

Zebra FX Series Embedded C/CPP SDK User Guide Linux

Version 1.0.1



ABOUT THIS GUIDE

Table of Content

1.0	Introduction
2.0	Revision History4
3.0	Pre-Requisites
4.0	SDK Install6
4.1	Install C/C++ SDK debian package (Ubuntu 16.04)6
4.2	Install Native C/C++ SDK tar file (Fedora/Ubuntu)7
5.0	Starting C/C++ SDK
6.0	C/C++ Sample Application Build and Debug13
6.1	Building C/C++ Executable Binary File14
6.2	Debug Embedded RFID C/C++ Application17
6.3	Setup RFID C Remote Debug Configuration18
7.0	Embedded RFIDSample4App C/C++ Application from scratch
7.1	Creating a Workspace
7.2	Creating an Embedded RFIDSample4App C/C++ Project35
7.3	Adding Source Files to Embedded RFIDSample4App C/C++ Project42
7.4	Setup Cross Compiler and Library Environment for Embedded Native C/C++ Project46
7.4	Build/Debug Embedded Native RFIDSample4App C/C++ Project58
8.0 Cr	eate Start and Stop Scripts of C/C++ Installation Package59
9.0 En	bedded Application Installation Package Creation60
9.1	Embedded application package creation60
9.2	Installation and Removal of application package On RFID reader using UI



INTRODUCTION

The 'Zebra FXSeries Embedded C/CPP SDK User Guide Linux' describes the detailed steps about how to use the FX Series Embedded native C/CPP SDK to develop the RFID sample application from scratch, debugging it and packaging it as debian package using Eclipse IDE based on Ubuntu 16.04/Fedora 27 (64 bits, x86) host and executing the RFID sample application in target 'FX9600/FX7500' Readers.

This user guide describes the following:

- Zebra Native C/C++ SDK in general describes how to create, build, and debug an embedded C/C++ application.
- Embedded Sample RFID C/C++ Application from scratch, create, build and debug and how to create debian package, how to create Start/Stop script files for the deployment of debian package and install it through Web interface of target 'Reader FX9600/FX7500' from Ubuntu 16.04/Fedora 27 64bit x86 Host.
- Debian packaging of embedded C/C++ Linux SDK package for newer version.
 Note: Uninstall any older Zebra SDK if installed on Linux host machine.



REVISION HISTORY

REV	DESCRIPTION	DATE	AUTHOR
1.0	Steps and procedure to develop, debug and package	04-Jan-	
	embedded application for Zebra C/C++ sample application	2019	
1.1	Updated steps with added gdb support	09-Apr-	
	 Modified steps for strict host key checking SCP for gdb file transfer 	2019	
1.2	Added C++ steps	03-	
	Added scp key for Ubuntu server	May-	
	Added Debian package creation	2019	
1.3	Added Debian package installation process	24-	
		May-	
		2019	
1.4	 dos2unix conversion executed on script files in debian 	23-	
	installables	June-	
	 Updated the SDK file names in new format 	2019	
	 Corrected hyphens to underscore in section 9 for basicdebtest-2.0-1 		
	 Java 8 installation added in prerequisites 		
1.5	Modified the document in removable of repeated	30-	
	build/debug steps in section 7.	April-	
	• Section 6 updated for C/C++ application compiler's include path, library path, libraries, compiler/linker flags.	2020	
	 Modified c/c++ application projects of SDK in single workspace instead of individual workspace and user guide contents are updated for same. 		
	 Procedure to execute RFID sample application at RFID reader. 		



PRE-REQUISITES

- Host Machine running with Ubuntu 16.04/Fedora 27 (64-bits x86)
- Host Machine with minimum of 8GB RAM (16GB recommended) preferred with 40GB free space, Intel Core i7 CPU
- Installation files:

Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux_V1.0.1.tar.gz

(tarred/compressed file) or

ZebraFXSeriesEmbeddedSDKCCPPLinux_1.0.1.deb (debian package)

provided by Zebra.

- Target Machine 'RFID Reader FX Series FX9600/FX7500' with firmware Version 3.x.x or higher
- Ensure that both target/host service port (22) is up
- Ensure latest Java (java 8) is installed in the host
- RFID API documentation (RFID3_SDK_C_Help.chm) in the doc folder can be viewed using kchmviewer. This can be installed as below.

Ubuntu 16.04:

sudo apt-get update

sudo apt-get install kchmviewer

Fedora:

yum install kchmviewer-qt



C/C++ SDK INSTALL

4.0 SDK Install

This section describes the steps involved for installing Zebra Native C/C++ SDK on Linux host Ubuntu 16.04 machine.

4.1 Install C/C++ SDK debian package (Ubuntu 16.04)

Note: Debian package installation is supported only with Ubuntu-Linux Host machine used is Ubuntu-16.04

Download the deb file "ZebraFXSeriesEmbeddedSDKCCPPLinux_1.0.1.deb" from Zebra ftp site.

Install Command

"sudo dpkg -i ZebraFXSeriesEmbeddedSDKCCPPLinux_1.0.1.deb"

Once installation is done, SDK will be installed in

/usr/share/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux directory In Debian Zebra installed packages above path will become the default installation path

Using GUI browse to the new path

/usr/share/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux

Jump to Figure 4 in the section below and follow the same steps thereafter considering /usr/share/Zebra-FXSeries-Embedded-Native-C-SDK-Linux as the default install-path for Debian Zebra package installations only.

Note:

For **removal** of Debian zebra C/C++ package is required, then come out of the installation path (/usr/share/Zebra-FXSeries..../) into some other directory and then use command

• "sudo dpkg -r zebrafxseriesembeddedsdkccpplinux"

This will remove the installed Debian Zebra C/C++ package from the default path.

Issue while installing deb package

Sometimes users may face the lock problem while trying to install using Debian package.

dpkg: error: dpkg status database is locked by another process

Solution to unlock and install

First run:

lsof /var/lib/dpkg/lock

Then make sure that process isn't running: ps cax | grep PID If it is running: kill PID #wait kill -9 PID



Make sure process is done: ps cax | grep PID Then remove the lock file: sudo rm /var/lib/dpkg/lock Let dpkg fix itself: sudo dpkg --configure -a After this dpkg installation should work fine

4.2 Install Native C/C++ SDK tar file (Fedora/Ubuntu)

Tar file can be installed in Ubuntu/Fedora as follows.

Copy the Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux_V1.0.1.tar.gz file provided by Zebra in any of the host machine directory; this will be the install base directory (i.e. <installation-path>). Extract the tar file as shown below:

Figure 1: Tar File



Right-click on the tar file, and select Extract Here as shown in Figure 2.



Figure 2: Untar SDK

<	> (•						٩		Ξ
0	Recent									
奋	Home			agention of						
	Desktop						00			
D	Documents	backup	Desktop	Documents	Downloads	hidden	Music			
÷	Downloads									
93	Music				t -	BB				
ø	Pictures				a Kai					
×	Videos	Pictures	Public	snap	Templates	Videos	Zebra-FXSeries-			
0	Trash						SDK-C-CPP_Linux			
+	Other Locations		Open With Archive Manage Open With Other Application	r Ctrl+O						
			Cut	Ctrl+X			(\circ)			
			Сору	Ctrl+C	a state and a second second					
		模板	Move to Copy to	. 19492	run.do_install	serialwrite_pallet_1.	ubuntu-16.04.1- desktop-amd64.iso			
			Move to Trash	Delete						
			Rename	F2						
		tar.a	Send to Extract Here							
		Zebra-FXS	Revert to Previous Version							
		Embedded-I SDK-C-CPP Lini	Properties	Ctrl+I						
		V1.0.4.tar.gz								
					"Zebra-EXSeries-	Embedded-Native-SDK-C-	CPP Linux V1.0.4 tar.gz"	selecte	1 (1.0	GB)

After successful extraction, the folder will be visible as shown below. Double-click the folder to see that the respective folders are present.

Figure 3: Untarred Directory

< > 4 12 Home	*						a		
Recent Home Desktop Documents	backup	Desktop	Documents	Downloads	hidden	Music			
Downloads Music Pictures Videos Trash	Pictures	Public	snap	Templates	Videos	Zebra-FXSeries- Embedded-Native- SDK-C-CPP_Linux			
+ Other Locations	模板	Examples	log.do_install.19492	run.do_install	serialwrite_pallet_1.	ubuntu-16.04.1- desktop-amd64.iso			
	Zebra-FXSeries- Ember-GPF-Linux V1.0.4.tar.gz								
				"Zebra-FXSerie	s-Embedded-Native-SDK-	C-CPP_Linux" selected (co	ntainin	g 9 iten	ns)

Double-click on the untarred folder 'Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux' Verify that required directories are available as per following Figure 4.



Figure 4: SDK Base Directory

< > ← ☆ Home	Zebra-FXSeriesK-C-	CPP_Linux >					৹ ः ≡
@ Recent			· · · · · · · · · · · · · · · · · · ·				
12 Home							
Desktop							
D Documents	buildPackages	doc	eclipse	java	RFID_C_API	rootfs	
🕹 Downloads							
Ja Music			Harden and Andrew Party				
Pictures			HALF CARACTER				
🛏 Videos	samples	tools	Version.txt				
🗇 Trash							
+ Other Locations							

Double-click the eclipse directory to verify that required files are available as per Figure 5.

Figure 5: Eclipse Directory

< > < 🏠 Home	Zebra-FXSeriesPP_Li	nux_V1.0.4 eclipse	•				٩	
⊙ Recent ☆ Home ■ Desktop								
Documents Documents Documents Documents Documents Documents Music Documents Pictures Videos Trash	artifacts xml	eclipse	eclipse ini		plugins	readme		
+ Other Locations		Lugse	conservation	ConApri				

Note:

Once installation is done, in case of non-Debian (using tar file) default installation, SDK will be installed in, **<installation-path>/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux** folder.



STARTING SDK

5.0 Starting C/C++ SDK

To start the SDK double-click or right-click and select 'Run' on the eclipse executable file inside eclipse directory as shown in Figure 6

Figure 6: Right-click on the Eclipse Executable File



The following eclipse will pop-up as per Figure 7 and later workspace window will be showcased as per Figure 8.

Figure 7: Eclipse Screen





As shown in Figure 8, click on the Browse button to select the workspace directory.

Figure 8: Workspace Popup

8 Eclipse Launcher						
Select a directory as workspace Eclipse IDE uses the workspace directory to store its preferences and development artifacts.						
Workspace: //home/user/eclipse-workspace	▼ Browse					
Use this as the default and do not ask again	Cancel Launch					

Select the workspace directory path as shown in Figure 9 below.

Select till C or C++ in case of C or C++ application respectively.

About default Eclipse Projects: Both debian package (i.e.

ZebraFXSeriesEmbeddedSDKCCPPLinux_1.0.4.deb) and tarred/compressed file (i.e. Zebra-FXSeries-Embedded-SDK-C-CPP_Linux_V1.0.4.tar.gz) contains C application project as 'RFIDSample4App-C' and C++ application project as 'RFIDSample4App-CPP' under Eclipse workspace at folder '<installation-path>/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/samples/workspace'.

Figure 9: Select Workspace Directory

8	Select Work	spac	e Directory			
0	Recent		Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux_V1.0.1 samples work	space	F	C 7
â	Home	Na		-	Size	Modified
Ginn	Desktop		RemoteSystemsTempFiles			22 Jan
D	Documents		RFIDSample4App-C			22 Jan
-	Downloads		RFIDSample4App-CPP			22 Jan
	Music					
00	Music					
0	Pictures					
-	Videos					
Gene	eclipse					
	Other Location					
Sele	ect the workspace	e dir	ectory to use.			
				Cance		OK



Select the workspace of

'[install-path]'/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/workspace for C/C++ applications and click OK.

After selection of workspace directory, click on Launch button as shown in figure 10.

Figure 10: Eclipse Launch

8 Eclipse Launcher						
Select a directory as workspace Eclipse IDE uses the workspace directory to store its preferences and development artifacts.						
Workspace:	/home/guest/Zebra-FXSeries-Embedded-Native-SDK	•	Browse			
🗆 Use this a	s the default and do not ask again					
			Cancel			

Once Eclipse opens up, please click on Project Explorer Tab as per Figure 11. The Figure 11, shows both C application project 'RFIDSample4App-C' and C++ application project 'RFIDSample4App-CPP'. The following steps like build, debug, packaging, deployment are based on C application project 'RFIDSample4App-C' is as well as applicable to C++ application project 'RFIDSample4App-CPP'.

Figure 11: RFIDSample4Appp-C/ RFIDSample4Appp-CPP Projects





C/C++ SAMPLE APPLICATION BUILD AND DEBUG

6.0 C/C++ Sample Application Build and Debug

This section describes build and debugging steps for C application project 'RFIDSample4App-C'.

Note: If default perspective is not C/C++ in eclipse, then we can enable by 2 methods

Method 1:

Click on the following icon in eclipse.

Figure 12: C/C++ Perspective Icon



Method 2:

Click on open perspective.

Figure 13: Open Perspective Icon



Select C/C++ (default) and click Open.

Figure 14: C/C++ Selection View



😣 🗉 Open Perspective	
🖥 C/C++ (default)	
Debug Docker Tooling GDB Trace Git Git Control Con	
	Cancel Open

6.1 Building C/C++ Executable Binary File

This section explains the steps on how to clean and build c application binary executable 'RFIDSample4App-C'. Select the project name 'RFIDSample4App-C' and go to menu 'Project' item and select 'Clean'.

Figure 15: Project Clean View

🐔 🎋 🔳 🎋 Debug 🗸 🖻 RFIDSa	Open Project 📩 🔽 🐨 🗟 🥻 New Connection
Image: Arrow of the second	Build All Build Project RFIDSample4App.cpp Set Cleap ntHandling = false; Suild Automatically C/C++Index Name[260]; Name[260];
	<pre>rt = 0; CONTROL singulationControl; 9 extern ANTENNA_INFO g_antennaInfo; 10 11 void InventoryFilterOption(RFID_HANDLE32 readerHandle); 12 void Createmenu(RFID_HANDLE32 readerHandle); 13 void ConfigurationMenu(RFID_HANDLE32 readerHandle); 14 void InventoryMenu(RFID_HANDLE32 readerHandle); 15 void AccessMenu(RFID_HANDLE32 readerHandle); 16</pre>
	<pre>17 #ifdef linux 18 int main(int argc, char* argv[]) 19 #else 20⊖int _tmain(int argc, wchar_t* argv[]) 21 #endif 22 {</pre>



Note: Disable "Start a build immediately" checkbox. Click on Clean button in clean popup which appears as per figure 16.



Figure 16: Project Clean Popup View

Select Project (RFIDSample4App-C) and go to Project menu and select Build Project.

Figure 17: Build Project View



🐔 🔯 🔳 🔅 Debug	✓ C RFIDSa		Dpen Project		📬 🖛 🔚 🖬 🖬 🖆 New Col
* • O • G • @@@ ~ !	🛛 🖙 🤣 👻 😽 👻 🏷				
* • • • • • • • • •			Build <u>A</u> ll		
눱 Project Explorer 🛛		E	<u>a</u> uild Project		RFIDSample4App.cpp ⊠
🕨 🚅 RFIDSample4App-C		E	Build <u>W</u> orking Set	*	
🕨 🚰 RFIDSample4App-CPP		C	:lea <u>n</u>		ntHandling = false ;
		E	Build Auto <u>m</u> aticall	y	
		¢	C/C++ <u>I</u> ndex	►	Name[260]; rt = 0;
		<u> </u>	<u>Properties</u>		CONTROL singulationCor
		9	extern ANTENNA	_INFO	5 g_antennaInfo;
		10	void Inventorv	Filte	rOption(RFID HANDLE32
		12	void Createmen	u(RFI	D HANDLE32 readerHandl
		13	void Configura	tion№	lenu(RFID_HANDLE32 read
		14	void Inventory	Menu (RFID_HANDLE32 readerHa
		15	vold AccessMen	u(RF1	D_HANDLE32 readerHandl
		10			

After build the results are shown in the Console tab.

Figure 18: Project Details Window View

★ ■ ★ Debug ✓ E RFID: ★ ● ★ ●	Sample4App-C Debug 🖂 🔅	📑 🔻 🔚 🕼 🔜 💒 New Col	nnection 🖓 🕅 🕅 🖶 🖜 🖜	· ⊑ ∕ (▶
** • • • • • • • • • • • • • • • • • •	<pre>Console % else</pre>	<pre> RFIDSample4App.cpp Ar EventHandling = false; ostName[260]; rPort = 0; ON_CONTROL_singulationCor NF0 g_antennaInf0; lterOption(RFID_HANDLE32 readerHandl onMenu(RFID_HANDLE32 readerHandl onMenu(RFID_HANDLE32 readerHandl c, char* argv[]) rgc, wchar_t* argv[]) rgc, wchar_t* argv[]) argc == 3) == 1) py(hostName, L"localhost" erPort = 0; rs Problems Executat ample4App-C] nueabi_gcc -L./././.// et: RFIDSample4App-C d. 0 errors</pre>	<pre>htrol; readerHandle); le); lerHandle); andle); le); '); '); bles Debugger Console ① Memo /RFID_C_API/lib -L///ro s. (took 1s.702ms)</pre>	<pre>◊>= Varia ⋈ v>= Varia ⋈ v v v v v v v v v v v v v</pre>

刹市• ZEBRA

6.2 Debug Embedded RFID C/C++ Application

This section explains detailed steps on debugging RFID sample C application. In the Project Explorer view, Right-click on the Project name RFIDSample4App-C and click on Debug As -> Debug Configurations as shown in the figure 19



🐔 掾 🔳 🎋 Debi	New	•	🔅 📑 🔻 🗟 💀 🚅 New Connection	V N 📲 - I 👁 -		= N 3. 7).e[i→ =5;	*
* - 0 - 9 - 1000	Go Into						Quick A	ccess 🛛 🖻 😭
 Project Explorer ³ ♣ RFIDSample4App ♣ RFIDSample4App 	Open in New Window Show in Local Terminal In Copy Im Paste X Delete Remove from Context Source Move Rename Im Import Build Project Clean Project Close Project Close Unrelated Projects Build Targets Index Build Configurations Show in Remote Systems view	Ctrl+C Ctrl+V F2 F5	<pre> ■ RFIDSample4App.cpp ⊠ on.h" 32EventHandling = false; hostName[260]; derPort = 0; 110N_CONTROL singulationControl; INFO g_antennaInfo; FilterOption(RFID_HANDLE32 readerHandle); u(RFID_HANDLE32 readerHandle); u</pre>		(x)= Varia 😫	♥ Brea	M Expr	A Modu
	Profiling Tools Validate D Run As	•	ters 🖹 Problems 🕥 Executables 🖼 Debugg Sample4App-C]	er Console 🚺 Memor	ry 🕹	û 🔁 📰	<u>a</u> = 12	1 🖻 🔻 📑 🔻
중 RFIDSample4App-C	 Debug As Profile Ac Restore from Local History Run C/C++ Code Analysis Team 	:	1 C/C++ Container Application 1/Li 2 Local C/C++ Application 2. 3 Remote Application 2.70 Debug Configurations 1.00	b -L///ro 2ms)	otfs/usr/lib) -L/.	.//	rootfs/lib

刹市• ZEBRA

6.3 Setup RFID C Remote Debug Configuration.

This section explains the steps on how to setup remote debug configuration on target 'Reader FX9600/FX7500'.

RFIDSample4App-C binary file built on Linux host machine from eclipse will be transferred to target's '/apps' directory

To connect to target and transfer the built binary, new connection should be made between target and host machine.

Follow the below steps to create new configuration as in figure 20.

Figure 20: Debug New Configuration.





Follow the below steps to select program 'RFIDSample4App-C' as shown in figure 21 (a). Similary also shown for 'RFIDSample4App-CPP' program selection for CPP application at figure 21(b).

Image: the set of the set	Choose a program to run: Binaries:		Access
> ⊕ RFIDSample4A ■ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	: 🕞 Binaries:		or 🛋 Modu 🗂 🗖
Eprendent Lext Pro E_C/C++ Application Pro C_C/C++ Attach to Application Pro C_C/C++ Container Launcher C/C C_C/C++ Postmortem Debugge Pro C_C/C++ Remote Application Pro C_C/C++ Remote Application Pro C_C/C++ Remote Application Pro C_C/C++ Remote Application Pro C_C/C++ Unit E GDB Hardware Debugging C_Launch Group Launch Group Launch Group Launch Group Built Pro Target Communication Fram Pro Filter matched 14 of 14 items Using	 RFIDSample4App-C RFIDSample4App-C Sample4App-C Qualifier: Qualifier: \$ armle - /RFIDSample4App-C/Debug/RFIDSample4/ \$ armle - /RFIDSample4App-C/Debug/RFIDSample4/ G Cancel OK 	es Search Project Browse auto build Vorkspace Settings ew Edit Properties other Revert Apply	v Modu - v Value
⑦ 17:05:17 Bu	ld Finished. 0 errors, 507 warnings. (took 827ms)	Close Debug	./rootfs/lib -DUN

Figure 21(a): Select C application Program View

🐔 O 🔳 O F	Debug Configurations		😣 🐵 Program Selection	
🕸 🕶 💽 🕶 💁 🕶 🎒	Create, manage, and run configution of the configut	pecified.	Choose a program to run:	Access 🗄 🖻 🗟 😵
) ∯RFIDSample4A	Image: Second Secon	Name: F Main Project: RFID3a C/C++ Ar Debug, Build (if Build C © Enal © Use Connect Remote	Binarles:	ko@konton
	Filter matched 14 of 14 items	Using G	Cancel OK Close Debug	./rootfs/lib -DUN:
€ RFIDSample4App	-CPP	30 Build I	Finished. 0 errors, 504 warnings. (took 817ms)	

figure 21(b): Select CPP application program View



Select "New" as shown in figure 22

Figure 22: Debug Configuration New Connection View

🐔 🗱 🔳 🎋 🛛	😣 🗉 Debug Configurations		
	Create, manage, and run configu	urations to the second s	Access
) ∰ RFIDSample4A	Image: Contract of the system of the sys	Name: RFIDSample4App-C Debug Main ↔ Arguments) ☆ Debugger Project: RFIDSample4App-C RFIDSample4App-C Browse C/C++ Application: Debug/RFIDSample4App-C Debug/RFIDSample4App-C Variables Build (if required) before launching Browse Build Configuration: Use Active ● Enable auto build Disable auto build ● Use workspace settings Configure Workspace Settings Connection: Remote Host New Remote Absolute File Path for C/C++ Application: Edit	
	Filter matched 14 of 14 items	Using GDB (DSF) Automatic Remote Debugging Launcher - Select other Revert Apply	
	?	Close Debug	./rootfs/lib -DUN:
€ RFIDSample4App	-c	-	

In Create a new connection pop-up, choose connection type, select "SSH" from drop down and click "OK"

Figure 23: SSH Connection View



S & B *	Debug Configurations		
* - 0 - 9 - 9	Create, manage, and run config	rations	Access
Project Explorer			or 🛋 Modu 🗖 🗖
▶ ♣ RFIDSample4A		Name: RFIDSample4App-C Debug	1000 - 11 - 2
	type filter text	Main M-Arguments * Debugger	
	C/C++ Application	Project:	
	C/C++ Container Launcher	Create a new connection Browse	
	 C/C++ Postmortem Debugge C/C++ Remote Application 	Choose connection type:	
	RFIDSample4App-C Debug	Serial Port Serial Port Browse Browse	
	GDB Hardware Debugging	Arduino Tologt OK	
	Launch Group	SSH	
	Remote Application	Configure Workspace Settings Configure Workspace Settings	
	Semote Debugger		
		Connection: FX7500 New Edit Properties	
		Remote Absolute File Path For C/C++ Application:	
	Filter matched 14 of 14 items	Using GDB (DSF) Automatic Remote Debugging Launcher - Select other Revert Apply	
	?	Close Debug	./rootfs/lib -DUN:
	14.07.1	A Build Finished A errors 507 warnings (took 1s 702ms)	
	14.07.	o barta rinishtan o triors, sor marrings, (took 15.702m3)	
RFIDSample4App-	c		

Enter Details of the RFID Reader FX9600 /FX7500 target

Enter Connection Name "FX9600"

Enter RFID Reader IP address "xxx.xxx.xxx" in the Host field

Enter User name "rfidadm"

And select "Password based authentication" radio button and leave Password field empty as shown

in figure 24 and click "Finish" button

Figure 24: New Connection Window View



Image: Specify properties of a new connection Im	
> for \$FIDSample4Al > for \$FIDSample4Al > for \$FIDSample4Al > connection name	ess 🕴 🖻 🖏
Image: Information Image: Im	
Target Communication Fram Connection: FX7500 New Edit Properties Remote Absolute File Path for C/C++ Application:	
Filter matched 14 of 14 items Using GDB (DSF) Automatic Remote Debugging Launcher - Select other Revert Apply	₽੶₽₽₽
Close Debug ./rootfs	otfs/lib -DUN:
14:07:50 Build Finished. θ errors, 507 warnings. (took 1s.702ms)	

Select Connection name, say FX9600 and provide the path of the RFID Reader target where the binary executable needs to be saved on the Remote Absolute Path ("/apps/RFIDSample4App-C") as shown in Figure 25(a) for C application project (RFIDSample4App-C). Similarly applicable for C++ application project (RFIDSample4App-CPP) where the binary executable needs to be saved on the Remote Absolute Path ("/apps/RFIDSample4App-CPP") as shown n Figure 25(b).

Figure 25(a): New Connection Apply Window View for C application project



🐔 🎋 🔳 🎋	😣 🗉 Debug Configurations		
* • 0 • % • @(Create, manage, and run configu	Irations to the second s	Access
Project Explorer		2	pr 🛋 Modu 🗖 🗖
▶ ∰ RFIDSample4Aj	type filter text	Name: RFIDSample4App-C Debug	
	C/C++ Application	RFIDSample4App-C Browse	
	C/C++ Attach to Application C/C++ Container Launcher	C/C++ Application:	
	© C/C++ Postmortem Debugg∉ ▼ ⓒ C/C++ Remote Application	Variables Search Project Browse	
	© RFIDSample4App-C Debug CüC/C++ Unit ⓒ GDB Hardware Debugging	Build (if required) before launching Build Configuration: Use Active	
	Gaunch Group Launch Group (Deprecated) Superstand	Enable auto build Disable auto build Use workspace settings Configure Workspace Settings	
	▲ Remote Application 茶 Remote Debugger 【 Target Communication Fram	Connection FX9600 New Edit Properties	
		Remote Absolute File Path for C/C++ Application: /apps/RFIDSample4App-C Browse	
	Filter matched 14 of 14 items	Using GDB (DSF) Automatic Remote Debugging Launcher - Select other Revert Apply	
	?	Close Debug	./rootfs/lib -DUN:
	14:07:5	0 Build Finished. 0 errors, 507 warnings. (took 1s.702ms)	
🔓 RFIDSample4App	-c		

Figure 25(b): New Connection Apply Window View for C++ application project

😣 🗉 Debug Configurations		
Create, manage, and run configu	urations	-
Image: Second secon	Name: RFIDSample4App-C Debug	
© C/C++ Application © C/C++ Attach to Application © C/C++ Container Launcher	RFIDSample4App-CPP C/C++ Application: Debug/RFIDSample4App-CPP	Browse
 C/C++ Postmortem Debugge C/C++ Remote Application RFIDSample4App-C Debug C/C++ Unit GDB Hardware Debugging 	Variables Search Project Build (if required) before launching Build Configuration:	Browse
Group Launch Group Launch Group (Deprecated) Deprecated	 Enable auto build Disable auto build Use workspace settings Configure Workspace Settings 	
 端 Remote Application 券 Remote Debugger Target Communication Fram 	Connection: FX9600 New Edit Remote Absolute File Path for C/C++ Application:	Properties
	/apps/RFIDSample4App-CPP	Browse
Filter matched 14 of 14 items	Using GDB (DSF) Automatic Remote Debugging Launcher - Select other Revert	Apply
?	Close	Debug

Command to execute before application run is explained below

Copy gdbserver file to the RFID Reader target board only for the first time for debugging. The following command copies the gdbserver file from SDK samples directory from the Linux Ubuntu/Fedora host machine onto the RFID reader target in /tmp directory



for Ubuntu Desktop 16.04

"scp -oStrictHostKeyChecking=no -oUserKnownHostsFile=/dev/null <user>@<Linux-host-IP>:/<installation path>/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/samples/gdbserver /tmp"

Note: for Ubuntu Server 16.04 edition use the command given below

"scp -oKexAlgorithms=+diffie-hellman-group1-sha1 -oStrictHostKeyChecking=no oUserKnownHostsFile=/dev/null <user>@<Linux-host-IP>:/<installation path>/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/samples/gdbserver /tmp"

Above command downloads gdbserver file from Linux Ubuntu/Fedora host machine (IP address whose IP address is given in the command) then followed by samples directory path on Linux Host Ubuntu/Fedora machine, "/tmp" directory specify the path on the RFID reader where gdbserver file is downloaded.

Paste the above command given in bold in the "Commands to execute before application is run" box and click Apply a shown below.

🐔 🗱 🔳 🎋 🛛	😣 💷 Debug Configurations		
* • 0 • • • (● (Create, manage, and run config	urations The second sec	Access
눰 Project Explorer		2	pr 🛋 Modu 🖳 🗖
► 🔐 RFID Sample4A		Name: RFIDSample4App-C Debug	
▶ RFIDSample4A	type filter text	Main ^(M) Arguments ^(M) Debugger ^(M) Source ^(M) <u>C</u> ommon	
	 C/C++ Application C/C++ Attach to Application C/C++ Container Launcher C/C++ Postmortem Debugge 	Variables Search Project Browse Build (if required) before launching Image: Comparison of the second	
	▼ C/C++ Remote Application CRFIDSample4App-C Debug CUC/C++ Unit CGDB Hardware Debugging	Build Configuration: Use Active Enable auto build Disable auto build Use workspace settings Configure Workspace Settings 	
	 ➡ Launch Group ➡ Launch Group (Deprecated) ➡ ➡ Remote Application 	Connection: FX9600 New Edit Properties Remote Absolute File Path for C/C++ Application:	
	称 Remote Debugger	/apps/RFIDSample4App-C Browse	
	Target Communication Fram	Commands to execute before application	
	Filter matched 14 of 14 items	scp -oStrictHostKeyChecking=no -oUserKnownHostsFile=/dev/null guest@192.168.1.10:/home/guest/Zebra- Using GDB (DSF) Automatic Remote Debugging Launcher - Select other Revert Apply	
	?	Close Debug	./rootfs/lib -DUN:
	14:07:	50 Build Finished. 0 errors, 507 warnings. (took 1s.702ms)	
🔓 RFIDSample4App	-c		

Figure 26: Commands to execute before application

Make sure "Gdbserver Settings" is set to /tmp/gdbserver

Figure 27: Debugger -> Debugger Options -> Gdb Server settings



🐔 🗱 🔳 🕸 🛛	😣 🗉 Debug Configurations		
* - 0 - 9 - 19(Create, manage, and run config	urations	Access
Project Explorer			or 🛋 Modu 🗖 🗖
) 🔂 RFIDSample4A) 🔂 RFIDSample4A	Image: Second Secon	Name: RFIDSample4App-C Debug Main (** Arguments * Debugger SourceCommon Debugger Options Gdbserver Settings Main Shared Libraries Gdbserver Settings Gdbserver path:	
	 E CDB Hardware Debugging ♣ Launch Group ▶ Launch Group (Deprecated) ▶ Remote Application ☆ Remote Debugger Target Communication Fram 	Remote timeout (seconds):	
	Filter matched 14 of 14 items	Using GDB (DSF) Automatic Remote Debugging Launcher - Select other Revert Apply	
	?	Close Debug	./rootfs/lib -DUN:
1 ^C DEIDCompletApp	14:07:	50 Build Finished. 0 errors, 507 warnings. (took 1s.702ms)	

Next select Debugger tab -> Debugger Options -> Main tab and click on Browse button Figure 28: New Connection Debugger Window View

😣 🗈 Debug Configurations		
Create, manage, and run configu	Irations	The second
Image: Second state Image: Secon	Name: RFIDSample4App-C Debug Main ௸ Arguments ♥ Debugger Debugger Options Main Shared Libraries Gdbserver Settings	
 C/C++ Container Launcher C/C++ Postmortem Debugge C/C++ Remote Application RFIDSample4App-C Debug Cii C/C++ Unit GDB Hardware Debugging Launch Group Launch Group (Deprecated) Remote Application Remote Debugger Target Communication Fram 	GDB debugger: gdb GDB command file: .gdbinit (Warning: Some commands in this file may interfere with the startup operation of the d example "run".) Non-stop mode (Note: Requires non-stop GDB) Enable Reverse Debugging at startup using: Software Reverse Debugging (detailed Force thread list update on suspend Automatically debug forked processes (Note: Requires Multi Process GDB) Tracepoint mode: Normal	Browse Browse ebugger, for d but slower) 🔻
Filter matched 14 of 14 items	Using GDB (DSF) Automatic Remote Debugging Launcher - Select other Revert	Apply
?	Close	Debug

Make sure Cross GDB path is set properly

Navigate to "<install_path>/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux_V1.0.4/tools/armhfp-gnueabi/bin" directory and select the ARM cross gdb debugger

("arm-montavista-linux-gnueabi-gdb") as shown in the Figure 29 and click OK.



Figure 29: GDB Debugger View

8	GDB Debugge	er							
0	Recent	ecent du guest Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux tools arm					nfp-gnueabi	bin	
ŵ	Home	Name					Size	Modifie	ed
	Desktop		arm-monta	vista-linux-onueabi				30 Jun	2019
n	Documents		python3-native					30 Jun	2019
			arm-monta	vista-linux-gnueabi-addr2line			749.7 kB	29 Jun 1	2019
Ý	Downloads		arm-monta	vista-linux-gnueabi-ar			779.6 kB	29 Jun 3	2019
1	Music		arm-monta	vista-linux-gnueabi-as			1.4 MB	29 Jun 2019	
ø	Pictures		arm-monta	vista-linux-gnueabi-c++			844.6 kB	29 Jun 2019	
	- inclusion		arm-monta	vista-linux-gnueabi-c++filt			747.7 kB	29 Jun 1	2019
	Videos	\diamond	arm-monta	vista-linux-gnueabi-cpp			842.5 kB	29 Jun 1	2019
	eclipse		arm-monta	vista-linux-gnueabi-dwp			2.4 MB	29 Jun 1	2019
		Ø	arm-monta	vista-linux-gnueabi-elfedit			28.0 kB	29 Jun 3	2019
+	Other Locations		🖗 arm-montavista-linux-gnueabi-g++					29 Jun (2019
		🗇 arm-montavista-linux-gnueabi-gcc				842.4 kB	29 Jun 1	2019	
		arm-montavista-linux-gnueabi-gcc-6.4.0					842.4 kB	29 Jun 1	2019
			arm-monta	vista-linux-gnueabi-gcc-ar			25.5 kB	29 Jun 1	2019
		$\langle \rangle$	arm-monta	vista-linux-gnueabi-gccgo			843.8 kB	29 Jun 1	2019
		Ø	arm-monta	vista-linux-gnueabi-gcc-nm			25.5 kB	29 Jun 1	2019
		Ø	arm-monta	vista-linux-gnueabi-gcc-ranlib			25.5 kB	29 Jun 3	2019
			arm-monta	vista-linux-gnueabi-gcov			425.6 kB	29 Jun 1	2019
			arm-monta	vista-linux-gnueabi-gcov-dump			364.2 kB	29 Jun 1	2019
			arm-monta	vista-linux-gnueabi-gcov-tool			388.8 kB	29 Jun	2019
			arm-monta	vista-linux-gnueabi-gdb			4.9 MB	29 Jun 3	2019
			arm-monta	vista-linux-gnueabi-gprof			814.5 kB	29 Jun 1	2019
	🗇 arm-montavista-linux-gnueabi-ld					1.3 MB	29 Jun 1	2019	
			arm-monta	vista-linux-gnueabi-ld.bfd			1.3 MB	29 Jun 1	2019
	arm-montavista-linux-gnueabi-ld.gold						4.7 MB	29 Jun 1	2019
			arm-monta	vista-linux-gnueabi-nm			761.5 kB	29 Jun 1	2019
							Cancel	OK	:

Click "Apply" button in Debug Configurations window Next click on Debug button as shown in the Figure 30.

Figure 30: GDB Debugger Apply View



🗧 🗊 Debug Configurations

* 🖻 🎦 🗶 🗖 🦫 🔻		
C/C++ Application C/C++ Attach to Application C/C++ Container Launcher	Name: RFIDSample4App-C Debug Main I I Arguments I Provide Arguments I Provide Arguments Main Shared Libraries Gdbserver Settings CDB debugger	
 C/C++ Postmortem Debugge C/C++ Remote Application RFIDSample4App-C Debug Cii C/C++ Unit GDB Hardware Debugging Launch Group Launch Group (Deprecated) Remote Application % Remote Debugger Target Communication Fram 	GDB command file: gdbinit (Warning: Some commands in this file may interfere with the startup operation of the del example "run".) Non-stop mode (Note: Requires non-stop GDB) Enable Reverse Debugging at startup using: Software Reverse Debugging (detailed l Force thread list update on suspend Automatically debug forked processes (Note: Requires Multi Process GDB) Tracepoint mode: Normal	Browse bugger, for but slower)
Filter matched 14 of 14 items	Using GDB (DSF) Automatic Remote Debugging Launcher - Select other Revert	Apply

The following figures of 31, 32, 33 and 34 may be encountered while setting up the remote connection for the first time with a fresh new Linux user. Click on "Yes", "OK" buttons as shown below.

Figure 31: Authentication 1

😣 Aut	hentication Message
?	WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY! Someone could be eavesdropping on you right now (man-in-the-middle attack)! It is also possible that the RSA host key has just been changed. The fingerprint for the RSA key sent by the remote host 192.168.1.9 is 13:d3:e7:f7:d6:ed:a6:a6:b4:d3:c1:55:71:23:a2:ce. Please contact your system administrator. Add correct host key in /home/guest/.ssh/known_hosts to get rid of this message. Do you want to delete the old key and insert the new key? No



Figure 32: Authentication 2



Figure 33: Authentication 3



Figure 34 : Authentication 4



The "Problem Occurred" error window will popup, click on "OK" as per the following figure 35.



Figure 35: Problem Occurred popup view

8 🗖 🗖	Problem Occurred
	'Launching RFIDSample4App-C Debug' has encountered a problem. Error during file upload.
	Details >> OK

Now click the top left second button, Launch in 'Debug' mode

Figure 36: Launch in 'Debug' mode

🐔 🐐 🔳 🎋 Debug 🛛 🗸 🖸 RFIDS	Sample4App-C Debug 🗠 🌞 📑 🛨 🗐 🗟 🕍 🗚 Nev
\$* - ○ - ᅆ - ᄵఴ / - * / * * * + * + *	← → ⇒
🔁 Project Explorer 🛛 🛛 🖻 🔄 🔽 🗖 🗖	RFIDSample4App.c RFIDSample4App.cpp
▶ 🥩 RFIDSample4App-C	1 #include "common.h"
▶ 🚰 RFIDSample4App-CPP	<pre>bool g_bUseWin32EventHandling = false bool g_bUseWin32EventHandling = falseWin32EventHandling = false bool g_bUseWin32EventHandling = false bool g_bUseWin32EventHandling = falseWin32EventHandling = falseWin32E</pre>
	16 17 #ifdef linux

The following console displays the password prompt

Figure 37.1: Password prompt



🐔 🔯 🔳 🎋 Debug 🛛 🗸 💽 🖡	RFIDSample4App-C Debug 🗸 🏟 🖃 🐨 🐨 🐨 📾 🌬 🔺 New Connection 💦 🖓 🕅 🛍 🔻 🕲 🔻 🖳 🕷 🕨 💷	IN 3. 0. r i> 5 7
│☆ ▼ O ▼ Q ▼ (౨⊜ A ▼ ⊿ ⋧ ݤ ▼ ∛		Quick Access
ବ Project Explorer 🛛 🗧 🕏 😨 =	ି 🗋 🖻 RFIDSample4App.c 🚺 RFIDSample4App.cp ଅ	🂊 Brea 🞋 Expr 🛋 Modu 🖓 🗖
▶ 🚅 RFIDSample4App-C	1 #include "common.h"	10 - 11 - 11 - 11 - 11 - 11 - 11 - 11 -
▶ ∰ RFIDSample4App-CPP	<pre>bool g_bUSeWin32EventHandling = false; bool g_bUSeWin32EventHandling = false; static int readerPort = 0; static SINGULATION CONTROL singulationControl; extern ANTENNA_INF0 g_antennaInfo; void InventoryFilterOption(RFID_HANDLE32 readerHandle); void Createmenu(RFID_HANDLE32 readerHandle); void InventoryMenu(RFID_HANDLE32 readerHandle); void InventoryMenu(RFID_HANDLE32 readerHandle); void AccessMenu(RFID_HANDLE32 readerHandle); filter InventoryMenu(RFID_HANDLE32 readerHandle); filter InventoryMenu(</pre>	
	Supervised and the second seco	▓▙▋▓▉▓▝▌▌▼▝▌▼▝▌
	<pre>KHIDSample4app-Cuebug [c/c++ Kennote Application] Remote Shell rfidadm@FX960@FF82BA:-\$ scp -oStrictHostKeyChecking=no -oUserKnownHostsFile=/dev /null guest@192.168.1.10:/home/guest/Zebra-FXSeries-Embedded-Native-SDK-C-CPP Li nux_V1.0.4/samples/gdbserver /tmp;/tmp/gdbserver :2345 /apps/RFIDSample4App-C;e xit Warning: Permanently added '192.168.1.10' (ECDSA) to the list of known hosts.</pre>	
L	guest@192.168.1.10's password:	
🔗 RFIDSample4App-C	Launching	RFIDSamplebug: (57%) 💷 🦷

Enter your linux host user password in the console window password prompt

Figure 37.2: Enter Password





Figure 38: Confirm Perspective Switch



8 Confirm Perspective Switch							
?	This kind of launch is configured to open the Debug perspective when it suspends.						
	This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.						
	Do you want to switch to this perspective now?						
Rem	Remember my decision						
	No Switch						

Right click RFIDSample4App-C's pre defined break point address and click Resume for the debug to continue as shown in the figure 39

Figure 39: Debug Console view



The application will halt at the pre-defined breakpoint Next click on the resume button again



Figure 40.1: RFIDSample4App-C console breakpoint



The application output will be seen as per figure 40.2

Figure 40.2: RFIDSample4App output View



३०४ अंदि राज्य रा

EMBEDDED RFID C/C++ APPLICATION

7.0 Embedded RFIDSample4App C/C++ Application from scratch

This section describes the detailed steps to create embedded C/C++ RFIDSample4App from scratch. The steps involved are:

- Create Workspace
- Create Project
- Add sources and header files
- Add header file include path, library path, libraries
- Add compiler and linker flag
- Clean and build steps
- Debug embedded C/C++ RFID application
- Creation of start and stop script for C/C++ installation package

NOTE :

Wherever the steps for C & C++ application project/workspace differ, it will be mentioned here with step/screenshot of C application project/workspace first, followed by the step/screenshot for C++ application project/workspace. If not stated separately then one can assume the steps are same. **Figure xx(a)** is for **C application project/workspace** and **Figure xx(b)** is for **C++ application project/workspace**.

(Also for C project work with .c files and for C++ project work with .cpp files).

The assumption is both C/C++ application project doesn't belong to same workspace. It is show both projects are under it own workspace.

7.1 Creating a Workspace

Navigate to '[installation-path]'/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/eclipse/'. Double-click on eclipse executable file. Under Eclipse Launcher popup window, click on Browse for workspace as per figure 41.



Figure 41: Eclipse Launcher

8 Eclipse Launcher							
Select a directory as workspace Eclipse IDE uses the workspace directory to store its preferences and development artifacts.							
Workspace: /home/guest/eclipse-workspace	▼ Browse						
Use this as the default and do not ask again							
	Cancel Launch						

Select the directory path where the RFID sample application has to be created from scratch. The following figure 42 shows the creation of new directory for this.

Figure 42: Creating new workspace

< > 《 论 Home	Þ						۹	::	≡
 Recent Home Desktop Documents 	Desktop	Documents	Downloads	hidden	Music	Pictures			
 Downloads Music Pictures Videos 	Public	6030							
 Videos Trash + Other Locations 	Public Examples	snap	ru	er py	Create Cancel desktop-amd64.iso	模板 tar.gz Zebra-FXSeries- Embedded-Native- SDK-C-CPP_Linux_ V1.0.4.tar.gz			

The following figure 41 depicts selection of newly created directory as workspace directory. Click on Launch

ZEBRA TECHNOLOGIES



Figure 43: Eclipse Launcher

8 Eclipse Launcher							
Select a directory as workspace Eclipse IDE uses the workspace directory to store its preferences and development artifacts.							
Workspace: /home/guest/new-workspace	▼ Browse						
Use this as the default and do not ask again							
	Cancel Launch						

Following eclipse welcome screen shows up

Figure 44: Welcome to the Eclipse

Se 🗇 🗇 New-Workspace - Eclipse IDE							
🚳 Welcome 🛿			🟠 🗘 🖧 📶 🗖 🗗 🗗	8			
ecli	DSE Welcome to the Eclipse IDE for	C/C++ [Developers	HE I			
:=	Tutorial: Import an existing project A guided walk-through how to import an existing project		Overview Get an overview of the features				
•	Review IDE configuration settings Review the IDE's most fiercely contested preferences		Tutorials Go through tutorials				
0	Create a new C project Create a new Eclipse project for C source code	1	Samples Try out the samples				
0	Create a new C++ project Create a new Eclipse project for C++ source code		What's New Find out what is new				
0	Import a project with a working Makefile Open the New item wizard						
•>	Checkout projects from Git Checkout Eclipse projects hosted in a Git repository		🗹 Always show Welcome at start up				
<u></u>							

Close this Welcome tab inside eclipse.

7.2 Creating an Embedded RFIDSample4App C/C++ Project

Create a new Project RFIDSample4App in the Eclipse Select File->New->Project

Figure 45: C Eclipse IDE View



File Edit Navigate Search Project R	Run	Window He	lp	
New	>	Project	o Debua	
Open File		Example	i c	
Recent Files	>	Other	• • : 🗠	<u> </u>

Expand C/C++ Folder and Select "C Project" and click "Next" button as shown in the figure 46(a)

Figure 46(a): New Project view for C application

😣 🗉 New Project	
Select a wizard	
Create a new C project	
Wizards:	
type filter text	•2
▶ 🧁 General	
▼	
Arduino Project	
C Project	
C/C++ Project	
C++ Project	
Makefile Project with Existing Code	
▶ 🗁 RPM	
▶ 🧁 Tracing	
▶ 🗁 Examples	
? < Back Next > Cancel Fi	inish

Figure 46(b): New Project view for C++ application



😣 🗉 New Project
Select a wizard
Create a new C++ project
Wizards:
type filter text
 ▶ General ▼ C/C++ ➡ Arduino Project
Î C Project 같 C/C++ Project
Image: C++ Project Image: Makefile Project with Existing Code Image: Point Project With Existing Code Image: Point Project With Existing Code Image: Point Project Project With Existing Code Image: Point Project Pr
? < Back Next > Cancel Finish

In the C Project Window or C++ Project Window, as the case may be, enter the Project name as "RFIDSample4App"

For Project Type select "Empty Project"

For Toolchain select "Cross GCC" and select "Next" button as shown in figure 49.

Figure 49: New Project Select Type view



😣 🗊 C Project	
C Project Create C project of selected type	
Project name: RFIDSample4App	
Use <u>d</u> efault location	
Location: /home/guest/new-workspace	/RFIDSample4App B <u>r</u> owse
Choose file system: default 🕶	
Project type:	Toolchains:
🕨 🗁 GNU Autotools	Cross GCC
🗢 🗁 Executable	Linux GCC
Empty Project	
Hello World ANSI C Project	
Shared Library	
Static Library	
Show project types and toolchains on	y if they are supported on the platform
?	lext > Cancel Finish
Select "Next" button in Figure 48	

Figure 48: Create C or C++ Project View

😣 💷 C Project	
Select Configurations Select platforms and configurations you wish to deploy on	
Project type: Executable Toolchains: Cross GCC Configurations:	
🖾 🤀 Debug	Select all
Carlos Release	Deselect all
	Advanced settings
Use "Advanced settings" button to edit project's properties.	
Additional configurations can be added after project creation. Use "Manage configurations" buttons either on toolbar or on	property pages.
? < Back Next > Car	Finish



Under C or C++ Project window, Set the following parameters.

- 1. Cross Compiler Prefix -> "arm-montavista-linux-gnueabi-".
- 2. Click on Browse button to set the Cross-compiler path as shown in figure 49.

Figure 49: Project Finish Button view



Select "[Installation-path]/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/tools/armhfpgnueabi/bin" directory and click OK



Figure 50: Cross Compiler Path Selection

8	0	-		_	
0	Recent	1	Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux tools armhfp-gnueabi	bin 🕨	53
仚	Home	Na	me 🔺	Size	Modified
	Desktop		arm-montavista-linux-onueabi		30 Jun 2019
n	Documents		python3-native		30 Jun 2019
			arm-montavista-linux-gnueabi-addr2line	749.7 kB	29 Jun 2019
Ý	Downloads		arm-montavista-linux-gnueabi-ar	779.6 kB	29 Jun 2019
99	Music	٨	arm-montavista-linux-gnueabi-as	1.4 MB	29 Jun 2019
ø	Pictures	٨	arm-montavista-linux-gnueabi-c++	844.6 kB	29 Jun 2019
		Ø	arm-montavista-linux-gnueabi-c++filt	747.7 kB	29 Jun 2019
	Videos		arm-montavista-linux-gnueabi-cpp	842.5 kB	29 Jun 2019
	eclipse	Ø	arm-montavista-linux-gnueabi-dwp	2.4 MB	29 Jun 2019
		Ø	arm-montavista-linux-gnueabi-elfedit	28.0 kB	29 Jun 2019
+	Other Locations		arm-montavista-linux-gnueabi-g++	844.6 kB	29 Jun 2019
			arm-montavista-linux-gnueabi-gcc	842.4 kB	29 Jun 2019
		Ø	arm-montavista-linux-gnueabi-gcc-6.4.0	842.4 kB	29 Jun 2019
			arm-montavista-linux-gnueabi-gcc-ar	25.5 kB	29 Jun 2019
		$\langle \rangle$	arm-montavista-linux-gnueabi-gccgo	843.8 kB	29 Jun 2019
		Ø	arm-montavista-linux-gnueabi-gcc-nm	25.5 kB	29 Jun 2019
			arm-montavista-linux-gnueabi-gcc-ranlib	25.5 kB	29 Jun 2019
			arm-montavista-linux-gnueabi-gcov	425.6 kB	29 Jun 2019
		٨	arm-montavista-linux-gnueabi-gcov-dump	364.2 kB	29 Jun 2019
		$\langle \rangle$	arm-montavista-linux-gnueabi-gcov-tool	388.8 kB	29 Jun 2019
		$\langle \rangle$	arm-montavista-linux-gnueabi-gdb	4.9 MB	29 Jun 2019
		$\langle \rangle$	arm-montavista-linux-gnueabi-gprof	814.5 kB	29 Jun 2019
		٨	arm-montavista-linux-gnueabi-ld	1.3 MB	29 Jun 2019
			arm-montavista-linux-gnueabi-ld.bfd	1.3 MB	29 Jun 2019
			arm-montavista-linux-gnueabi-ld.gold	4.7 MB	29 Jun 2019
			arm-montavista-linux-gnueabi-nm	761.5 kB	29 Jun 2019
				Cancel	ОК



Click on Finish button.

Figure 51: Cross Compiler Path Selection

😣 🗉 C Project	
Cross GCC Command	-
Cross compiler prefix: arm-montavista-linux-gnueabi-	
Cross compiler path:tive-SDK-C-CPP_Linux/tools/armhfp-gnueabi/bin	Browse
Cancel	Finish

Open Associated Perspective window may appear, click on Open Perspective button.

Figure 52: Open Perspective view





7.3 Adding Source Files to Embedded RFIDSample4App C/C++ Project

In this section source files will be added to RFIDSample4App project Add "inc" and "src" directory

In the Project Explorer View, right-click on the project RFIDSample4App, Select->New->Folder.

Figure 53: RFID Folder Creation view



Select "RFIDSample4App"

Enter Folder name as "inc" as shown in the figure 52 and click "Finish" button.

Figure 54: RFID Folder Name view

😣 🗊 New Folder	
Folder	
Create a new rolder resource.	
Enter or select the parent folder:	
RFIDSample4App	
➢ RemoteSystemsTempFiles	
• 🚔 RFIDSample4App	
Folder name: ind	
Advanced >>	
\odot	Cancel Finish



Add "src" directory following same above steps as mentioned to add "inc" directory. Similarly, create "src" folder under "RFIDSample4App" project.

Add source files to "inc" and "src" directory

Right-click on "inc" directory, select New->File as shown in the figure 55

Figure 55: RFID inc view



Enter the File name as "common.h" the header file as shown in the figure 56 and click "Finish" button.

Figure 56: common.h view

se IDE			
🌾 🔳 🎋 Debug 🗸 💽	RFIC	🛞 🗊 New File	
::::::::::::::::::::::::::::::::::::::	9	File Create a new file resource.	
Project Expl 🛛 😫 Connection 🖓 🗖		Enter or calact the parent folder:	
■ ➡ ■ ■ ■ ■ ■ Includes > ■ Includes		RFIDSample4App/inc [™]	
▶ æsrc		 ➢ RemoteSystemsTempFiles ☞ ➢ RFIDSample4App ➢ .settings ➢ inc ➢ src 	_
	No	File name: common.h Advanced >>	
		? Cancel Fin	nish



Similarly, add "common.c" and "RFIDSample4App.c" source files to "src" directory Listing of files is shown below in the figure List of Files added to project

Figure 57(a): List of Files in RFIDSample4App project for C application



Figure 57(b): List of Files in RFIDSample4App project for C++ application



Copy the source code from the samples C directory (from the untarred package "[install-path]/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/samples/workspace/RFIDSample4App-C", copy the respective files from inc and src directories) provided and save it by clicking Save All.

File Edit Source Refactor Navigate	Search Project Run Window Help
New Open File 🦻 Open Projects from File System	> o Launch Configurations > on: → > ⊗ + ⊗ </td
Recent Files	
Close Close All Save Save All Save All Revert Move Refresh Convert Line Delimiters To Print Import Export Properties Switch Workspace Restart Exit	<pre> *common.h @ *common.c @ *RFIDSample4App.c % clean_stdin(); clean_stdin(); clean_stdin(); clean_stdin(); clean_stdin(); clean_stdin(); case 1; rfidstatus = AddPreFilter(readerHandle); case 2; case 2; case 3; rfidstatus = RemovePrefilter(readerHandle); clear case 3; case 3; return; clear case case case case case case case case</pre>
	278 Problems 13 Tasks Console Properties W Call Graph 76 errors, 0 warnings, 0 others Description • Errors (76 items)

Figure 58(a): Adding source files for C application

For C++ application project, copy the source code from the samples C++ directory (from the untarred package "[install-path]/Zebra-FXSeries-Embedded-Native-SDK-C-

CPP_Linux/samples/workspace/RFIDSample4App-CPP",

copy the respective files from inc and src directories) provided and save it by clicking Save All.

Figure 58(b): Adding source files for C++ application



File Edit Source Refactor Navigate Sea	rch Project Run Window Help	🏚 🛛 0.9 KiB/s 🔶	En 📧 (100%) 🗤) 5:23 PM 🤱
<u>N</u> ew	Launch Configurations 🗸 on:	- 🔚 🐚 📎 - 🍕 - 📾	
Open File <u>.</u>		· 👝 🚓 🖌 • 🖂 🖘 🖬 🔳 🖷	
Recent Files			
Close			
<u>Cl</u> ose All	In common.h Is common.cpp Is *RFIDSample4App.cpp IS		E Ou 🛛 🗐 Ta 💿 Bu 🗖 🗎
ave 3	2 heal a blackin225 wastlandling false.		P □ 4z R X • # ▼
	4		 g bUseWin32EventHandling :
🐘 Sav <u>e</u> All	<pre>5 6 static wchar_t hostName[260];</pre>		<pre></pre>
Rever <u>t</u>	<pre>7 static int readerPort = 0; 3 8 static SINGULATION CONTROL singulationControl;</pre>		readerPort : int singulation Control : SINCLUM
Move	<pre></pre>		 g_antennalnfo: ANTENNA_IN
Rename	11 void InventoryFilterOption(RFID HANDLE32 readerHandle) 12 void Createmenu(RFID HANDLE32 readerHandle);	;	++ InventoryFilterOption(RFID_F
Convert Line Delimiters To	13 void ConfigurationMenu(RFID HANDLE32 readerHandle); 14 void InventorvMenu(RFID HANDLE32 readerHandle):		Createmenu(RFID_HANDLE32 ConfigurationMenu(RFID_HA
🚔 Print	<pre>\$ 15 void AccessMenu(RFID_HANDLE32 readerHandle); 16</pre>		+ InventoryMenu(RFID_HANDLI
	17 #ifdef linux 18@ int main(int argc_char* argv[])		++ AccessMenu(RFID_HANDLE32
🖆 Export	19 #else 20 int_tmain(int_argc_wchar_t*_argv[])		main(int, char*[]): int Createmenu(RFID_HANDLE32
P <u>r</u> operties	21 #endif		ConfigurationMenu(RFID_HA
Switch Workspace	23 if(argc == 1 argc == 3)		InventoryMenu(RFID_HANDLI AccessMenu(DEID_HANDLE3)
Restart	25 if(argc == 1)		 InventoryFilterOption(RFID_F
E <u>x</u> it	<pre>20 1 wcscpy(hostName, L"localhost");</pre>		
	readerPort = 0;		
	Problems 🛛 🧟 Tasks 📮 Console 🗔 Properties 🕮 Call Graph		
	109 errors, 0 warnings, 0 others (Filter matched 100 of 109 items)	th Location Tu	
	Resource Pa	Location Ty	pe
	Writable Smart	Insert 270:1	

7.4 Setup Cross Compiler and Library Environment for Embedded Native C/C++ Project

In the Project Explorer view, select RFIDSample4App project, click on Project->Properties as shown in Figure 59.

Figure 59: Project Properties



In the Properties Window of RFIDSample4App project Select and expand "C/C++ General" in left panel

Click on "Path and Symbols"

Under "Includes" tab, Select "GNU C" as language for C application

Figure 60(a): Properties Window RFIDSample4App for C application





In the Properties Window of RFIDSample4App project Select and expand "C/C++ General" in left panel Click on "Path and Symbols"

Under "Includes" tab, Select "GNU C++" as language for C++ application

Figure 60(b): Properties for RFIDSample4App for C++ application

type filter text	Paths and Symbols		↓ ↓ ↓
 Resource Builders C/C++ Build C/C++ General Code Analysis 	Configuration: Debug [Active]	▼ ■ Elibrary Paths Bource Location B References	Manage Configurations.
Documentation File Types Formatter Indexer Language Mappings Preprocessor Include Profiling Categories Linux Tools Path Project Natures Project References Run/Debug Settings > Task Repository Task Tags > Validation WikiText	Languages Assembly GNU C GNU C++	Include directories	Add Edit Delete Export Move Up Move Down

NOTE: For C & C++ applications, rest of the steps will be similar except Language selection and until further depicted. Hereby, continuing further description with C language selected for C application.

Click "Add" button, which will pop up the "Add directory path" Window as shown below in the figure

Click "workspace" button.

Figure 61(a): Add Directory Path-workspace





Expand "RFIDSample4App" folder, Select "inc" folder from workspace and Click "OK"

Figure 61(b): inc folder selection from workspace



🕲 🙂 Properties for R	FIDSample4App		
type filter text 🛛	Paths and Symbols	😵 🗊 Folder selection	↓ ↓ ↓
 Resource Builders C/C++ Build Build Variables Environment 	Configuration: Debug [Active]	Select a folder from workspace:	Manage Configurations)
Logging	Languages	> 🔁 Debug	Add
Settings Tool Chain Editor	Assembly		Edit
▼ C/C++ General	GNU C	Corroject	Edit
Code Analysis	GNU C++	☑ .project	Delete
Documentation File Types			Export
Indexer			Move Up
Language Mapping Paths and Symbols Preprocessor Inclu Profiling Categorie Linux Tools Path Proiert Natures			Move Down
Project References Run/Debug Settings		Cancel OK	
 Task Repository Task Tags 	(i) "Preprocessor Include Paths, N	Macros etc." property page may define additional entries	
 Validation WikiText 	Show built-in values		
	📽 Import Settings 🖗 Export	Settings	
			Restore <u>D</u> efaults <u>Apply</u>
1			Cancel Apply and Close

Ensure the directory is taken up correctly and "is a workspace path" gets enabled

Click "OK"

Figure 61(c): Added inc folder in workspace

Properties for R	FIDSample4App		
type filter text 🛛	Paths and Symbols		⇔ → ⇔ → →
 Resource Builders 	Configuration: Debug [Active]	→ Library Paths BSource Location B References	Manage Configurations
	Languages Assembly CNU C GNU C++	Include directory path Directory: (RFIDSample4App/Inc Add to all configurations Add to all languages ✓ S is a workspace path ✓ File system OK Cancel File system	Add Edit Delete Export Move Up Move Down
Validation WikiText	☑ Show built-in values ෯ Import Settings) இ Export Set	ttings Restor	re <u>D</u> efaults <u>Apply</u> cel <u>Apply and Close</u>

Click "Add" button, which will pop up the "Add directory path" Window as shown below in the figure Click "Filesystem" button.

Figure 61(d): Add Include Directory Path-File system

🛛 🐵 Properties for R	FIDSample4App		
type filter text 🛛	Paths and Symbols		↓ ↓ ↓ ↓
 Resource Builders C/C++ Build 	Configuration: Debug [Active	1	▼) (Manage Configurations)
Build Variables Environment	BIncludes	aries 😞 Library Paths 😂 Source Location 🖻 References	
Logging Settings	Languages	Include directories	Add
Tool Chain Editor	Assembly	<pre>%/RFIDSample4App/inc</pre>	Edit
 C/C++ General Code Analysis 	GNU C++	Add directory path	Delete
Documentation File Types		Directory:	Export
Formatter Indexer Language Mapping Paths and Symbols Preprocessor Incluin Profiling Categorie		 Add to all configurations Add to all languages Workspace File system 	Move Up Move Down
Project Natures Project References Run/Debug Settings Task Repository		OK Cancel	
Task Tags • Validation WikiText	 "Preprocessor Include Paths, Show built-in values Import Settings 	Macros etc." property page may define additional entries	
			Restore <u>D</u> efaults <u>Apply</u>
?			Cancel Apply and Close

Add include path by navigating to

"[Installation-path]/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/RFID_C_API/include" directory as shown below in the figure. And click "OK" button.



Figure 62: RFID A API include path

80		_	_
🛇 Recent	guest Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux RFID_C_API	include	► C 7
✿ Home	Name	Size	Modified
🗖 Desktop	📰 rfidapi.h	42.0 kB	1 Арг
Documents	rfidapiConstants.h	36.5 kB	1 Арг
🕹 Downloads	rfidapiErrors.h	9.6 kB	1 Арг
- Music	📷 rfidapiStructs.h	125.0 kB	1 Apr
	indapitypes.n	2.2 KD	ГАрг
Pictures			
Videos			
eclipse			
+ Other Locations			
Select a folder from	ile system:		
		Cancel	ок

Click "Ok" Button under Add directory path window

Figure 63: Add directory path-File system

SOB new							
	type filter text	Paths and Symbols			, , , , , , , , , , , , , , , , , , ,		
Project 5	 Resource Builders C/C++ Build 	Configuration: Debug [Active]	•	Manage Co	nfigurations	cess	
Project E S	 C/C++ General Code Apalysis 	Add directory path	ALibrary Pat	hs 😕 Sourc	ce Location 💙	R 2 (× ≤
▼ BRFIDSam ► Sinclude	Documentation	Directory:			Add	on.h	vchar tí
▼⊜inc	File Types Formatter	ries-Embedded-Native-SDK-C-CPP_Linux/RFI	D_C_API/include	2	Edit	VENT	S
🕨 🍋 comi Comi 🗸 🔂 src	Indexer	 Add to all configurations Add to all languages 	Variables		Delete	idex : l intCall	JINT32[] back(RF
🕨 🔂 comi	Paths and Symbols	 □ ⇒ Is a workspace path 	Workspace		Export	entLoo	k : pthre
REID	Preprocessor Incluc Profiling Categories		File system		Move Up	EM_C	DUNT
	Linux Tools Path	ОК	Cancel	lentries	Move Down	DCOU	NT
	Project Natures Project References Refactoring History Run/Debug Settings	Market Strow Builden Values)			orySta oryCor	rtSemap npleteSe
			Resto	ore Defaults	Apply	Compl entSer	eteSem; naphore
	?		Car	ncel	pply and Close	Event/ orySta	Awaitin <u>c</u> rtSemap
					- Inven	τογγοι	npleteS
		Problems 🛱 🧔 Tasks 📮 Console 🗔 Properti	ies 👭 Call Grap	h		다 🕈	~ 0
	742	errors, 0 warnings, 0 others (Filter matched 10	0 of 742 items)				
	D	escription	Resource	Path	Locatio	n	Туре
	•	Errors (100 of 742 items)					
💕 RFIDSample	4App		1				



Figure 64(a): C project include path view

e	Paths and Symbols		$\phi + \phi +$
Resource Builders	Configuration: Debug [Active]	•] [•	lanage Configurations.
C/C++ General	🕒 Includes 🛛 # Symbols 🛋 Librar	ries 👼 Library Paths 😕 Source Location 🖻 References	
 Code Analysis Documentation 	Languages	Include directories	Add
File Types	Assembly	<pre>/RFIDSample4App/inc</pre>	Edit
Indexer	GNU C	//home/guest/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux_V1.0.4/RFID_C_API/inc	Delete
Language Mapping Paths and Symbols			Export
Preprocessor Incluc			Move Up
inux Tools Path			Move Down
Project Natures			
Project References			
Task Repository			
Task Tags			
Validation			
VIKITEXC			
	 "Preprocessor Include Paths, I 	Macros etc." property page may define additional entries	
	Show built-in values		
	🚳 Import Settings) 🌾 Expor	t Settings	
		Restore	Defaults Apply

Figure 64(b): C++ project include path view

type filter text 🛛 🔊	Paths and Symbols		⇔ + ⇔ + +
 Resource Builders C/C++ Build 	Configuration: Debug [Active]	•]	Manage Configurations
✓ C/C++ General	Mincludes # Symbols	es Elibrary Paths Bource Location References	
Code Analysis			
Documentation	Languages	Include directories	Add
File Types	Assembly	/RFIDSample4App/inc	Edit
Formatter	GNUC	//www.st/zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/RFID_C_API/include	
Indexer	GNU C++		Delete
Language Mapping			Export
Paths and Symbols			
Preprocessor Incluc			Move Up
Profiling Categorie:			
Project Natures			Move Down
Project References			
Refactoring History			
Run/Debug Settings			
Task Repository			
Task Tags			
Validation			
WikiText			
	"Preprocessor Include Paths A	" acros etc." property page may define additional entries	
	Show built-in values	neros cer property page may derme adartematements	
	import Settings	Settings	
		Resto	re Defaults Apply
3		Can	cel Apply and Close

To Add "Library Paths", click on "Library Paths" tab as shown in the figure 65. Add following Library paths

- 1. "[Installation-path]/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/RFID_C_API/lib"
- 2. "[Installation-path]/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/rootfs/usr/lib"
- 3. "[Installation-path]/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/rootfs/lib"



Figure 65: Add library paths in C application and as well as the same applicable to C++ application.

type filter text 🛛 🔊	Paths and Symbols	⇔ + ⇔ + +
 Resource Builders C/C++ Build 	Configuration: [Debug [Active]	onfigurations
▼ C/C++ General	🔒 Includes # Symbols 🛋 Libraries 🖪 Library Paths 😂 Source Location 🕑 References	
Documentation	/home/guest/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/RFID_C_API/lib	Add
File Types	/home/guest/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/rootfs/usr/lib	Edit
Formatter	/home/guest/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/rootfs/llb	Edit
Indexer		Delete
Language Mapping		Export
Paths and Symbols		
Profiling Categorie:		Move Up
Linux Tools Path		Move Down
Project Natures		
Project References		
Refactoring History		
Run/Debug Settings		
Task Tags		
Validation		
WikiText		
	The "Programmer include Baths, Massaciats, " proparty page may define additional extrins	
	Show built-in values	
	Restore Defaults	Apply
~		
(?)	Cancel	Apply and Close

Click Libraries tab, Click on Add button,

In the Pop Up "Add" Window

Enter "rfidapi32" as the library name as shown in the figure 66 and click OK button.

Figure 66: rfidapi32 view

4	🚇 Includes 🛛 # Symbols 🛋	ibraries 🛛 👼 Library Paths 🛛 😂 Source Location	n 🖻 References	
n				Add Edit
pping hbols Incluc gorie: th s nces tings y		 Add File: rfidapi32 Add to all configurations Add to all languages > Is a workspace path 	Variables Workspace File system OK Cancel	Delete Export Move Up Move Down

Likewise add other set of libraries (gnutls, nettle, curl, xml2, ssl, ltk, crypto, ssh2, z, pthread, idn, hogweed, gmp, unistring). Below are figure 67(a) and 67(b) shown after adding the list. This is applicable to both C/C++ application.



Figure 67(a): library list view 1



Figure 67(b): library list view 2

😑 💷 Properties for RFI	DSample4App
type filter text	Paths and Symbols $\diamond \neg \diamond \neg \neg$
 Resource Builders C/C++ Build 	Configuration: Debug [Active]
▼ C/C++ General	 ✓ Bincludes # Symbols ■ Libraries ● Library Paths ⊘ Source Location
Code Analysis	
Documentation	BAltk Add
File Types	Edit
Formatter	₽Å ssh2
Indexer	Delete Delete
Language Mapping	Phread Export
Paths and Symbols	að idn
Preprocessor Incluc	The second secon
Profiling Categories	DA gmp
Linux Tools Path	Move Down Move Down
Project Natures	Using relative paths is ambiguous It can cause unexpected effects.
Project References	Show built-in values
 Run/Debug Settings Task Repository 	Restore Defaults Apply
?	Cancel Apply and Close

In Properties Window for RFIDSample4App

Expand "C/C++ Build" section

Navigate to "C/C++ Build" ->Settings->Tool Settings->Cross Settings

Click "Cross Settings" as shown in the figure below

Make sure Prefix is "arm-montavista-gnueabi-"

Make sure Toolchain path is set to "[Installation-path]/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/tools/armhfp-gnueabi/bin"

Figure 68: Cross Settings





For C application, set Cross GCC Compiler flags:

In Settings->Tool Settings->Cross GCC Compiler->Miscellaneous-> Other flags,

enter extra flags of below as shown in Figure 69(a).

"-c-fmessage-length=0-DUNICODE -mfloat-abi=hard"

Figure 69(a): Cross GCC Compiler flag settings





For C application, set Cross GCC Linker flags:

In Settings->Tool Settings->Cross GCC Linker->Miscellaneous-> Linker flags,

enter extra flags of below as shown in Figure 69(b).

"-DUNICODE -mfloat-abi=hard"

Figure 69(b): Cross GCC linker flag settings



For C++ application, set Cross G++ Compiler flags:

In Settings->Tool Settings->Cross G++ Compiler->Miscellaneous-> Other flags,

enter extra flags of below as shown in Figure 69(c).

"-c -fmessage-length=0 -DUNICODE -mfloat-abi=hard"

Figure 69(c): Cross G++ compiler flag settings





For C++ application, set Cross G++ Linker flags:

In Settings->Tool Settings->Cross G++ Linker->Miscellaneous-> Other flags,

enter extra cross linker flags of below as shown in Figure 69(d).

"-DUNICODE -mfloat-abi=hard"

Figure 69(d): Cross G++ linker flag settings:

type filter text	Settings		⇔ + ⇔ + +
 kype filter text Resource Builders C/C++ Build Build Variables Environment Logging Settings Tool Chain Editor C/C++ General Linux Tools Path Project Natures Project References Refactoring History Run/Debug Settings Task Repository Task Tags Validation WikiText 	Settings © Tool Settings Container Sett © Cross GCC Compiler © Dialect © Preprocessor © Includes © Optimization © Debugging © Warnings © Miscellaneous © Cross G++ Compiler © Dialect © Preprocessor © Includes © Optimization © Debugging © Warnings © Miscellaneous © Consect + Linker © Consect + Linker © Consect + Linker © Consect + Linker	tings PBuild Steps PBuild Artifact BBinary Parsers Fror Parsers Linker flags -DUNICODE -mfloat-abi=hard Other options (-Xlinker [option]) Other objects Other objects	
	 ➢ Miscellaneous ➢ Shared Library Settings ▼ ➢ Cross GCC Assembler ➢ General 		
?			Restore Defaults Apply Cancel Apply and Close



You may find this pop up, click "Yes" button on the Settings window.

Figure 70: Cross GCC Linker view

😣 Settings				
?	Changes made will not be reflected in the index until it is rebuilt. Do you wish to rebuild it now?			
Re	member my decision			
	No Yes			

7.4 Build/Debug Embedded Native RFIDSample4App C/C++ Project.

The building and debugging the C/C++ application project is applicable as mentioned in section 6.0.

ৠ**™• ZEBRA**

START AND STOP SCRIPTS FOR RFID INSTALLATION PACKAGE

8.0 Create Start and Stop Scripts of C/C++ Installation Package

Creating Start and Stop Scripts for C/C++ Installation Package

1. Copy start_sampleapp.sh and stop_sampleapp.sh from: [Embedded SDK Install folder]/Zebra-FXSeries-Embedded-Native-SDK-C-CPP_Linux/samples/sampleScripts/c_c++/ into the build directory, which is the application directory (i.e., /apps)

2. Rename the script files start_sampleapp.sh and stop_sampleapp.sh to start_appname.sh and stop_appname.sh with the executable file name (existing as "RFIDSample4App") as appname.elf OR appname.

3. Replace line /apps/%sampleapp% & in start_appname.sh with /apps/appname.elf & or /apps/appname & (same as the executable name).

4. Replace the line EXECUTABLE_NAME=%sampleapp% in stop_appname.sh with EXECUTABLE_NAME=appname.elf or EXECUTABLE_NAME=appname (same as the executable name).



EMBEDDED APPLICATION INSTALLATION PACKAGE CREATION

9.0 Embedded Application Installation Package Creation

9.1 Embedded application package creation

To create an FX RFID Reader Embedded Application, install package on Ubuntu 16.04 OS based host system, follow the steps

1. Create the Debian package directory structure as shown below

RFIDSample4App_2.0.1

- ---- DEBIAN
- | └── control
- ----- RFIDSample4App (any C/C++ executable file)
- ---- start_RFIDSample4App.sh
- └── stop_RFIDSample4App.sh

Inside RFIDSample4App_2.0.1 directory there are

1 directory and 4 files

In the above directory structure, 2.0.1 is the version

"RFIDSample4App" is the build directory name. It contains one directory "DEBIAN" with single control file is explained later.

"RFIDSample4App" directory contains start and stop script along with executable file (C/C++).

2. Example details of control File



Description: Basic Debian Test

====================

Create control file containing the following fields to be updated as shown above

- 1. Package
- 2. Version
- 3. Priority
- 4. Architecture
- 5. Maintainer
- 6. Description

For further details please refer https://www.debian.org/doc/debian-policy/ch-controlfields.html

3. Create Start and Stop scripts for the embedded application in [Build folder], which is the application folder as mentioned in section 8.

4. Ensure that **dpkg-deb** is installed on the host

5. Go to the parent directory of folder 'RFIDSample4App_2.0.1/'.

6 Run the below command

dpkg-deb --build -Zgzip RFIDSample4App_2.0.1
deb package will be created in the parent directory

7. Using web UI, install the deb package on the RFID reader

NOTE: Ensure execution permission is provided for the file, the Start and Stop script. If not, use the chmod +x command to change permission of files.

NOTE: The name of the package and name of the application are the same.

NOTE: Package, Version, and Maintainer are mandatory. There are many optional fields in the control file.

9.2 Installation and Removal of application package On RFID reader using UI

Below listed steps will help in installing and uninstalling application package on RFID reader

Once login to web console of RFID reader

Step 1: Application -> Install New Package -> Browse (Select the created deb package).



Step 2: Application package should be listed in List of installed apps as shown in figure below.

Step 3: Click on Start or Stop circular button to Start or Stop the application executable.

Step 4: To Uninstall or remove application package, click on uninstall button in the figure below.

← → C ③ Not secure	192.168.6.58/readerindex.html	Image: Constructive EZE Sci Image: Constructive EZE Sci <t< th=""></t<>
र्शेन् JEBRA		FX9600
Home	User Application Page	Applications ?
Satus > Operation Statistics > Configure Reader Read Tags > Communication Date Time IP Sec License Manager Change Password GPIO Applications Profilies > Firmware Commit/Discard > System Log Diagnostics Shutdown Logout	Existing Packages:	 This page provides the details of installed application and also to install applications in the reader. List of installed Apps. This drop down menu shall list the current packages installed in the reader. Start/Stop. The image detaysity the running status as indicated below. Click the image to toggle the status. Indicates App is NOT running Indicates App is NOT running. AutoStart - Selecting this check box shall enable the application to run at startup. Uninstall - Shall remove the package from reader. How to create packages: Packages can be created using any of standard deban package creation tooks or manually. The guidelines for package reaction for K Start Start, Deckage 1, 2, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
	© Copyright 2019 Zebra Technologies, All Rights Reserved	

Figure 71: User Application Page

Login to Reader through remote terminal/console as user 'rfidadm' and execute RFID Sample application as '/apps/RFIDSample4App. Note: The using of start and stop circular button/auto start is not applied to RFID sample C/C++ application since it will be executed as background process, which requries user inputs from console.