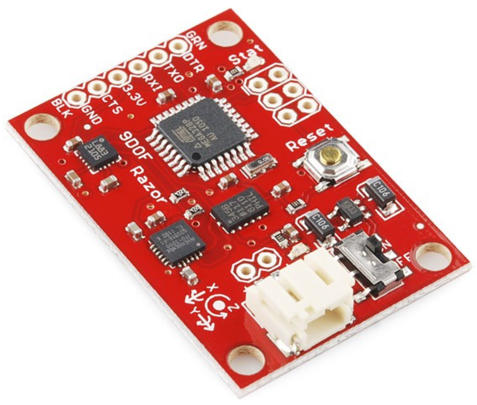
DATASHEET
Sparkfun Razor 9-DOF IMU

I – Description



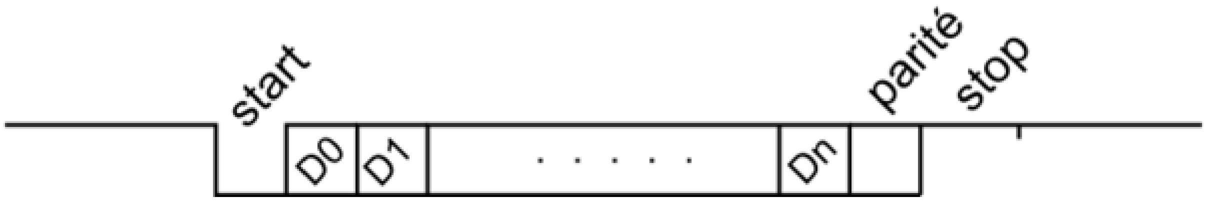
La Razor est une centrale d'attitude 9 axes composée d'un accéléromètre (ADXL345), d'un magnétomètre (HMC5883L) et d'un gyroscope (ITG-3200) reliés en I2C à un processeur on-board de type ATmega328.

La communication entre la carte et le DAARRT se fait cependant par le port série de la carte pcDuino en UART.

Ce protocole de communication utilise 2 fils pour communiquer : un pour la transmission (TX) et un pour la réception (RX). Il faut cependant faire attention aux branchements car le canal RX de la pcDuino se branche sur le TX de la Razor, et le TX de la pcDuino, sur le RX de la Razor.

Une trame UART est constituée des bits suivants :

- un bit de start toujours à 0 : servant à la synchronisation du récepteur
- les données : la taille peut varier (généralement entre 5 et 9 bits)
- éventuellement un bit de parité paire ou impaire
- un bit de stop toujours à 1 (la durée peut varier entre 1, 1,5 et 2 temps bit)



II – Communication avec la carte

1 – Configuration de la pcDuino

Pour utiliser le port série de la pcDuino, il faut en premier lieu configurer les entrée et sortie standard (GPIO) 0 et 1 de manière à fonctionner en UART. Pour cela il y a 3 commandes à exécuter :

```
# Chargement du module du noyau approprié (si non chargé)
sudo modprobe gpio

# Passage des pins 0 et 1 en mode UART
echo "3" > /sys/devices/virtual/misc/gpio/mode/gpio0
echo "3" > /sys/devices/virtual/misc/gpio/mode/gpio1
```

2 – Protocole de communication

Le protocole de communication avec la Razor est régi par le code Arduino qui y est implémenté. Celui-ci admet les commandes suivantes :

```
"#o<params>" - Set OUTPUT mode and parameters. The available
               options are:
      // Streaming output
      "#o0" - DISABLE continuous streaming output. Also see #f
               below.
      "#o1" - ENABLE continuous streaming output.

      // Angles output
      "#ob" - Output angles in BINARY format (yaw/pitch/roll as
              binary float, so one output frame is 3x4 = 12 bytes
              long).
      "#ot" - Output angles in TEXT format (Output frames have
              form like "#YPR=-142.28,-5.38,33.52", followed by
              carriage return and  line feed [\r\n]).

      // Sensor calibration
      "#oc" - Go to CALIBRATION output mode.
      "#on" - When in calibration mode, go on to calibrate NEXT
              sensor.

      // Sensor data output
      "#osct" - Output CALIBRATED SENSOR data of all 9 axes in
                TEXT format. One frame consist of three lines -
                one for each sensor: acc, mag, gyr.
      "#osrt" - Output RAW SENSOR data of all 9 axes in TEXT
                format. One frame consist of three lines - one for
                each sensor: acc, mag, gyr.
      "#osbt" - Output BOTH raw and calibrated SENSOR data of all
                9 axes in TEXT format. One frame consist of six
                lines - like #osrt and #osct combined (first RAW,
                then CALIBRATED). NOTE: This is a lot of number to
                text conversion work for the little 8MHz chip on
                the Razor boards.
```

```
    "#oscb" - Output CALIBRATED SENSOR data of all 9 axes in
              BINARY format. One frame consist of three 3x3
              float values = 36 bytes. Order
              is: acc x/y/z, mag x/y/z, gyr x/y/z.
    "#osrb" - Output RAW SENSOR data of all 9 axes in BINARY
              format. One frame consist of three 3x3 float
              values = 36 bytes. Order is: acc x/y/z, mag x/y/z,
              gyr x/y/z.
    "#osbb" - Output BOTH raw and calibrated SENSOR data of all
              9 axes in BINARY format. One frame consist of 2x36
              = 72 bytes – like #osrb and #oscb combined (first
              RAW, then CALIBRATED).

    // Error message output
    "#oe0" - Disable ERROR message output.
    "#oe1" - Enable ERROR message output.


"#f" - Request one output frame - useful when continuous output
       is disabled and updates are required in larger intervals
       only. Though #f only requests one reply, replies are
       still bound to the internal 20ms (50Hz) time raster. So
        worst case delay that #f can add is 19.99ms.


"#s<xy>" - Request synch token - useful to find out where the
           frame boundaries are in a continuous binary stream or
           to see if tracker is present and answering. The
           tracker will send "#SYNCH<xy>\r\n" in response (so
           it's possible to read using a readLine() function). x
           and y are two mandatory but arbitrary bytes that can
           be used to find out which request the answer belongs
           to.
```

3 – Exemple simple (python)

```python
import serial

# Initialisation (après configuration de l'UART)
bus = serial.Serial('/dev/ttyS1', 9600, timeout = 10)

# Ecriture/lecture
bus.write('#ob')          # Setting output to binary angles
bus.write('#f')           # Requesting an output frame
print bus.read(12)        # Reading 12 bytes
```

III – Données constructeur

Description: The 9DOF Razor IMU incorporates three sensors - an ITG-3200 (MEMS triple-axis gyro), ADXL345 (triple-axis accelerometer), and HMC5883L (triple-axis magnetometer) - to give you nine degrees of inertial measurement. The outputs of all sensors are processed by an on-board ATmega328 and output over a serial interface. This enables the 9DOF Razor to be used as a very powerful control mechanism for UAVs, autonomous vehicles and image stabilization systems.

The board comes programmed with the 8MHz Arduino bootloader (stk500v1) and some example firmware that demos the outputs of all the sensors. Simply connect to the serial TX and RX pins with a 3.3V FTDI Basic Breakout, open a terminal program to 57600bps and a menu will guide you through testing the sensors. You can use the Arduino IDE to program your code onto the 9DOF, just select the 'Arduino Pro or Pro Mini (3.3v, 8mhz) w/ATmega328' as your board.

The 9DOF operates at 3.3VDC; any power supplied to the white JST connector will be regulated down to this operating voltage - our LiPo batteries are an excellent power supply choice. The output header is designed to mate with our 3.3V FTDI Basic Breakout board, so you can easily connect the board to a computer's USB port. Or, for a wireless solution, it can be connected to the Bluetooth Mate or an XBee Explorer.

Having a hard time picking an IMU? Our Accelerometer, Gyro, and IMU Buying Guide might help!

Note: This product is a collaboration with Jordi Munoz of 3d Robotics. A portion of each sales goes back to them for product support and continued development.

Features:

- 9 Degrees of Freedom on a single, flat board:
    - ITG-3200 - triple-axis digital-output gyroscope
    - ADXL345 - 13-bit resolution, ±16g, triple-axis accelerometer
    - HMC5883L - triple-axis, digital magnetometer
- Outputs of all sensors processed by on-board ATmega328 and sent out via a serial stream
- Autorun feature and help menu integrated into the example firmware
- Output pins match up with FTDI Basic Breakout, Bluetooth Mate, XBee Explorer
- 3.5-16VDC input
- ON-OFF control switch and reset switch

Dimensions: 1.1" x 1.6" (28 x 41mm)

Documents:

- Schematic
- Eagle Files
- Datasheet (ITG-3200)
- Datasheet (ADXL345)
- Datasheet (HMC5883L)
- Python Graphic interface
- AHRS Code
- AHRS/Head-tracker Tutorial (Thanks Peter!)
- GitHub