Using Embedded Tools for I2C, SPI, and USB Debugging for the Renesas RX63N RDK

Scott Loring, Engineering Manager, Total Phase
Scott Loring

- Engineering Manager at Total Phase

Education
- B.S. Computer Engineering from the University of California, Riverside
- M.S. Electrical Engineering from Stanford University

Work Experience
- 10 years experience with embedded systems development
- 5 years at Total Phase designing hardware and software development tools
Renesas Technology & Solution Portfolio
Agenda

- Introduction to the Renesas RX63N RDK
- Introduction to Total Phase Development Tools
- Lab
  - Beagle protocol analyzers and Data Center Software
  - Aardvark I2C/SPI Host Adapter and Flash Center Software
  - Development Start Up
  - Bug 1: I2C → LCD
  - Bug 2: Read I2C
- Q&A
Renesas RX63N RDK
Total Phase Development Tools
Total Phase Solutions

By using debugging tools manufactured by Total Phase, you can:

- Debug in real time
- Quickly evaluate embedded systems
- Program EEPROMS and flash memories
- Easily collaborate with colleagues
- Maximize productivity
Aardvark I²C/SPI Host Adapter

- General purpose I²C/SPI master or slave
  - Active communication on the I²C bus up to 800 kHz
  - Active communication up to 8 MHz as an SPI master and up to 4 MHz as an SPI slave
Control Center Software

- Designed to work with the Aardvark I²C/SPI Host Adapter
  - Read and Write I²C/SPI messages
  - XML Batch Script support
  - Built-in Help System
  - Multiple adapter support
  - Windows, Linux, Mac OS X
Beagle I²C/SPI Protocol Analyzer

- Non-intrusively monitor an I²C or SPI bus
  - Interactive Real-Time Display, Filter, and Search
  - Monitors I²C data up to 4 MHz
  - Monitors SPI data up to 24 MHz

Beagle I²C/SPI Protocol Analyzer
Beagle USB 480 Protocol Analyzer

- Non-intrusively monitor high-, full-, and low-speed USB 2.0
  - Interactive Real-Time Display, Filter, and Search
  - Real-time class-level decoding
  - 64 MB on-board hardware buffer
Data Center Software

- Designed to work with the Beagle Protocol Analyzers
  - LiveDisplay
  - LiveFilter
  - LiveSearch
  - 32-bit and 64-bit support
  - Tree View and Block View
  - Windows, Linux, Mac OS X
Lab Introduction
Lab Introduction

Physical Set Up
Lab Introduction

Logical Set Up

I²C Bus

I²C EEPROM

RX63N RDK

Beagle USB 480 Protocol Analyzer

USB

Aardvark I²C/SPI Host Adapter

Beagle I²C/SPI Protocol Analyzer

PC
Beagle Protocol Analyzers
Beagle Protocol Analyzers

- Objective: Introduce the Beagle protocol analyzers

- Task: Set up and start a live capture using the Data Center Software
Beagle Protocol Analyzers

Configuring the Beagle analyzer for use

1. Open the **Data Center Software**.

2. Click **Connect to Analyzer**.

3. Select the I2C/SPI analyzer and Click **OK**.
Beagle Protocol Analyzers

Configuring the Beagle analyzer for use

4. Click **Device Settings**.

5. Change the capture protocol to **I2C**.

6. Set the sampling rate to **50 MHz**.

7. Set the protocol lens to **I2C**.
Beagle Protocol Analyzers

Configuring the Beagle analyzer for use

4. Click **Device Settings**.

5. Change the capture protocol to **I2C**.

6. Set the sampling rate to **50 MHz**.

7. Set the protocol lens to **I2C**.

8. **Start** the capture.
Aardvark I²C/SPI Host Adapter
Aardvark I²C/SPI Host Adapter

Objective: Introduce Aardvark I²C/SPI Host Adapter.

Task: Use the Aardvark adapter and Flash Center Software to read and write from the I²C EEPROM.
Aardvark I²C/SPI Host Adapter

Configuring the Aardvark I²C/SPI Host Adapter for use

1. Open the **Flash Center Software**
2. Click on **Add Adapters**
3. Select Aardvark I²C/SPI Host Adapter
4. Click **Add**
Aardvark I²C/SPI Host Adapter

5. Turn on the **Target Power** button

6. Click on **Choose Target** to specify which part you will be using
Aardvark I²C/SPI Host Adapter

7. Select I²C EEPROM under Device Type
8. Under Manufacturer, select Atmel
9. Under Part Number, choose AT24C02
10. Click OK
Aardvark I²C/SPI Host Adapter

Reading and Erasing the SPI EEPROM Contents

11. Click the **Read Target** icon

12. Click **Erase**

13. Select **OK**

14. Click **Read Target** again
Aardvark $I^2C$/SPI Host Adapter

**Writing Data to the $I^2C$ EEPROM**

15. Click the **Clear** button to clear the local buffer

16. Click into the ASCII editor

17. On the first line, type **abcdefghijklmnop**
Aardvark I²C/SPI Host Adapter

18. Select the **Program and Verify** icon

19. Click **OK** if presented with a warning about exceeding capacity

20. Verify the transactions in Data Center
Development Start Up
Development Start Up

- Objective: Access the project content and application

- Task: Load and set up the lab project using the e²Studio designed by Renesas
Development Start Up

1. Open e²Studio

2. Select Browse in the Workspace Launcher

3. Browse to C:\<PATH_TO_FILES>\e2studioworkspace\TPRX63N.work
Development Start Up

4. Select **Project → Build All**

5. Select **Run → Debug (F11)**

6. Select **Go (F8)**
7. The “Renesas DevCon 2012” message will be displayed on the LCD along with a prompt to reinsert the USB cable.
Development Start Up

Installing the Driver

8. Reinsert the USB cable (near the Ethernet connector)

9. In the Windows ‘Found New HW wizard’, select “**No, not this time**”  
   *(If you are running Windows 7, Select “Browse my computer.”)*

10. Click **Next**
Development Start Up

Installing the Driver

11. Select "Install from a specific location"
    (If you are running Windows 7, select “Let me pick.”).

12. Click Next
Development Start Up

Installing the Driver

13. If asked for the libusb driver, select ‘Browse’ and point to the directory `<PATH>\material\libusb-drivers\x86`. 
Development Start Up

Installing the Driver

14. In Windows Device Manager, check that Windows has enumerated the RX properly. You should have it listed under the “libusb-win32 device” class.
Development Start Up

15. Launch the RX63N LibUSB GUI by double-clicking the launcher

C:\<PATH_TO_FILES>\libusb-gui\launch.bat
Bug 1 – I$^2$C to LCD
Bug 1 – I²C to LCD

- Problem: The EEPROM data is incorrectly displayed on the LCD screen

- Task: Use Total Phase tools to diagnose and solve the problem
Bug 1 – I$^2$C to LCD

1. Click **LibUSB Connect** in the RX63N GUI.
Bug 1 – I²C to LCD

Testing Application Functions

2. Click LED Control

3. Type Aardvark and click Write LCD

4. Click Clear LCD

5. Click I²C to LCD
Spend the next few minutes solving the bug in sending data from the EEPROM to the LCD
Bug 1 – I²C to LCD

**Solution**

- Using the Beagle I²C analyzer, we can quickly see this problem.
  - When we used Flash Center, the Data Center displays this transaction.

| 9 B | SP 50 | Write Transaction | 00 61 62 63 64 65 66 67 68 |
| 9 B | SP 50 | Write Transaction | 08 69 6A 6B 6C 6D 6E 6F 70 |
| 1 B | SP 50 | Write Transaction | 00 |
| 16 B | SP 50 | Read Transaction | 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70* |

- The first byte of data is 00. Remember that 00 is our offset.

- When we click I²C to LCD, the Data Center displays these transactions.

| 1 B | S 50 | Write Transaction | 08 |
| 16 B | SP 50 | Read Transaction | 69 6A 6B 6C 6D 6E 6F 70 FF FF FF FF FF FF FF FF FF FF* |

- Note that the data is being pushed to the LCD starting at offset 08. The correct offset is 00 as shown in the first write transaction.
Bug 1 – I²C to LCD

*Correcting the source code and verifying the solution*

6. Make the corrections to source code in line 254.

```c
253  case USB_LIBUSB_CMD_I2C_LCD:
254  i2c_eeprom_read(0x50, 8, data, 12);
255  lcd_display(LCD_LINE1, "I2C -> LCD");
256  lcd_display(LCD_LINE2, data);
257  break;
```
Bug 1 – I²C to LCD

Correcting the source code and verifying the solution

7. Save and rebuild the project referring to the handout

8. Click Connect in the Application Box

9. Click I2C to LCD to verify the solution.
Bug 2 - Read I²C
Bug 2 – Read I\(^2\)C

- Problem: Data is incorrectly read from the EEPROM

- Task: Use Total Phase tools to diagnose and solve the problem
Bug 2 – Read I²C

Writing and reading from the EEPROM

1. Click **I2C to GUI**

You can see the data is incorrectly displayed in the application.
Spend the next few minutes solving the bug in reading data from the EEPROM.
Bug 2 – Read I\textsuperscript{2}C

*Evaluating the I\textsuperscript{2}C data*

- Using the Beagle analyzers, we can quickly isolate this problem.
  - When we Read I2C, the Data Center displays these transactions.

<table>
<thead>
<tr>
<th>1 B</th>
<th>S</th>
<th>50</th>
<th>Write Transaction 00</th>
</tr>
</thead>
<tbody>
<tr>
<td>17 B</td>
<td>SP</td>
<td>50</td>
<td>Read Transaction 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 FF*</td>
</tr>
</tbody>
</table>

- Using the Details pane, we can see the data is being read correctly from the I2C EEPROM.
Bug 2 – Read I²C

*Evaluating the USB data using the Beagle USB 480 analyzer*

2. Open another Data Center Software

3. Click **Connect to Analyzer**.

4. Select the USB analyzer and Click **OK**.
Bug 2 – Read I²C

Evaluating the USB data using the Beagle USB 480 analyzer

5. Go to **Capture Settings** and adjust the Data Capture Limit to **700 MB**

6. **Start** the Capture
Bug 2 – Read I²C

Reset the device to see the enumeration details

7. Go to the Bus Pane

8. In e²studio click Reset then Go (F8)

9. Reinsert the USB cable

10. RX600 LibUSB Demo should appear in the bus pane
Bug 2 – Read I²C

Filter out extraneous USB data

11. Go to the Filter tab

12. Uncheck Collapsed & SOFs

13. Apply filter

Only data of interest is displayed in the capture log
Spend the next few minutes analyzing the USB data to identify the bug.
Bug 2 – Read I\textsuperscript{2}C

**Evaluating the USB data**

- Using the Beagle USB 480 analyzer, we can diagnose this problem.
  - When we Read I2C, the Data Center displays these input reports.

```
<p>| | | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2B</td>
<td>05</td>
<td>01</td>
<td>OUT</td>
<td>txn</td>
<td></td>
<td></td>
<td>02</td>
<td></td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4B</td>
<td>05</td>
<td>02</td>
<td>IN</td>
<td>txn</td>
<td>[10</td>
<td>POLL]</td>
<td>64</td>
<td>63</td>
<td>62</td>
<td>61</td>
<td>dcba</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2B</td>
<td>05</td>
<td>01</td>
<td>OUT</td>
<td>txn</td>
<td></td>
<td></td>
<td>02</td>
<td></td>
<td>04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4B</td>
<td>05</td>
<td>02</td>
<td>IN</td>
<td>txn</td>
<td></td>
<td></td>
<td>68</td>
<td>67</td>
<td>66</td>
<td>65</td>
<td>hgf e</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2B</td>
<td>05</td>
<td>01</td>
<td>OUT</td>
<td>txn</td>
<td></td>
<td></td>
<td>02</td>
<td></td>
<td>08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4B</td>
<td>05</td>
<td>02</td>
<td>IN</td>
<td>txn</td>
<td></td>
<td></td>
<td>6C</td>
<td>6B</td>
<td>6A</td>
<td>69</td>
<td>lkji</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

- Using the Details pane, we can see the data is being read backwards over USB.
Bug 2 – Read I²C

Correcting the source code and verifying the solution

14. Make the corrections to source code.

```c
232 uint32_t usb_bulk_send_in_packet (uint16_t ip, int16_t fn, uint32_t val) {
233     USB_ER_t err_code;
234     int16_t state;
235
236     static uint8_t data[4];
237     for (int i = 0, s = 0; i < 4; i++, s+=8) {
238         data[i] = (uint8_t)((val >> s) & 0xff);
239     }
240     write(fn, data, 4);
```
Bug 2 – Read I²C

Correcting the source code and verifying the solution

15. Save and rebuild the project referring to the handout

16. Click **Connect** in the GUI

17. Click **I²C to GUI** to verify the solution.

![GUI Screenshot](image-url)
Questions and Answers
Please Provide Your Feedback...

- Please utilize the ‘Guidebook’ application to leave feedback

  ![Guidebook Logo]

  or

- Ask me for the paper feedback form for you to use...