

Connecting the Sensor Platform Kit to the RIoTboard



Above: ROHM Sensor Platform Base Board connected to the RIoTboard

12 January 2015, Revision A



Table of Contents

Copyright and License	2
Revision History	2
Introduction	3
Getting Started	3
Connecting the Sensor Platform UART Output to the RIoTboard	3
Other Notes and Considerations	6

Copyright and License

RIoTboard Open Source Development Platrform by **RIoTboard.org** is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**. All references to the "RIoTboard" are based on a work at <u>RIoTboard.org</u>.

Please note that all references to ROHM's Sensor Platform Kit are also shared under open source guidelines under the GNU General Public License, Version 3. Details can be found at the following link: https://github.com/ROHMUSDC/ROHMSensorPlatformEVK.

Revision History

Date	Description	Revision ID
12 January 2015	First Draft	A



Introduction

The following document was written to provide a brief connection guide and starting point for using ROHM's Sensor Platform Kit with the RIoTboard. This guide assumes that the user has basic functional knowledge of both the Sensor Platform Kit and the Riot Board itself. If this is not correct, please see the following links for other guides and information on these products.

ROHM's Sensor Platform Kit: <u>https://github.com/ROHMUSDC/ROHMSensorPlatformEVK</u>

RIoTboard: http://riotboard.org/

Getting Started

- 1. Initial Setup
 - a. ROHM Sensor Platform Kit
 - i. The Sensor Platform Base Board can be connected to the RIoTboard directly via USB; thus, there are no pre-requisites for this side of the system
 - ii. We would recommend a USB-A to USB-B extender to allow for movement of the sensor base board while connected to the RIoTboard.
 - b. The following are recommended for using the RIoTboard for this guide
 - i. Barrel Connector for Power
 - ii. HDMI Cable and Monitor for Display (Or LVDS)
 - iii. USB Mouse and Keyboard for User Input
 - iv. WiFi Dongle for Internet Connectivity

Connecting the Sensor Platform UART Output to the RIoTboard

- 1. The Sensor Platform Board uses FTDI's FT230XS-R as the on board UART to USB Bridge for data transmission
- 2. FTDI has also built drivers and sample android applications that help to interface onto Android based operating systems.
 - a. Download and install FTDI's "UART Terminal Application" APK from the following website (<u>http://www.ftdichip.com/Android.htm</u>)
 - i. Source Code:
 - 1. <u>http://www.ftdichip.com/Support/SoftwareExamples/Android/J2xxHyp</u> <u>erTerm.7z</u>
 - ii. Direct Link to APK:
 - 1. <u>http://www.ftdichip.com/Support/SoftwareExamples/Android/J2xxHyp</u> <u>erTerm.zip</u>
 - iii. Link to Play Store:
 - 1. <u>https://play.google.com/store/apps/details?id=com.ftdi.j2xx.hyperterm</u> <u>&hl=en</u>



U.S. DESIGN CENTER



- 3. After completing the installation of this application, connect the Sensor platform Base board to the RIoTboard using one of the available USB Ports.
- 4. Open the "FTDI UART Terminal Application". Upon correct connection, you will see two messages flash on the screen:
 - a. "1 port device attached"



U.S. Design Center

FTDI UART Terminal v1.0		
Key Code	Save to File	Send File
Format - Character Port 0; UART Setting - Baudrate:9600 StopBit:1 DataBit:8 Parity:None Flow		
[JKUX61> AccelX_raw = 27, AccelX_scaled = 0.02637[g], AccelY_ KUX61> MagX_raw = 152, MagX_scaled = 22.19200[uT], MagY_raw =	_raw = -480, AccelY_scaled = -0.46875[g], AccelZ_raw = 915, A 289, WagY_scaled = 42.19400[uT], MagZ_raw = -167, MagZ_scale	ccelZ_scaled = 0.89355[g] ed = -24.38200[uT][F
	1 port device attached	
CHAR		White

 b. "Port 0; UART Setting – Baudrate:9600 StopBit:1 DataBit:8 Parity:None FlowControl:CTS/RTS"

				▼⊿ 🗎 6:2
FTDI UART Terminal v1.0				:
Key Code		Save to File		Send File
Format - Character Port 0; UART Setting - Baudrate:9600 StopBit:1	DataBit:8 Parity:None FlowControl:CTS	/RTS		
<pre>[JKINK61>AccelX_row = 26, AccelX_scale (RIK61> lagX_csaled = 23, 0.02637[g], AccelY_row = -479, AccelY (RIK61> lagX_row = 154, lagX_scaled = 22 = 0.02539[g], AccelY_row = -481, AccelY (RIK61> lagX_row = 153, lagX_scaled = 22</pre>	d = 0.02539[g], Accel\raw = -48 ,1920[0T], Mag\raw = 28, Mag <scaled -0.46777[g],="" =="" accel\raw<br="">2.4840[0T], Mag\raw = 285, Mag <realed -0.46973[g],="" =="" accel\raw<br="">2.33800[uT], Mag\raw = 285, Mag</realed></scaled>	2. Accelly_scaled = -0.47070[g]). (*scaled = 42.04800[uT] MagZ_rri wm = 915. AccelZ_scaled = 0.8933 (*scaled = 41.61000[uT], MagZ_rri wm = 915. AccelZ_scaled = 0.8933 (*scaled = 41.61000[uT], MagZ_rri	Accel2_raw = 914, Accel2_s we = -167, WagZ_scaled = -20 [5[g] we = -164, WagZ_scaled = -20 [5[g] w = -166, WagZ_scaled = -20	caled = 0.89258[g] 4.38200[uT] [F [JKUX61> AccelX_raw = 27. AccelX_scal. 3.94400[uT] [F [JKUX61> AccelX_raw = 26. AccelX_scal. 4.23600[uT] [F
	Port 0; UART Setting FlowControl:CTS/R	g - Baudrate:9600 StopBit:1 DataB TS	it:8 Parity:None	
	¢,			
CHAR				Write
		f á		

- 5. Edit the control settings by clicking the "settings button" (three dots on the upper right side of the application) and click "settings". Below the writing portion of the console, you will see additional UART settings
 - a. Adjust these settings to the below options:



U.S. Design Center

K	ey Code		Save to File		Send F	le
rmat - Character rt 0; UART Setting - Baudrate:9	600 StopBit:1 DataBit:8 Parity:I	None FlowControl:CTS/RTS				
T1; Hiraware - Sensor P Revision : REV00 Release date: 0ct 7 201 By : ROHH Senic Standard Seni	Latform EVK 4 16:47:39 onductor USA, LLC ***********************************	AccelY_raw = -117, AccelY_ ngY_raw = 173, UngY_scaled = 9[g], AccelZ_raw = 1001, Ac ngY_raw = 171, UngY_scaled = 2[g], AccelZ_raw = 1001, Ac ngY_raw = 171, UngY_scaled = 5[g], AccelZ_raw = 1001, Ac	scaled = -0.11426[g]. Acc = 25.25800[uT]. WagZ_rew celZ_scaled = 0.97754[g] = 24.9600[uT]. WagZ_rew celZ_scaled = 0.97754[g] = 24.96600[uT]. WagZ_rew celZ_scaled = 0.97754[g]	:elZ_raw = 1002, AccelZ_s = -T20, WagZ_scaled = -T; = -120, WagZ_scaled = -T; = -119, WagZ_scaled = -T;	caled = 0.97852[g] !.52000[uT] [F [JKUX61> Acco !.52000[uT] [F [JKUX61> Acco .37400[uT] [F [JKUX61> Acco	<pre>21X_raw = 4, AccelX_scalec 21X_raw = 3, AccelX_scalec 21X_raw = 3, AccelX_scalec</pre>
		agY_raw = 170, WagY_scaled =	- 24.82000[uT], MagZ_raw	= -119, WagZ_scaled = -17	'.37400[uT][F	
CHAR	_scazeu - 23-30000[u1], w	gg_raw = 1/0. Wagy_scaled =	- 24.82000[uT], MagZ_raw	= -119, WagZ_scaled = -17	, 37400[uT] [F	Write
CHAR Baud Rate	Stop(bit)	Data(bit)	- 24.82000[uT], MagZ_row Parity	Flow Control	2. 37400[uT] [F	Write
CHAR Baud Rate 9600	Stop(bit)	Data(bit)	- 24.82000[uT], MagZ_row Parity None	Flow Control	Port	Writ

- ii. Note: the only major change is the "flow control" set to "None"
- b. Upon the Sensor Reset (pressing the PB on the board), firmware information along with additional sensor platform sensor information should be returned by the sensor platform itself.

Other Notes and Considerations

i.

- 1. Please note that the FTDI UART application is not optimized for some of the special function commands sent via UART (such as the clear line, etc..); thus, you will see the sensor data return in a streaming format.
- 2. From this guide you can now generate your own applications using these sensor solutions by using and understanding the below three topics
 - a. Building an Android App on the RIoTboard
 - i. <u>http://www.element14.com/community/community/designcenter/single-</u> <u>board-computers/riotboard//blog/2014/01/24/publishing-my-first-android-app-</u> <u>on-riotboard</u>
 - b. Source Code for FTDI Console:
 - i. <u>http://www.ftdichip.com/Support/SoftwareExamples/Android/J2xxHyperTerm.</u> 7z
 - c. Source Code for ROHM's Sensor Platform Kit:
 - i. https://github.com/ROHMUSDC/ROHMSensorPlatformEVK