LORD TECHNICAL NOTE

3DM-GX3®

Using an Hardware Datalogger with Inertial Sensors

Overview

Many inertial applications incorporate dataloggers. The datalogger can take many forms: hardware or software, analog or digital, simple or complex, and/or combinations of all. In this technical note, let's work through a digital datalogger with an OEM (circuit board) form factor. The inertial sensor is connected to the datalogger via an RS-232 communication interface. The inertial sensor is pre-programmed to automatically send data when powered. The datalogger receives the stream of data and continuously records the stream to a data file on a flash memory card. The data file is post-processed with software to extract the actual data values.

Configure the 3DM-GX3®

In our example, we are using a Lord MicroStrain® 3DM-GX3®-45 Miniature GPS-Aided Inertial Navigation System with RS-232 interface and its accompanying MIP Monitor software to configure the inertial sensor.

- Connect the RS-232 communication cable to the host computer as normal and power the inertial sensor.
- Launch MIP Monitor software and establish communication with the inertial sensor as normal.
- Click Settings.
- Click AHRS and the AHRS Settings screen will appear.
- Select Euler Angles and 10 Hz in the AHRS Message Format drop-down.
- Click OK and the AHRS Settings screen will disappear.
- Go to the GPS and NAV Settings screen and insure that no other data quantities are selected for output.
- Click the blue Start Streaming Data icon to begin sampling and outputting Euler Angles at 10 Hz.
- Click Settings.
- Click Save Current Settings and an information window will appear indicating that the current settings will be written to non-volatile memory.
- Click OK and the information window will disappear.
- **Note:** By clicking OK, we have configured the inertial sensor to automatically and continuously output Euler Angles at 10 Hz each time it is powered up.
- Disconnect the power and reconnect the power. Now observe the activity LED and its rapid blinking will indicate that it is now automatically and continuously outputting Euler Angles.

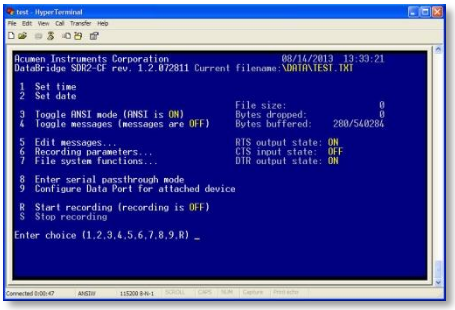
Configure and Operate the Datalogger

In our example we are using an Acumen Instruments Corporation DataBridge SDR2-OEM-CF Serial Data Recorder. The SDR2-OEM-CF has an OEM form factor with standoffs for mounting in a larger system, captures data from any device equipped with a serial port, and stores it in a PC-compatible file on a CompactFlash® card. The datalogger has a second serial port for connection to a host computer and is configured using any terminal program. To configure the datalogger:

Figure 1: *Datalogger*

- Launch a terminal program; we are using Windows Hyperterminal.
- The Connection Description window opens; enter a name for the Connection; we are using 'test'.
- Click OK and the Connect To window opens; select the serial port to which the datalogger is connected; we are on 'COM1'.
- Click OK and the COM1 Properties window opens; select a baud rate of 115200; the datalogger host connection is set to 115200 by default.
- Click OK and the terminal program connects with the datalogger.
- Press any key and the datalogger main menu screen appears.
- Following the datalogger instruction manual:
  - Set time
  - Set date

  - File system functions; in our example, we are naming the data file ‘Test.txt’
  - Configure Data Port for attached device; in our example, we have set ‘115200,n,8,1’, the default settings of the 3DM-GX3®-45
- The datalogger is now ready to start recording.
- Connect the 3DM-GX3®-45 RS-232 communication cable to the datalogger and power the inertial sensor.
- Either select ‘R’ in the datalogger menu to start recording, or push the start/stop button on the datalogger itself.
- The datalogger menu will indicate activity in the communication buffer and in the data file.


test - HyperTerminal
File Edit View Call Transfer Help
Connected 0:00:47 ANSIW 115200 8-N-1 SCROLL CAPS NUM Capture Print echo

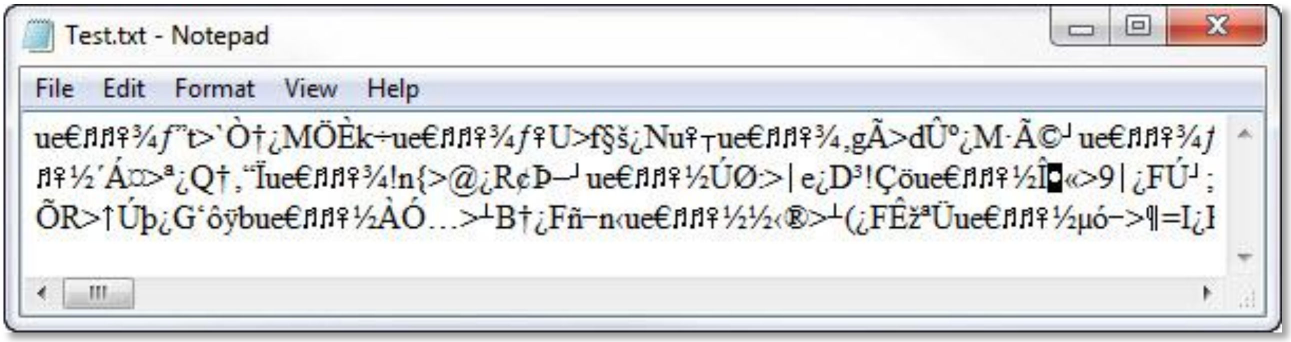
```
Acumen Instruments Corporation                         08/14/2013  13:33:21
DataBridge SDR2-CF rev. 1.2.072811 Current filename:\DATA\TEST.TXT

 1  Set time
 2  Set date
                                          File size:               0
 3  Toggle ANSI mode (ANSI is ON)         Bytes dropped:           0
 4  Toggle messages (messages are OFF)    Bytes buffered:    280/540284

 5  Edit messages...                      RTS output state: ON
 6  Recording parameters...               CTS input state:  OFF
 7  File system functions...              DTR output state: ON

 8  Enter serial passthrough mode
 9  Configure Data Port for attached device

 R  Start recording (recording is OFF)
 S  Stop recording

Enter choice (1,2,3,4,5,6,7,8,9,R) _
```

Figure 2: *Datalogger Main Menu*

- Either select ‘S’ in the datalogger menu to stop recording, or push the start/stop button on the datalogger itself.
- Remove the CompactFlash® card from the datalogger and use the USB card reader shipped with the datalogger to copy the ‘Test.txt’ data file onto a computer for post-processing.

Understanding the Datalogger Data File

- Open Test.txt and you will be presented with a view, as shown in our example figure 3.
- We are looking at a series of ASCII characters which at first sight seem to be gibberish. However, these characters actually represent our data and with a bit of manipulation we will find our Euler Angles.



```
ue€ЯЯ‽¾ƒ˜t>`Ò†¿MÖÈk÷ue€ЯЯ‽¾ƒ‽U>f§š¿Nu‽┬ue€ЯЯ‽¾,gÃ>dÛº¿M·Ã©┘ue€ЯЯ‽¾ƒ
Я‽½´Á⌧>ª¿Q†‚“Ïue€ЯЯ‽¾!n{>@¿R¢Ð─┘ue€ЯЯ‽½ÚØ>|e¿D³!Çöue€ЯЯ‽½Î◘«>9|¿FÚ┘;
ÕR>↑Úþ¿G‘ôÿbue€ЯЯ‽½ÀÓ…>┴B†¿Fñ–n‹ue€ЯЯ‽½½‹®>┴(¿FÊžªÜue€ЯЯ‽½µó–>¶=I¿I
```

Figure 3: *Datalogger File*

- Let’s pull out the first several characters and dissect these further by comparing them with an ASCII chart.
- Open this ASCII chart in a separate browser window to follow along.
- Our first 3 characters are: ue€
- By looking through the ASCII chart, we find the following, as shown in figure 4.

DEC	OCT	HEX	BIN	Symbol	HTML Number	HTML Name	Description
117	165	75	01110101	u	&#117;		Lowercase u
101	145	65	01100101	e	&#101;		Lowercase e
128	200	80	10000000	€	&#128;	&euro;	Euro sign

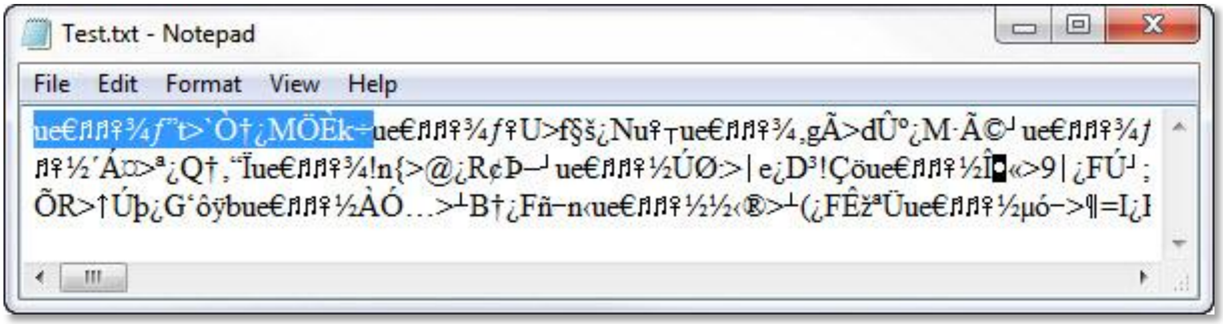
Figure 4: *ASCII table*

- In the Symbol column, we see our characters and in the DEC column, we see that each character has a DECIMAL value, i.e., u represents a value of 117, e represents a value of 101, € represents a value of 128 and so forth. Also note that each symbol has a HEX value. We will come back to this subject in a moment.

Understanding the Inertial Sensor Data Packets

- The 3DM-GX3®-45 inertial sensor outputs its data in ‘packet’ form.
- A data packet contains several characters in it and at the level of the RS-232 communication interface, these characters are referred to as ‘bytes’; a data packet contains several bytes.

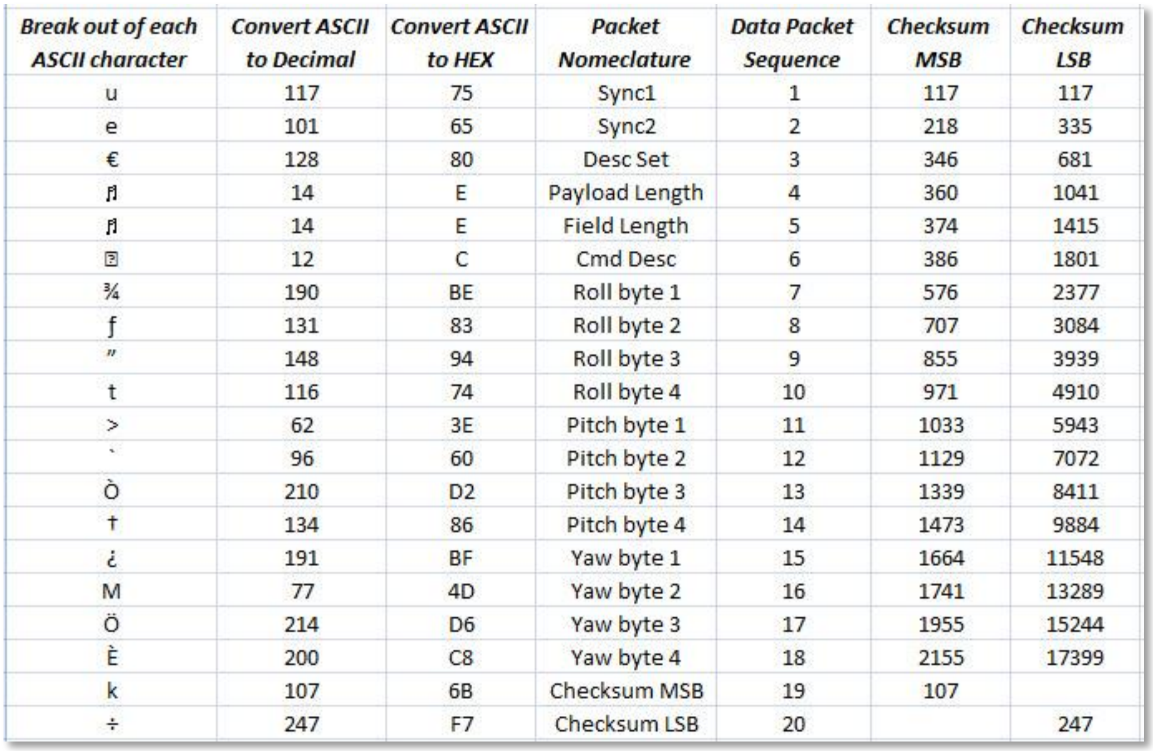
- A detailed discussion of this subject begins on page 11 of the 3DM-GX3®-45 Data Communications Protocol manual. In this discussion we see among other things that each data packet has 2 starting bytes, 2 so-called ‘header’ bytes, being hex 75 and hex 65 which we see are our symbols ‘u’ and ‘e’.
- In the *Configure the 3DM-GX3®* section above, we configured the 3DM-GX3®-45 to continuously output Euler Angles at 10 Hz each time it is powered up. Suffice it to say that our datalogger is recording data packet after data packet after data packet. That is in fact what we are seeing in figure 5.
- Look thorough the string of characters and you will see ‘ue’ pairs indicating the start of the next data packet.
- Let’s parse the first data packet that we see, as highlighted in figure 5.



```
ue€♫♫♀¾ƒ”t>`Ò†¿MÖÈk÷ue€♫♫♀¾ƒ♀U>f§š¿Nu♀┬ue€♫♫♀¾,gÃ>dÛº¿M·Ã©┘ue€♫♫♀¾ƒ
♫♀½´Á□>ª¿Q†‚“Ïue€♫♫♀¾!n{>@¿R¢Ð–┘ue€♫♫♀½ÚØ>|e¿D³!Çöue€♫♫♀½Î■«>9|¿FÚ┘;
ÕR>↑Úþ¿G‘ôÿbue€♫♫♀½ÀÓ…>┴B†¿Fñ–n‹ue€♫♫♀½½‹®>┴(¿FÊžªÜue€♫♫♀½µó–>¶=I¿I
```

Figure 5: *Data packet highlighted*

- In figure 6, our ASCII column breaks out each ASCII character from the data packet, our Convert ASCII to Decimal column shows us the decimal equivalent, our Convert ASCII to Hex column shows the hex equivalent, and our Packet Nomenclature column shows us the role of each byte in the data packet as indicated in the data communications protocol. We also see the checksum calculations which indicate we have a good data packet.



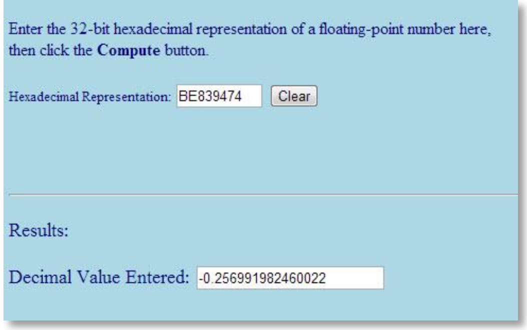
Break out of each ASCII character	*Convert ASCII to Decimal*	*Convert ASCII to HEX*	*Packet Nomeclature*	*Data Packet Sequence*	*Checksum MSB*	*Checksum LSB*
u	117	75	Sync1	1	117	117
e	101	65	Sync2	2	218	335
€	128	80	Desc Set	3	346	681
♫	14	E	Payload Length	4	360	1041
♫	14	E	Field Length	5	374	1415
⍰	12	C	Cmd Desc	6	386	1801
¾	190	BE	Roll byte 1	7	576	2377
ƒ	131	83	Roll byte 2	8	707	3084
”	148	94	Roll byte 3	9	855	3939
t	116	74	Roll byte 4	10	971	4910
>	62	3E	Pitch byte 1	11	1033	5943
`	96	60	Pitch byte 2	12	1129	7072
Ò	210	D2	Pitch byte 3	13	1339	8411
†	134	86	Pitch byte 4	14	1473	9884
¿	191	BF	Yaw byte 1	15	1664	11548
M	77	4D	Yaw byte 2	16	1741	13289
Ö	214	D6	Yaw byte 3	17	1955	15244
È	200	C8	Yaw byte 4	18	2155	17399
k	107	6B	Checksum MSB	19	107	
÷	247	F7	Checksum LSB	20		247

Figure 6: *Data packet breakout*

- Drilling down further, we see in our Packet Nomenclature and Data Packet Sequence columns that our Roll value is derived from bytes 7, 8, 9 and 10. Pitch is derived from bytes 11, 12, 13 and 14. Yaw is derived from bytes 15, 16, 17 and 18. As described on page 14 of the Data Communications Protocol and elsewhere in the manual, these groups of 4 bytes are defined as *single precision IEEE-754 floating point numbers*.

Converting 4 bytes to Roll, Pitch and Yaw

- Here is Wikipedia's definition of IEEE-754 floating point.
- Here is a great site to confirm your IEEE-754 conversions.
- If we run our four ROLL bytes BE 83 94 74 through this calculator, we get a ROLL value of -0.25699 radians as shown in figure 6.
- Radians, of course, can be converted to degrees: -0.25699 radians x 57.2957795 degrees/radian = -14.7 degrees Roll.
- Note: If you are writing your own software, the coding language will typically provide a 4 byte float function to make these conversions.



Enter the 32-bit hexadecimal representation of a floating-point number here, then click the **Compute** button.

Hexadecimal Representation: BE839474

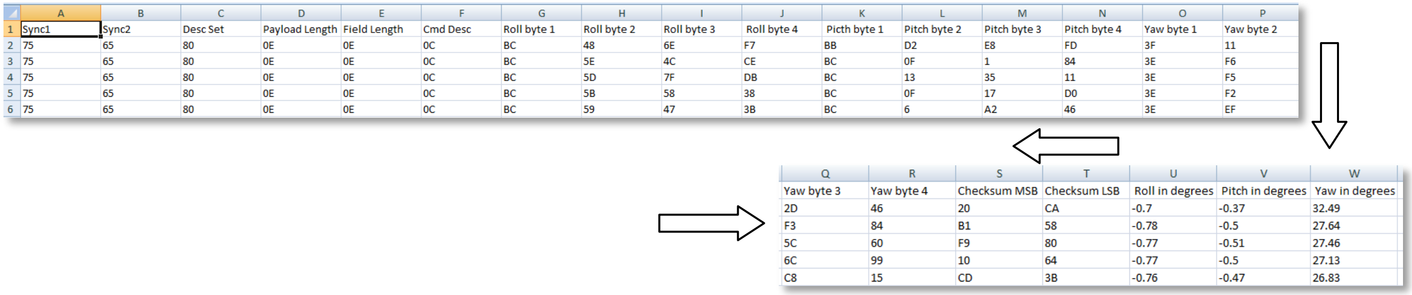
Results:

Decimal Value Entered: -0.256991982460022

Figure 7: *IEEE-754 calculation*

Post-Processing Software

As we have described, the data that is contained in the datalogger data files requires further processing to be meaningful. Given the myriad of applications, dataloggers do not typically provide this post-processing software. In our example we wrote a VB.Net program to parse the data file into a usable format. The program reads in the raw datalogger file, parses out the structured data packets, calculates the roll, pitch and yaw radians, converts the radians to degrees, and outputs the processed data to a new CSV file. The CSV file, of course, imports into any standard analysis program such as Excel, Matlab, etc. Figure 8 shows the CSV after it is imported into Excel. The VB.Net sample code is available for download here.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Sync1	Sync2	Desc Set	Payload Length	Field Length	Cmd Desc	Roll byte 1	Roll byte 2	Roll byte 3	Roll byte 4	Picth byte 1	Pitch byte 2	Pitch byte 3	Pitch byte 4	Yaw byte 1	Yaw byte 2	Yaw byte 3	Yaw byte 4	Checksum MSB	Checksum LSB	Roll in degrees	Pitch in degrees	Yaw in degrees
2	75	65	80	0E	0E	0C	BC	48	6E	F7	BB	D2	E8	FD	3F	11	2D	46	20	CA	-0.7	-0.37	32.49
3	75	65	80	0E	0E	0C	BC	5E	4C	CE	BC	0F	1	84	3E	F6	F3	84	B1	58	-0.78	-0.5	27.64
4	75	65	80	0E	0E	0C	BC	5D	7F	DB	BC	13	35	11	3E	F5	5C	60	F9	80	-0.77	-0.51	27.46
5	75	65	80	0E	0E	0C	BC	5B	58	38	BC	0F	17	D0	3E	F2	6C	99	10	64	-0.77	-0.5	27.13
6	75	65	80	0E	0E	0C	BC	59	47	3B	BC	6	A2	46	3E	EF	C8	15	CD	3B	-0.76	-0.47	26.83

Figure 8: *CSV output showing raw data packet and scaled Euler angles*

Support

We have dealt with dataloggers in very broad brush strokes in this technical note. LORD MicroStrain® support engineers are always available to expand on this subject and support you in any way we can.