**University of Minnesota**

**FMU Communication Protocol**

Table of Contents

[1 Ethernet Interface 4](#_Toc427836711)

[1.1 Packet Format 4](#_Toc427836712)

[1.2 Packet Summary 4](#_Toc427836713)

[1.3 Packet Definitions 5](#_Toc427836714)

[1.3.1 Host Heartbeat (0x00) 5](#_Toc427836715)

[1.3.2 Control Surface Command (0x01) 5](#_Toc427836716)

[1.3.3 GPS Command (0x02) 6](#_Toc427836717)

[1.3.4 Host Debug / Exception (0x7F) 6](#_Toc427836718)

[1.3.5 FMU Heartbeat (0x80) 7](#_Toc427836719)

[1.3.6 IMU Data (0x81) 8](#_Toc427836720)

[1.3.7 GPS Data (0x82) 9](#_Toc427836721)

[1.3.8 Control Surface Data (0x84) 9](#_Toc427836722)

[1.3.9 FMU Debug / Exception (0xFF) 10](#_Toc427836723)

[2 CAN Interface 11](#_Toc427836724)

[2.1 Message Protocol 11](#_Toc427836725)

[2.2 Message Summary 11](#_Toc427836726)

[2.3 Message Definitions 12](#_Toc427836727)

[2.3.1 Servo Command 12](#_Toc427836728)

[2.3.2 Servo Status 12](#_Toc427836729)

[2.3.3 V\_SENSE Data 12](#_Toc427836730)

[2.3.4 Node Status 12](#_Toc427836731)

[2.3.5 Node Version 13](#_Toc427836732)

[2.3.6 Configuration Modification 13](#_Toc427836733)

[2.3.7 Configuration Write 15](#_Toc427836734)

[2.3.8 Configuration Read 15](#_Toc427836735)

[3 Document Revision Log 16](#_Toc427836736)

# Ethernet Interface

The Ethernet interface is the primary means of communication with the flight management unit (FMU). Data is transmitted using UDP packets. A server port number of 55455 is used for both the Host and FMU.

## Packet Format

The UMN protocol format is as follows:

|  |  |  |
| --- | --- | --- |
| Byte # | Description | Value |
| 0 | Header 0 | ASCII ‘U’ (0x55) |
| 1 | Header 1 | ASCII ‘M’ (0x4D) |
| 2 | Header 2 | ASCII ‘N’ (0x4E) |
| 3 | Payload Type | See packet definition. |
| 4 | Length (LSB) | Payload length (# of bytes 6 to N). |
| 5 | Length (MSB) |
| 6 … N | Payload | See packet definition. Multi-byte values use little endian. |
| N + 1 | CRC (LSB) | CRC-16-CCITT of bytes 3 to N. |
| N + 2 | CRC (MSB) |

## Packet Summary

|  |  |  |
| --- | --- | --- |
| Payload Type | Data Direction | Description |
| 0x00 | Host to FMU | Host Heartbeat |
| 0x01 | Host to FMU | Control Surface Command |
| 0x02 | Host to FMU | GPS Command |
| 0x7F | Host to FMU | Host Debug / Exception |
| 0x80 | FMU to Host | FMU Heartbeat |
| 0x81 | FMU to Host | IMU Data |
| 0x82 | FMU to Host | GPS Data |
| 0x84 | FMU to Host | Control Surface Data |
| 0xFF | FMU to Host | FMU Debug / Exception |

## Packet Definitions

### Host Heartbeat (0x00)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Message Type | Host Heartbeat (0x00) | | | | |
| Description | This packet provides periodic status and monitoring of the host. | | | | |
| Data Direction | Host to FMU | | | | |
| Frequency | 1 Hz | | | | |
| Message Structure | Header | Type | Length (Bytes) | Payload | CRC |
| 0x55, 0x4D, 0x4E | 0x00 | 16 | See below. | CRC-16-CCITT |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Data Format | Scaling | Name | Unit | Description |
| 0 | UINT32 | - | fwVersion | - | Firmware version ID. |
| 4 | UINT32 | - | hwVersion | - | Hardware version ID. |
| 8 | UINT32 | - | serialNum | - | Serial number. |
| 12 | UINT32 | - | msUptime | ms | System uptime in milliseconds. |

### Control Surface Command (0x01)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Message Type | Control Surface Command (0x01) | | | | |
| Description | This packet provides control surface command information. | | | | |
| Data Direction | Host to FMU | | | | |
| Frequency | 100 Hz typical | | | | |
| Message Structure | Header | Type | Length (Bytes) | Payload | CRC |
| 0x55, 0x4D, 0x4E | 0x01 | 6N1 | See below. | CRC-16-CCITT |

1 Up to 10 control surface fields are supported per ‘Control Surface Command’ message.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Data Format | Scaling | Name | Unit | Description |
| 3(N-1) + 0 | UINT8 | - | surfaceID | - | Control surface ID, 0-127. |
| 3(N-1) + 1 | UINT8 | - | cmdType | - | 0 = Use PWM command for control.  1 = Use Position command for control. |
| 3(N-1) + 2 | UINT16 | - | cmdPwm | us | Commanded PWM value. |
| 3(N-1) + 4 | INT16 | 1e3 | cmdPos | rad | Control surface position in scaled radians, input range: +/- 1570. |

### GPS Command (0x02)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Message Type | GPS Command (0x02) | | | | |
| Description | This packet provides GPS command information. | | | | |
| Data Direction | Host to FMU | | | | |
| Frequency | Asynchronous | | | | |
| Message Structure | Header | Type | Length (Bytes) | Payload | CRC |
| 0x55, 0x4D, 0x4E | 0x02 | 1-1024 | See below. | CRC-16-CCITT |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Data Format | Scaling | Name | Unit | Description |
| 0 … N | UINT8[] | - | gpsData | - | GPS receiver command. |

### Host Debug / Exception (0x7F)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Message Type | Host Debug / Exception (0x7F) | | | | |
| Description | This packet provides host debug and exception information. | | | | |
| Data Direction | Host to FMU | | | | |
| Frequency | Asynchronous | | | | |
| Message Structure | Header | Type | Length (Bytes) | Payload | CRC |
| 0x55, 0x4D, 0x4E | 0x7F | 1-1024 | See below. | CRC-16-CCITT |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Data Format | Scaling | Name | Unit | Description |
| 0 | UINT8[] | - | debugData | - | Application defined debug and exception data. Examples include numeric data memory dumps or ASCII formatted text strings. |

### FMU Heartbeat (0x80)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Message Type | FMU Heartbeat (0x80) | | | | |
| Description | This packet provides periodic status and monitoring of the FMS. | | | | |
| Data Direction | FMU to Host | | | | |
| Frequency | 1 Hz | | | | |
| Message Structure | Header | Type | Length (Bytes) | Payload | CRC |
| 0x55, 0x4D, 0x4E | 0x80 | 20 | See below. | CRC-16-CCITT |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Data Format | Scaling | Name | Unit | Description |
| 0 | UINT32 | - | fwVersion | - | Firmware version ID. |
| 4 | UINT32 | - | hwVersion | - | Hardware version ID. |
| 8 | UINT32 | - | serialNum | - | Serial number. |
| 12 | UINT32 | - | msUptime | ms | System uptime in milliseconds. |
| 16 | UINT16 | - | inputVoltage | mV | Input voltage in millivolts. |
| 18 | INT16 | 1e2 | boardTemp | C | Board temperature. |

### IMU Data (0x81)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Message Type | IMU Data (0x81) | | | | |
| Description | This packet provides IMU sensor data. | | | | |
| Data Direction | FMU to Host | | | | |
| Frequency | 100 Hz typical | | | | |
| Message Structure | Header | Type | Length (Bytes) | Payload | CRC |
| 0x55, 0x4D, 0x4E | 0x81 | 76 | See below. | CRC-16-CCITT |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Data Format | Scaling | Name | Unit | Description |
| 0 | UINT64 | - | fmuTime | us | FMU timestamp of queued packet. |
| 8 | UINT16 | - | imuType | - | IMU Type:  0 = VN-100  1 = MPU-9150 |
| 10 | UINT16 | - | imuValid | - | Set as 1 if data is valid, 0 otherwise.  Bit 0: Mag  Bit 1: Accel  Bit 2: Gyro  Bit 3: Temp  Bit 4: Press  Bit 5: Attitude  Bit 6-15: Reserved |
| 12 | UINT32 | - | imuTimeSyncIn | us | Elapsed time between last IMU sync pulse trigger and sampling of IMU measurements. |
| 16 | FP32 | - | magX | Gauss | Uncompensated magnetic X-axis. |
| 20 | FP32 | - | magY | Gauss | Uncompensated magnetic Y-axis. |
| 24 | FP32 | - | magZ | Gauss | Uncompensated magnetic Z-axis. |
| 28 | FP32 | - | accelX | m/s/s | Uncompensated acceleration X-axis. |
| 32 | FP32 | - | accelY | m/s/s | Uncompensated acceleration Y-axis. |
| 36 | FP32 | - | accelZ | m/s/s | Uncompensated acceleration Z-axis. |
| 40 | FP32 | - | gyroX | rad/s | Uncompensated angular rate X-axis. |
| 44 | FP32 | - | gyroY | rad/s | Uncompensated angular rate Y-axis. |
| 48 | FP32 | - | gyroZ | rad/s | Uncompensated angular rate Z-axis. |
| 52 | FP32 | - | temp | C | IMU temperature. |
| 56 | FP32 | - | pressure | kPa | Barometric pressure. |
| 60 | UINT32 | - | attTimeSyncIn | us | Elapsed time between last IMU sync pulse trigger and sampling of attitude measurements. |
| 64 | FP32 | - | yaw | deg | Estimated yaw attitude. |
| 68 | FP32 | - | pitch | deg | Estimated pitch attitude. |
| 72 | FP32 | - | roll | deg | Estimated roll attitude. |

### GPS Data (0x82)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Message Type | GPS Data (0x82) | | | | |
| Description | This packet provides GPS sensor data. | | | | |
| Data Direction | FMU to Host | | | | |
| Frequency | 1 Hz typical | | | | |
| Message Structure | Header | Type | Length (Bytes) | Payload | CRC |
| 0x55, 0x4D, 0x4E | 0x82 | Variable | See below. | CRC-16-CCITT |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Data Format | Scaling | Name | Unit | Description |
| 0 | UINT64 | - | fmuTime | us | FMU timestamp of queued packet. |
| 8 | UINT16 | - | gpsType | - | GPS Type:  0: Novatel OEMStar  1: U-blox |
| 10 … N | UINT8[] | - | gpsData | - | GPS receiver data. Though not guaranteed, a best effort will be made to segment GPS data on packet boundaries. |

### Control Surface Data (0x84)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Message Type | Control Surface Status (0x84) | | | | |
| Description | This packet provides control surface status | | | | |
| Data Direction | FMU to Host | | | | |
| Frequency | 100 Hz typical | | | | |
| Message Structure | Header | Type | Length (Bytes) | Payload | CRC |
| 0x55, 0x4D, 0x4E | 0x84 | 13N | See below. | CRC-16-CCITT |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Data Format | Scaling | Name | Unit | Description |
| 3(N-1) + 0 | UINT8 | - | surfaceID | - | Control surface ID, 0-127. |
| 3(N-1) + 1 | UINT16 | - | cmdTypeEcho | - | Echo of the command type used.  0 = Use PWM command for control.  1 = Use Position command for control. |
| 3(N-1) + 3 | UINT16 | 1e3 | actPwm | us | Actual surface PWM application. |
| 3(N-1) + 5 | UINT16 | - | inputVoltage | mV | Input voltage in millivolts. |
| 3(N-1) + 7 | UINT16 | - | inputCurrent | mA | Input current in milliamps. |
| 3(N-1) + 9 | INT16 | 1e2 | vsense1Cor | - | The calibration corrected value of V\_SENSE1. |
| 3(N-1) + 11 | INT16 | 1e2 | vsense2Cor | - | The calibration corrected value of V\_SENSE2. |

### FMU Debug / Exception (0xFF)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Message Type | FMU Debug / Exception (0xFF) | | | | |
| Description | This packet provides FMU debug and exception information. | | | | |
| Data Direction | FMU to Host | | | | |
| Frequency | Asynchronous | | | | |
| Message Structure | Header | Type | Length (Bytes) | Payload | CRC |
| 0x55, 0x4D, 0x4E | 0xFF | 0-1024 | See below. | CRC-16-CCITT |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Data Format | Scaling | Name | Unit | Description |
| 0 | UINT8[] | - | debugData | - | Application defined debug and exception data. Examples include numeric data memory dumps or ASCII formatted text strings. |

# CAN Interface

## Message Protocol

The communication protocol uses 29-bit extended identifier format. The 29-bit identifier is divided into six fields:

|  |  |  |
| --- | --- | --- |
| Bit Index | Field | Description |
| 28-19 | Data Type ID | The type of the data structure (i.e. how to interpret it). |
| 18-17 | Transfer Type | This two bit field is used for identifying the type of message being transferred. The four possible values are:  0b00 – Service response.  0b01 – Service request.  0b10 – Message broadcast.  0b11 – Message unicast. |
| 16-10 | Source Node ID | The node ID of the transmitting node. |
| 9-7 | Reserved | These bits are not used. |
| 6-0 | Destination Node ID | The node ID of the receiving node. (N/A for broadcast messages) |

## Message Summary

The following table provides a summary of the CAN messages transmitted between the Flight Management Unit (FMU) and the Servo-Nodes (S-Node). The payload definition for the messages are detailed later.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Type ID | Transfer Type | Rate (Hz) | Transmitter | Receiver | Message Name |
| 10 | Message unicast | 100 | FMU | S-Node | Servo Command |
| 20 | Message broadcast | 100 | S-Node | - | Servo Status |
| 21 | Message broadcast | 100 | S-Node | - | VSENSE Data |
| 770 | Message broadcast | 2 | FMU/S-Node | - | Node Status |
| 771 | Message broadcast | 2 | FMU/S-Node | - | Node Version |
| 800 | Service request | Async  (< 10) | FMU | S-Node | Configuration Write |
| Service response | S-Node | FMU |
| 801 | Service request | Async  (< 50) | FMU | S-Node | Configuration Read |
| Service response | S-Node | FMU |

Note: Asynchronous messages have the potential of over-loading the hardware’s processing throughput. If asynchronous messages are received at too fast of a rate, the later message(s) will be ignored.

Note: ‘Configuration Write’ messages result in Non-Volatile Memory (NVM) being updated. Writing of NVM causes the processor to be stalled (for approximately 40ms) for each configuration parameter update (i.e. each message). Therefore, periodic software execution should not be expected immediately after issuing a ‘Configuration Write’ message.

Note: No bounds checking on input parameters (e.g. cmdPwm) is performed by the S-Node. It’s the responsibility of the source controller to verify values are within expected ranges.

## Message Definitions

Multi-byte data fields are populated in little endian format.

### Servo Command

|  |  |  |  |
| --- | --- | --- | --- |
| Data Format | Name | LSb Value | Description |
| UINT16 | cmdType | - | 0 = Use PWM command for control.  1 = Use Position command for control. |
| UINT16 | cmdPwm | 1 us | Commanded PWM signal value. |
| INT16 | cmdPos | 1 mrad | Commanded surface position. Values in the range [-1000:1000] are required. |

### Servo Status

|  |  |  |  |
| --- | --- | --- | --- |
| Data Format | Name | LSb Value | Description |
| UINT16 | cmdTypeEcho | - | Echo of ‘cmdType’ used by node. |
| UINT16 | actPWM | 1 us | The actual PWM value applied to the servo. Calibration corrected value (based on cmdPos) or commanded PWM value (cmdPWM) depending on cmdType. |
| UINT16 | servoVoltage | 1 mV | Voltage at servo (V\_SERVO\_OUT). |
| UINT16 | servoCurrent | 1 mA | Current applied to servo. |

### V\_SENSE Data

|  |  |  |  |
| --- | --- | --- | --- |
| Data Format | Name | LSb Value | Description |
| UINT16 | vsense1Raw | 1 | The raw ATD count value of V\_SENSE1. |
| INT16 | vsense1Cor | 0.01 | The calibration corrected value of V\_SENSE1. |
| UINT16 | vsense2Raw | 1 | The raw ATD count value of V\_SENSE2. |
| INT16 | vsense2Cor | 0.01 | The calibration corrected value of V\_SENSE2. |

### Node Status

|  |  |  |  |
| --- | --- | --- | --- |
| Data Format | Name | LSb Value | Description |
| UINT16 | resetCondition | - | 1 = Power on reset.  2 = Brown out reset.  3 = Software induced reset (expected).  4 = Software fault reset (not-expected). |
| UINT16 | resetDetail | - | Annunciation of ‘RCON’ register value. See dsPIC33EV256GM102 datasheet for detail. |

### Node Version

|  |  |  |  |
| --- | --- | --- | --- |
| Data Format | Name | LSb Value | Description |
| UINT8 | nodeType | - | 0 = FMU  1 = Servo |
| UINT8 | revVer | - | Software revision version number. |
| UINT8 | minVer | - | Software minor version number. |
| UINT8 | majVer | - | Software major version number. |
| UINT32 | serialNum | - | Serial number. |

### Configuration Modification

The following table list the configuration elements. The ‘cfgSel Value’ is used with read and write commands to access the required configuration data element.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cfgSel Value | Data Format | Selection | LSb Value | Description |
| 0 | UINT8 | nodeID | - | The CAN-Node Identifier. Note: This value must be less than 0x80. Value 0x7F represents an unconfigured node. |
| 1 | INT32 | pwmCoeff0 | 0.01 us | The zero degree coefficient of the PWM calibration correction. |
| 2 | INT32 | pwmCoeff1 | 0.01 us/rad | The 1st degree coefficient of the PWM calibration correction. |
| 3 | INT32 | pwmCoeff2 | 0.01 us/rad2 | The 2nd degree coefficient of the PWM calibration correction. |
| 4 | INT32 | pwmCoeff3 | 0.01 us/rad3 | The 3rd degree coefficient of the PWM calibration correction. |
| 5 | INT32 | pwmCoeff4 | 0.01 us/rad4 | The 4th degree coefficient of the PWM calibration correction. |
| 6 | INT32 | pwmCoeff5 | 0.01 us/rad5 | The 5th degree coefficient of the PWM calibration correction. |
| 7 | INT32 | vsense1Coeff0 | 0.0001\*Note | The zero degree coefficient of the V\_SENSE1 calibration correction. |
| 8 | INT32 | vsense1Coeff1 | 0.0001\*Note | The 1st degree coefficient of the V\_SENSE1 calibration correction. |
| 9 | INT32 | vsense1Coeff2 | 0.0001\*Note | The 2nd degree coefficient of the V\_SENSE1 calibration correction. |
| 10 | INT32 | vsense1Coeff3 | 0.0001\*Note | The 3rd degree coefficient of the V\_SENSE1 calibration correction. |
| 11 | INT32 | vsense1Coeff4 | 0.0001\*Note | The 4th degree coefficient of the V\_SENSE1 calibration correction. |
| 12 | INT32 | vsense1Coeff5 | 0.0001\*Note | The 5th degree coefficient of the V\_SENSE1 calibration correction. |
| 13 | INT32 | vsense2Coeff0 | 0.0001\*Note | The zero degree coefficient of the V\_SENSE2 calibration correction. |
| 14 | INT32 | vsense2Coeff1 | 0.0001\*Note | The 1st degree coefficient of the V\_SENSE2 calibration correction. |
| 15 | INT32 | vsense2Coeff2 | 0.0001\*Note | The 2nd degree coefficient of the V\_SENSE2 calibration correction. |
| 16 | INT32 | vsense2Coeff3 | 0.0001\*Note | The 3rd degree coefficient of the V\_SENSE2 calibration correction. |
| 17 | INT32 | vsense2Coeff4 | 0.0001\*Note | The 4th degree coefficient of the V\_SENSE2 calibration correction. |
| 18 | INT32 | vsense2Coeff5 | 0.0001\*Note | The 5th degree coefficient of the V\_SENSE2 calibration correction. |

\*Note: VSENSE1 and VSENSE2 are left as generic signals – i.e. any analog input could be connected to these inputs. This yields a more configurable design, but one which is less intuitive as the units for VSENSE inputs, coefficients, and corrected outputs are currently unidentifiable.

#### VsenseX Calculation

The calculation is implemented assuming the following scaling:

* Coefficient scaling of 1E4 (per § 2.3.6): e.g. 5.4321 is represented with an integer value of 54321.
* VsenseX scaling of 2^12, e.g. 0.25 is represented with an integer value of 1024.
* Output scaling (i.e. vsenseXCor) of 1E2 (per § 2.3.3): e.g. 3.14 is represented with an integer value of 314.

***Example***

With a VsenseX value of ‘0.25’ (i.e. an integer value of ‘1024’) and coefficient values of those specified in the following table, the calculated vsenseXCor value is ‘6.00’ (i.e. an integer value of ‘600’).

|  |  |  |
| --- | --- | --- |
| Coefficient | Integer Value | Decimal Equivalent |
| vsenseXCoeff0 | 10,000 | 1.0000 |
| vsenseXCoeff1 | 40,000 | 4.0000 |
| vsenseXCoeff2 | 160,000 | 16.0000 |
| vsenseXCoeff3 | 640,000 | 64.0000 |
| vsenseXCoeff4 | 2,560,000 | 256.0000 |
| vsenseXCoeff5 | 10,240,000 | 1024.0000 |

### Configuration Write

Request

|  |  |  |  |
| --- | --- | --- | --- |
| Data Format | Name | LSb Value | Description |
| UINT16 | cfgSel | - | See following table for valid values. |
| variable | cfgVal | variable | The value to set the configuration field. |

Note: After updating the nodeID the software performs a self-induced reset. This causes the node to reinitialize and the updated nodeID value to be used for CAN message filtering.

Response

|  |  |  |  |
| --- | --- | --- | --- |
| Data Format | Name | LSb Value | Description |
| UINT16 | cfgSel | - | See previous table for valid values. |
| UINT16 | faultStatus | - | 0 = Success in updating the value.  1 = Failure in updating the value. |

### Configuration Read

Request

|  |  |  |  |
| --- | --- | --- | --- |
| Data Format | Name | LSb Value | Description |
| UINT16 | cfgSel | - | See previous table for valid values. |

Response

|  |  |  |  |
| --- | --- | --- | --- |
| Data Format | Name | LSb Value | Description |
| UINT16 | cfgSel | - | See previous table for valid values. |
| variable | cfgVal | variable | The value of the configuration data. |

# Document Revision Log

1. Initial draft.
2. Updated with CAN bus protocol specification.
3. Updated CAN bus protocol with new header formatting the message content (review as new). Also updated Ethernet IMU Data message to include a sync time for each data packet from the IMU – i.e. IMU data and attitude data.
4. The following updates were performed:
   1. Added specification of server port used by both FMU and Host.
   2. Removed definition of ‘Air Data’ Ethernet message since not used. Note: Air data sensors could be attached to any CAN nodes Vsense1 or Vsense2 inputs for relaying of information to the Host.
   3. Update Ethernet Control Surface Command messages to include option to directly command a CAN Servo-Node PWM value.
   4. Added definition of Ethernet GPS Command message for allowing the host to send a command to the GPS unit.
   5. Updated Ethernet Control Surface Data message to include calibration corrected data for Vsense1 and Vsense2 signals.