

| | | | |
|--|---|--|--|
| Document Number | | RG_SPEC-0027 | |
| Title | | Inertial Measurement Unit - IMU | |
| CAN Speed: 1Mbps | Base CAN ID: 0x470h | Serial Number: | Checked By: |
| Accelerometer Range: +/- 4G | Angular Rate Range: +/- 250deg/s | CAN Mode: Standard | |
| Accelerometer Filter Type: Differential Avg | | Accelerometer Filter Time: 10ms | |
| Firmware Revision | Date | Prepared By | Change History |
| 1.0 | 11/27/2013 | James Nero | Initial release |
| 1.5 | 7/8/2015 | Steven Bravek | Fixed accelerometer calibration |
| 1.6 | 7/21/2015 | Steven Bravek | Updated manager based calibration |
| 1.7 | 4/19/2016 | Steven Bravek | Added more filtering options for accelerometer |

The RaceGrade IMU contains a three axis accelerometer and a three axis angular rate sensor designed to measure and communicate all values over the CAN bus.

Part # RG IMU

Specifications:

| | |
|----------------------------|---|
| Acceleration Ranges: | +/- 2, 4, 8, 16 G |
| Angular Rate Ranges: | +/- 250, 500, 2000 deg/s |
| CAN bus speeds: | 1 Mbps, 500 Kbps, 250 Kbps, 125 Kbps |
| Supply Voltage: | 9 - 36 VDC |
| Supply Current: | 30 mA |
| Temperature Range: | -20°C to +85°C |
| Dimensions: | 92 x 57 x 15 mm 3.625 x 2.25 x 0.59 in |
| Wire Lead Length: | 4" |
| Weight: | 95 grams |
| Ingress Protection Rating: | IP 68 |

CAN Messaging:

All channels are 16 bit signed Motorola style.
 Accelerations are transmitted in G per bit with a resolution of 0.001.
 Angular rates are transmitted in hertz with a resolution of 0.001.
 (multiple by 360 for deg/s)
 Transmission rate for 1 Mbps and 500 Kbps CAN speed is
 500Hz for acceleration, 125Hz for angular rates.
 Transmission rate for 250 Kbps and 125 Kbps CAN speed is
 166Hz for acceleration, 100Hz for angular rates.



Standard:

Base CAN ID:

Byte 0/1 = Lateral Acceleration

Byte 2/3 = Longitudinal Acceleration

Byte 4/5 = Vertical Acceleration

Base CAN ID+1:

Byte 0/1 = Yaw Rotational Rate

Byte 2/3 = Pitch Rotational Rate

Byte 4/5 = Roll Rotational Rate

Compound:

Base CAN ID:

Message 0:

Byte 0 = message ID

Byte 1 = N/A

Byte 2/3 = Lateral Acceleration

Byte 4/5 = Longitudinal Acceleration

Byte 6/7 = Vertical Acceleration

Message 1:

Byte 0 = message ID

Byte 1 = N/A

Byte 2/3 = Yaw Rotational Rate

Byte 4/5 = Pitch Rotational Rate

Byte 6/7 = Roll Rotational Rate

Connection:

Mating Connector: ASL606-05SN

pin 1 – Ground

pin 2 – N/C

pin 3 – Power

pin 4 – CAN Lo

pin 5 – CAN Hi

Mounting Considerations:

Do not firmly mount sensor to the chassis as this will induce unwanted engine vibration. Use the soft type of hook & loop tape for mounting, not the industrial type. This will reject the most amount of engine vibration through the chassis. Mount on a solid structure of the chassis. Do not mount on any type of thin sheet metal.

Manager:

The management software uses a PCAN-USB by [Peak Systems](#) to communicate with the IMU over CAN. The manager lets the user change accelerometer range, gyro range, CAN ID, CAN bus speed, and has the ability to zero the accelerometer sensor.

The manager uses the following CAN ID's to communicate with the IMU: 0x006, 0x007, 0x008, 0x009, and 0x014.

To download the latest software go to: <http://www.racegrade.com/downloads.html>

Accelerometer Filter:

The IMU has two low pass filter options for the accelerometer data. The first method is a Differential Average and the second is a Moving Average. The Differential Average takes the difference between the current value and the previous value; divides by the filter value and then adds to the previous value. The Moving Average, averages the accelerometer data based on filter time.

Zero Accelerometer:

The accelerometer can be zeroed using the IMU manager or by sending the following CAN message on ID 0x010. The zeroing feature assumes that the vertical axis is reading one G and the other two are reading zero G's.

Message structure (ID 0x010):

```

BYTE (0) = 0x69 // i
BYTE (1) = 0x6d // m
BYTE (2) = 0x75 // u
BYTE (3) = 0x7a // z
BYTE (4) = 0x65 // e
BYTE (5) = 0x72 // r
BYTE (6) = 0x6f // o
BYTE (7) = 0x7a // z

```

Drawing: