

Application Note

AN_325

FT9xx Toolchain Installation Guide

Version 1.04

Issue Date: 2017-03-08

This guide documents the tools and methods required for building, programming and debugging the FT9xx series devices from BRT Chip.

Use of Bridgetek devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold Bridgetek harmless from any and all damages, claims, suits or expense resulting from such use.

> Bridgetek Pte Limited (BRT Chip) 178 Paya Lebar Road, #07-03, Singapore 409030 Tel: +65 6547 4827 Fax: +65 6841 6071 Web Site: <u>http://www.brtchip.com</u> Copyright © Bridgetek Pte Limited



Table of Contents

1 F	T9xx Toolchain Introduction	3
1.1	Compiler: ft32-elf-gcc	.3
1.2	Assembler: ft32-elf-as	.3
1.3	Linker: ft32-elf-ld	.3
1.4	Debugger: ft32-elf-gdb	.4
1.5	A useful utility: ft32-elf-objdump	.4
2 S	etting up the FT9xx Toolchain	5
2.1		
	1.1 Installing Java Runtime Environment Manually	
2.2	Verifying the installation	
	uick Start Guide: From creating to getting your	
-	ication to run on the FT9xx MCUs	.6
	Creating a new project	
3.2		
3.3		
3.3	3.1 GUI Version	
3.3	3.2 Command Line Version	24
3.4	"Hello World" in action, and more	25
	etting up Eclipse for Debugging2	
	Build the application using the Debug configuration	
4.2	Create a new debug configuration	28
4.3	Running the GDB Bridge	30
4.4		
4.4	4.1 Watch variables in Eclipse Debug Perspective	
4.4	4.2 Og Compiler Option when debugging	33
4.5	Eclipse features supported by ft32-elf-gdb	33
5 B	ridgetek Projects	34
	dvanced Topics	
•		-



6.1	Ru	nning the toolchain from the command prompt	36
6.	1.1	Compiling the sample applications using a Makefile	
6.	1.2	Programming a binary file into the chip	
6.	1.3	Debugging the sample applications with ft32-elf-gdb	
6.2	In	stalling Eclipse and the FT9xx plugin manually	37
6.	2.1	Eclipse Installation	
6.	2.2	FT9xx Eclipse Plugin Installation	39
6.3	Со	mmon project settings in Eclipse	40
6.	3.1	Include paths	
6.	3.2	Toolchain settings	
6.	3.2.1	FT32B options (only for FT93x)	
6.	3.3	C/C++ Indexer Settings	43
7 T	roul	bleshooting	44
7.1	Ма	kefile error	44
8 C	onta	act Information	45
Δnn	endi	ix A – References	
_		ent References	-
Acr	onyı	ms and Abbreviations	46
Арр	end	ix B – List of Tables & Figures	47
List	t of F	-igures	47
Арр	endi	ix C – Revision History	49



1 FT9xx Toolchain Introduction

The free FT9xx toolchain is a port from the popular GNU toolchain which includes the following components:

- GCC based compiler
- GNU Binary Utilities (binutils) based tools, most notably:
 - as the assembler
 - o Id the linker
 - and some other useful tools such as objdump, ar, ranlib, addr2line, etc.
- GDB based debugger
- In addition, a plugin for the Eclipse IDE is also provided. This plugin allows the FT9xx toolchain to integrate seamlessly into Eclipse and as a result, greatly simplifies the development works for the FT9xx MCUs.

1.1 Compiler: ft32-elf-gcc

The FT9xx compiler is used similarly to standard GCC. It supports most GCC options such as -Wall, -O1, -O2...

Example: To compile a C file into an object file:

ft32-elf-gcc -c -o file.o file.c

1.2 Assembler: ft32-elf-as

The FT9xx assembler functions in the same way as the standard GNU assembler (GAS). The assembly files should be written using the GAS general syntax.

Example: To compile an assembly file into an object file:

ft32-elf-as -o file.o file.s

1.3 Linker: ft32-elf-ld

Typically running behind ft32-elf-gcc, the FT9xx linker performs two tasks. It first links all object files and libraries into a.out and then convert's a.out into an executable file for FT9xx. Similar to the FT9xx compiler and assembler, the FT9xx linker supports most standard GNU linker options.

Example:

- To link various object files / libraries into an .elf file:
 ft32-elf-gcc -nostartfiles file1.o file2.o -L <libfolder> -l lib1 -l lib2 -o file.elf
- To convert file.elf into a FT9xx binary file, which can be programmed into the chips: ft32-elf-ld --oformat binary -o file.bin file.elf



1.4 Debugger: ft32-elf-gdb

The Bridgetek programmer/debugger module is needed for the communication between ft32-elfgdb and the chip. The communication follows the GDB remote protocol. In addition to the debugger module, two software components are needed:

- GDB Bridge: for converting GDB commands into the debugger module commands
- Bootloader: for receiving & executing the debugger module commands

More information on how to use the FT9xx debugger can be found in $\frac{\text{section } 5.1.3}{\text{document.}}$ of this document.

1.5 A useful utility: ft32-elf-objdump

ft32-elf-obj dump displays various information about object files. Its usage is the same as standard GNU objdump.

Example: To disassemble file.elf into a text file

ft32-elf-objdump -d file.elf > disassembly.txt



2 Setting up the FT9xx Toolchain

2.1 Installing the toolchain

The toolchain can be installed by running the setup wizard "FT9xx Toolchain Setup_version.exe", which can be downloaded from the <u>Bridgetek website</u>. Please follow the steps in the wizard to complete the installation process. It is recommended to use the default settings for simplicity.

Note: all applications should be closed before the installation or a restart may be required.

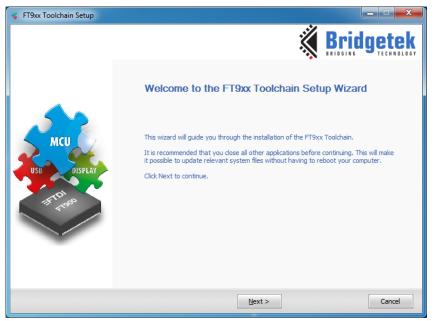


Figure 1 Toolchain Setup Wizard Dialog box

In the License Agreement dialog box, click I Agree.

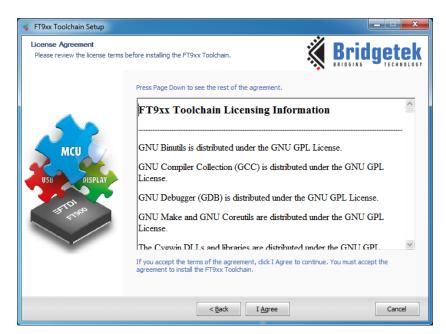


Figure 2 License Agreement Dialog box



Go through the Revision and Release information and click Next.



Figure 3 Revision and Release Information Dialog box

Select the Components and click **Next**.

🔹 FT9xx Toolchain Setup			
Choose Components Choose which features of the FT	9xx Toolchain you want to install.	Ķ	
	Check the components you want Click Next to continue.	to install and uncheck the compon	ents you don't want to install.
MCU USB DISPLAY TH TO THE TO SEA	Select components to install: Space required: 736.9MB	FT9xx Toolchain F F19xx Toolchain F Eclipse IDE for FT9xx F Clipse IDE for FT9xx F GnuWin32 Make 3 GnuWin32 CoreUt F Java 8 Update 12 F Python 2.7.10 F Python 2.7.10 F FT9xx Example Applica FT9xx Example Applica V Update PATH variable F Create icons on deskto	Description Position your mouse over a component to see its description.
	< <u>B</u> ack	Next >	Cancel

Figure 4 Components Dialog box



Click **Browse** and select a different file path for the FT9xx Toolchain installation. Alternately, continue installing in the specified folder by clicking **Next**.

🤹 FT9xx Toolchain Setup		
Choose Install Location Choose the folder in which to in:	stall the FT9xx Toolchain.	Bridgetek
	Setup will install the FT9xx Toolchain in the following folder. To inst Browse and select another folder. Click Next to continue.	all in a different folder, dick
MCU USB DISPLAY TATI DI SPS	Destination Folder	Browse
	< <u>B</u> ack <u>N</u> ext >	Cancel

Figure 5 FT9xx Toolchain Install Location Dialog box

Click **Browse** and select a different file path for installing FT9xx examples and documents. Alternately, continue installing in the specified folder, by clicking **Install**.

🤹 FT9xx Toolchain Setup		
Choose Install Location Choose the folder in which to in	istall the FT9xx Toolchain.	Bridgetek
MCU USB DISPLAY TETTONSSS	Please select the installation folder for the FT9xx examples and docu	iments.
	< Back Install	Cancel

Figure 6 FT9xx Toolchain-Examples & Documents Install Location Dialog box



The FT9xx Toolchain installation progress bar is displayed.

🤹 FT9xx Toolchain Setup		
Installing Please wait while the FT9xx Too	chain is being installed.	r k Y
MCU USB DISPLAY THT TO GOD	Extract: hashtable.h Extract: tipe 100% Extract: type_traits 100% Extract: type_maits 100% Extract: typeindex 100% Extract: unordered_map 100% Extract: unordered_map 100% Extract: valarray 100% Extract: valarray 100% Extract: valarray 100% Extract: vector 100% Extract: auto_ptr.h 100% Extract: backward_warning.h 100% Extract: hash_fun.h 100% Extract: hash_fun.h 100% Extract: hash_spet 100% Extract: hash_spet 100%	
	< Back Next > Cancel	

Figure 7 FT9xx Toolchain - Installation Progress Window

If Java is selected for installation, the following message is displayed. Please close any open Java Applications and click $\mathsf{O}\mathsf{K}$

🔹 FT9xx Toolchain Setup	
Installing Please wait while the FT9xx Toolchain is being installed.	getek
USB Setup will now run the Java Runtime (JRE) Installer. Please close any open Java Applications (like Eclipse) before proceeding. OK	
Extract: coreutils.mo 100% Output folder: C: \Program Files\Bridgetek\FT9xx Toolchain\External\GnuWin32\s Output folder: C: \Program Files\Bridgetek\FT9xx Toolchain\External\GnuWin32 Output folder: C: \Program Files\Bridgetek\FT9xx Toolchain\External\Java8 Extract: jre-8u121-windows-i586.exe	ihare∖oca ∨
< Back Next >	Cancel

Figure 8: Close any running Java Applications before launching JRE Installer



Setup will then launch the JRE installer and the following window will be displayed



Figure 9 Java Setup Window

Click **Install** and follow the instructions to install Java on the machine.



Figure 10 Java Setup Progress Window

During installation, if a **Python 2.7.10 Setup** dialog box is displayed, select the appropriate option as required and click **Next**.



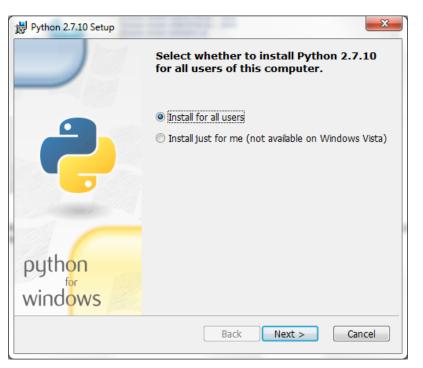


Figure 11 Python Setup Dialog box

Select a different Destination Directory to setup Python. Alternately, continue installing in the specified folder by clicking **Next**.

岃 Python 2.7.10 Setup		×
	Select Destination Director	
e.	Python27	- Up New
python windows	C:\Python27\	Cancel

Figure 12 Destination Directory Selection Dialog box for Python Setup

Select the Python features to be installed or continue with the default features and click **Next**.





Figure 13 Python Features Customization Dialog box



Python installation progress bar is displayed.

Bython 2.7.1	.0 Setup	×
Install Pyt	hon 2.7.10	
	wait while the Installer installs Python 2.7.10. This may take minutes.	
Status:	Copying new files	
	< Back Next > Cancel	

Figure 14 Python Installation Progress Dialog box

Click **Finish** to complete the Python installation.



Figure 15 Python Installation Completion Dialog box



The FT9xx Toolchain installation is continued.

🤹 FT9xx Toolchain Setup		- 🗆 X
Installing Please wait while the FT9xx Tool	chain is being installed.	getek
MCU USB DISPLAY 75 TO 75 SP	Extract: structboms_scsi_c_r_b_inquiry_t.html Extract: struct_u_s_b_hc_d_c_a_c_m_context.html 100% Extract: struct_u_s_b_hf_232_context.html 100% Extract: struct_u_s_b_h_ctx.html 100% Extract: struct_u_s_b_h_device_info.html 100% Extract: struct_u_s_b_h_interface_info.html 100% Extract: struct_u_s_b_h_interface_info.html 100% Extract: struct_u_s_b_h_interface_info.html 100% Extract: structboms_command_status_wrapper_t.html 100% Extract: structboms_command_status_wrapper_t.html 100% Extract: structboms_command_status_wrapper_t.html 100% Extract: structboms_com_command_status_wrapper_t.html 100% Extract: structboms_scsi_c_d_b_inquiry_t.html 100% Extract: structboms_scsi_c_d_b_read_capacity_t.html 100% Extract: structboms_scsi_c_d_b_read_thill 100%	
	< <u>B</u> ack <u>N</u> ext >	Cancel

Figure 16 FT9xx Toolchain - Installation Progress Window

Select the **Open AN_325** checkbox to start immediately after closing the Setup Wizard. Else leave it unchecked. Click **Finish** to complete the FT9xx Toolchain Setup.



Figure 17 FT9xx Toolchain Setup Completion Dialog box



After the installation, the toolchain can be found in the installation directory. The default location is "C:\Program Files\Bridgetek\FT9XX Toolchain" for 32-bit Windows and "C:\Program Files (x86)\Bridgetek\FT9XX Toolchain" for 64-bit Windows. This directory also contains the external utilities needed. The FT9xx drivers, sample applications and documents (if selected for installation) can be found in "My Documents\Bridgetek\FT9xx".

2.1.1 Installing Java Runtime Environment Manually

The toolchain requires the Windows MSI Installer (msiexec.exe) while installing the Java Runtime Environment (JRE). The MSI installer can only process one installation at a time. Under some conditions, msiexec.exe may have already been started by another Windows process during automatic Windows Update for example. If the installer detects another instance of msiexec.exe running in the background, the user will be prompted to either wait for the background MSI Installer to complete and retry after 5 seconds or to skip the JRE installation entirely. This is shown in Figure 18.

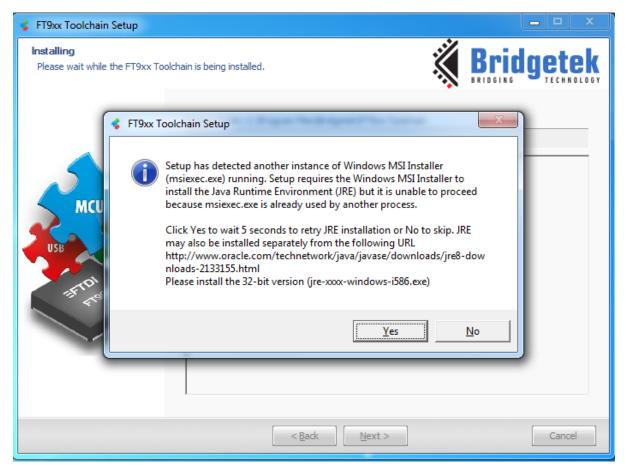


Figure 18: MSI Installer is busy

If the user skips JRE installation, JRE can be installed manually from the Oracle Website (<u>http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html</u>) or by re-running the FT9xx Toolchain Installer later. Please ensure to install the 32-bit version of JRE (jre-xxx-windows-i586.exe) as the Eclipse installed as part of the FT9xx Toolchain install is 32-bit.



2.2 Verifying the installation

- 1. Open a Command Prompt window by typing "cmd" in "Windows Start button \rightarrow Search box".
- 2. Type "ft32-elf-gcc --version" in the command prompt. It should give the following message:

ft32-elf-gcc (GCC) 7.0.0 20161219 (experimental) Copyright (C) 2016 Free Software Foundation, Inc. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

If this message appears, then the toolchain has been successfully setup.



3 Quick Start Guide: From creating to getting your application to run on the FT9xx MCUs

This chapter guides you through the steps to create a new application, compile and program it into the chip. To debug your application, please refer to <u>chapter 4 - "Setting up Eclipse for Debugging"</u>. For more information about the tools, as well as the advanced features, refer to <u>chapter 5 - "Advanced Topics"</u>.

3.1 Creating a new project

Double click on the icon "Eclipse for FT9xx" to launch the Eclipse IDE.



Figure 19 Eclipse for FT9xx Icon

When you run Eclipse for the first time, it will ask you for the location of the workspace. Eclipse will create some files within this directory to manage the projects. Specify a folder of your choice and click OK.

G Workspace Launcher	×
Select a workspace	
Eclipse stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace:	▼ <u>B</u> rowse
<u>U</u> se this as the default and do not ask again	OK Cancel

Figure 20 Eclipse Workspace Selection

Note: The following message will be displayed if an existing workspace, which was created by an older version of Eclipse, is specified. As there may be some configurational changes in files related to workspace in the newer version of Eclipse which may cause issues, it is recommended to create a new workspace and imported the existing projects there.



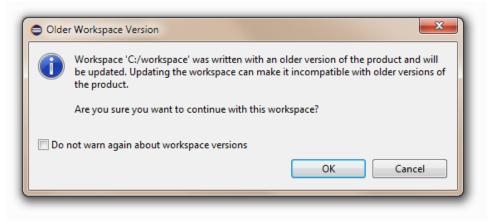


Figure 21 Eclipse Workspace Update

To create a new C project in Eclipse, on the menu bar click "File \rightarrow New \rightarrow C Project". The C Project wizard will open.

Give a name to the project, for example "Hello World". By default, the new project will be created inside the workspace you have chosen. If you want to change it, uncheck the box "Use default location" and specify another location. Choose **"Empty Project"** for the project type and **"Bridgetek FT9xx GCC"** for the toolchain. This ensures all the relevant FT9xx include files are part of the project. Click Next.

C Project				
C Project Create C project of selected type				
Project name: Hello World				
Use default location				
Location: C:\workspace\Hello Wo	rld B <u>r</u> owse			
Choose file system: defa	ault 🔻			
Project type:	Toolchains:			
Project type: Fight t				
Show project types and toolchains only if they are supported on the platform				
? < <u>B</u> ack	<u>N</u> ext > <u>F</u> inish Cancel			

Figure 22 C Project Wizard



In the next window, select FT900 _* configurations if your project is to target FT900 series of MCU or FT930_* to target FT930 series, or both if you wish to target both series and click **Next**.

C Project	
Select Configurations Select platforms and configurations you wish to deploy on	
Project type: Executable Toolchains: Bridgetek FT9xx GCC Configurations:	
♥ ♥ FT900_Debug ♥ ♥ FT930_Release ♥ ♥ ♥ FT930_Debug ♥ ♥ FT930_Release	Select all Deselect all
Use "Advanced settings" button to edit project's properties.	Advanced settings
Additional configurations can be added after project creation. Use "Manage configurations" buttons either on toolbar or on prop	erty pages.
? < Back Next > Finis	h Cancel

Figure 23 Project Wizard - Build Configurations Selection

The last window is for the toolchain prefix and location. By default, the values will be prefilled as follows.

FT90x compiler prefix:	ft32-elf-	
FT90x compiler path:	C:\Program Files (x86)\FTDI\FT90x Toolchain\tools\bin	Browse
_		

Figure 24 C Project Wizard - Toolchain Details

Click Finish to complete the New Project Wizard. A new FT9xx project will be created in Eclipse.

3.2 Building the project

After the wizard completes, some folders and an empty source file (main.c) will be created.



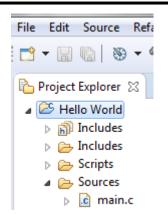


Figure 25 New empty project structure

The main.c contains some default content:

```
#include <stdint.h>
#include <ft900.h>
#if defined( FT900
                  )
#define GPIO_UART0_TX
                       48
#define GPIO UART0 RX
                       49
#elif defined( FT930
#define GPIO_UART0_TX
                       23
#define GPIO_UART0_RX
                       22
#endif
void main(void){
   /* Enable the UART Device... */
   sys_enable(sys_device_uart0);
   /* Set UART0 GPIO functions to UART0_TXD and UART0_RXD... */
   gpio_function(GPIO_UART0_TX, pad_uart0_txd); /* UART0 TXD */
   gpio_function(GPIO_UART0_RX, pad_uart0_rxd); /* UART0 RXD */
   uart_open(UART0,
                                    /* Device */
                                   /* Prescaler = 1 */
            1,
            UART_DIVIDER_19200_BAUD, /* Divider = 1302 */
            uart_data_bits_8, /* No. Data Bits */
uart_parity_none, /* Parity */
                                  /* No. Stop Bits */
            uart_stop_bits_1);
   /* Print out a welcome message... */
   uart_puts(UART0,
       "----- \r\n"
       "Hello World! \r\n"
       "----- \r\n"
       );
   /* Now keep looping */
   while (1);
```

}

Now select the Build Configuration for either FT90x or FT93x via the project menu:



	Open Project Close Project	8 - 9 - 12 ⊘ ⊘ ≁ №	1	[]图·图·学令·
010	Build All Ctrl+B			
	Build Configurations	Set Active	•	1 FT900_Debug
 Image: A start of the start of	Build Project Build Working Set Clean Build Automatically	Manage Build by Working Set Set Active by Working Set Manage Working Sets	•	2 FT900_Release 3 FT930_Debug 4 FT930_Release
	Build Targets			

Figure 26 Build Configuration

This can also be done via the Manage Configurations toolbar icon:

9	- 🔦 - 🗟 🔍 🖗 🛷
~	1 FT900_Debug
	2 FT900_Release
	3 FT930_Debug
	4 FT930_Release

Figure 27 Build Configuration

Now the project can be built by clicking on the menu **Project** \rightarrow **Build Project**, but note that there are a few options like right-click on the project \rightarrow Build Project and the icon.

Proj	ect Run	Target	Window	Help
	Open Pro	ject		
	Close Pro	ject		
D	Build All		Ct	rl+B
	Build Cor	nfiguratio	ns	-
	Build Pro	ject		

Figure 28 Building the Project

The console window at the bottom of the IDE shows the build status. If the build completes successfully, two files will be created - "Hello World.elf" and "Hello World.bin". The file to be programmed into the chip is "Hello World.bin". The .elf file is used for the debugger, as detailed in the next chapter.



📳 Problems 🧔 Tasks 📃 Console 🔀 🔲 Properties

CDT Build Console [Hello World]

15:52:13 **** Incremental Build of configuration FT930_Release for project Hello World ****
make all
'Building file: ../Sources/main.c'
'Invoking: FT9xx GCC Compiler'
ft32-elf-gcc -D_FT930____IC:/Program Files/Bridgetek/FT9xx Toolchain/Toolchain/hardware/include" -Os -Wall -c -fmessage-len{
'Finished building: ../Sources/main.c'
''
'Building target: Hello World.elf'
'Invoking: FT9xx GCC Linker'
ft32-elf-gcc -L"C:/Program Files/Bridgetek/FT9xx Toolchain/Toolchain/hardware/lib/Release" -Wl,--gc-sections -Wl,--entry=_star

'Finished building target: Hello World.elf'

'Invoking: FT9xx Flash File Generator'

ft32-elf-objcopy --output-target binary "Hello World.elf" "Hello World.bin"

'Finished building: Hello World.bin'

Figure 29 Build Status

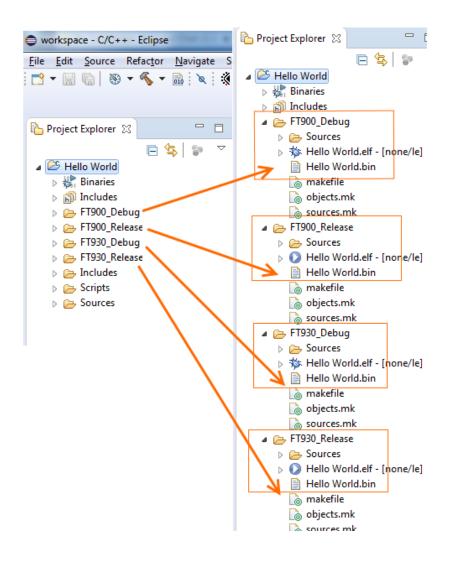


Figure 30 List of Files after building (if all 4 configurations were selected)

For syntax highlighting to work correctly as configurations are switched, the C/C++ Indexer must be configured to "work with the active build configuration". Refer to section 6.3.3 for more details on how to do this.



3.3 Programming the binary file into the chip

The FT9xx Programmer is provided together with the toolchain. There are a couple of options available.

3.3.1 GUI Version

To run it, double click on the icon "FT9xx Programming Utility" created on your desktop, if selected during install, otherwise it can be found in:

C:\Program Files (x86)\Bridgetek\FT9xx Toolchain\Toolchain\programmer\dist



Figure 31 FT900 Programming Utility Icon

You can also open the programming utility from Eclipse by selecting it in the Bridgetek Utilities menu or the toolbar icon as highlighted in Figure 32:

Brid	Bridgetek <u>U</u> tilities <u>W</u> indow <u>H</u> elp				
۲	FT9xx Programming Utility	Ctrl+Alt+U			
ſ	FT9xx GDB Bridge	Ctrl+Alt+R			

Figure 32 Bridgetek Utilities Menu

After the splash message the following screen will appear.



\$ FT9xx Programming Utility	
Program via One-Wire interface	Bridgetek
Program via USB (DFU) interface	
Add/Customize DFU-suffix (DFU binary)	
Customize Bootloader file	
Next	

Figure 33: FT9xx Programmer - Work with One-Wire

Select the "Work with One-Wire" option and click Next. The next screen shows a list of supported devices that you might wish to program.

When a valid FT900 and Programmer module are detected, the information will be displayed in the list. Select the device you wish to program and click Next to launch the programmer window.

🔹 FT9xx Programming	g Utility		X
Interface One-Wire		X Bridgetek	
Device	Programmer	Programmer Serial	
FT900	UMFTPD2A A	FTK1RIE5A	
Back	Scan	Next	

Figure 34: FT9xx Programmer - Device Selection

In the programmer window, leave everything as default. Specify the location of the binary file and click Start. If the Verify check box is selected, an icon will show up next to the status bar to indicate whether the flash memory has been properly programmer.



🔹 FT9xx Programmin	g Utility		
Flash & PM D2XX Dat	ta Log About		
Device FT900		\$	Bridgetek
Interface		Target	
One-Wire		Flash	
Binary file			Browse
Config file			Browse
		Config Address (hex)]
Verify	Exclude bootloader	Keep existing bootloader	
Progress Restore b		Start	Cancel
Scan for devic	e		Back

Figure 35 FT9xx Programmer – Flash and PM Screen

More information on the utility can be found in the 'About' tab, then click on Help.

3.3.2 Command Line Version

FT900Prog.exe is available to run at a command prompt. Enter FT900Prog.exe to see the options available. See <u>section 5.1.2</u> for more details.

This can also be run within Eclipse as an External Tool. See Figure 36 for settings found in Run \rightarrow External Tools \rightarrow External Tools Configurations. The argument string is:

-f "\${project_loc}\\${config_name:\${project_name}}\\${project_name}.bin" -0 -v



reate, manage, and run conf	figurations			_
Run a program				
° 🕼 🗶 🕒 🐎 ▼	Name: FT9xx Program MTP			
type filter text	📄 Main 🕜 Refresh 🚮 Bui	ld 🚾 Environment 🔲 Comr	non	
🛛 💁 Program	Location:			
4 FT9xx Program MTP	C:\Program Files (x86)\Bridge	tek\FT9xx Toolchain\Toolchain	\programmer\dist\FT900	Prog.exe
		Browse Workspace	Brows <u>e</u> File System	Variables
	Working Directory:			
		Browse Wor <u>k</u> space	Browse File System	Varia <u>b</u> les
	<u>A</u> rguments:			
	-f "\${project_loc}\\${config_na	ime:S{project_name}}project_	_name}.bin" -O -v	*
	Note: Enclose an argument co	ntaining spaces using double-q	uotes (").	 Variable <u>s</u>
	-			
			Revert	Analy
< <u>Ⅲ</u> ► ilter matched 2 of 2 items			reven	Apply

Figure 36 FT9xx Programmer in Eclipse

3.4 "Hello World" in action, and more...

The "Hello World" example above will send a message to a serial terminal via the FT9xx UARTO port. Open a terminal on your computer, for example Tera Term or HyperTerminal. Apply the following settings:

- Baud Rate: 19200 •
- Parity Bit: None • 8
- Data Bit: ٠
- Stop Bit: •
- Flow Control: None •

1

Now when you reset the MCU, the message will be printed to the terminal.



🇞 FT90x - HyperTerminal	x
File Edit View Call Transfer Help	
D 🛎 🐵 🕉 🗈 🎦 🖬	
>Hello World! -	-
	*

Figure 37 Hello World

Congratulations! You have just completed your first project for FT9xx. The FT9xx toolchain comes with plenty of examples, which demonstrate a variety of features. If you have selected to install them in the Toolchain Installation Wizard, by default they can be found in:

"My Documents\Bridgetek\FT9xx\version\Examples"

The Eclipse project has already been setup for these examples, as suggested by the presence of two files - ".cproject" and ".project". Instead of creating a new project, you can simply import these projects into the workspace. To do this:

- 1. On the Menu bar, choose "File \rightarrow Import"
- 2. In the Import window, choose "General \rightarrow Existing Projects into Workspace" and click "Next".
- 3. In the next window, set the root directory to "My Documents\Bridgetek\FT9xx*version*\Examples". The projects will be detected by Eclipse.
- Select which projects you wish to import and click Finish to complete the importing process. This is an example of how Eclipse would look like with the sample applications. Refer to AN 360 for more details about these applications.



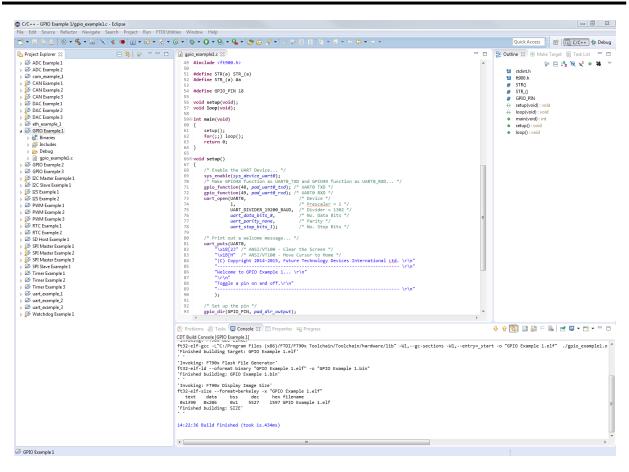


Figure 38 FT9xx Examples



4 Setting up Eclipse for Debugging

Eclipse comes with an intuitive GUI for debugging applications. To enable this feature, eclipse requires additional information about our debugger. The steps are presented below.

4.1 Build the application using the Debug configuration

The application should be built using the Debug configuration so that the debug information is available. It is the default build configuration but can be verified in the Project menu.

ch	<u>P</u> roj	ect <u>R</u> un Bridgetek <u>U</u> tilities <u>W</u> in	dow	<u>H</u> elp		
r :		Open Project Close Project		▼ 🌯 ▼ 😕 🗁 🖋 ▼ 🌛 😜 _2_testing.c 🙁	P	■ ¶ ! 2 ▼ % ▼ % <
	010	Build All Ctrl+B	hhie	_z_testing.c &		
		Build Configurations	•	Set Active	\checkmark	1 FT900_Debug
	Build Project			Manage		2 FT900_Release
	Build Working Set Clean	•	Build by Working Set Set Active by Working Set		3 FT930_Debug 4 FT930_Release	
	✓	Build Automatically		Manage Working Sets		

Figure 39 Build Configuration

Please note for FT93x: As of toolchain v2.3.0 it is recommended to disable the -mcompress option when using GDB as single stepping does not work properly with -mcompress. The option can be disabled in the project settings. See section 6.3.2.1 for more details.

4.2 Create a new debug configuration

A debug configuration is used by Eclipse to launch the debug GUI and only needs to be created once for the FT9xx Debugger. To create it:

- 1. On the Menu bar, select **Run > Debug Configurations...**
- In the Debug Configurations window, double click on "C/C++ Remote Application"
 Press the 'New' button to create a new debug configuration
- 4. In the next window, a Debugging Launcher will need to be specified. Click on "Select others..." at the bottom of the window.

Using GDB (DSF) Automatic Remote Debugging Launcher - <u>Select oth</u>	er Apply	Revert
	Debug	Close

Figure 40 Choosing a Debugging Launcher (1)

5. In the "Select Preferred Launcher" window, check "Use configuration specific settings". Then choose "GDB (DSF) Manual Remote Debugging Launcher". Click OK.



Select Preferred Launcher				
This dialog allows you to specify which launcher to use when multiple launchers are available for a configuration and launch mode.				
Use configuration specific settings Change Workspace Settings				
Launchers:				
GDB (DSF) Automatic Remote Debugging Launcher				
Standard Remote Create Process Launcher				
GDB (DSF) Manual Remote Debugging Launcher				
(?) OK Cancel				

Figure 41 Choosing a Debugging Launcher (2)

- Now provide the details for the configuration. Specify the name of the Debug configuration, for example "FT9xx Remote Debug". Use this configuration to debug FT9xx projects from now on.
- 7. Under Main tab, specify the **project** and the **.elf file** for **Application**. The "Browse..." button next to the project field will list all active projects. The .elf file can be found easily after the project has been selected, by clicking on "**Search Project...**" button.

Name: FT9xx Remote Debug	
🖺 Main 🛛 🏇 Debugger 🤤 Source 🔲 Common	
Project:	
Hello World	Browse
C/C++ Application:	Л
Debug/Hello World.elf	V
	Variables Search Project Browse

Figure 42 Eclipse Debugging Application Settings

8. Under the Debugger tab, the user needs to provide some Debugger Options. Specify the path to ft32-elf-gdb.exe (or simply "ft32-elf-gdb.exe") in the Main sub-tab and make sure the "GDB command file" field is empty. ft32-elf-gdb.exe can be located in the toolchain installation folder, under "tools\bin".

Name:	FT9xx Remote D	ebug
骨 Mai	in 🕸 Debugge	🛛 🦆 Source 🔲 Common
	p on startup at:	main
Debu	gger Options	
Mai	n Shared Libr	aries Connection
	B debugger: B command file	C:\Program Files (x86)\Bridgetek\FT9xx Toolchain\Toolchain\tools\bi Browse Browse

Figure 43 Eclipse GDB Settings



9. Under the Connection sub-tab, choose the connection type to be **TCP**. Enter **"localhost"** for the **host name** and **9998** for the **port number**.

Name: FT9xx Remote Debug				
🖺 Main 🕸 Debugger 👍 Source 🔲 Common				
Stop on startup at: main				
Debugger Options				
Main Shared Libraries Connection				
Type: TCP -				
Host name or IP address: localhost				
Port number: 9998				

Figure 44 Eclipse TCP Port Settings

10. Click "Apply" and close the window.

A debug configuration for FT9xx has now been created. To use the same configuration for other projects, simply open it and select the right project and application, as presented in step 6 above.

4.3 Running the GDB Bridge

The GDB Bridge is needed for ft32-elf-gdb to talk to the MCU. To run it, simply double click on the desktop icon "GDB Bridge".



Figure 45 GDB Bridge Icon

You can also launch the GDB Bridge from Eclipse by selecting it in the Bridgetek Utilities menu or the toolbar icon as highlighted in **Figure 46** Bridgetek Utilities Menu

Brid	getek <u>U</u> tilities <u>W</u> indow	<u>H</u> elp
٢	FT9xx Programming Utilit	y Ctrl+Alt+U
ſ	FT9xx GDB Bridge	Ctrl+Alt+R

Figure 46 Bridgetek Utilities Menu

The following window should appear:



G .						x
C:\Program Files ded OK	<pre>(x86)\FTDI\FT90x</pre>	Toolchain\To	olchain\util:	ities>D2XX	library	loa
ueu ox						-
						*

Figure 47 GDB Bridge in action

Now the tools are ready to debug the application in Eclipse.

Note: User must close this debug GDB script when debugging is finished, otherwise it may not be possible to program the device for example.

4.4 Debugging the application in Eclipse

1. Open the debug configuration that was created (FT9xx Remote Debug) and click Debug. Note that this will appear in the Debug button on the toolbar after running once.

Name: Hello World.elf	
📄 Main 🛛 🕺 Arguments 🔚 Environment 🕸	🖇 Debugger 🙀 Source 🔲 Common
Project:	
Hello World	Browse
	biowsen
C/C++ Application:	
FT900_Debug/Hello World.elf	
	Variables Search Project Browse
Build (if required) before launching	
Build Configuration: FT900_Debug	•
Enable auto build	O Disable auto build
Use workspace settings	Configure Workspace Settings
Cose workspace sectings	<u>compact vorapore stangan</u>
	Rev <mark>irt</mark> Apply
	Debug Close

Figure 48 Eclipse Run Remote Debugging

2. The Debug perspective will be opened. The execution will stop at the first line in main(), as shown below. Various debug commands (step into/over, resume, halt, stop, etc.) can now be accessed from the toolbar via buttons. Function variables, setting breakpoints and viewing



physical memory in the memory tab, along with some other debug features are also available now.

🖨 Debug - Hello World/main.c - Eclipse Platform					
File Edit Source Refactor Navigate Search Project Run Window Help					
📑 🗝 🔚 🕼 🕒 📄 🔛 💷 🛤 🖎 👁	e i⇒ ≣	ह 💌 💠 🔹 🖸	• 🏊 • 🙋 🍅 🛷 • ಶ)	
包 ▼ 得 ▼ ∜		Quick	Access 📑 🖻 🔒 Resourc	e 🗟 C/C++ 🔯 Deb	bug
🎄 Debug 🔀 🧏 🦌 🖬 🐄 🔻		(x)= Variables 🔀	● Breakp 1010 Registers 📷	Modules 🗖 🗖	
▲ C FT90x Remote Debug [C/C++ Remote Application]			🏝 📲 📄 🍧 🕽	: 🔌 🛅 🖻 🔻	
Hello World.elf		Name	Туре	Value	
 Thread #1 (Suspended : Breakpoint) main() at main.c:12 0x2f8 					
C:/Program Files (x86)/FTDI/FT90x Toolchain/tools	/bin/ft32-	•		•	
<	۴	•		Þ.	
🖻 main.c 🔀 💽 main.c			😑 🗖 📑 Outline 🗧	×	
⊕ * main.c				× ¥ ● ¥ ▽	
<pre>#include <stdio.h></stdio.h></pre>			🖬 std		
			● ma	in(void) : int	
⊖int main(void)					
<pre>printf("Hello World!\n"); while (1);</pre>					
return 0;					
}					
📮 Console 🕴 🖉 Tasks 🦹 Problems () Executables () Memory 🔳 💥 🦹 🖳 🔛 💭 🛃 💭 🛃 🖉 🖛 🗇 🗖 🗖					
FT90x Remote Debug [C/C++ Remote Application] C:/Program Files (x86)/FTDI/FT90x Toolchain/tools/bin/ft32-elf-gdb.exe (7.9.50.20150310) Find the GDB manual and other documentation resources online at:					
<pre><http: documentation="" gdb="" software="" www.gnu.org=""></http:>.</pre>					
For help, type "help". Type "apropos word" to search for commands related to "word".					
Writable	Smart In	sert 12:1			

Figure 49 Eclipse Debug Environment

Note: If there is an error message about missing source file as below, locate the source file that contains the main() function using the "Locate file..." button.

Locate the file or edit the source lookup path to include its location.
View Disassembly
Locate File
Edit Source Lookup Path

Figure 50 Eclipse Missing Source File

4.4.1 Watch variables in Eclipse Debug Perspective

If the watch variables fail to update or display incorrect values, check that the following flags exist for the debug build (they are present by default in all projects created with Bridgetek eclipse plugin)

-fvar-tracking -fvar tracking-assignments



onfiguration: FT900_Debug [Active	=]	▼ Manage Configuration
🕴 Tool Settings 🎤 Build Steps 🍳	P Build Artifact 🗟 Bin	ary Parsers 😣 Error Parsers
 FT9xx Toolchain Settings Target Board FT32B Options FT9xx GCC Compiler Dialect Preprocessor Includes Optimization Debugging Warnings 	Generate prof inform	

Figure 51 Debug flags

4.4.2 Og Compiler Option when debugging

When compiling a project with no optimization (or -00) some useful debugging information may not be generated at all, leading to possible unexpected results while debugging. To avoid this, it is recommended to turn on -0g option when no other optimization flags are used. The Bridgetek eclipse plugin does this automatically.

Note that if multiple optimization options are used, only the last option will be effective.

Configuration: FT900_Debug [Active]	▼ Manage Configurations
🛞 Tool Settings 🎤 Build Steps 🖷	Build Artifact 🗟 Binary Parsers	O Error Parsers
 FT9xx Toolchain Settings Target Board FT32B Options FT9xx GCC Compiler Dialect 	Other optimization flags -Og	

Figure 52 Og compiler optimization option

More information can be found in the GCC documentation - <u>https://gcc.gnu.org/onlinedocs/gcc/Debugging-Options.html</u> and <u>https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html</u>

4.5 Eclipse features supported by ft32-elf-gdb

At the moment, not all features of the Eclipse debug perspective are supported by ft32-elf-gdb. The current list of supported features is:

- Breakpoint creation.
- Single stepping/stepping in/stepping out of functions
- Watch variables
- Assembly instruction stepping



5 Bridgetek Projects

Besides the empty project used as the example in Chapter 3, there are several project types specific to Bridgetek. They can be found under "Others" in the project type selection window. Currently, there are two project types:

- D2XX Project
- Data Log (DLOG) Project

For more details about these project types, refer to "<u>AN 360 FT900 Example Applications</u>", which is included in the toolchain installation.

 ▲ È Executable ♦ Empty Project ♦ Hello World ANSI C Project ▷ Shared Library ▷ Static Library ▲ Others ♦ FT9xx D2XX Project ♦ FT9xx Data Logger Project ▷ Makefile project ▷ GNU Autotools 	

Figure 53 Bridgetek Project Types



The procedure to create a new project is similar to the empty project. When the wizard completes, a template source file will be generated. Below is the template generated for the D2XX project.

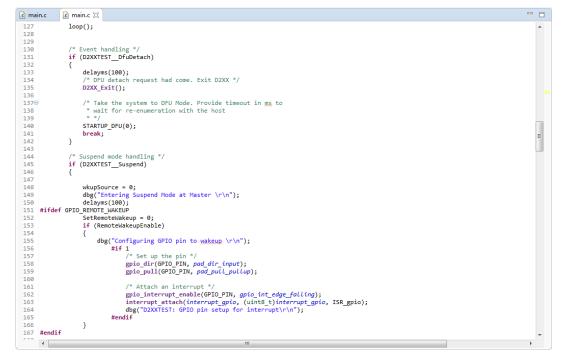


Figure 54 D2XX Project Template

The template can be compiled as it is but additional code is needed to customize it according to the user's need.

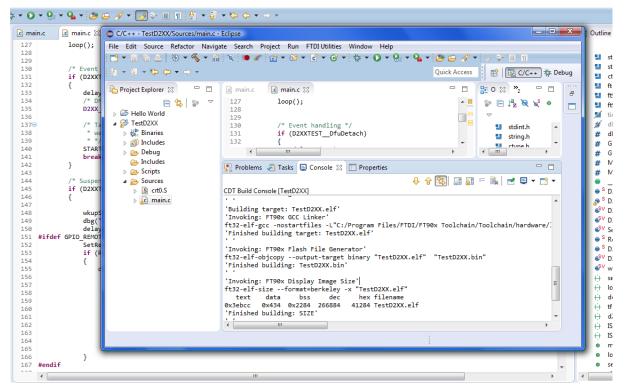


Figure 55 Compiling D2XX Template Code



6 Advanced Topics

6.1 Running the toolchain from the command prompt

6.1.1 Compiling the sample applications using a Makefile

The FT9xx GNU toolchain can be used to compile source code from a command prompt in the same way the official GNU Toolchain is used, often with the help of a Makefile or a batch file.

The sample applications are available in "My Documents\Bridgetek\FT9xx\Examples\ if you have installed them using the installation wizard.

NOTE: makefiles are not included with the toolchain installer.

6.1.2 Programming a binary file into the chip

The programmer can be found in the folder "programmer\dist" in the program installation directory (C:\Program Files (x86)\Bridgetek\FT9xx Toolchain). The command line programmer is FT900Prog.exe. The toolchain is provided with a default bootloader. The bootloader is located at the top 4 KB of the flash memory (address 0x3F000 to 0x3FFFF). At boot, the FT9xx resets and executes instruction at 0x00000, jumping into the bootloader. The bootloader then performs the initializations needed and jumps to location 0x8c, which is the start of the user program. The bootloader is also needed to support debugging with the FT9xx port of GDB.

1. Run the tool FT900Prog.exe without any arguments, the options and usage will be printed. They will also be printed if the specified options are not valid. The most common usage is programming a binary file through the one-wire interface with the supplied bootloader. To do this, the command is:

FT900Prog.exe -f < .bin file with path if needed> -0 in which the options are:

-f: programming the binary file into the **f**lash. The path to the binary file must follow. -O: using the **o**ne-wire interface.

If you want to ${\bf v} erify$ the content of the flash memory after programming, specify ``-v'' in the command:

FT900Prog.exe -f <.bin file with path if needed> -0 -v

2. If the bootloader is not required, option "-x'' can be specified, in which case the program will start executing from address zero and the command is:

FT900Prog.exe -f <.bin file with path if needed> -0 -x

The supports for GDB debugging will not be available however.



6.1.3 Debugging the sample applications with ft32-elf-gdb

1. The applications must to be compiled with -g option (i.e. ft32-elf-gcc -g ...). An .elf file will be created which includes the debug information, for example GPIO/gpio_example1.elf. Note that this file is not used for programming the chip.

Note: If the output file name for the linker is not specified in the Makefile (i.e. option -o is missing), a.out will be created instead of an .elf file. They are the same and these steps can be applied to a.out as well.

- 2. Flash the .bin file into the chip. Refer to <u>section 5.1.2</u> above.
- Open a command line window, run: "python <Installation directory>\Toolchain\utilities\gdb_bridge.py live"

Note: An alternative is to double click on the shortcut "GDB Bridge" created after the installation.

The correct response should be:

	And the Party of t	the subscription of the su			x
C:\Program Files y loaded OK	<pre>(x86)\Bridgetek\FT9xx</pre>	Toolchain\Toolchain\utilities	>D2XX	libra	ar ^

Figure 56 FT9xx Debugging Status

Note 1: If there is an error message about permission being denied, the command line window may need to be opened with administrator rights by right-clicking and selecting 'Run as administrator'.

Note 2: It is also possible to run the GDB Bridge using the shortcut created after the installation.

Note 3: If the path to gdb_bridge.py contains spaces, enclose it with double quotes ("").

- 4. Open another command line window, go to the folder that includes the .elf file, run "ft32-elf-gdb <.elf file>", for example "ft32-elf-gdb gpio_example1.elf".
- 5. After ft32-elf-gdb starts, type in "target remote localhost:9998" to establish a connection to the MCU.
- 6. Use standard GDB commands to debug the program. Note that the command to start execution should be "continue", not "run".

6.2 Installing Eclipse and the FT9xx plugin manually

When running the installer, it is possible to choose not to install Eclipse as part of the installation. This might be useful if the user have already installed Eclipse for other purposes. This section details how to set it up for use with the FT9xx.



6.2.1 Eclipse Installation

1. Go to Eclipse website, download "Eclipse IDE for C/C++ Developers". At the time of this writing, Eclipse Mars is the latest release and is the recommended version.

Ť	Eclipse IDE for Java Developers 166 MB 1,144.540 DOWNLOADS The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven integration and WindowBuilder	Windows 32 bit 64 bit
©.	ITE MIN 449.358 DOWNLOADS An IDE for C/C++ developers with Mylyn integration.	Windows 32 bit 64 bit
	Eclipse IDE for Eclipse Committers 4.5.1 244 MB 297,100 DOWNLOADS Package suited for development of Eclipse itself at Eclipse.	Windows 32 bit 64 bit
	Eclipse for PHP Developers 153 MB 281.587 DOWNLOADS The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and	Windows 32 bit 64 bit
	Eclipse IDE for Java and DSL Developers 296 MB 170.559 DOWNLOADS The essential tools for Java and DSL developers, including a Java & Xtend IDE, a DSL Framework (Xtext), a Git client.	Windows 32 bit 64 bit

Figure 57 Eclipse Versions

2. When Eclipse is run for the first time, it will ask for the workspace location.

Workspac	e Launcher	×
Select a wo	rkspace	
	es your projects in a folder called a workspace. orkspace folder to use for this session.	
Workspace:		Browse
🔲 Use this a	as the default and do not ask again	OK Cancel

Figure 58 Eclipse Workspace Location

A workspace is a directory on the hard drive where Eclipse stores the projects defined to it. More specifically, a workspace is a logical collection of projects. When you specify this directory name to Eclipse, Eclipse will create some files within this directory to manage the projects. The projects controlled by this workspace may or may not reside in this directory. Specify a directory name and click OK.



Note: To run Eclipse, it is required to download and install the Java Run Time Environment (JRE) or Java Developer Kit (JDK). Eclipse should display a warning if this is not installed. Oracle provides these tools for free.

6.2.2 FT9xx Eclipse Plugin Installation

To assist with completing the configuration of Eclipse for FT9xx coding an extra plug-in is provided as part of the download. To install the plug-in the following steps are required:

1. From the Eclipse toolbar select Help -> Install New Software which will pop up the window as below.

➡ Install	
Available Software Check the items that you wish to install.	
Work with: Bridgetek - file:, com.ftdichip	ft90x/ Add Find more software by working with the <u>"Available Software Sites"</u> preferences.
type filter text	
Name	Version
▶ 📝 100 Bridgetek FT9xx Eclipse Plugins	
Select All Deselect All	
- Details	1
Show only the latest versions of available software	Hide items that are already installed
Group items by category	What is <u>already installed</u> ?
Show only software applicable to target environment	
Contact all update sites during install to find required software	< Back Next > Finish Cancel

Figure 59 Eclipse Plugin Setup Wizard

- 2. Select the ADD button, and browse to the LOCAL location of the folder `com.ftdichip.ft90x' which can be found in "Toolchain\eclipse plugins" in the toolchain installation directory.
- 3. Press "SELECT ALL" followed by NEXT to install the plugin
- 4. Close the window when complete.



6.3 Common project settings in Eclipse

6.3.1 Include paths

Eclipse uses its built-in indexer to resolve dependencies between files. In order for the indexer to work correctly, paths that contain the header files in the project need to be added as follows:

1. Right-click on the project and select Properties

C/C++ - FT90x_Exan	nples/GPIO/gpio_example1.c - Eclipse			- • • ×
	Refactor Navigate Search Project Run Window H			
11 - H R A	8) • 🔦 • 🔒 🔌 🍠 🖗 🔳 👔 👩 • 🗯 • 🕻	「・ G・ 教・ Q・ Q・ (2) + (2) G ダ・ 2 ・ 2 ・ (- + + +	12	
			Quick Access	😭 🔤 C/C++ 🕸 Debug
🎦 Project Explorer 🔅	🛛 🖳 🕞 Makefile 🔀 gpio_example	c 🖾		🔗 Search 🐹 🖆 Inclu 😐 🗖 🔐
		of goodwill or reputation; and/or loss of, damage to or	*	sh 🔳 🖗 🕶 🐨 👔
مسج FT90x_Example		ap on FTDI's liability.		No search results available. Start a
Incluc	New	subsequently been amended by another user and then		search from the search dialog
> 🗁 ADC	Go Into	ther user ("Adapted Software"). If so that user may ce terms that apply to those amendments. However, FTDI		
Came	Open in New Window	elation to those amendments.		
DAC	Copy Ctrl+0			
👂 🗁 Ethen 🚞	Paste Ctrl+1			
D 🕞 GPIO 🔛	Delete Delet		=	
> 👝 I2C_M 🗮 > 👝 I2C_SI 🧕				
b ≥ 125	Source			
🛛 🕞 lib	Move			
> 🗁 PWM	Bename			
⊳ 🕞 RTC				
<u><u></u></u>				
🗄 0 🛛 💿 ⊿	Export			
	Build Project			
ft900.h	Clean Project			
# GPIO F	Refresh F.		-	
# STR()	Close Project			
# STR_0	Close Unrelated Projects	Tasks 🕎 Console 🐹 🗔 Properties 🔳	🗙 🍇 🖹 🔓	1 (R) 🕑 🖻 🕶 🗂 🗖 👘 📄 📄
++ loop(ve loop() :	Build Configurations	, ation] C:\FT900\toolchain\programmer\dist\FT900Prog.exe (26/1/15 5:56 pm)		
 main(v 	Make Targets	x		<u>^</u>
++ setup(v	Index	<pre>> g Utility!!!</pre>		
 setup() 	Show in Remote Systems view			
	Profiling Tools			
	Convert To			
	Profile As	, TA		
	Debug As	• A		
	Run As	•		
😂 FT90x_Example	Compare With	· · · · · · · · · · · · · · · · · · ·		
	Restore from Local History			
*				
~	Team	x		
	Properties Alt+Ente			

Figure 60 Eclipse Project Properties

2. In the Properties window, select C/C++ General > Paths and Symbols



Document Reference No.: BRT_000116 Clearance No.: BRT#074

Properties for helloworld			- • ×
type filter text	Paths and Symbols		(+ + - + →
 ▷ Resource Builders ▷ C/C++ Build △ C/C++ General ▷ Code Analysis 	Configuration: Default	it [Active]	onfigurations
Documentation File Types	🕒 Includes 🛛 # Syml	ıbols 🛋 Libraries 🔀 Library Paths 🔀 Source Location 😂 Output	t Location 🔹 🔸
Formatter Formatter Language Mappings Paths and Symbols Preprocessor Include P₂ Profiling Categories Project References Run/Debug Settings ▷ Task Repository WikiText	Languages Assembly GNU C GNU C++		Add Edit Delete Export Move Up Move Down
4 1111		Restore Defaults	Apply
?		ОК	Cancel

Figure 61 Eclipse Paths and Symbols

- 3. Under the **Includes** tab, choose **"GNU C"** under **Languages**, then click **"Add..."** on the right side of the window
- 4. In the **"Add directory path"** window, specify the path to the folder that contains the header files. If the same path is used for some C++ files, check the box **"Add to all languages"**, then click **OK**.

Add directory path	×
Directory:	
 Add to all configurations Add to all languages 	Variables Workspace File system OK Cancel

Figure 62 Eclipse Add Directory Path

Note: The paths should be added one at a time. The use of semicolon is not supported.

6.3.2 Toolchain settings

The FT9xx toolchain supports most GNU toolchain options. To specify an option that is not included by default, for example to create a map file, do it as follows:

- 1. Right click on the project and select Properties
- 2. In the Properties window, select C/C++ Build > Settings. The toolchain settings can be adjusted in the Settings window.



Document Reference No.: BRT_000116 Clearance No.: BRT#074

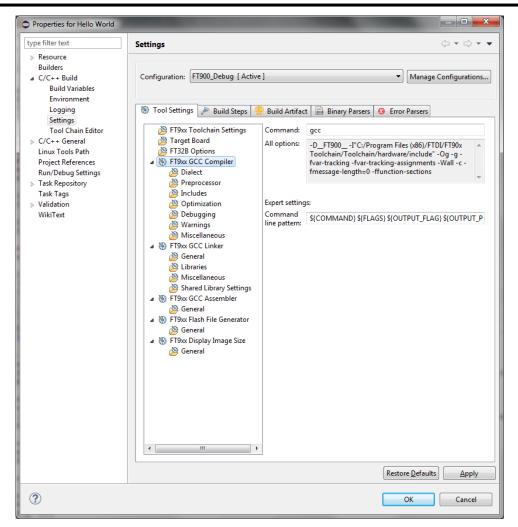


Figure 63 Eclipse Toolchain Settings

6.3.2.1 FT32B options (only for FT93x)

The FT93x target has two special compiler options -mft32b and -mcompress. -mft32b enables the FT32B instruction set and -mcompress enables code compression, which can typically result in a code size reduction of about 20 – 30%. These options can be updated in the toolchain settings as shown in *Figure 64*. Note that this option is only available in the FT930_* configurations.

type filter text	Settings $(\neg \neg \neg \neg \neg$
 Resource Builders C/C++ Build Build Variables Environment 	Configuration: FT930_Debug
Logging Settings Tool Chain Editor	 Tool Settings Build Steps Build Artifact Binary Parsers Settings FT9xx Toolchain Settings FT32B Instruction Set (-mft32b)
 C/C++ General Linux Tools Path Project References 	Target Board Code Compression (-mcompress)
,	Figure 64: FT32B options (for FT93x)



6.3.3 C/C++ Indexer Settings

For eclipse syntax highlighting to work correctly as you switch configurations, the Indexer has to be configured to work with the *active build configuration* as shown in Figure 65. This is a workspace specific setting and can be accessed in eclipse via Window | Preferences | C/C++ | Indexer

Preferences	
type filter text	Indexer $\diamond \star \star$
 General C/C++ Appearance Autotools Build Code Analysis Code Style Debug Editor File Types Indexer Language Mappings New C/C++ Project Wi: Profiling Property Pages Settings Task Tags Template Default Value ChangeLog Help Install/Update Library Hover Mylyn Oomph Remote Development Remote Systems RPM Run/Debug Team Tracing Validation XML 	Intexet 64 MB Index Enable indexer Index source files not included in the build Index unused headers Index all variants of specific headers: Index all variants of specific headers: Index all variants of specific headers: Index source and header files opened in editor Allow heuristic resolution of includes MB Skip files larger than: 8 MB Skip included files larger than: 16 MB Skip type and macro references (Search for these references will not work) Skip type and macro references (Search for these references will not work) Indexing strategy Automatically update the indexer Index use active build configuration Use active build configuration specified in the project's indexer settings Cache limits Index database cache: Limit relative to the maximum heap size: 10 % Absolute Limit: 256 MB Header file cache (used by refactoring): Absolute Limit: 64 MB
< >	Restore Defaults Apply
? 🔘	OK Cancel

Figure 65: Configure eclipse indexer to use the active build configuration



7 Troubleshooting

This section documents the problems you may encounter when using the FT9xx toolchain.

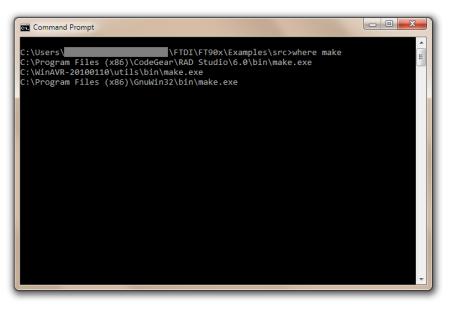
7.1 Makefile error

If using a makefile to build an application, some makefile errors may be reported, for example:

Command Prompt	
C:\Users\ HAKE Version 5.3 Copyright (c) 1987, 2008 CodeGear Error makefile 50: Command syntax error Error makefile 53: Command syntax error Error makefile 53: Command syntax error Error makefile 59: Command syntax error Error makefile 59: Command syntax error Error makefile 316: No match found for wildcard '*/%.o' Error makefile 316: Command syntax error Error makefile 316: Command syntax error Error makefile 316: Command syntax error Error makefile 317: No match found for wildcard '*/%.o' Error makefile 317: Command syntax error *** 10 errors during make ***	< III

Figure 66: Makefile Error

This is usually because some existing toolchain on the system may be using its own "make" utility which is also referred to in the PATH variable. The FT9xx examples need to be built by the GnuWin32 "make" utility, which can be installed during the toolchain installation. To solve this problem, adjust the PATH variable so that the correct "make" utility is called by the toolchain. Note that it may be necessary to adjust PATH again for the other toolchain. Type "where make" in a command prompt to find out which "make" utilities are present on the system.







8 Contact Information

Head Quarters – Singapore

Bridgetek Pte Ltd 178 Paya Lebar Road, #07-03 Singapore 409030 Tel: +65 6547 4827 Fax: +65 6841 6071

> <u>sales.apac@brtchip.com</u> <u>support.apac@brtchip.com</u>

Branch Office – Taipei, Taiwan

Bridgetek Pte Ltd, Taiwan Branch 2 Floor, No. 516, Sec. 1, Nei Hu Road, Nei Hu District Taipei 114 Taiwan, R.O.C. Tel: +886 (2) 8797 5691 Fax: +886 (2) 8751 9737

Lutaco Tower Building, 5th Floor, 173A Nguyen Van

E-mail (Sales) E-mail (Support)

Branch Office – Vietnam

Ward 11, Phu Nhuan District,

Ho Chi Minh City, Vietnam

Bridgetek VietNam Company Limited

sales.apac@brtchip.com
support.apac@brtchip.com

Branch Office - Glasgow, United Kingdom

Bridgetek Pte. Ltd. Unit 1, 2 Seaward Place, Centurion Business Park Glasgow G41 1HH United Kingdom Tel: +44 (0) 141 429 2777 Fax: +44 (0) 141 429 2758

E-mail (Sales) E-mail (Support)

E-mail (Sales)

E-mail (Support)

sales.emea@brtichip.com support.emea@brtchip.com

E-mail (Sales) E-mail (Support)

Tel: 08 38453222 Fax: 08 38455222

Troi,

sales.apac@brtchip.com
support.apac@brtchip.com

Web Site

http://brtchip.com/

Distributor and Sales Representatives

Please visit the Sales Network page of the <u>Bridgetek Web site</u> for the contact details of our distributor(s) and sales representative(s) in your country.

System and equipment manufacturers and designers are responsible to ensure that their systems, and any Bridgetek Pte Ltd (BRT Chip) devices incorporated in their systems, meet all applicable safety, regulatory and system-level performance requirements. All application-related information in this document (including application descriptions, suggested Bridgetek devices and other materials) is provided for reference only. While Bridgetek has taken care to assure it is accurate, this information is subject to customer confirmation, and Bridgetek disclaims all liability for system designs and for any applications assistance provided by Bridgetek. Use of Bridgetek devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold harmless Bridgetek from any and all damages, claims, suits or expense resulting from such use. This document is subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Neither the whole nor any part of the information contained in, or the product described in this document, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. Bridgetek Pte Ltd, 178 Paya Lebar Road, #07-03, Singapore 409030. Singapore Registered Company Number: 201542387H.



Appendix A – References

Document References

http://brtchip.com/m-ft9/ TN 160 Eclipse Projects

Acronyms and Abbreviations

Terms	Description
CMD	Command-line interface
DLL	Dynamic-link Library
DLOG	Data Log (Project)
GAS	GNU Assembler
GCC	GNU Compiler Collection
GDB	GNU Project Debugger
GNU	GNU (Gnu's Not Unix) Operating System
GUI	Graphical User Interface
IDE	Integrated Development Environment
JDK	Java Development Kit
JRE	Java Runtime Environment
МСО	Microcontroller Unit
PATH	PATH Environment Variable
ТСР	Transmission Control Protocol



Appendix B – List of Tables & Figures

List of Figures

Figure 1 Toolchain Setup Wizard Dialog box5
Figure 2 License Agreement Dialog box5
Figure 3 Revision and Release Information Dialog box
Figure 4 Components Dialog box
Figure 5 FT9xx Toolchain Install Location Dialog box7
Figure 6 FT9xx Toolchain-Examples & Documents Install Location Dialog box7
Figure 7 FT9xx Toolchain - Installation Progress Window
Figure 8: Close any running Java Applications before launching JRE Installer
Figure 9 Java Setup Window9
Figure 10 Java Setup Progress Window
Figure 11 Python Setup Dialog box 10
Figure 12 Destination Directory Selection Dialog box for Python Setup 10
Figure 13 Python Features Customization Dialog box 11
Figure 14 Python Installation Progress Dialog box 12
Figure 15 Python Installation Completion Dialog box12
Figure 16 FT9xx Toolchain - Installation Progress Window13
Figure 17 FT9xx Toolchain Setup Completion Dialog box
Figure 18: MSI Installer is busy 14
Figure 19 Eclipse for FT9xx Icon 16
Figure 20 Eclipse Workspace Selection 16
Figure 21 Eclipse Workspace Update 17
Figure 22 C Project Wizard 17
Figure 23 Project Wizard - Build Configurations Selection 18
Figure 24 C Project Wizard - Toolchain Details
Figure 25 New empty project structure 19
Figure 26 Build Configuration 20
Figure 27 Build Configuration 20
Figure 28 Building the Project
Figure 29 Build Status
Figure 30 List of Files after building (if all 4 configurations were selected)
Figure 31 FT900 Programming Utility Icon 22
Figure 32 Bridgetek Utilities Menu 22
Figure 33: FT9xx Programmer - Work with One-Wire
Figure 34: FT9xx Programmer - Device Selection
Figure 35 FT9xx Programmer – Flash and PM Screen 24



Document Reference No.: BRT_000116 Clearance No.: BRT#074

Figure 36 FT9xx Programmer in Eclipse
Figure 37 Hello World
Figure 38 FT9xx Examples
Figure 39 Build Configuration
Figure 40 Choosing a Debugging Launcher (1)
Figure 41 Choosing a Debugging Launcher (2)
Figure 42 Eclipse Debugging Application Settings
Figure 43 Eclipse GDB Settings
Figure 44 Eclipse TCP Port Settings
Figure 45 GDB Bridge Icon
Figure 46 Bridgetek Utilities Menu
Figure 47 GDB Bridge in action
Figure 48 Eclipse Run Remote Debugging
Figure 49 Eclipse Debug Environment
Figure 50 Eclipse Missing Source File 32
Figure 51 Debug flags
Figure 52 Og compiler optimization option
Figure 53 Bridgetek Project Types 34
Figure 54 D2XX Project Template
Figure 55 Compiling D2XX Template Code
Figure 56 FT9xx Debugging Status
Figure 57 Eclipse Versions
Figure 58 Eclipse Workspace Location
Figure 59 Eclipse Plugin Setup Wizard
Figure 60 Eclipse Project Properties 40
Figure 61 Eclipse Paths and Symbols 41
Figure 62 Eclipse Add Directory Path 41
Figure 63 Eclipse Toolchain Settings 42
Figure 64: FT32B options (for FT93x) 42
Figure 65: Configure eclipse indexer to use the active build configuration
Figure 66: Makefile Error
Figure 67 "make" Locations



Appendix C – Revision History

Document Title:	AN_325 FT9xx Toolchain Installation Guide		
Document Reference No.:	BRT_000116		
Clearance No.:	BRT#074		
Product Page:	http://brtchip.com/m-ft9/		
Document Feedback:	Send Feedback		

Revision	Changes	Date
1.0	Initial release	2014-05-02
1.01	Expanded screenshots of Installation Wizard in Section 2	2015-08-21
1.02	Updated Version for Toolchain 2.1.0	2016-02-22
1.03	Added section 2.1.1 to document the handling of another instance of MSI installer running in the background while installing JRE (results in JRE install error 1618) Debugger related information moved to section 4.4. from Troubleshooting Updated screenshots for programmer	2016-09-19
1.04	Updated release Migration of the product from FTDI to Bridgetek name – logo changed, copyright changed, contact information changed	2017-03-08