken to ensure that the settings selected are compatible with the system that is being configured. In most cases a "FAIL" message will appear if incompatible selections are made by the user, however it is the users responsibility to ensure proper configuration of the unit.

IMPORTANT

Unit orientation selections must conform to the right hand coordinate system as noted in Section 3.1 of this user manual. Selecting orientations that do not conform to these criteria are not allowed.

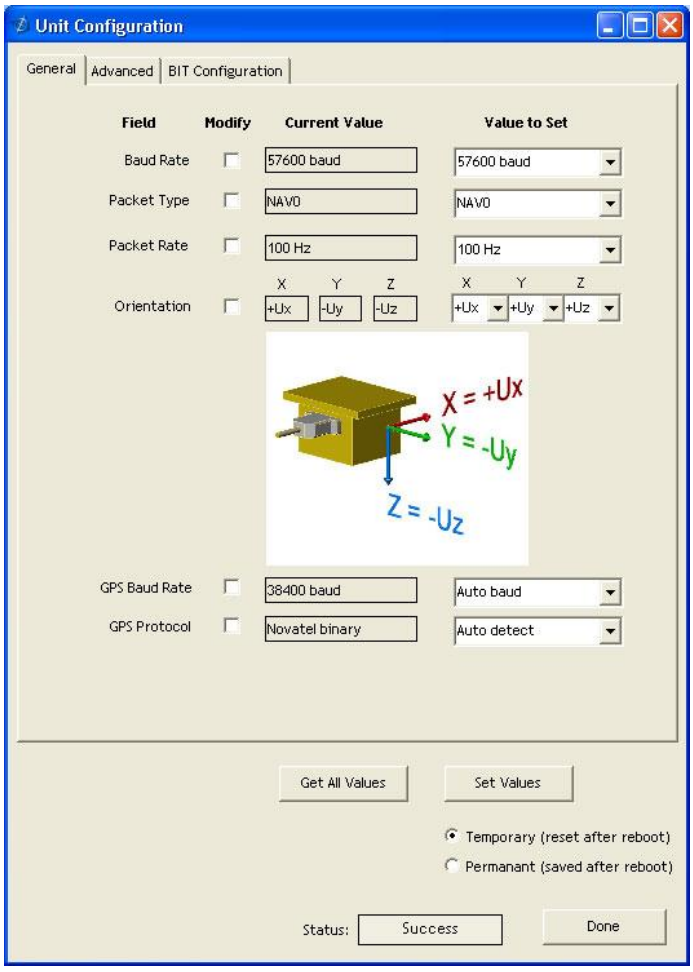


Figure 24 Unit Configuration

Advanced Configuration

Users who wish to access some of the more advanced features of NAV-VIEW and the DMUx81ZA Series products can do so by selecting the "Advanced" tab at the top of the "Unit Configuration" window.

WARNING

Users are strongly encouraged to read and thoroughly understand the consequences of altering the settings in the "Advanced" tab before making changes to the unit configuration. These settings are discussed in detail in Chapter 4 below.

Behavior switches are identified at the top of the page with marked boxes. A blue box will appear if a switch has been enabled similar to Figure 25 below. The values can be set in the same manner as noted in the previous section. To set a value, users select the appropriate "Modify" checkbox on the left side of the menu and select or enable the appropriate value they wish to set. At the bottom of the page, users have the option of temporarily or permanently setting values. When all selections have been finalized, simply press the "Set Values" button to change the selected settings.

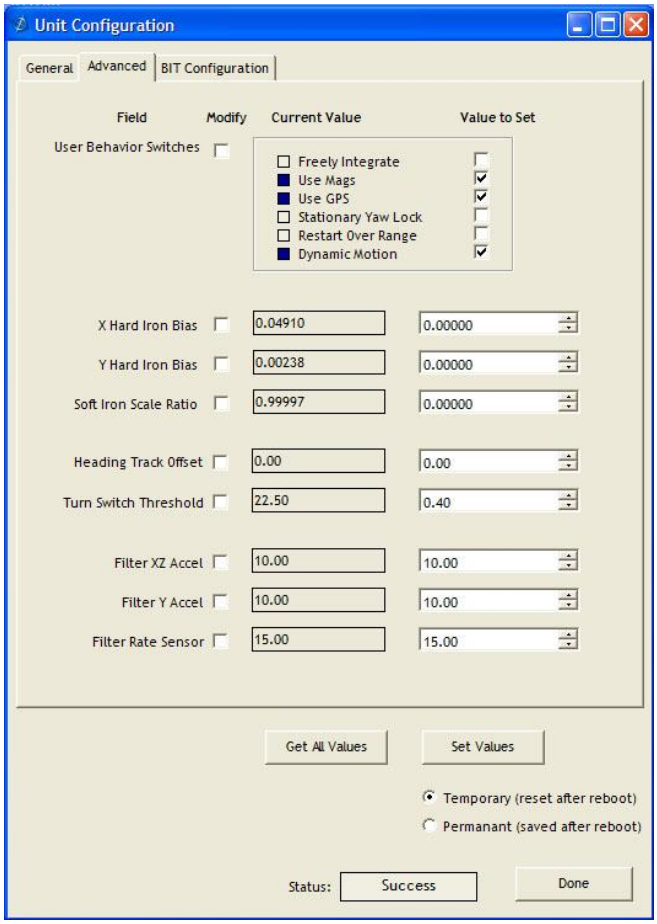


Figure 25 Advanced Settings

Bit Configuration

The third and final tab of the unit configuration window is "Bit Configuration" (See Figure 26). This tab allows the users to alter the logic of individual status flags that affect the masterStatus flag in the master BIT status field (available in most output packets). By enabling individual status flags users can determine which flags are logically OR'ed to generate the masterStatus flag. This gives the user the flexibility to listen to certain indications that affect their specific application. The masterFail and all error flags are not configurable. These flags represent serious errors and should never be ignored.

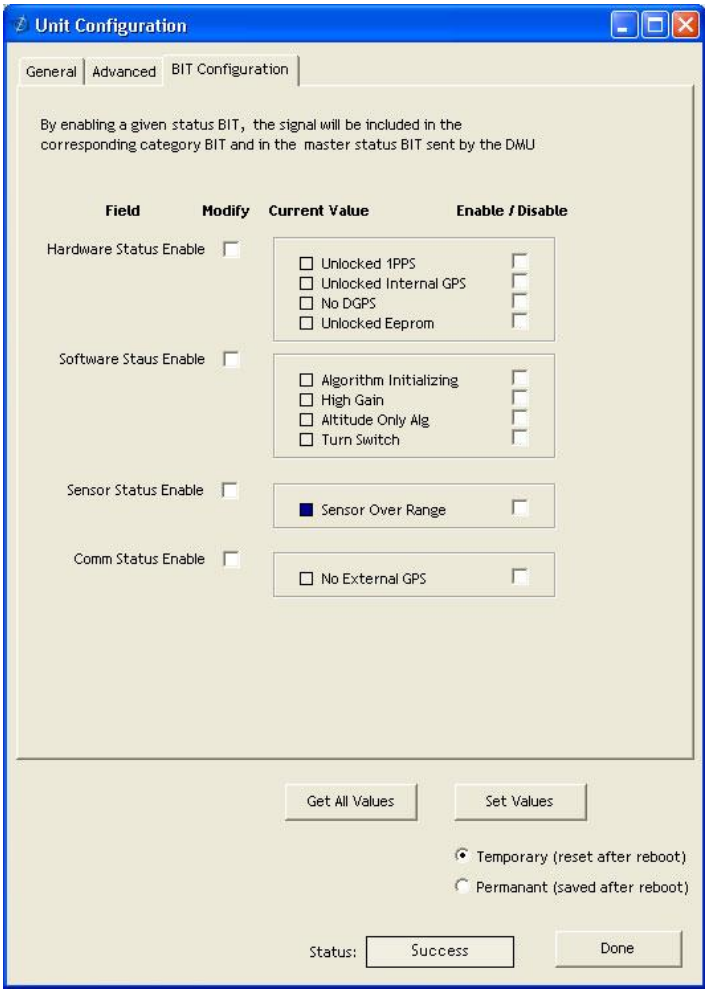


Figure 3.

Figure 26 BIT Configuration

Mag Alignment Procedure

IMPORTANT

The following section only applies to DMUx81ZA Series products with magnetometers (AHRS and INSx81ZA). If your particular model does not utilize magnetometers for heading or performance you can disregard the following section.

Hard Iron/Soft Iron Overview

The AHRS and INSx81ZA products use magnetic sensors to compute heading. Ideally, the magnetic sensors would be measuring only earth's magnetic field to compute the heading angle. In the real world, however, residual magnetism in your system adds to the total magnetic field measured. This residual magnetism (called hard iron and soft iron) will create errors in the heading measurement if it is not accounted for. In addition, magnetic material can change the direction of the magnetic field as a function of the input magnetic field. This dependence of the local magnetic field on input direction is called the soft iron effect.

The AHRS and INSx81ZA products can actually measure the constant magnetic field that is associated with your system and correct for it. The AHRS and INSx81ZA products can also make a correction for some soft iron effects. The process of measuring these non-ideal effects and correcting for them is called the "Mag Alignment Procedure". Performing a "Mag Alignment Procedure" will help correct for magnetic fields that are fixed with respect to the DMUx81ZA Series product. It cannot correct for time varying fields, or fields created by ferrous material that moves with respect to the DMUx81ZA Series product.

The AHRS and INSx81ZA products account for the extra magnetic field by making a series of measurements, and using these measurements to model the hard iron and soft iron environment in your system using a two-dimensional algorithm. The AHRS and INSx81ZA products will calculate the hard iron magnetic fields and soft iron corrections and store these as calibration constants in the EEPROM.

The "Mag Alignment Procedure" should always be performed with the AHRS or INSx81ZA product installed in the user system. If you perform the calibration process with the DMUx81ZA Series product by itself, you will not be correcting for the magnetism in the user system. If you then install the DMUx81ZA Series product in the system (i.e. a vehicle), and the vehicle is magnetic, you will still see errors arising from the magnetism of the vehicle.

Mag Alignment Procedure Using NAV-VIEW

The Mag Alignment Procedure using NAV-VIEW can be performed using the following steps below:

1. Select "Mag Alignment" from the "Configuration" drop down menu at the top.
2. If you can complete your 360 degree turn within 120 seconds, select the "Auto-Terminate" box.
3. Select the "Start" button to begin the "MagAlign" Procedure and follow the instructions at the bottom of the screen as shown in Figure 27 below.

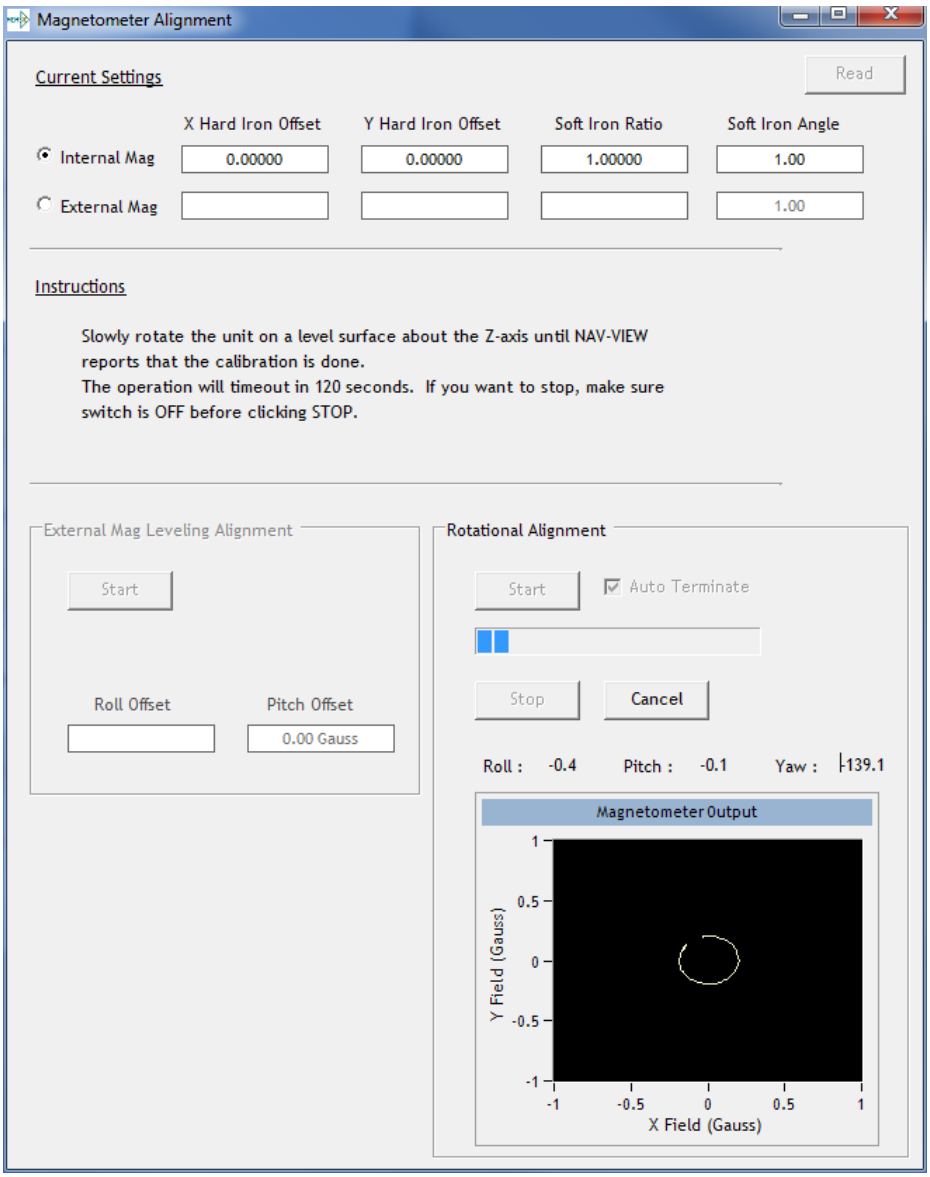


Figure 27 Mag Alignment

4. Rotate the AHRS or INSx81ZA product through x81 degrees of rotation or until you receive a message to stop.
5. Once you have completed your rotation, you will be given data concerning the calibration accuracy. The X and Y offset values indicate how far the magnetic field has been offset due to hard iron affects from components surrounding the unit. In addition, you will see a soft iron ratio indicating the effect of soft iron on the AHRS of INSx81ZA product.
6. Save this data to the AHRS or INSx81ZA product by selecting the "Apply" button (See Figure 28).

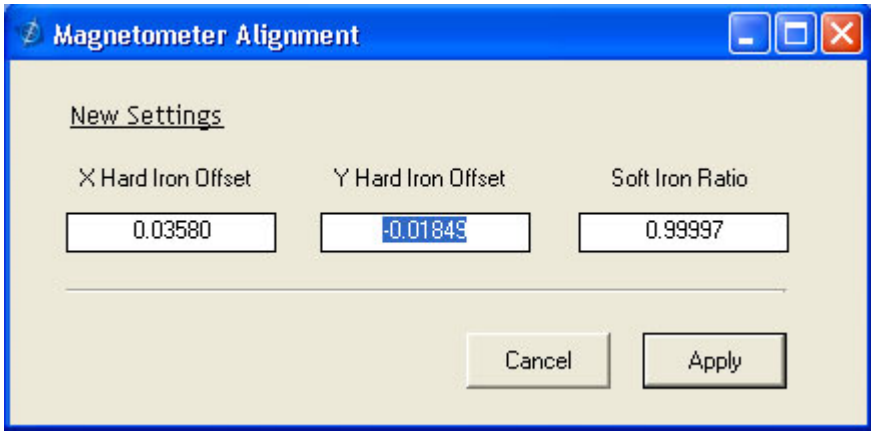


Figure 28 Magnetometer Alignment

7. Upon completion of the "Mag Alignment Procedure", the heading accuracy should be verified with all third party systems active using a known reference such as a compass rose, GPS track or a calibrated compass. Heading inaccuracies greater than the values specified on the data sheet or fluctuating heading performance may be an indication of magnetic field disturbances near the unit.

IMPORTANT

An acceptable calibration will provide X and Y Hard Iron Offset Values of < 2.5 and a Soft Iron Ratio >0.95. If this procedure generates any values larger than stated above, the system will assert the softwareError→dataError→magAlignOutOfBounds error flag. See section 9 for details on error flag handling. Note that the current release of the software does not have this functionality. Future releases of software will restore this functionality. The magnetometer ranges is +/- 4 gauss, thus 2.5 gauss is the recommended maximum hardiron that should be tolerated for the installation and still provide ample resolution and headroom to properly determine the earth's magnetic field (strength < 0.5 gauss). If the hard iron estimates are larger than 2.5 gauss, then a different installation location should be investigated. The hard iron and softiron data, while used internally to achieve a heading reference, do not get applied to the magnetometer data output in message A1 (see Section 7.4.3 and Section 8.3).

Read Unit Configuration

NAV-VIEW allows users to view the current settings and calibration data for a given DMUx81ZA Series unit by accessing the "Read Configuration" selection from the "Configuration" drop down menu (See Figure 29). From this dialog, users can print a copy of the unit's current configuration and calibration values with the click of a button. Simply select the "Read" button at the top of the dialog box and upon completion select the "Print" or "Print Preview" buttons to print a copy to your local network printer. This information can be helpful when storing hard copies of unit configuration, replicating the original data sheet and for troubleshooting if you need to contact ACEINNA's Support Staff.

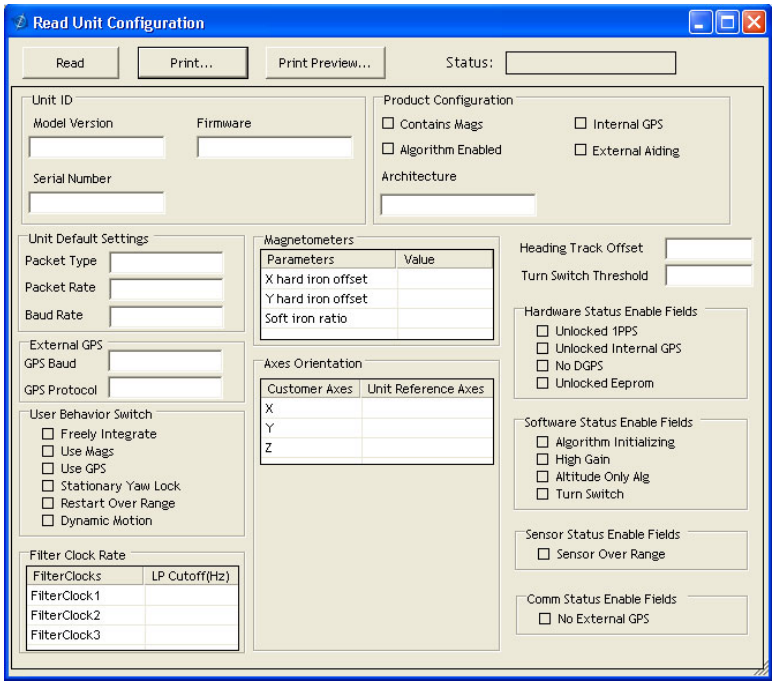
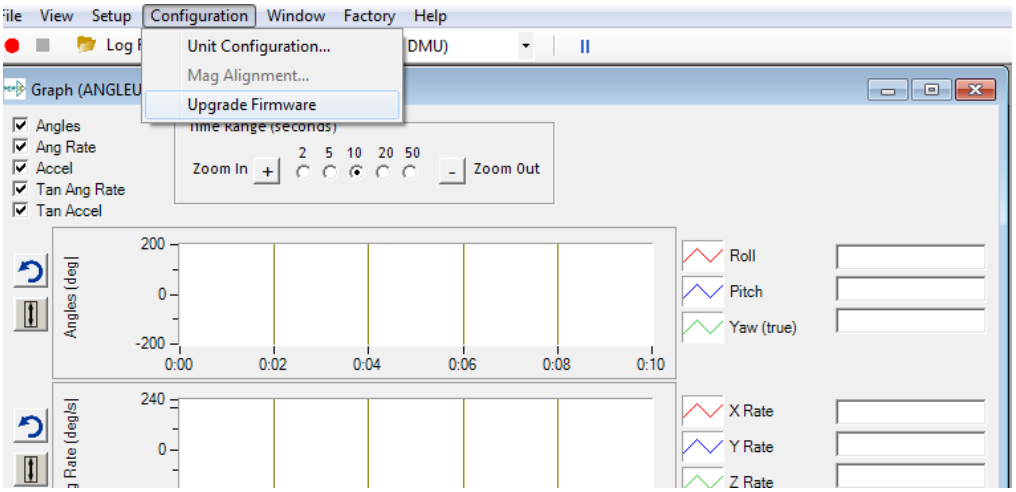


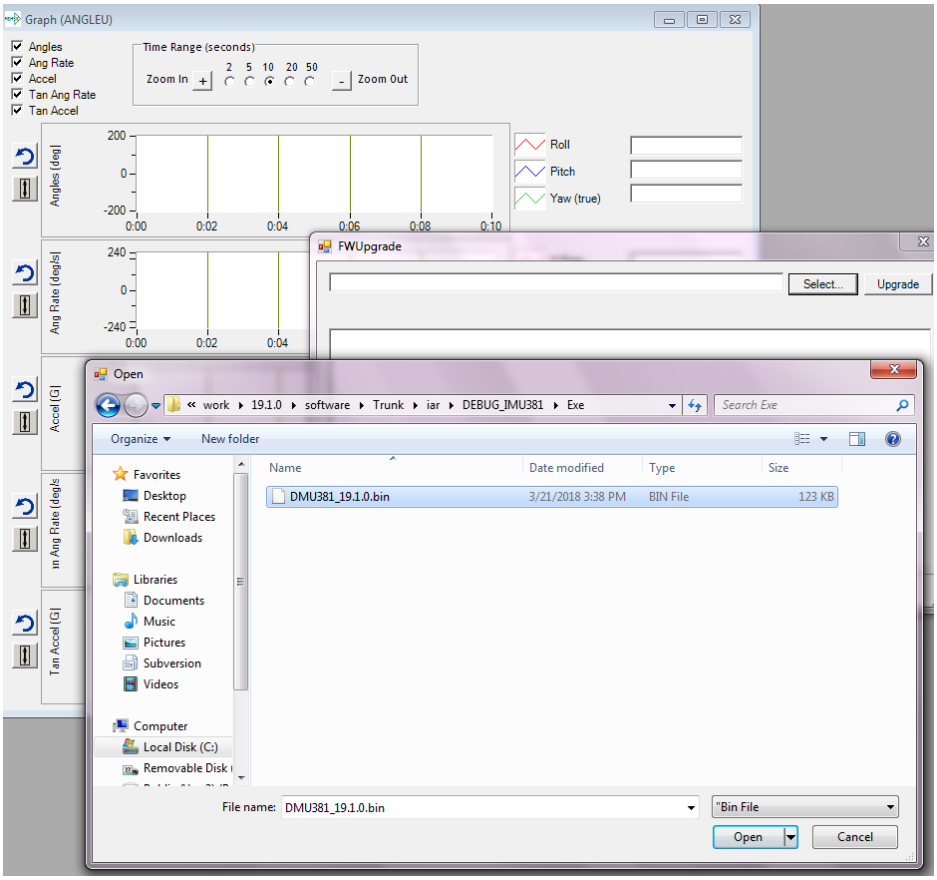
Figure 29 Read Configuration

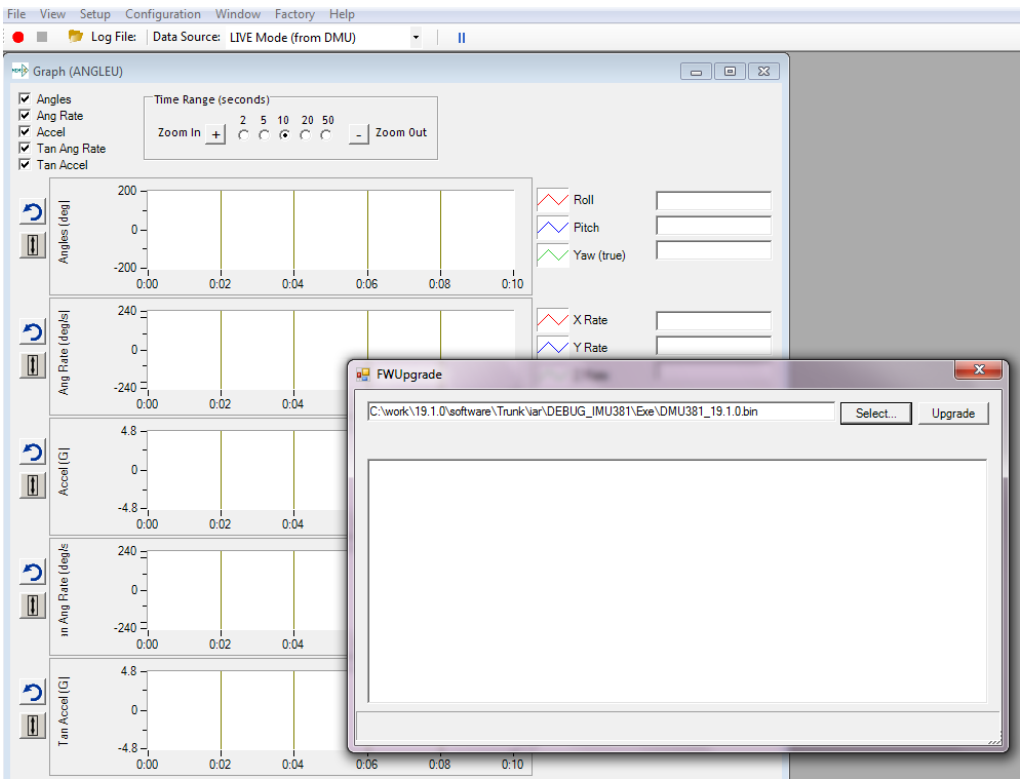
Firmware upgrade

Step 1, select Firmware upgrade from configuration menu.

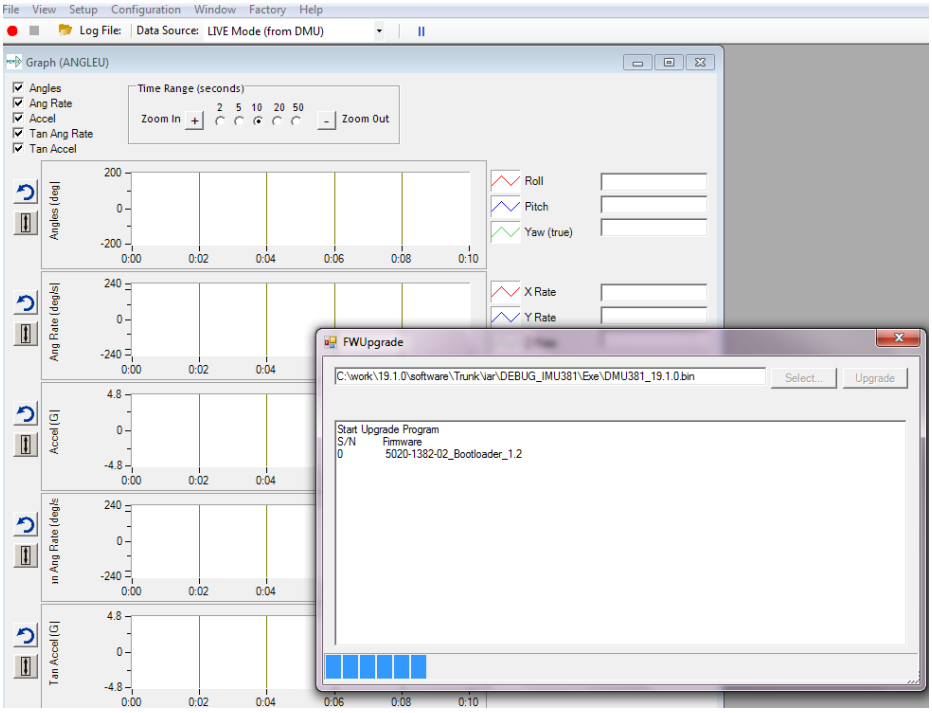


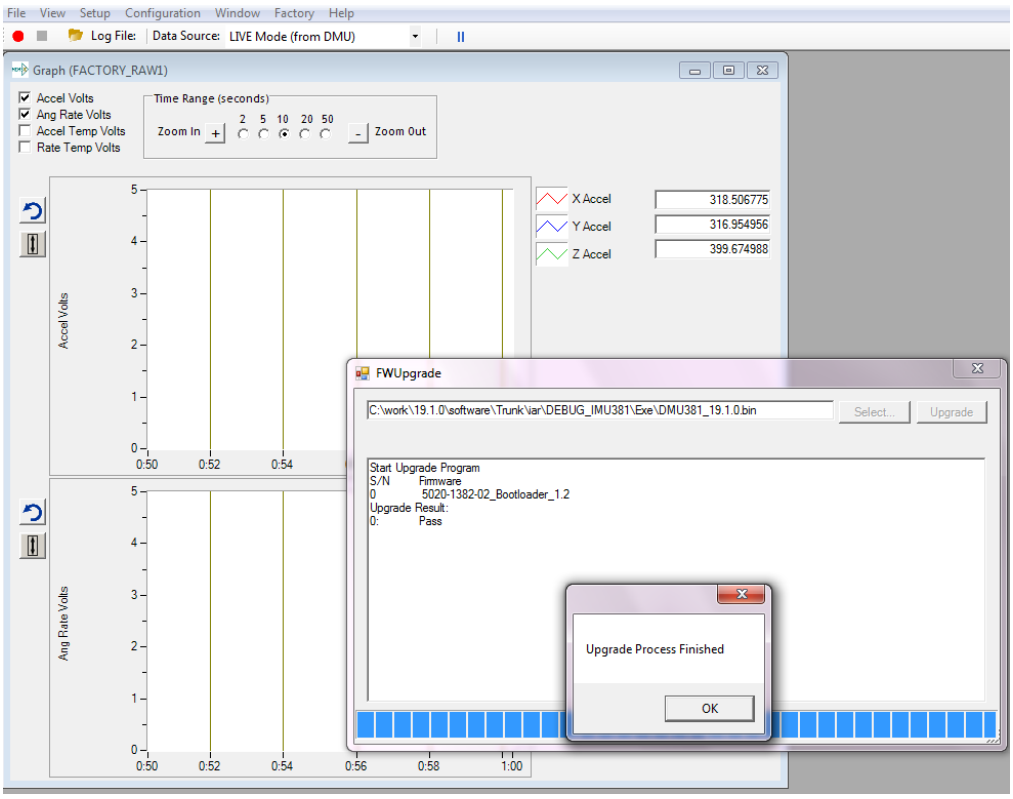
Step 2, On pop-up window, select a new version binary file by clicking SELECT button, then click Upgrade button.


Graph (ANGLEU)
Angles
Ang Rate
Accel
Tan Ang Rate
Tan Accel
Time Range (seconds)
Zoom In
Zoom Out
2 5 10 20 50
Angles [deg]
Ang Rate [deg/s]
Accel [G]
in Ang Rate [deg/s
Tan Accel [G]
Roll
Pitch
Yaw (true)
FWUpgrade
Select...
Upgrade
Open
work ▸ 19.1.0 ▸ software ▸ Trunk ▸ iar ▸ DEBUG_IMU381 ▸ Exe
Search Exe
Organize
New folder
Favorites
Desktop
Recent Places
Downloads
Libraries
Documents
Music
Pictures
Subversion
Videos
Computer
Local Disk (C:)
Name
Date modified
Type
Size
DMU381_19.1.0.bin
3/21/2018 3:38 PM
BIN File
123 KB
File name:
*Bin File
Open
Cancel


File View Setup Configuration Window Factory Help
Log File:
Data Source:
LIVE Mode (from DMU)
Graph (ANGLEU)
Angles
Ang Rate
Accel
Tan Ang Rate
Tan Accel
Time Range (seconds)
Zoom In
Zoom Out
2 5 10 20 50
Angles [deg]
Ang Rate [deg/s]
Accel [G]
in Ang Rate [deg/s
Tan Accel [G]
Roll
Pitch
Yaw (true)
X Rate
Y Rate
FWUpgrade
C:\work\19.1.0\software\Trunk\iar\DEBUG_IMU381\Exe\DMU381_19.1.0.bin
Select...
Upgrade

Step 3, wait for the process ongoing until a successful or failure message pops up.





Appendix B: NMEA Message Format

The GPS receiver outputs data in NMEA-0183 format at 9600 Baud, 8 bits, no parity bit, and 1 stop bit. The GGA and RMC message packet formats are explained in this section.

GGA - GPS fix data

Time and position, together with GPS fixing related data (number of satellites in use, and the resulting HDOP, age of differential data if in use, etc.).

```
$GPGGA,hhmmss.ss,Latitude,N,Longitude,E,FS,NoSV,HDOP,msl,m,Altref
,m,DiffAge,DiffStation*cs<CR><LF>
```

Name	ASCII String		Description
	Format	Example	
$GPGGA	string	$GPGGA	Message ID: GGA protocol header
hhmmss.ss	hhmmss.sss	092725.00	UTC Time: Current time
Latitude	dddmm.mmmm	4717.11399	Latitude: Degrees + minutes
N	character	N	N/S Indicator: N=north or S=south
Longitude	dddmm.mmmm	00833.91590	Longitude: Degrees + minutes
E	character	E	E/W indicator: E=east or W=west
FS	1 digit	1	Position Fix Indicator (See Table below)
NoSV	numeric	8	Satellites Used: Range 0 to 12
HDOP	numeric	1.01	HDOP: Horizontal Dilution of Precision
msl	numeric	499.6	MSL Altitude (m)
m	character	M	Units: Meters (fixed field)
Altref	blank	48.0	Geoid Separation (m)
m	blank	M	Units: Meters (fixed field)
DiffAge	numeric		Age of Differential Corrections (sec): Blank (Null) fields when DGPS is not used
DiffStation	numeric	0	Diff. Reference Station ID

cs	hexadecimal	*5B	Checksum
<CR> <LF>			End of message

Fix Status	Description
0	No fix / Invalid
1	Standard GPS (2D/3D)
2	Differential GPS
6	Estimated (DR) Fix

Appendix C: Sample Packet-Parser Code

Overview

This appendix includes sample code written in ANSI C for parsing packets from data sent by the DMUx81ZA Series Inertial Systems. This code can be used by a user application reading data directly from the DMUx81ZA Series product, or perhaps from a log file. Check at https://github.com/Aceinna for other reference code.

The sample code contains the actual parser, but also several support functions for CRC calculation and circular queue access.:

- **process_xbow_packet** – for parsing out packets from a queue. Returns these fields in structure XBOW_PACKET (see below). Checks for CRC errors
- **calcCRC** – for calculating CRC on packets.
- **Initialize** - initialize the queue
- **AddQueue** - add item in front of queue
- **DeleteQueue** - return an item from the queue
- **peekWord** - for retrieving 2-bytes from the queue, without popping
- **peekByte** – for retrieving a byte from the queue without popping
- **Pop** - discard item(s) from queue
- **Size** – returns number of items in queue
- **Empty** – return 1 if queue is empty, 0 if not
- **Full** - return 1 if full, 0 if not full

The parser will parse the queue looking for packets. Once a packet is found and the CRC checks out, the packet's fields are placed in the XBOW_PACKET structure. The parser will then return to the caller. When no packets are found the parser will simply return to the caller with return value 0.

The XBOW_PACKET stucture is defined as follows:

```
typedef struct xbow_packet
{
    unsigned short packet_type;
    char           length;
    unsigned short crc;
    char           data[256];
} XBOW_PACKET;
```

Typically, the parser would be called within a loop in a separate process, or in some time triggered environment, reading the queue looking for packets. A separate process might add data to this queue when it arrives. It is up to the user to ensure circular-queue integrity by using some sort of mutual exclusion mechanism withing the queue access funtions.

Code listing

```
#include <stdio.h>
/* buffer size */
#define MAXQUEUE 500
/*
 * circular queue
 */
typedef struct queue_tag
{
    int count;
    int front;
    int rear;
    char entry[MAXQUEUE];
} QUEUE_TYPE;

/*
 * ACEINNA packet
 */
typedef struct xbow_packet
{
    unsigned short packet_type;
    char                        length;
    unsigned short crc;
    char                        data[256];
} XBOW_PACKET;

QUEUE_TYPE circ_buf;

/*****************************************************************************
 * FUNCTION: process_xbow_packet looks for packets in a queue
 * ARGUMENTS: queue_ptr: is pointer to queue to process
 *                     result: will contain the parsed info when return value is 1
 * RETURNS:    0 when failed.
 *                     1 when successful
 *****************************************************************************/
int process_xbow_packet(QUEUE_TYPE *queue_ptr, XBOW_PACKET *result)
{
    unsigned short myCRC = 0, packetCRC = 0, packet_type = 0, numToPop=0, counter=0;
    char packet[100], tempchar, dataLength;

    if(Empty(queue_ptr))
    {
```

```
        return 0;   /* empty buffer */
    }

    /* find header */
    for(numToPop=0; numToPop+1<Size(queue_ptr) ;numToPop+=1)
    {
        if(0x5555==peekWord(queue_ptr, numToPop)) break;
    }

    Pop(queue_ptr, numToPop);

    if(Size(queue_ptr) <= 0)
    {
        /* header was not found */
        return 0;
    }

/* make sure we can read through minimum length packet */
if(Size(queue_ptr)<7)
{
   return 0;
}

/* get data length (5th byte of packet) */
dataLength = peekByte(queue_ptr, 4);

/* make sure we can read through entire packet */
if(Size(queue_ptr) < 7+dataLength)
{
   return 0;
   }


   /* check CRC */
   myCRC = calcCRC(queue_ptr, 2,dataLength+3);
   packetCRC = peekWord(queue_ptr, dataLength+5);

   if(myCRC != packetCRC)
   {
       /* bad CRC on packet - remove the bad packet from the queue and return */
       Pop(queue_ptr, dataLength+7);
       return 0;
   }
```

```
    /* fill out result of parsing in structure */
    result->packet_type = peekWord(queue_ptr, 2);
    result->length      = peekByte(queue_ptr, 4);
    result->crc         = packetCRC;
    for(counter=0; counter < result->length; counter++)
    {
        result->data[counter] = peekByte(queue_ptr, 5+counter);
    }

    Pop(queue_ptr, dataLength+7);

    return 1;
}

/*****************************************************************************
 * FUNCTION: calcCRC calculates a 2-byte CRC on serial data using
 *      CRC-CCITT 16-bit standard maintained by the ITU
 *                    (International Telecommunications Union).
 * ARGUMENTS: queue_ptr is pointer to queue holding area to be CRCed
 *                    startIndex is offset into buffer where to begin CRC calculation
 *                    num is offset into buffer where to stop CRC calculation
 * RETURNS:    2-byte CRC
 *****************************************************************************/
unsigned short calcCRC(QUEUE_TYPE *queue_ptr, unsigned int startIndex, unsigned int num)
{
    unsigned int i=0, j=0;
    unsigned short crc=0x1D0F; //non-augmented inital value equivalent to augmented
initial value 0xFFFF

    for (i=0; i<num; i+=1) {
        crc ^= peekByte(queue_ptr, startIndex+i) << 8;

        for(j=0;j<8;j+=1) {
            if(crc & 0x8000) crc = (crc << 1) ^ 0x1021;
            else crc = crc << 1;
        }
    }
    return crc;
}

/*****************************************************************************
 * FUNCTION: Initialize - initialize the queue
```

```
 * ARGUMENTS: queue_ptr is pointer to the queue
 ******************************************************************************/
void Initialize(QUEUE_TYPE *queue_ptr)
{
    queue_ptr->count = 0;
    queue_ptr->front = 0;
    queue_ptr->rear = -1;
}

/******************************************************************************
 * FUNCTION: AddQueue - add item in front of queue
 * ARGUMENTS: item holds item to be added to queue
 *                     queue_ptr is pointer to the queue
 * RETURNS:    returns 0 if queue is full. 1 if successful
 ******************************************************************************/
int AddQueue(char item, QUEUE_TYPE *queue_ptr)
{
    int retval = 0;
    if(queue_ptr->count >= MAXQUEUE)
    {
        retval = 0;/* queue is full */
    }
    else
    {
        queue_ptr->count++;
        queue_ptr->rear = (queue_ptr->rear + 1) % MAXQUEUE;
        queue_ptr->entry[queue_ptr->rear] = item;
        retval = 1;
    }
    return retval;
}

/******************************************************************************
 * FUNCTION: DeleteQeue - return an item from the queue
 * ARGUMENTS: item will hold item popped from queue
 *                     queue_ptr is pointer to the queue
 * RETURNS:    returns 0 if queue is empty. 1 if successful
 ******************************************************************************/
int DeleteQueue(char *item, QUEUE_TYPE *queue_ptr)
{
    int retval = 0;
    if(queue_ptr->count <= 0)
    {
```

```
        retval = 0; /* queue is empty */
    }
    else
    {
        queue_ptr -> count--;
        *item = queue_ptr->entry[queue_ptr->front];
        queue_ptr->front = (queue_ptr->front+1) % MAXQUEUE;
        retval=1;
    }
    return retval;
}

/*****************************************************************************
 * FUNCTION: peekByte returns 1 byte from buffer without popping
 * ARGUMENTS: queue_ptr is pointer to the queue to return byte from
 *                    index is offset into buffer to which byte to return
 * RETURNS:    1 byte
 * REMARKS:    does not do boundary checking. please do this first
 *****************************************************************************/
char peekByte(QUEUE_TYPE *queue_ptr, unsigned int index) {
    char byte;
    int firstIndex;

    firstIndex = (queue_ptr->front + index) % MAXQUEUE;

    byte = queue_ptr->entry[firstIndex];
    return byte;
}


/*****************************************************************************
 * FUNCTION: peekWord returns 2-byte word from buffer without popping
 * ARGUMENTS: queue_ptr is pointer to the queue to return word from
 *                    index is offset into buffer to which word to return
 * RETURNS:    2-byte word
 * REMARKS:    does not do boundary checking. please do this first
 *****************************************************************************/
unsigned short peekWord(QUEUE_TYPE *queue_ptr, unsigned int index) {
    unsigned short word, firstIndex, secondIndex;

    firstIndex = (queue_ptr->front + index) % MAXQUEUE;
    secondIndex = (queue_ptr->front + index + 1) % MAXQUEUE;
```

```
    word = (queue_ptr->entry[firstIndex] << 8) & 0xFF00;
    word |= (0x00FF & queue_ptr->entry[secondIndex]);
    return word;
}

/*****************************************************************************
 * FUNCTION: Pop - discard item(s) from queue
 * ARGUMENTS: queue_ptr is pointer to the queue
 *                     numToPop is number of items to discard
 * RETURNS:    return the number of items discarded
 *****************************************************************************/
int Pop(QUEUE_TYPE *queue_ptr, int numToPop)
{
    int i=0;
    char tempchar;
    for(i=0; i<numToPop; i++)
    {
        if(!DeleteQueue(&tempchar, queue_ptr))
        {
            break;
        }
    }
    return i;
}

/*****************************************************************************
 * FUNCTION: Size
 * ARGUMENTS: queue_ptr is pointer to the queue
 * RETURNS:    return the number of items in the queue
 *****************************************************************************/
int Size(QUEUE_TYPE *queue_ptr)
{
    return queue_ptr->count;
}

/*****************************************************************************
 * FUNCTION: Empty
 * ARGUMENTS: queue_ptr is pointer to the queue
 * RETURNS:    return 1 if empty, 0 if not
 *****************************************************************************/
int Empty(QUEUE_TYPE *queue_ptr)
{
```

```
    return queue_ptr->count <= 0;
}


/*****************************************************************************
 * FUNCTION: Full
 * ARGUMENTS: queue_ptr is pointer to the queue
 * RETURNS:    return 1 if full, 0 if not full
 *****************************************************************************/
int Full(QUEUE_TYPE *queue_ptr)
{
    return queue_ptr->count >= MAXQUEUE;
}
```

Appendix D: Sample Packet Decoding

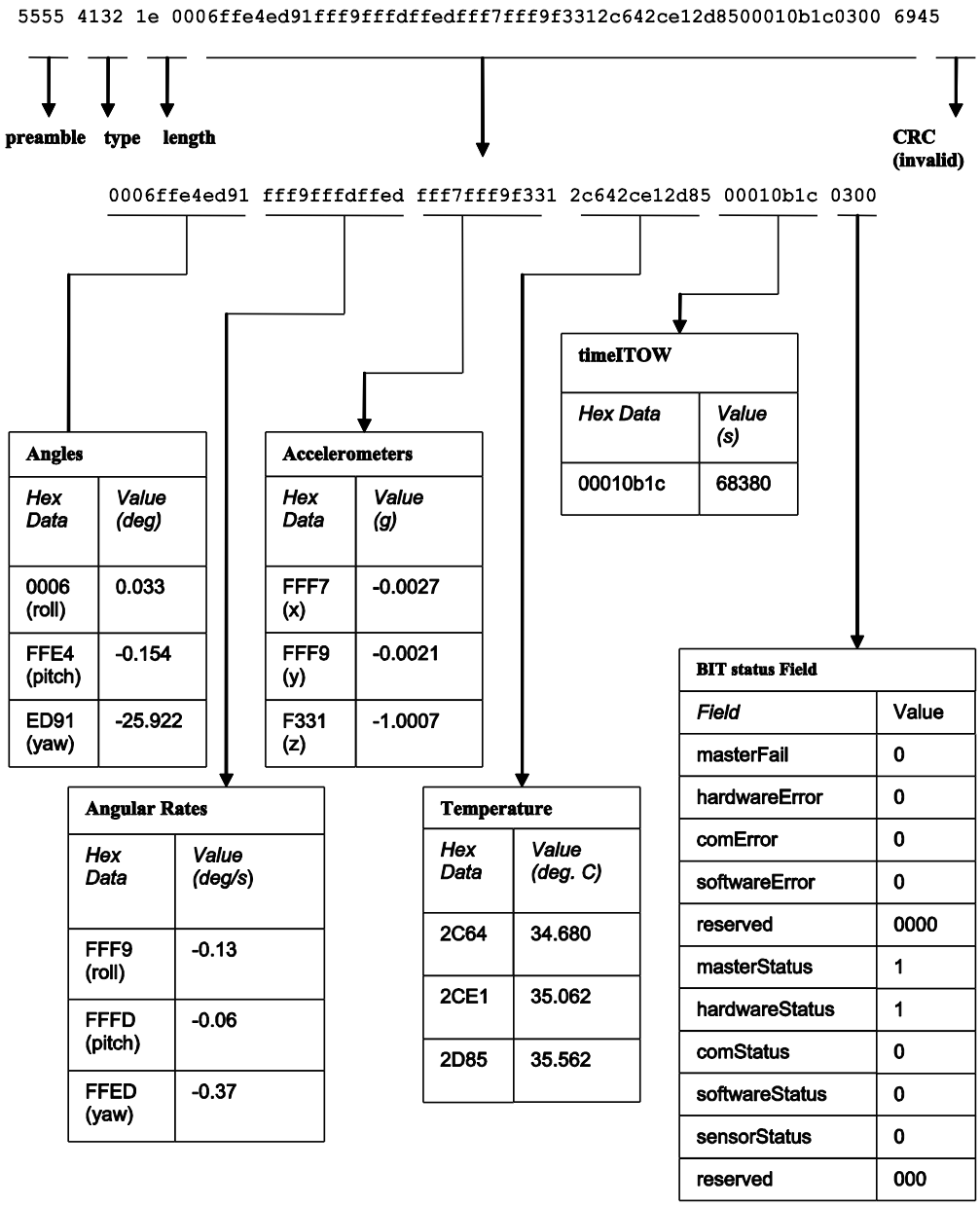
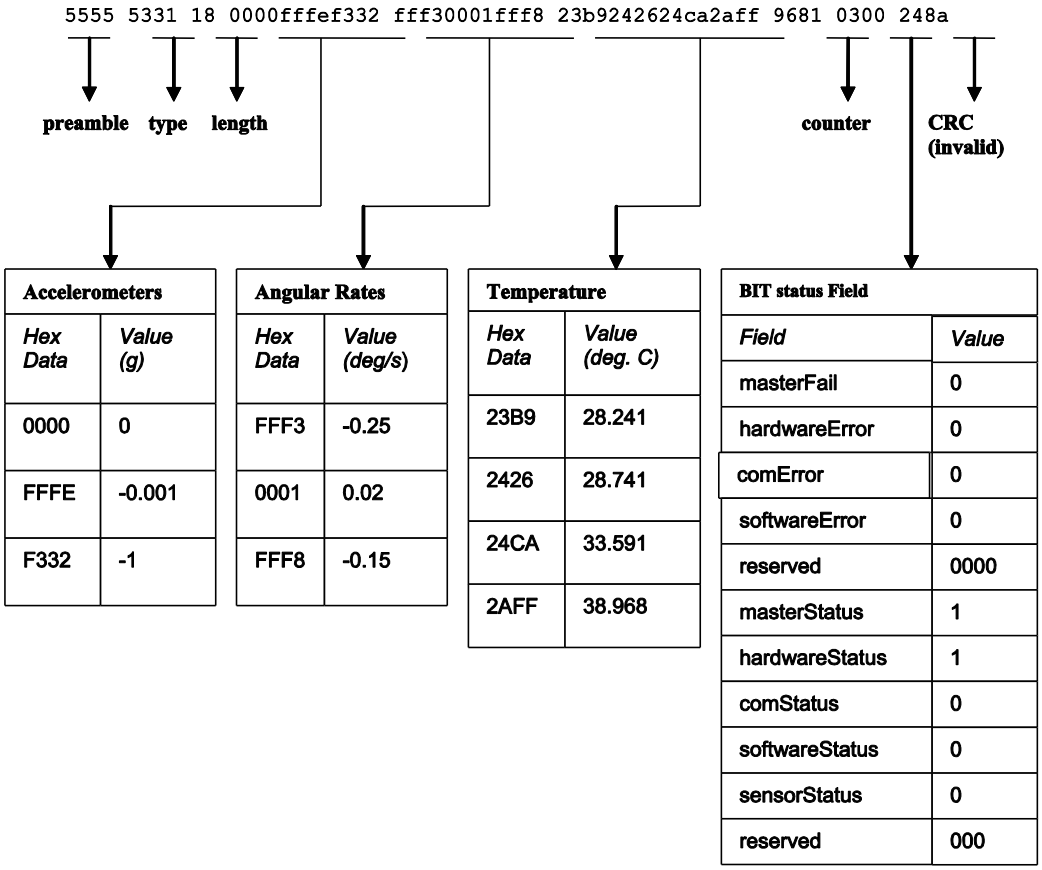


Figure 30 Example payload from Angle Data Packet 2 (A2)



Accelerometers	
Hex Data	*Value (g)*
0000	0
FFFE	-0.001
F332	-1

Angular Rates	
Hex Data	*Value (deg/s)*
FFF3	-0.25
0001	0.02
FFF8	-0.15

Temperature	
Hex Data	*Value (deg. C)*
23B9	28.241
2426	28.741
24CA	33.591
2AFF	38.968

BIT status Field	
Field	*Value*
masterFail	0
hardwareError	0
comError	0
softwareError	0
reserved	0000
masterStatus	1
hardwareStatus	1
comStatus	0
softwareStatus	0
sensorStatus	0
reserved	000

Figure 31 Example payload from Scaled Data Packet 1 (S1)

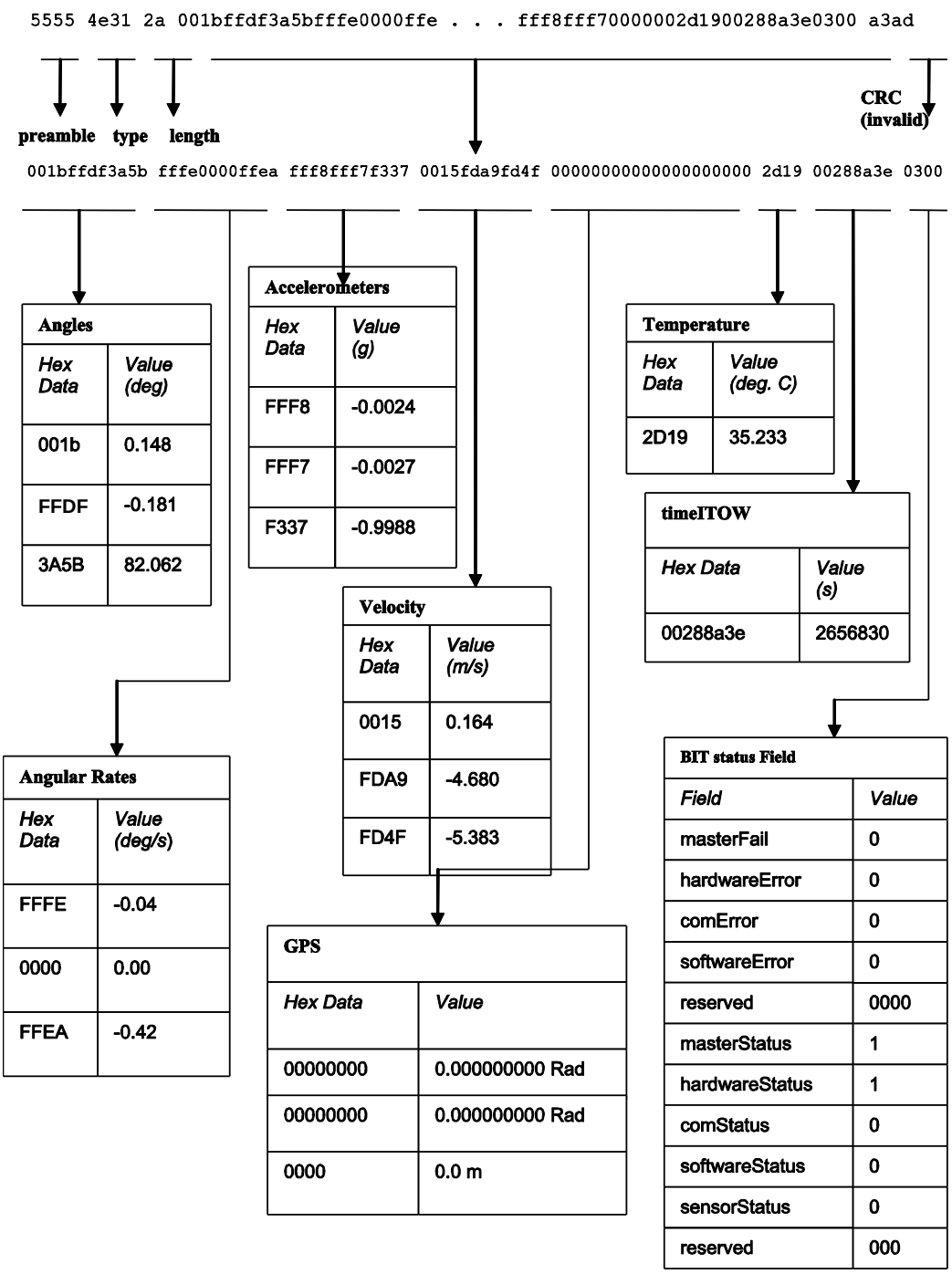


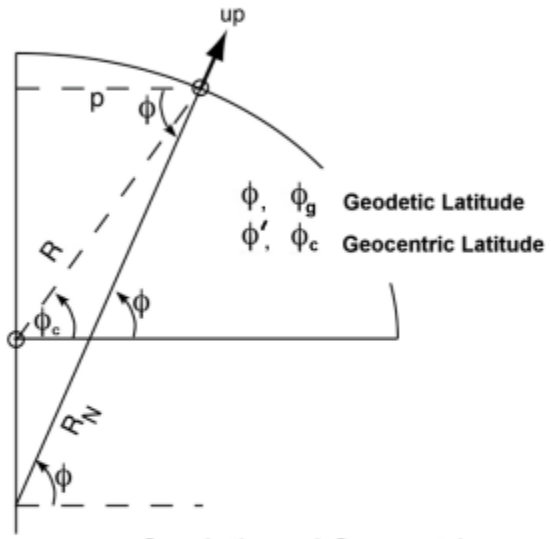
Figure 32 Example payload from Nav Data Packet 1 (N1)

Appendix E: Geodetic Coordinate Conversions

Geodetic Coordinate Conversions

James R. Clynch
February 2006

I. Geodetic to/from Geocentric Latitude



Geodetic and Geocentric Latitude

A. Geodetic Latitude (ϕ, or ϕ_g) to Geocentric Latitude (ϕ', or ϕ_c)

There are many equations that can be used. One of the most common involves the tangent of the latitude. At a geodetic or ellipsoidal height h,

$$\tan\phi_c = \left[1 - e^2 \frac{R_N}{R_N + h}\right] \tan\phi$$

where the radius of curvature in the prime vertical, R_N, is given by

$$R_N = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}}$$

B. Geocentric Latitude (ϕ', or ϕ_c) to Geodetic Latitude (ϕ, or ϕ_g)

This case uses the same equations. The value of R_N is found using the geocentric latitude. The error in this approximation is second order in smallness and is usually ignored. The ratio factor is, of course divided into the right hand side of the tangent equation in this case

$$\tan\phi = \left[1 - e^2 \frac{R_N}{R_N + h}\right]^{-1} \tan\phi_c$$

Note on common errors:

The heights used in these equations is ellipsoidal or geodetic height. It is not the height seen on maps and may differ from that height by 100 m. See the note on heights for more details. However as the height only enters in a ratio after being added to a quantity approximately the radius of the earth, the result is fairly insensitive to small errors in h.

A similar magnitude error happens if you use geocentric latitude in computing Rn. (A 70 m error at 45 deg latitude). In both cases the maximum North-South position error occurs at an altitude of 1 earth radius at 45 deg latitude. This error is 1.1 meter. One iteration on latitude types makes this much less than a cm. The error remains if you use the orthometric height even if you iterate. Both these errors result in zero position error on the ellipsoid.

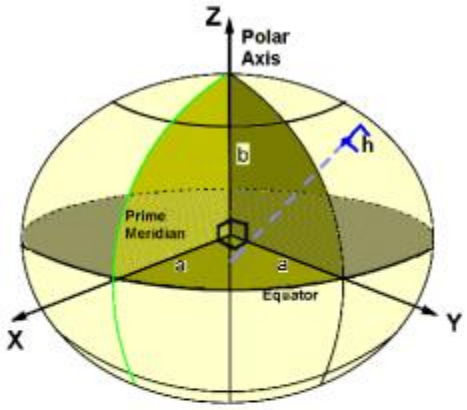
There is a common approximation for small h (near earth),

$$\tan\phi_c = [1 - e^2 + 1.1x10^{-9} h(m)]\tan\phi$$

where h must be in meters.

The constant multiplying h is e^2/Rn and has the units of inverse length. Clearly this approximation will not work well at altitudes that are a significant fraction of a earth radius, such as GPS or geosynchronous satellites.

II. Latitude, Longitude and Height to/from ECEF (x,y,z)



Latitude, Longitude, Height To/From (X,Y,Z)

A. Latitude, Longitude, Height to ECEF xyz

There is a closed form solution for this transformation. Given geodetic latitude, ϕ, (what you find on maps), longitude, λ, and ellipsoidal height h, then

$$x = (R_N + h)\cos\phi\cos\lambda$$
$$y = (R_N + h)\cos\phi\sin\lambda$$
$$z = ([1-e^2]R_N + h)\sin\phi$$

Note that the longitude will be East Longitude. This is the convention for geodesy.

B. ECEF xyz to Latitude, Longitude, Height

There is no closed form solution for this transformation if the altitude is not zero. The problem is that the radius R_N is needed to find geodetic height h and geodetic latitude is needed to find R_N. The usual procedure is to iterate beginning with the assumption that there is no difference between geodetic and geocentric latitude.

$$h = \sqrt{L^2 + z^2} - R_N ,$$

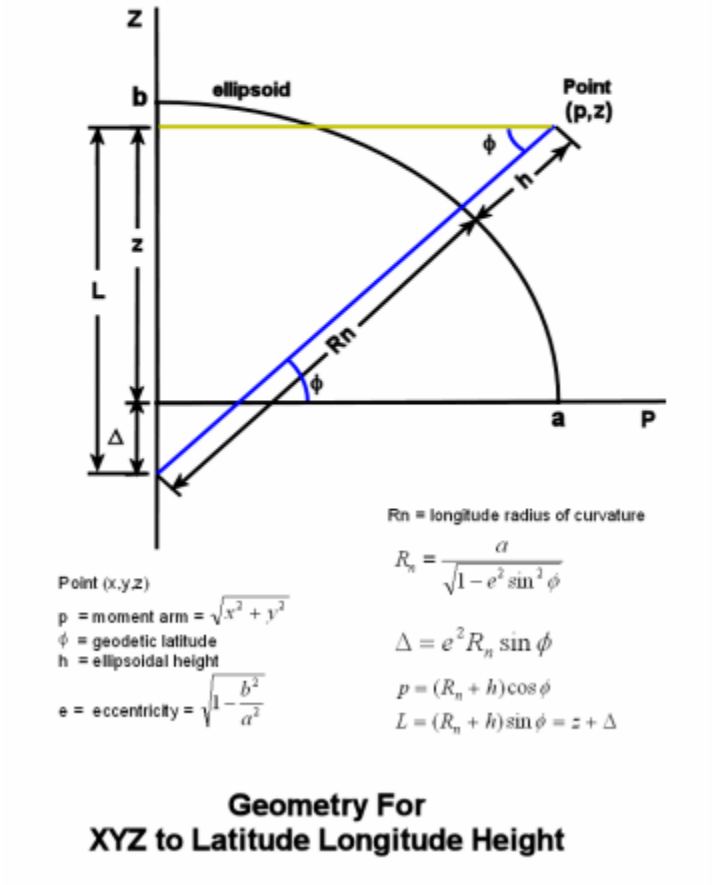
where

$$L = z + e^2 R_N \sin\phi .$$

Where

$L = z + e^2 R_N \sin\phi$

These equations are based on the geometry in the figure below.



Geometry For
XYZ to Latitude Longitude Height

First compute the longitude, which can be precisely done.

$$\begin{aligned}\lambda &= \text{a tan}(y/x) \\ &= \text{a tan 2}(y,x)\end{aligned}$$

The second form is the usual computer call for a 4-quadrant arctangent. Note that computer code usually returns angles in radians. These must be converted to degrees. Note also that this procedure produces East Longitude.

Next the physical radius of the point and the radius in the x-y plane are computed and used in an initial estimate of the altitude.

$$\begin{aligned}r &= \sqrt{x^2+y^2+z^2} \\ p &= \sqrt{x^2+y^2}\end{aligned}$$

The geocentric latitude is computed exactly, and used as the initial value for the geodetic latitude in the iteration loop.

$$\begin{aligned}\phi_c &= \text{a tan}(p/z) \\ &= \text{a tan 2}(p,z) \\ \phi_{now} &= \phi_c\end{aligned}$$

The loop is:

$$h = \frac{p}{\cos\phi_{now}} - R_N(\phi_{now})$$

$$\phi_{next} = \text{a tan}\left[\frac{z}{p}\left(1-e^2\frac{R_N}{R_N+h}\right)^{-1}\right] .$$

This converges in a few iterations (4 at most) to a few centimeters. This is for positions even at earth satellite altitudes. After the geodetic latitude, ϕ, is found the ellipsoidal height, h is obtained from

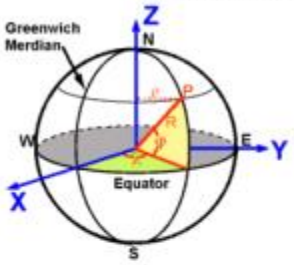
$$h = \frac{p}{\cos\phi} - R_N .$$

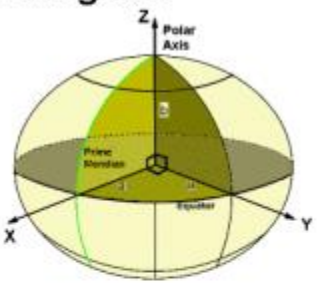
This equation for h diverges at the poles. There are two alternatives. One is

$$h = \frac{L}{\sin\phi} - R_N ,$$

and the other is

Cartesian to Angular





$$x = (r+h)\cos\phi\cos\lambda$$
$$y = (r+h)\cos\phi\sin\lambda$$
$$z = (r+h)\sin\phi$$

Spherical

$$x = (R_N+h)\cos\phi\cos\lambda$$
$$y = (R_N+h)\cos\phi\sin\lambda$$
$$z = [(1-e^2)R_N+h]\sin\phi$$

Ellipsoidal

NOTE: The heights used in these equations is ellipsoidal or geodetic height. It is not the height seen on maps and may differ from that height by 100 m. See the note on heights for more details.

LifeElectronics
Живое партнерство

ООО «ЛайфЭлектроникс» **“LifeElectronics” LLC**

ИНН 7805602321 КПП 780501001 Р/С 40702810122510004610 ФАКБ "АБСОЛЮТ БАНК" (ЗАО) в г.Санкт-Петербурге К/С 30101810900000000703 БИК 044030703

Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.

С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров

Мы предлагаем:

- *Конкурентоспособные цены и скидки постоянным клиентам.*
- *Специальные условия для постоянных клиентов.*
- *Подбор аналогов.*
- *Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.*

- *Приемлемые сроки поставки, возможна ускоренная поставка.*
- *Доставку товара в любую точку России и стран СНГ.*
- *Комплексную поставку.*
- *Работу по проектам и поставку образцов.*
- *Формирование склада под заказчика.*

- *Сертификаты соответствия на поставляемую продукцию (по желанию клиента).*
- *Тестирование поставляемой продукции.*
- *Поставку компонентов, требующих военную и космическую приемку.*
- *Входной контроль качества.*
- *Наличие сертификата ISO.*

В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.

Конструкторский отдел помогает осуществить:

- *Регистрацию проекта у производителя компонентов.*
- *Техническую поддержку проекта.*
- *Защиту от снятия компонента с производства.*
- *Оценку стоимости проекта по компонентам.*
- *Изготовление тестовой платы монтаж и пусконаладочные работы.*

Тел: +7 (812) 336 43 04 (многоканальный)
Email: org@lifeelectronics.ru

www.lifeelectronics.ru