

## FXOS8700CQ Evaluation Kit Description and Usage Sensors Solutions Division

Jacques Trichet – Sensors Application Engineer

NOV.14.2014





External Use

Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire, CoWare, the Energy Efficient Solutions logo, Kinetis, mobileGT, PEG, PowerQUICC, Processor Expert, QorQ, Qortiva, SafaAssure, the SafaAssure logo, StanCore, Symphony and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg, U.S. Pat, & Tm. Off. Alrifast, BeeKt, BeeStack, CoreNet, Flexis, Layerscape, MagnV, MXC, Plattom in a Package, QorQ Qonverge, QUICC Engine, Ready Play, SMARTMOS, Tower, TurboLink, UMEMS, Vybrid and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2014 Freescale Semiconductor, Inc.

## Introduction

- This Document first gives a brief description of the Freescale 6-axis sensor evalkit (reference RD4247FXOS8700) and more specifically the DIP16 daughter board that features the magnetometer+accelerometer combo device
- Then we'll give simple information about the protocol that allows communication with RD4247FXOS8700 evalkit and in particular is used by the PC GUI (i.e. XTRINSIC-ACCEL-MAG-SENSOR-FUSION-DEMO.exe software) to achieve continuous and autonomous raw data streaming
- We will illustrate transaction sequences with examples using a simple Serial Terminal Software to connect and communicate with the "Bridge" MCU through the USB serial port. This "bridge", located on the RD4247FXOS8700 kit intermediate Board, is a MC9S08QE08 device that realizes the USB -actually SCI- to I2C conversion and handles dumb sensors management (initialization and data collection)
- Information provided will allow user to control Freescale 6-axis development kit RD4247FXOS8700 and collect raw data without Sensor Toolbox PC software Environment



## Software and Hardware used

#### • RealTerm

For demonstration purpose, we will use RealTerm free Serial Terminal Software available at <a href="http://realterm.sourceforge.net/index.html#downloads\_Download">http://realterm.sourceforge.net/index.html#downloads\_Download</a>. Note that most Terminal Programs (such as Microsoft Window HyperTerminal) can be used as well.

LFSTBUSB

#### **Communication Board** Standard Connector for All RD4247FXOS8700 EVK Sensor Toolbox Pressure Sensor Development Boards The complete evalkit contains: LFSTBUSB communication board 1. Sensor interface board 2. FXOS8700CQ 6-axis sensor 3. development board This hardware (see picture) can be Sensor ordered directly at Interface **USB** Mini-B Board http://www.freescale.com/webapp/sps/site/prod Connector summary.jsp?code=RD4247FXOW8700&tab= Buy Parametric Tab&fromSearch=false Power Switch

Freescale FXOS8700CQ 6-Axis Sensor 3 x 3 x 1.2 mm QFN Single Package Accelerometer and Magnetometer





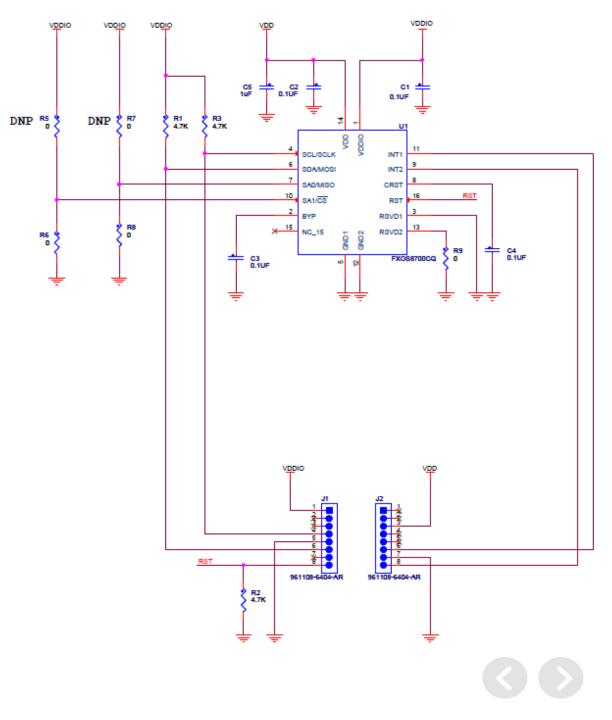
FXOS8700CQ DIP16 Board Description



## FXOS8700CQ DIP16 Board Schematic

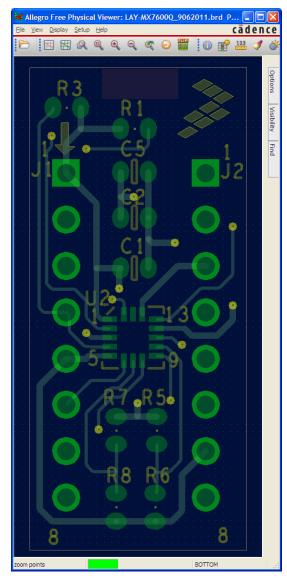
SA0=SA1=0

Device I2C address: 0x4C (pre-production parts) 0x1E (production parts)



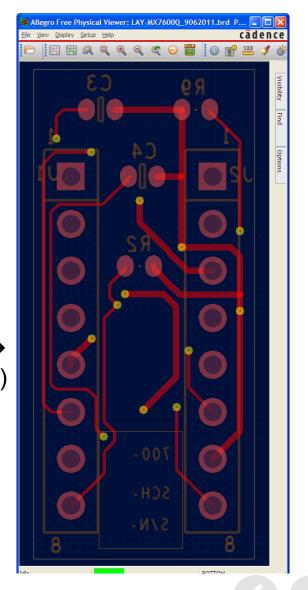


### **FXOS8700CQ DIP16 Board Layout**



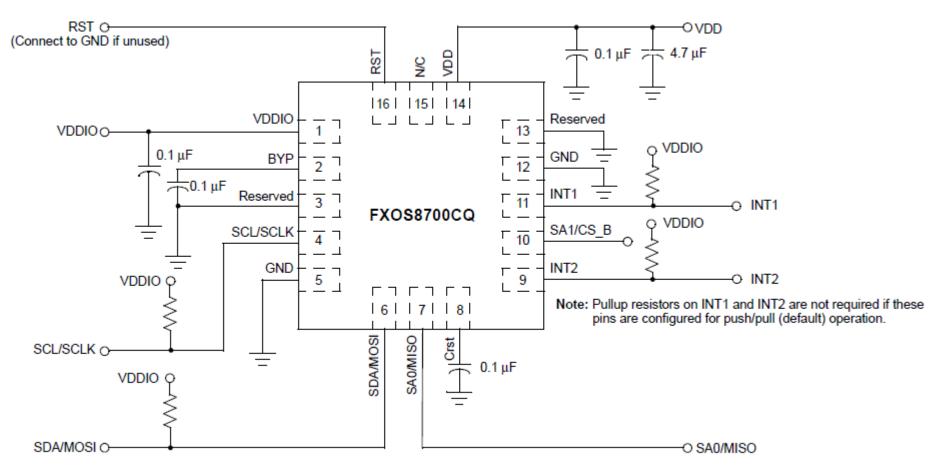
← PCB Top View

PCB Bottom → ("transparency" view)



6

## **FXOS8700CQ** typical application (from Datasheet)



Note: Pullup resistors on SCL/SCLK and SDA/MOSI are not required if the device is operated in SPI Interface mode.

| 7



## **FXOS8700CQ DIP16 Board J1 Connector**

DIP16 Board Pin Number / Name	FXOS8700CQ Pin Number / Name	Description
J1-1 / VDDIO	1 / VDDIO	Digital IOs Supply Rail 1.62 to 3.6V
J1-4 / SCL	4 / SCL	I <sup>2</sup> C Slave Data Line Pulled-up to VDDIO by R3
J1-5 / GND	5,12 / GND1,2	Main Ground
J1-6 / SDA	6 / SDA	I <sup>2</sup> C Slave Data Line Pulled-up to VDDIO by R1
J1-8 / RST	16 / RST	Pulled to GND by R2
J1-2,3,7 / NC	N/A	Unused



## FXOS8700CQ DIP16 Board J2 Connector

DIP16 Board Pin Number / Name	FXOS8700CQ Pin Number / Name	Description
J2-3 / VDD	14 / VDD	Main Supply 1.95 to 3.6V
J2-6 / INT1	11 / INT1	First Interrupt Line
J2-7 / GND	5,12 / GND1,2	Main Ground
J2-8 / INT2	9 / INT2	Second Interrupt Line
J2-1,2,4,5 / NC	N/A	Unused









## RD4247FXOS8700 Communication Protocol Details and Examples





## **Bridge Protocol Basic Commands Description**

The syntax of commands used to access the sensor registers through the bridge is shown below

#### Device Write Register

- Command sent from PC to MCU
  - Byte 0 : SPACE
  - Byte 1 : 'm'
  - Byte 2 : Slave I2C address
  - Byte 3 : Size
  - Byte 4 : Register Addr
  - Byte 5 : Data x size
  - Byte n : CR
- Answer sent from MCU to PC
  - Byte 0 : SPACE
  - Byte 1 : 'M'
  - Byte 2 : CR

#### Device Read Register

- Command sent from PC to MCU
  - Byte 0 : SPACE
  - Byte 1 : 'l' (lower case L)
  - Byte 2 : Slave I2C address
  - Byte 3 : Size
  - Byte 4 : Register Addr
  - Byte 5 : CR
- Answer sent from MCU to PC
  - Byte 0 : SPACE
  - Byte 1 : 'L'
  - Byte 2: Size
  - Byte 3: Register Addr
  - Byte 4: Data x size
  - Byte n: CR

- message header (0x20)
- character command for device Write Reg
- slave I2C address of the device
- number of consecutive bytes to write
- device first register address to write into
- data to be written (size x bytes)
- terminating character (0x0D)
- message header (0x20)
- character answer for device Write Reg
- terminating character (0x0D)
- message header (0x20)
- character command for device Read Reg
- slave I2C address of the device
- number of consecutive bytes to read. Maximum 20.
- device first register address to read from
- terminating character (0x0D)
- message header (0x20)
- character answer for dévice Read Reg
- number of bytes read (should be the same as in command)
- device first register address read (should be the same as in command)
- consecutive data read (size x bytes)
- terminating character (0x0D)

External Use

| 11



## **Bridge Protocol Basic Commands Description**

This slide provides the syntax of some useful commands allowing to check the serial link, guery Bridge FW revision, initiate and close data streaming

#### Handshake

- Command sent from PC to MCU - message header (0x20)
  - Byte 0 : SPACE
  - Byte 1 : 'h'
  - Byte n : CR

Byte 2 : CR

- Answer sent from MCU to PC
  - Byte 0 : SPACE Byte 1 : 'H'

- message header (0x20)

- terminating character (0x0D)

- character answer for Handshake

- character command for Handshake

terminating character (0x0D)

#### Bridge FW information

- Command sent from PC to MCU
  - Byte 0 : SPACE
  - Byte 1 : 'i' (lower case I)
  - Byte 2 : CR
- Answer sent from MCU to PC
  - Byte 0 : SPACE
  - Byte 1 : 'I' (uppercase I)
  - Byte 2-9
  - Byte 10 : CR

- message header (0x20)
- character command for bridge FW information
- terminating character (0x0D)
- message header (0x20)
- character answer for bridge FW information
- bridge FW information details
- terminating character (0x0D)

### Start/Stop 6DOF data streaming

- Command sent from PC to MCU
  - Byte 0 : SPACE
  - Byte 1 : 'd' (lower case D)
     character command for streaming 6-axis device Data
  - Byte 2 : 0x

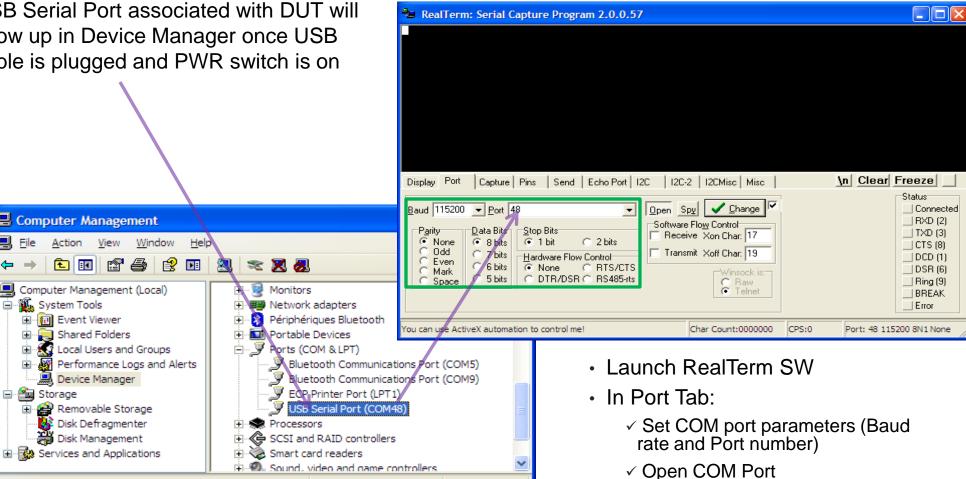
- message header (0x20)
- Byte 3 : CR

- streaming flag (01 to Start, 00 to Stop)
- terminating character (0x0D)
- Answer sent from MCU to PC (See examples in next slides)



## **Configure and Open Serial Com Port**

USB Serial Port associated with DUT will show up in Device Manager once USB cable is plugged and PWR switch is on





📙 Computer Management

主 📧

View

r Star **a** 

Action

System Tools

Event Viewer

Real Shared Folders

🔜 Device Manager

🚔 Removable Storage 🚱 Disk Defragmenter

🚟 Disk Management

Services and Applications

Ē. ۳ų,

File

+

🚊 🊵 Storage

## **Check Serial Port Communication (Handshake)**

 Suggested RealTerm Display tab settings: Display Port Capture Pins - tick Half Duplex to show outgoing characters in green Display As Half Duplex ~ Ascii newLine mode Ansi - Display As <u>Hex[space]</u> to see characters hexadecimal code Invert Data Hexispace Big Endian Hex + Ascii uint8 Data Frames int8 Hex 2 **B**vtes int16 RealTerm: Serial Capture Program 2.0.0.57 Single uint16 Ascii In RealTerm Send tab: 20 68 0D 20 48 0D Binary Nibble Terminal Font tick EOL +CR to automatically add a Float4 carriage return to sent data - tick \n After to automatically change to a new line - type "<space>h" in field (you can use backslash with Display Port Capture Pins Send Echo Port 12C 12C-2 12CMisc Misc \n Clear Freeze Hexadecimal code "\xHH" to send any value, i.e. \x20\x68 sends the Status EUL -∖n Connected Send Numbers Send ASCII h ~ +CR same string) Before | BXD (2) +LF After Send Numbers +CR Send ASCII TXD (3) Hit Send ASCII button (<CR> is +LF | CTS (8) 0 °C LF Repeats 1 🜩 SMBUS 8 🕞 automatically added as EOL) Literal +crc Strip Spaces ] DCD (1) Dump File to Port | DSR (6) The DUT will echo 🗶 Stop Delays 0 **\$** 0 \$ -Send File lc:\temp\capture.txt | Ring (9) ----"<space>H<CR>" as confirmed by BREAK ...... \$ 0 Repeats 1 ŧ Error the vellow codes "20 48 0D" Char Count:6 CPS:0 ou can use ActiveX automation to control me Port: 48 115200 8N1 None



## **DUT Initialization for Data Streaming**

- In order to prepare the device for data streaming, a specific Configuration Sequence must be sent to the DUT after power-up:
  - This commands/answers sequence is reflected in the RealTerm log window below

- Use backslash with Hexadecimal code "\xHH" to send any value. The string associated to 2<sup>nd</sup> command is shown here
- DUT will acknowledge each "Register Write" command by issuing "<space>M<CR>" answer i.e. "20 4D 0D"

RealTerm: Serial Capture Program 2.0.0.70	
20 6D 3C 01 2A 00 0D 20 4D 0D 20 6D 3C 02 5B 1F 20 0D 20 4D 0D 20 6D 3C 02 2D 01 01 0D 20 4D 0D 20 6D 3C 02 2D 01 01 0D 20 4D 0D 20 6D 3C 01 2A 21 0D 20 4D 0D	
Display Port Capture Pins Send Echo Port 12C	Ders Send <u>A</u> SCII
	bers Send A <u>S</u> CII
0 ^C LF Repeats 1 文	🗖 Strip Spaces 🗖
Dump File to Port	
c:\temp\capture.txt 🗨 Sen	d <u>F</u> ile 🗙 Stop De
	<u>R</u> epeats
▲ III	



## **Data Streaming Start/Stop Commands**

Start data Streaming -

- This command will trigger the transfer of raw measured data as19 bytes data packets transmitted continuously at 25Hz rate
- Stop data Streaming-
- Streaming will stop after ongoing packet transfer is complete

RealTerm: Serial Capture Program 2.0.0.70		
20       64       01       00       20       44       0F       02       01       00       D4       FF       6F       01       CB       02       01       38       00         14       3F       08       0D       20       44       0F       02       01       00       D5       FF       71       01       CA       02       01       60       FF         F4       3F       18       0D       20       44       0F       02       01       00       D5       FF       71       01       CA       02       01       60       FF         F4       3F       18       0D       20       44       0F       02       01       00       D5       FF       6F       01       C7       02       01       84       FF         E4       3F       28       0D       20       44       0F       02       01       00       D9       FF       74       01       CB       02       01       78       00         10       3F       28       0D       20       44       0F       02       01       00       D8       FF	\n Clear Freeze ?	
Baud       115200 <ul> <li>Port</li> <li>Parity</li> <li>Parity</li></ul>	Status Disconnect RXD (2) TXD (3) CTS (8) DCD (1) DSR (6) Ring (9) BREAK Error	
You can use ActiveX automation to control me! Char Count:304	CPS:0 Port: 13 115200 8	



## **6DOF Data Packet Format used for Data streaming**

# 20 44 0F 02 01 00 D5 FF 75 01 C6 02 01 34 FF DC 3F 38 0D

### Header - 1 - Magnetometer X,Y,Z data - 2 - Accelerometer X,Y,Z data - <CR>

- A data packet has a fixed length of 19 bytes. Magnetometer data are transmitted 1<sup>st</sup>, followed by Accelerometer data
- the 2 sensors X,Y,Z data are coded as 2's complement 16bits signed integers (left justified if sensor resolution is less than 16 bits)
- last char of data packet (i.e. <CR>) shall not be used as sync byte as it could also be found in sensors data fields. Header is more appropriate as sync pattern. Anyway as packet size and format is invariant, a sync pattern is even not necessary
- <u>Trick:</u> In RealTerm Display tab, select 57 as \_\_\_\_\_ Columns number to align data packets

📲 RealTerm: Serial Capture Program 2.0.0.70				
20       44       0F       02         20       44       0F       02	01       00       D3       FF       72       01       CE       02       0         01       00       D6       FF       70       01       C4       02       0         01       00       D6       FF       70       01       C4       02       0         01       00       D6       FF       70       01       C4       02       0         01       00       DE       FF       6F       01       CC       02       0         01       00       DE       FF       6F       01       CC       02       0         01       00       DE       FF       6F       01       CC       02       0         01       00       D5       FF       75       01       C6       02       0         01       00       D5       FF       6F       01       CC       02       0         01       00       D5       FF       6F       01       CC       02       0         01       00       CE       FF       72       01       C9       02       0         01       00 <td< th=""><th>11       28       FF       F8       3F       18       0D         11       28       FF       F8       3F       18       0D         11       28       FF       F8       3F       18       0D         11       28       FF       D4       3F       10       0D         11       44       FF       D4       3F       10       0D         11       44       FF       D4       3F       10       0D         11       40       FF       EC       3F       18       0D         11       78       FF       E0       3F       00       0D         11       34       FF       DC       3F       38       0D         11       34       FF       DC       3F       38       0D         11       88       00       20       3F       38       0D         11       60       FF       E0       3E       CC       0D         11       60       FF       E0       3E       CC       0D         11       50       FF       D8       3F       24       0D    <!--</th--></th></td<>	11       28       FF       F8       3F       18       0D         11       28       FF       F8       3F       18       0D         11       28       FF       F8       3F       18       0D         11       28       FF       D4       3F       10       0D         11       44       FF       D4       3F       10       0D         11       44       FF       D4       3F       10       0D         11       40       FF       EC       3F       18       0D         11       78       FF       E0       3F       00       0D         11       34       FF       DC       3F       38       0D         11       34       FF       DC       3F       38       0D         11       88       00       20       3F       38       0D         11       60       FF       E0       3E       CC       0D         11       60       FF       E0       3E       CC       0D         11       50       FF       D8       3F       24       0D </th		
Display Port Display As C Ascii C Ansi C Hex[space] C Hex[space] C Hex + Ascii C uint8 C int8 C Hex C uint16 C Ascii C Binary C Nibble C Float4 C Hex CSV	Rows Cols			
You can use ActiveX automation to control me! Char Count:15048				
		F		



## **Summary of Communication Sequence for Data Streaming**

Step #	Command	Answer	Purpose
1	20 68 OD	20 48 OD	Handshake (to verify if communication is ok)
2	20 69 0D	20 49 60 01 10 01 40 02 FF FF 0D	Get Bridge Rev Info (optional)
3	20 6D <b>3C</b> 01 2A 00 0D	20 4D 0D	I2C Addr.= x <b>3C</b> (7-bits Addr.= x1E) Write 00 in Register 2A to set device in Standby Mode
4	20 6D 3C 02 5B 1F 20 0D	20 4D 0D	Select Max OSR and Hybrid Mode with Auto-Increment (Write 1F and 20 in Registers 5B & 5C)
5	20 6D 3C 02 2D 01 01 0D	20 4D 0D	Configure Data-ready interrupt to INT1 pin (Registers 2D & 2E)
6	20 6D 3C 01 2A 21 0D	20 4D 0D	Select active mode with Output data rate of 25Hz (Hybrid Mode)
7	20 64 01 0D	20440F0201FC70FE0E0485020D44E61C32F00D20440F0201FC70FE0E0485020D44E61C32300D	Start data streaming
8	20 64 00 0D	No specific answer	Stop data streaming



## **Complete Communication Log for Data Streaming**

 Note that during data streaming, the bridge MCU is collecting the 6axis measurement from FXOS8700 device according to INT1 interrupt signal (data ready event).

Consequently the communication on the I2C interface is kept minimal

📲 RealTerm: Serial Capture Program 2.0.0.70			
20 68 0D 20 48 0D 20 69 0D 20 49 60 01 10 01 40 02 FF FF 0D 20 6D 3C 01 2A 00 0D 20 4D 0D 20 6D 3C 02 5B 1F 20 0D 20 4D 0D 20 6D 3C 02 5B 1F 20 0D 20 4D 0D 20 6D 3C 02 5B 1F 20 0D 20 4D 0D 20 6D 3C 01 2A 21 0D 20 4D 0D			
20       64       01       00       24       0F       02       01       00       D4       FF       77       01       D2       02       01       44       FF       D0       31         44       0F       02       01       04       PF       78       01       D1       02       01       44       FF       D0       3F       30       0D       20       44       01         D4       FF       77       01       D8       02       01       5C       FF       F8       3F       28       0D       20       44       0F       02       01       00       D4       FI         02       01       5C       FF       F8       3F       28       0D       20       44       0F       02       01       00       D4       FI         02       01       5C       FF       F8       3F       28       0D       20       44       0F       02       01       00       D4       FI         03       5C       FF       78       01       00       D2       04       0F       02       01       00       D2       01	F 02 01 00 F 77 01 D8 1 28 FF F0 ≡ 8 0D 20 44 2 01 00 D2 6 01 CE 02		
01       6C       FF       DC       3F       64       0D       20       44       0F       02       01       00       CF       FF       78       01       D1       02       01       3C       FF       FC       3F       2C       01       02       01       3C       FF       FC       3F       FF       7C       01       02       01       3C       FF       FC       3F       FF       7C       01       02       01       02       01       02       01       02       01       02       01       02       01       02       01       02       01       7C       01       01       02       01       7C       01       02       01       02       01       02 <td< td=""><td>D 20 44 0F 1 00 D0 FF 1 CC 02 01 F F4 3F 18</td></td<>	D 20 44 0F 1 00 D0 FF 1 CC 02 01 F F4 3F 18		
44         0F         02         01         00         0F         77         01         04         02         01         64         FF         DC         3F         18         0D           Display         Port         Capture         Pins         Send         Echo Port         12C         12C-2         12CMisc         Misc         Misc	Clear Freeze ?		
d\x01     ▼     Send Numbers     Send ASCII     EOL +L F     \n     Status       Before     Bsconnect			
Send Numbers Send ASUI +LF	After TXD (3)		
Literal Strip Spaces I +crc DCD (1)			
c:\temp\capture.txt 💽 Send <u>File</u> 🗶 Stop Delays 0	_ DSR (6) _ Ring (9) _ BREAK		
<u>R</u> epeats 1			
You can use ActiveX automation to control me! Char Count:368 CPS	5:0 Port: 13 115.		



## **Communication Sequence Summary**

- Note that Magnetometer raw data can have significant offset that needs to be calibrated (Hard Iron)

Command		Answer f	rom DUT	
20 68 0D	handshake (optional	20 48 0D		
20 6D 3C 01 2A 00 0D	Standby Mode	20 4D 0D		
20 6D 3C 02 5B 1F 20 0D	Initalize DUT 1/2	20 4D 0D	20 4D 0D	
20 6D 3C 02 2D 01 01 0D	Initalize DUT 2/2	20 4D 0D	20 4D 0D	
20 6D 3C 01 2A 21 0D	Active Mode	20 4D 0D	20 4D 0D	
20 64 01 0D	Start data Streaming	20 44 OF 0	20 44 0F 02 01 00 D5 FF 75 01 C6 02 01 34 FF DC 3F 38 0D	
			Mag. X Y Z data + Accel. X Y Z data	
		19 bytes data packet received continuously at 25Hz rate		
20 64 00 0D	Stop data Streaming			
data	decimal	value	unit	
Mag X	213	21.30	μT (scale=0.1μT/LSB)	
Mag Y	-139	-13.90	μΤ	
Mag Z	454	45.40	μΤ	
Accel X	308	18.80	mg (scale= 2g/2^15/LSB)	
Accel Y	-36	-2.20	mg	
Accel Z	16184	987.79	mg	



## Conclusion

- Freescale 6-axis Sensor development kit (RD4247FXOS8700) can easily be configured by sending a few commands via a serial terminal Software
- Once Data streaming is configured and triggered, DUT continuously provides 6DOF data packets to the Host PC through the COM port. Extraction of the 6 measurement values out of a data packet is quite straightforward
- Instructions provided in the document can be applied to more sophisticated programs that are able to use a serial port object such as Matlab script or Microsoft Visual Studio project. This enables automatic collection of the 6DOF raw measurement as well as real time data processing and plotting



