Application Project Guide

Renesas PE-HMI1 Synergy S7 with Clarinox SPP Application

Contents

1.	Introduction	3
2.	Prerequisites	3
3.	Requirements	3
4.	Installation and Importing for e2 Studio	3
4.1.	Setting up Hardware	4
	Importing/Creating the project	
	Configuring the project	
4.4.	Building the Project	. 10
	Running the application	
	Customizing the Application Project	
5.1.	Application Source Files and Purpose	. 19
	Callback functions	
5.3.	Threads	. 20

1. Introduction

This application demonstrates Bluetooth Serial Port Profile (SPP) of ClarinoxBlue stack with Bluetooth Classic running on the Renesas PE-HMI1 Synergy S7 platform. The application provides the use with the ability pair the Renesas PE-HMI1 with nearby Bluetooth devices and to transfer messages between devices.

A video explaining how to run SPP application on Renesas PE-HMI1 Synergy S7 can be found under following link.

http://www.clarinox.com/videos-clxblue_renesas_s7g2

2. Prerequisites

This document describes building the application project on e2 Studio and running it on Renesas PE-HMI1 Synergy S7 platform. This process requires the following prerequisites.

- Installing Renesas e2 studio and SSP Distribution on PC
- Installing "Bluetooth SPP Pro" app on Android mobile device
- Installing Clarinox Debugger tool on PC (Optional)

Installation instructions and the user guide for Clarinox debugger tool can be found in the document "Clarinox Debugger User Manual".

Also, being familiar with running applications on e2 studio and having PE-HMI1 Synergy S7 platform tested for basic functionality would be useful. Users can run some sample applications on the Synergy platform to check its functionality and to be familiar with the process.

3. Requirements

This application has the following hardware requirements.

- Renesas PE-HMI1 Synergy S7 kit
- Smart Phone or mobile device

Installing, building and running the application require the following tools and software to be pre-installed.

- e2 studio (tested with version 5_4_0_015)
- SSP Distribution (tested with version 1.2.0)
- "Bluetooth SPP Pro" Android Mobile App
- Clarinox Debugger (tested with version 3.2.219)

4. Installation and Importing for e2 Studio

This section includes step-by-step process of importing the project and running the application on Renesas PE-HMI1 Synergy S7.

4.1. Setting up Hardware

The hardware setup for running the application is shown in Figure 01. The PE-HMI1 Synergy S7 board can be powered over ethernet as shown in the picture. Also for debugging via J-Link, user can connect the J-Link connection via J12 which is not shown in the below figure.



Figure 01: Renesas PE-HMI1 Synergy S7 Setup for Programming and Debugging

In order to debug the application, user can use Clarinox Debugger tool which comes with a full detailed protocol analyzer when integrated with Wireshark allowing the users to analyze Bluetooth and Wi-Fi messages.

This application uses JLINK interface available on PE-HMI1 Synergy S7 for debugging.

When the board is powered over ethernet, Clarinox Debugger can be configured for the J-Link debug connection via *Tools -> Configuration -> SEGGER JLINK RTT Interface*. Configure the J-Link interface as shown in below screen capture.

neral Configuration	Parameter	Value	
a Command Line Pane	Host Interface	USB	
ssage Browser sic Message Handler	JLink Emulator Serial Number	0x0	
ssage Details Pane	JLink Emulator IP Address	127.0.0.1	
arinox Symbol File Manager	Target Interface	SWD	
P/IP Server RT COM Port	Target Device Name	R7FS7G27H	
P Server	Target Interface Speed	64000	
GGER JLink RTT Interface		0x1ffe0000	
	RTT Control Block Address Range Start		
	RTT Control Block Address Range End	0x2007#fff	
	Target RTT Channel	1	

Figure 02: SEGGER J-Link Interface Configuration on Clarinox Debugger

After setting the configurations if the board is powered on, user can start the J-Link debugger connection to the hardware via *Connection -> Start -> SEGGER JLINK RTT Interface*. User can interact with the application via debugger virtual console. More details on using Clarinox Debugger can be found in the document "Clarinox Debugger User Manual"

4.2. Importing/Creating the project

The structure of the project folder is shown in the following figure.



Figure 03: Project Folder Structure

Following table gives the details of the content of these folders.

Folder	Content
Арі	Clarinox APIs for Bluetooth, WLAN, Common and BSP
Demo	Project files and lib file for SppApp
Framework	Renesas Synergy project S7G2_PE_HMI1 framework
Lib	Clarinox Bluetooth and Softframe libraries
Source	Project source code with BSP (J-link and UART) etc

Table 01: Project Folder Structure

Following steps describe how to import BLE Central and Peripheral application project into e2 Studio workspace.

1. Click on File -> Import -> Existing Projects into Workspace

Import	
elect Create new projects from an archive file or directory.	N
Select an import source:	
type filter text	
 General Archive File CMSIS Pack Existing Projects into Workspace File System HEW Project Preferences Renesas CA78K0R Project Renesas Common Project File C/C++ C/C++ C/C++ C/C++ Executable C/C++ Project Settings Existing Code as Makefile Project Install 	H

Figure 04: Import Existing Project into Workspace

2. Select the root directory of the project and then two projects will appear under "Projects". Select all of them and click Finish. Then two projects named "SppApp" and "S7G2_PE_HMI1" will be loaded into the workspace.

mport Projects			
Select a directory to sear	ch for existing Eclipse projects.		
Select root directory:	D:\SppApplication	•	Browse
🖱 Select archive file:		-	Browse
Projects:			
	D:\SppApplication\FrameWork)		Select All
SppApp (D:\Spp/	Application (Demo)		Deselect All
		6	Refresh
Options Search for nested pro Copy projects into w	orkspace		
Search for nested pro Copy projects into w Hide projects that all	Terrere and the second s		
 Search for nested pro Copy projects into w Hide projects that all Working sets 	ready exist in the workspace		
 Search for nested pro Copy projects into w Hide projects that all 	ready exist in the workspace	*	Select
 Search for nested pro Copy projects into w Hide projects that all Working sets Add project to work 	ready exist in the workspace	*	Select

Figure 05: Locating the Project Root Directory

4.3. Configuring the project In order to build the projects, the path for e2 Studio utilities should be set under project environment variables. Right click on SppApp project and select *properties*. Then edit "PATH" variable under Environment as shown in the following figure. Click on "Edit" button add or modify the existing path.

type filter text	Environment			1	(⇒ = ⇒ = :
 Resource Builders C/C++ Build Build Variables Environment 	Configuration: Debug [/	Active]		▼ Manage C	onfigurations
Logging	Environment variables to s	et			Add
Settings Tool Chain Editor Tools Paths C/C++ General Project References Run/Debug Settings	Variable AMS_KEEP_FILE AMS_LICENSE_PATH CWD PATH PWD	Value S{synergyKeepFile} S{synergyLicenseFile} D:\SppApplication\Dem C:\Program Files (x86)\ D:\SppApplication\Dem	Origin USER: PREFS USER: PREFS BUILD SYSTEM USER: CONFIG BUILD SYSTEM]	Select Edit Delete Undefine
	 Append variables to na Replace native environ 			Restore Defaults	Apply

Figure 06: Edit PATH Variable for SppApp

If the path for utilities folder is already added then check for the correct path, if not add the correct path, eg: "D:\Renesas\e2_studio\Utilities".

Name:		PATH	
Value:		ıs\e2_studio\eclipse;D:\Renesas\e2_studio\Utilities	Variables
ОК	Cancel	1	

Figure 07: Add or Modify PATH Variable

The same should be added under PATH variable for S7G2_PE_HMI1 project as well.

The libraries and preprocessor definitions should already be included under SppApp project's build settings as shown below.



Figure 08: Included Libraries under BleGattApp Build Settings

4.4. Building the Project

The order of building two projects is as follows.

- 1. SppApp
- 2. S7G2_PE_HMI1

In order to build the SppApp project, right click on the project and select "Build project". This should generate SppApp.a library in the Debug folder of the project.

Copy the generated SppApp.a library to the Lib folder in S7G2_PE_HMI1 project which already contains WLAN, Bluetooth, Softframe and WiLink libraries. The Lib folder and the libraries should be included to the project as shown in Figure 09.

Right click on S7G2_PE_HMI1 project and select "Build project" to build the project.

SppApp and S7G2_PE_HMI1should build with few compiler warnings.



Figure 09: Included Libraries for S7G2_PE_HMI1 Project

4.5. Running the application

In order to run the application, the PE-HMI1 Synergy S7 board should be assembled as shown in Figure 01. Power on the board via DEBUG_USB by connecting the micro-USB to PC.

To debug the application right click on S7G2_PE_HMI1 Synergy project and select *Debug* As -> 2 Renesas GDB Hardware Debugging. User can also click on the debug icon on e2 studio to debug the application.

If this project is run for the first time then it will ask for the debug hardware. Click on the J-Link ARM. Then select from the given list of devices as shown below. Then it will start downloading the application on the PE-HMI1 Synergy S7.

R7FS12476	
R7FS12477	_
R7FS12878	=
R7FS3A37A	
R7FS3A7	
R7FS3A77C	
R7FS5D97C	
R7FS5D97E	
R7FS7G2	
R7FS7G27G	
R7FS7G27H	
R7S721000	
R7S721000_DualSPI	
R7S721001	

Figure 10: Select the Device for PE-HMI1 Synergy S7

If Clarinox Debugger is used for debugging the application make sure to start the J-Link debugging connection just after downloading the program on PE-HMI1 Synergy S7. Start Clarinox Debugger connection via *Start -> Connection -> SEGGER JLink RTT Interface*.

When the SPP application starts running on the Renesas PE-HMI1 Synergy S7, a menu will be displayed on the debugger console as shown in the following picture. The selections will provide the user with options to use the SPP application for initialization and termination Bluetooth stack, searching for Bluetooth devices, connecting to Bluetooth devices and using the provided profiles.

ake sure to initialize the Bluetooth stack first.	
nter your selection:	
1. Initialize Bluetooth Stack	
2. Terminate Bluetooth Stack	
3. Connect to a paired device	
4. Wait for an incoming connection request	
5. Search for devices in proximity	
6. Make this device discoverable	
7. Make this device non-discoverable	
8. Go to connected device menu	
9. Delete all paired devices	
10. Display Camera	

Figure 11: Main Menu for SPP Application

Below steps show running of an example scenario with SPP application.

The first step is to initialize the Bluetooth stack. The stack initiates connection to the Controller via UART (or USB or any other method specified) at this point. Below shows the console output when executing this option.



Figure 12: Initializing Bluetooth Stack

Then the user can make the device (PE-HMI1 Synergy S7) discoverable and connectable with the option 4 in the menu. This will make the device appear in Bluetooth scan results on mobile devices.

User can tap on the name appeared on mobile device as "ClxSppTest" to pair with PE-HMI1 Synergy S7 and this will ask to confirm a passkey as shown in Figure 13. The same confirmation procedure will happen on Synergy SK-S7G2 as well.

Once this pairing process is happened ClxSppTest can connect to any paired device at any time by entering the menu option 3.



Figure 13: Synergy SK-S7G2 Pairing Process on Mobile Device

The following figure shows the result on debugger console when the pairing process completed.



Figure 14: Synergy SK-S7G2 with Mobile Device Pairing Process on Debugger Console This example uses "Bluetooth Spp Pro" mobile app to communicate data between two devices. Please note that first of all, Synergy SK-S7G2 Bluetooth device should be made discoverable by entering option 6 in the menu. This will allow mobile app to detect the device.

On the mobile app, select the device (Synergy SK-S7G2 as ClxSppTest) once it appears on the screen and tap on the connect button. Then a sub menu to communicate with the connected device will appear on the debugger console. Tap on the "Byte Stream Mode" button on the mobile app to initiate the communication by opening a message console as shown in the following figure.

Ý 🗈 🔺 🕷 🛜 📶 🖻	6:59 AM	\$ ₽ \$ ₽	7:00 AM 🌵 民 🖬	🗚 🕬 🗊 📶 🖬 7:00 AM
Scan Device scan	CLOSE	📣 Bluetooth spp pro	RESCAN 💦 Bluetooth :	spp pro rescan
ClxSppTest MAC: 00:1B:DC:06:5D:E5 CoD: ff2420 Device Type: BR/EDR Bluetooth	RSSI -50 Bonded	Connect the device: Device name: ClxSppTest Mac addr: 00:1B:DC:06:5D:E5	Connect the de Device name: ClxSpp Mac addr: 00:1B:DC:	Test
Simon's Mac Pro MAC: 18:AF:61:B9:23:D3 CoD: 1f00 Device Type: BR/EDR Bluetooth	RSSI -85 Nothing	Class of device:ff2420 Signal: -50 Type: BR/EDR Bluetooth	Class of device:ff242 Signal: -50 Type: BR/EDR Blueto	0
TLCN130-93B752 MAC: A0:E6:F8:93:B7:52 CoD: 1f00 Device Type: BR/EDR Bluetooth	RSSI -78 Nothing	Bind state: Bonded Service's UUID: 00001101-0000-1000-8000-00805f9b34fb	Bind state: Bonded Service's UUID :	0-8000-00805f9b34fb
MI Band 2 MAC: E1:D2:98:8C:89:E5 CoD: 1f00 Device Type: BR/EDR Bluetooth	RSSI -98 Nothing			
ClxSppTest MAC: 00:1B:DC:06:77:98 CoD: ff2420 Device Type: BR/EDR Bluetooth	RSSI -68 Nothing	Connect		nmunication mode
ClxSpp_Windows MAC: 00:1B:DC:06:5C:27 CoD: ff2420 Device Type: BR/EDR Bluetooth	RSSI -64 Nothing		Keyboard mode	e CMD line mode
Simon's Mac Pro MAC: 41:DD:22:DC:13:35 CoD: 1f00 Device Type: BR/EDR Bluetooth	RSSI -85 Nothing			
Select [110] : 4 Waiting for incoming (Please confirm the pa		ion request "435863" to connect to	"CLARINOX G S4" ?	[yYnN] : y
Authentication complet	ted			
SPP connected to Blue	tooth D	evice.		
Enter your selection:				=
 Connect to SPP Send ASCII Text Send Prebuilt 1 				
 Send Frebuilt 4. Disconnect from 5. Return to prev: 	m SPP			
	ious me	14(6)		

Figure 15: Connecting to ClxSppTest via Bluetooth SPP Pro and the Debugger Console Output Then the user can send some text from ClxSppTest to the mobile device by entering the 2nd option "Send ASCII Text" in the sub menu. User can enter a message with up to 48 characters on the debugger console and this message will be received by the mobile app as shown in the following figures.



Figure 16: Sending a Text Message from ClxSppTest (Synergy SK-S7G2)

User can also send a block of prebuilt test data to find the speed of the connection by entering the 3rd option on ClxSppTest sub menu. This will take few seconds to transmit the data and when it is finished the data will appear on the mobile device and the speed of the connection will be displayed on the debugger console as shown below.

Enter your selection:			
 Connect to SPP Send ASCII Text Send Prebuilt Test Disconnect from SP 			
5. Return to previous			
Select [15] : 3	10 J		
Speed = 131000 Bytes/Sec			_
Enter your selection:			
1. Connect to SPP			
2. Send ASCII Text			
3. Send Prebuilt Test	Data		
4. Disconnect from SI			
5. Return to previous			
o. Revain oo pictica			
Select [15] :			-
	Ý == 🔜 🔺 🗱	🛠 🕄 📶 🖸 7:07 AM	•
	< 🧩 Byte stream mode	CLEAR	
	Txd: 0B Rxd: 43333B	Running: 94s	
	Waiting to receive…		
	Hi This is a message fro	om Synergy S7G2-	
	SK!"#\$%&'()*+,- ./0123456789:;<=>?@ABCDEFG		
	YZ[\]^ `abcdefghijklmnopqr		
	000000000000000000000000000000000000000		
	000000000000000000000000000000000000000	000000000000000000000000000000000000000	



Figure 17: Sending a Block of Prebuilt Test Data from ClxSppTest (Synergy SK-S7G2)

The reverse scenario of sending messages from the mobile device to ClxSppTest can also be achieved if the user types the message on the mobile device and tap on send icon which will then be received by the other side. This scenario is shown in the following figure.

	Ý 📟 🖳 🖬	🛠 🕅 🗊 7:08 AM		
	< 🐉 Byte stream mo	ode clear		
	Txd: 0B Rxd: 433	33B Running: 94s		
	Hi this is a message	from the		
	mobile device !	\triangleright		
	his date	46.00		
	his this	thus 🗸		
	1 2 3 4 5	6 7 8 9 0		
	qwert	y u i o p		
	asdfg	<u>ј пјкі</u>		
		/bnm 💌		
	Sym 💥 Englis	:h(US)		
ter your selection:				
 Connect to SPP Send ASCII Text 				
 Send Ascil lext Send Prebuilt T 				
4. Disconnect from	SPP			
5. Return to previ	ous menu			
elect [15] : Hi t	hig ig a maggage f	rom the mobile devi	.ce !	
	nessaye r.	com one moorre devi		

Figure 18: Sending a Text Message from Mobile Device

Following screen capture shows an example of Bluetooth protocol messages captured in Clarinox Debugger Bluetooth Monitor showing lower level message details when the data is transferred with SPP Application.

Name of Dist in the) 😫 🛃 🕸 🖬 👔 🔍 🛦 🛞 🚺 📕 📮									
	e Details + 4.3	Message Browse	r1 Clarinox	Blue Monitor1 🗙 📕 Clar	rinox SDIO Monit	tor1 Delarinox Tasking Monitor1	Memory Monitor1 Memory Pool	lset1		
			D	Upper Lie Rei Trap						
		00100107.945	4	1 1 (c)		ACL_DATA (LICAP_START)	王 .	B	48 20 18 00 14 00 42 00 06 00 00 00 ++ ++ ++ ++	
		00100107.945	0	<	(L2C)	LOCAP_SD0 (SDP)	2.8	d		
		00:00:07.945	0	[<===]	(SDE)	SUP SERVICESEARCHATTE REQ				
The state back state				1 1 1 1						
Data Indu Lingh 10 Suma PLA ADDR (MASI ADD (M	-00 = BC Flag: Point-To-Point (0)	00100107.945	0	****>	(ope)	SDP_BERVICESEARCHATTE_RSP		y.v6.#6.p	07 00 00 00 79 00 76 36 00 73 36 00 70 09 00 00	
- Source UK-ADDP. Worker,	Data Total Length: 130			1 1 1 1					64 35 6a 35 63 19 01 00 35 03 19 00 01 09 00 05	
ame is building in the interval in the	Connect in frame: 40			1 1 1 1			5		35 03 19 10 02 09 00 06 35 09 09 65 6e 09 00 6e	
Sum Procession Sum P	- Source BD ADDR: Vencer 06:5d:e5 (00:1b:dc:06:5d:e5)			1 1 1 1					05 01 00 09 00 08 08 Cf 09 00 09 35 08 35 06 19	
Destruction D0, ADD, Serving 1, 104/1 (M1 104) Distruction D0, Serving 1, 104/1 (M1 104) Distructio				i i i i					04 10 09 02 04 29 01 09 02 05 09 00 01 00	
Destruction for whene		00100107.945	9	1 [see 5] [(120)	LICAP_SIG (SDF)	5.0 	J	76 00 4a 00 67 00 00 00 79 00 76 36 00 73 36 00	
		Electropic sets			10075	LOS DURA INSTRUME	p.,			
Lether holds Sing (2) Lether holds Sing (2) Lether holds (2)		00100107.946	0		(HCI EV	T) NBR OF COMPLETED PACKETS		Beer	13 05 01 48 08 01 00	
 		00:00:07.947	1		(HCI)	ACL DATA (LOCAP START)	н.	· · · · B. 78 · ·	49 20 08 00 04 00 41 00 35 73 01 cd	
Land Tools Calledon Findel Sol Land Tools Calledon Findel Sol	Last Role Change in Frame: 38	00:00:07.947	0	<	(L2C)				04 00 41 00 3b 73 01 cd	
• Let Mod Change in Forme: 0:0	Current Mode: Active Mode (0)	00100107.947	0	dana l	(RPC)	RECOME UN			35 73 01 ed 03 -6 06 -3 05 35 64 70	
Nature 1 1<		00:00:07.947	ă	>		LOCAP SCO (RECOMMS		0Z+D	08 00 4F 00 03 #F 09 #3 05 3b 8d 70	
Long: 126 Image: 126 <td></td> <td>00:00:07.948</td> <td>1</td> <td>1 1 1>1</td> <td>(HCT)</td> <td>ACL DATA (UNKNOWN)</td> <td>Я.,</td> <td></td> <td>48 00 Co 00 08 00 4f 00 03 ef 09 e3 05 3b 8d 70</td> <td></td>		00:00:07.948	1	1 1 1>1	(HCT)	ACL DATA (UNKNOWN)	Я.,		48 00 Co 00 08 00 4f 00 03 ef 09 e3 05 3b 8d 70	
• Company Market de Carde ()			2			T) NER OF COMPLETED PACKETS		.8		
Convertion Convertion <td></td> <td></td> <td>1</td> <td></td> <td></td> <td>ACL DATA (LICAP START)</td> <td>н.</td> <td>B.J.</td> <td>48 20 0± 00 58 00 01 00 56 07 04 00 42 05 4a 00</td> <td>and the second se</td>			1			ACL DATA (LICAP START)	н.	B.J.	48 20 0± 00 58 00 01 00 56 07 04 00 42 05 4a 00	and the second se
Concern trainer 80 (FOR Step A trainer 80) Descriptions L <thl< th=""> L <thl< th=""> <</thl<></thl<>			ä			LICEP DISCONVECTION PERFONSE		8.3.		
Lanced: University 0000 University 00000 University 000000 Unive		00:00:07,952	1		(HCI)	ACL DATA (UNKNOWN)	X.,	B.J.	48 00 Co 00 08 00 01 00 07 07 04 00 42 00 4a 00	
united of sectors	PSM: SDP (0x0001)		0			T) NER_OF_COMPLETED_PACKETS				
FOL Service Search Mitcle Reports Mitcle Report Mitcle Reports Mitcle Report Mitcle Reports Mitcle Reports Mitcle Reports Mitcle Reports Mit	luetooth SDP Protocol	00100107.954	2			ACL_DATA (LICAP_START)	н.		49 20 0c 00 CH 00 41 00 51 cf 09 c1 05 35 9d as	
Transform [1]	PDLE Service Search Attribute Response (0x07)									
Amount Length 131 Altababa Lai Dy Cart 11		00:00:07.955	1	(<===)	(HCI)	ACL DATA (L2CAP START)	х.	B	40 20 Cc 00 00 00 41 00 01 cf 09 c3 05 3b 0d as	
Attable atta fue Coast 11 		00:00:07.955	0	1 1<1 1		L2CAP_SED (RECOMM)			08 00 41 00 01 ef 09 e3 05 36 8d me	
a reflection in the outer			0	[<] []	(RFC_UT	E) MBC				
a closed will will will be will		00100107.955	0	and a second sec	INPO UD	TSPAN OPIL / KEYPOWY		in the	03 et 09 el 05 35 04 70 00 00 45 00 03 ef 02 el 05 35 04 70	
E De fineret Segurare auf 1915 byte 1 De foneret Segurare auf 1915 byte 1 De fonere		00:00:07.955	0		(HCT HC) MRITE SCAN ENABLE			1a 0c 01 00	
 	E Data Element: Sequence uint16 115 bytes	00:00:07.960	5	1 1 1	(HCI)	ACL DATA (LOCAP START)	ж.	· · · · · · · · · · · · · · · · · · ·	48 20 Co 00 08 00 01 00 02 08 04 00 01 00 4b 00	
- UND = UND Element Size (115) Descriptions of the first size	0011 0 = Data Element Type: Sequence (6)		0	I I Correl I						
Lag December Var Spen 19, ***********************************	- 110 - Data Flament Size: wint16 (6)	00100107.961	1		(BCI SV	T) COMMAND COMPLETE			De 04 01 18 02 00	
in Data Value		00100107.961	0		(HCI HC	WRITE SCAN ENABLE			14 00 01 00	
Note: Image:		00100107.965	5	(****)	(BCI IV	T) COMMAND COMPLETE			0c 01 01 1a 0c 00	
Note V = 1 Lot of any former former nois 1<			1					C.K		
Defendition to provide the to the total and the total a	Continuation State: no (00)		0					· · · · · · · C.K. · · · · ·	Do 00 01 00 03 05 09 00 43 00 45 00 00 00 00 00	
* Distribution Bit is and in the second diversion Bit is and in the second diversion <t< td=""><td>*</td><td>00:00:07.967</td><td>á</td><td></td><td></td><td>ACL DATA (UNKNOWN)</td><td></td><td></td><td>40 00 10 00 0= 00 01 00 03 00 08 00 ++ ++ ++ ++</td><td></td></t<>	*	00:00:07.967	á			ACL DATA (UNKNOWN)			40 00 10 00 0= 00 01 00 03 00 08 00 ++ ++ ++ ++	
Note: I i i i i i i i i i i i i i i i i i i i	>	00:00:07.968	1	1 1 1>1	(BCI)	ACL DATA (UNKNOWN)	8	C.K.	49 00 10 00 Ce 00 01 00 03 06 08 00 ++ ++ ++ ++	
<pre>sign Coll point in the secies depice in to Coll point 'III Provide the secies depice in to Coll point 'III Provide the secies depice in the secies depice</pre>					1000010		*	٣	40 00 0- 00 00 00 00 00 00 00 00 00 00 00	
Lind To Offset "HIPPProve" on the sense device Lind To Offset To HIPPProve" on the sense device Lind To Offset To HIPPProve" on the sense device Prove state time in the sense device Lind To Offset To HIP Device	nsole					▼ ª X Lus	Command Line			
pur slåtlini det som	vering CCN ports on the remote device	657				~ (cr	eateTypeMatchingSule			
Scored Scillage Sciend Scillage Scored Scillage Sciend						00	tVersion			
Concet to FF modDMensages						ap	plyHighlightRule			
Ged ACT Text halp Ged Activity For creationstrikingFile File Spread/Sec Cili of halp (fraction, sectionstrikingFile) File Spread/Sec Cili of halp (fraction, sectionstrikingFile) File Spread/Sec File Spread/SectionstrikingFile) * MOD Dyna//Sec File Spread/SectionstrikingFile * MOD Dyna//SectionstrikingFile File Spread/SectionstrikingFile ConstriktingFile File Spread/SectionstrikingFile) ConstriktingFile File Spread/SectionstrikingFile ConstriktingFile File Spread/SectionstrikingFile) ConstriktingFile File Spread/SectionstrikingFile ConstriktingFile File Spread/SectionstrikingFile Spread/Sectionstris </td <td></td> <td></td> <td></td> <td></td> <td></td> <td>CI</td> <td>eateStringMatchingRule</td> <td></td> <td></td> <td></td>						CI	eateStringMatchingRule			
Rad ("Dealing South Radie creations/statistication Rad ("Dealing South Radie South Rad	Send ASCII Test					he	1p			
Sacara K provision away Coll of Ability (Transition, jame") in profer to get a description of vesh function. J.S.C.B. J. 3 The connon literary (or connon. just tacked). - J.S.C.B. J. J. S.	Send Prebuilt Test Data					cr.	cateLuaMatchingRule			
11.51 r 3 The common libriry 'fei_common.link' Leddel. 1.5002 Byrea/Acc The libriry 'fei_common.link' Leddel. Construct to Byrea The libriry 'fei_common.link' Leddel.	Disconnect from SPP								Construction of a construction of the second s	
> 1002 Byrew/Awa Dyne Saletlays Condet: Los JP Gen ACLT Per Los De La Condet i Dencia La Realizational de la Condet de	Meturn to previous menu					Ca	iii cd.heip("instion_name")	in order to get a d	oscription of esca function.	
your saleChann function i functio						75	e common library 'od_common.	.lua' Loaded.		
your saleChann function i EachighteolMoster f							and the second se	Charles and the second s		
Constitution 10 FG Seed ACCI 2007 - Constitution and Con	 135000 Bytes/sec 					75	function : highlightRoolWe	AGE STR		
Red AVCII Text ol_initia 800 Usersacri tra BPO > ol_initia 800 Statistics to previous ment 3.5.230							function : calculatePool%a	aste		
Jand Polnkill Park Dula Disconnet fun SPP Disconnet fun SPP Disconnet fun Polnkill Park Disconnet fun Disconnet (D										
December Iron 279 bit december 200 bit d							- cd_init.lus END			
Retart to previous menu 2.2.239	Send ASCII Text									
	Send Prebuilt Text Data									
The Difference of the Differen	. Send Prebuilt Test Data Disconnect from SPP					3.	cd.getVeraios() 2,219			
	Send Probuilt Test Data Disconsect from SPP Retarn to previous menu					à,	01.941Veraion() 2.219			

Figure 19: ClarinoxBlue Protocol Monitor on Clarinox Debugger

Please note that a user manual explaining all the functionality of SPP application is available in the project folder.

5. Customizing the Application Project

This section guides the user to exploit the complete functionality of the system.

5.1. Application Source Files and Purpose

Clarinox IoT Application source files in "source" folder are provided under three categories;

- 1. Clarinox wireless application source files
- 2. Renesas Synergy platform support files
- 3. Third party interface files

5.1.1. Clarinox wireless application source files

In the first category, there are Bluetooth and Bluetooth low energy application files provided.

Main.cpp file which provides the Bluetooth stack configuration and initialization functionality. In addition, a console based simple menu allows basic inquiry, inquiry scan and connection functionalities.

SPP.cpp file provides the creation, deletion of the SPP profile in addition to a simple SPP menu. The menu provides connect/disconnect and send/receive functions. In the same file, a callback function is provided to handle events delivered by Clarinox middleware.

5.1.2. Renesas Synergy platform support files

BspOs.cpp file provides the Renesas platform ThreadX RTOS interface functionality.

Bsp.cpp file provides memory pool setup for Clarinox wireless components. In addition, terminal input and console print functionalities can be configured in this file. Platform specific setup of the Bluetooth configuration parameters are also set as part of the Board Support Package (BSP) initialization. Other Renesas Synergy Starter Kit or HMI board specific hardware settings can be found in this file.

5.1.3. Third party interface files

Segger JLink/RTT files are used to provide a back channel for connecting the Renesas Synergy platforms to PC based Clarinox debugger. An alternative mechanism is to use of UART interface.

5.2. Callback functions

In the Spp.cpp file, "stackMessageHandler" callback function is provided to handle Bluetooth related events delivered by Clarinox middleware. Any events raised by GAP and SPP profiles cause this call-back function executed with the associated event and parameters.

Users can customize this callback function to perform a task based on the type of indication. For an example, in SPP Application when there is an Indication of a SPP connection, user can turn off discoverability and connectability features to save power.

5.3. Threads

Threads are dynamically created as required, e.g. if the Bluetooth stack is started, then a thread is created and the associated scheduler is run on this thread. Thread priorities and Thread stack sizes are provided as part of the board support package (in Bsp.cpp file).

Configurable items are set by using "clxConfigInitIntegerParam" function call, an example is shown as follows;

```
clxConfigInitIntegerParam(&linkRequestTimeout, "LinkRequestTimeout", 16000,
configList); /* Link request timeout is set to 16 * 1.25 seconds */
```