



# **Future Technology Devices International Ltd.**

## **TN\_160 Eclipse Projects**

**Document Reference No.: FT\_001175**

**Version 1.0**

**Issue Date: 2015-10-13**

This is a technical note describing the use Eclipse IDE which can be used with FTDI FT51A and FT90x MCU products.

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

### **Future Technology Devices International Limited (FTDI)**

Unit1, 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 (Support): [support1@ftdichip.com](mailto:support1@ftdichip.com) Web: <http://www.ftdichip.com>

Copyright © 2015 Future Technology Devices International Limited

## **Table of Contents**

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
1.1	Overview .....	4
1.2	Scope .....	4
<b>2</b>	<b>Eclipse IDE Installation and Configuration .....</b>	<b>5</b>
2.1	Java Development Kit.....	5
2.2	Eclipse Workspaces.....	5
2.3	Eclipse IDE Startup.....	6
<b>3</b>	<b>MCU Specific Installation Requirements .....</b>	<b>8</b>
3.1	FT51A.....	8
3.2	FT90x .....	8
<b>4</b>	<b>Eclipse Projects .....</b>	<b>9</b>
4.1	Importing Example Code.....	9
4.2	Creating New Projects.....	10
4.2.1	FT51A .....	10
4.2.2	FT90x .....	10
<b>5</b>	<b>Adding Files to Projects .....</b>	<b>11</b>
5.1	Add New Files.....	11
5.2	Add Existing files.....	11
5.3	Adding Include Directories.....	11
<b>6</b>	<b>Building Projects .....</b>	<b>12</b>
<b>7</b>	<b>Programming devices .....</b>	<b>13</b>
7.1	FT51A.....	13
7.2	FT90x .....	14
<b>8</b>	<b>Debug .....</b>	<b>15</b>
8.1	FT51A.....	16
8.2	FT90x .....	17
<b>9</b>	<b>Tips and Tricks .....</b>	<b>18</b>

---

<b>9.1</b>	<b>Editor .....</b>	<b>18</b>
<b>9.2</b>	<b>Build Automatically .....</b>	<b>18</b>
<b>9.3</b>	<b>Debug.....</b>	<b>18</b>
<b>9.4</b>	<b>Eclipse Specifics .....</b>	<b>18</b>
<b>10</b>	<b>Troubleshooting .....</b>	<b>19</b>
<b>10.1</b>	<b>Access is Denied .....</b>	<b>19</b>
<b>10.2</b>	<b>Breakpoints .....</b>	<b>19</b>
<b>10.3</b>	<b>Java Error .....</b>	<b>19</b>
<b>10.4</b>	<b>Build Error .....</b>	<b>19</b>
<b>10.5</b>	<b>Viewing Definitions.....</b>	<b>19</b>
<b>10.6</b>	<b>Directory Names .....</b>	<b>20</b>
<b>10.7</b>	<b>Environment Variables.....</b>	<b>20</b>
<b>10.8</b>	<b>FT51A Build Error 1.....</b>	<b>20</b>
<b>10.9</b>	<b>FT51A Build Error 2.....</b>	<b>20</b>
<b>10.10</b>	<b>FT51A Programming Error 1.....</b>	<b>21</b>
<b>10.11</b>	<b>FT51A Programming Error 2.....</b>	<b>21</b>
<b>11</b>	<b>Contact Information .....</b>	<b>22</b>
<b>Appendix A</b>	<b>– References .....</b>	<b>23</b>
	<b>Document References .....</b>	<b>23</b>
	<b>Acronyms and Abbreviations.....</b>	<b>23</b>
<b>Appendix B</b>	<b>– List of Tables &amp; Figures .....</b>	<b>24</b>
	<b>List of Figures .....</b>	<b>24</b>
<b>Appendix C</b>	<b>– Revision History .....</b>	<b>25</b>

## 1 Introduction

The document details the use of Eclipse which can be used with FTDI FT51A and FT90x MCU products to test and develop applications and software.

### 1.1 Overview

Eclipse is a free open source Integrated Development Environment (IDE) which is used to provide a cross platform development and debugging environment. It features an integrated project manager, code editor, build configurator and debugger.

FTDI also offer an alternative IDE from MikroElektronika which is not covered in this document. Please refer to the [MikroElektronika](#) website for further information.

### 1.2 Scope

The scope of this document is to

- briefly discuss installation and FTDI MCU specific configuration
- show how to create or import projects
- show how to build projects
- show how to program and debug
- provide tips and tricks and troubleshooting help

## 2 Eclipse IDE Installation and Configuration

The toolchain installers give an option to install a version of Eclipse as part of the installation process. However you can also use your own version.

The latest Eclipse IDE is recommended. At the time of writing, Luna is the current version and this is referenced in this document, which includes screenshots.

The Eclipse IDE can be downloaded from the Eclipse website:

<https://www.eclipse.org/downloads/>

Select 'Eclipse IDE for C/C++ Developers' and choose the relevant windows download (32 Bit or 64 Bit). Download the file (.zip), extract and run 'eclipse.exe' found in the top level folder. Note that the eclipse IDE download is not an 'installer' program. The Eclipse IDE program is contained within the .zip file. A user can create a shortcut on their desktop to run Eclipse IDE for ease of access.

This also means that multiple versions of Eclipse can exist on the same PC.

### 2.1 Java Development Kit

To be able to run the Eclipse IDE, this additional program is also required to be installed on your PC. Eclipse tells you if this is not installed.

The Java Developer Kit (JDK) can be found on Oracle's website:

<http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>

Click on download, accept the license agreement, select the platform required and follow the installer's instructions.

### 2.2 Eclipse Workspaces

The default workspace location is decided when the program is first executed as shown in Figure 2.1.

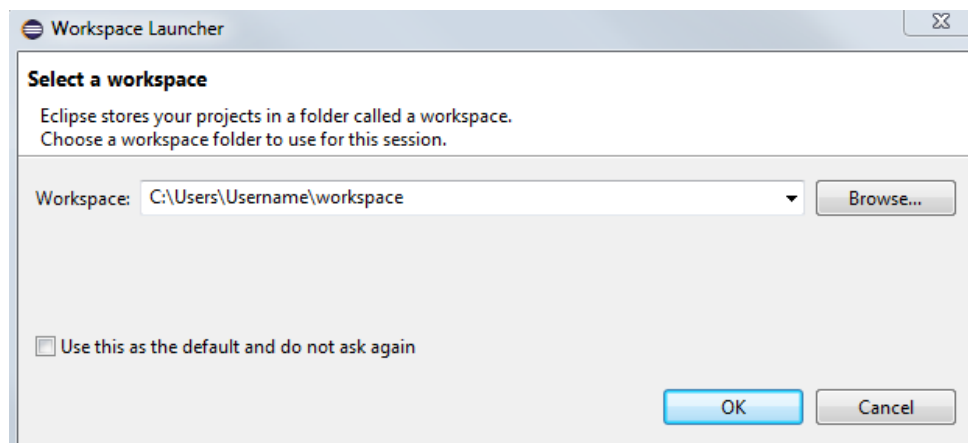


Figure 2.1 Eclipse Workspace Launcher

Avoid directories with spaces in the names which can cause issues when it comes to build, program and debug.

Project settings are contained within the specific project, so multiple projects for multiple MCUs can reside in the same workspace.

However, you can create multiple workspaces if required, for example separate workspaces for FTDI FT51A and FT90x MCUs. This should be done before creating or importing projects. Once Eclipse is started, Select File → Switch Workspace → Other and select the desired location on the PC.

## 2.3 Eclipse IDE Startup

When eclipse is started for the first time, the welcome screen is shown in Figure 2.2. This allows the user to learn about Eclipse to get an overview of the features, try out samples and access tutorials.

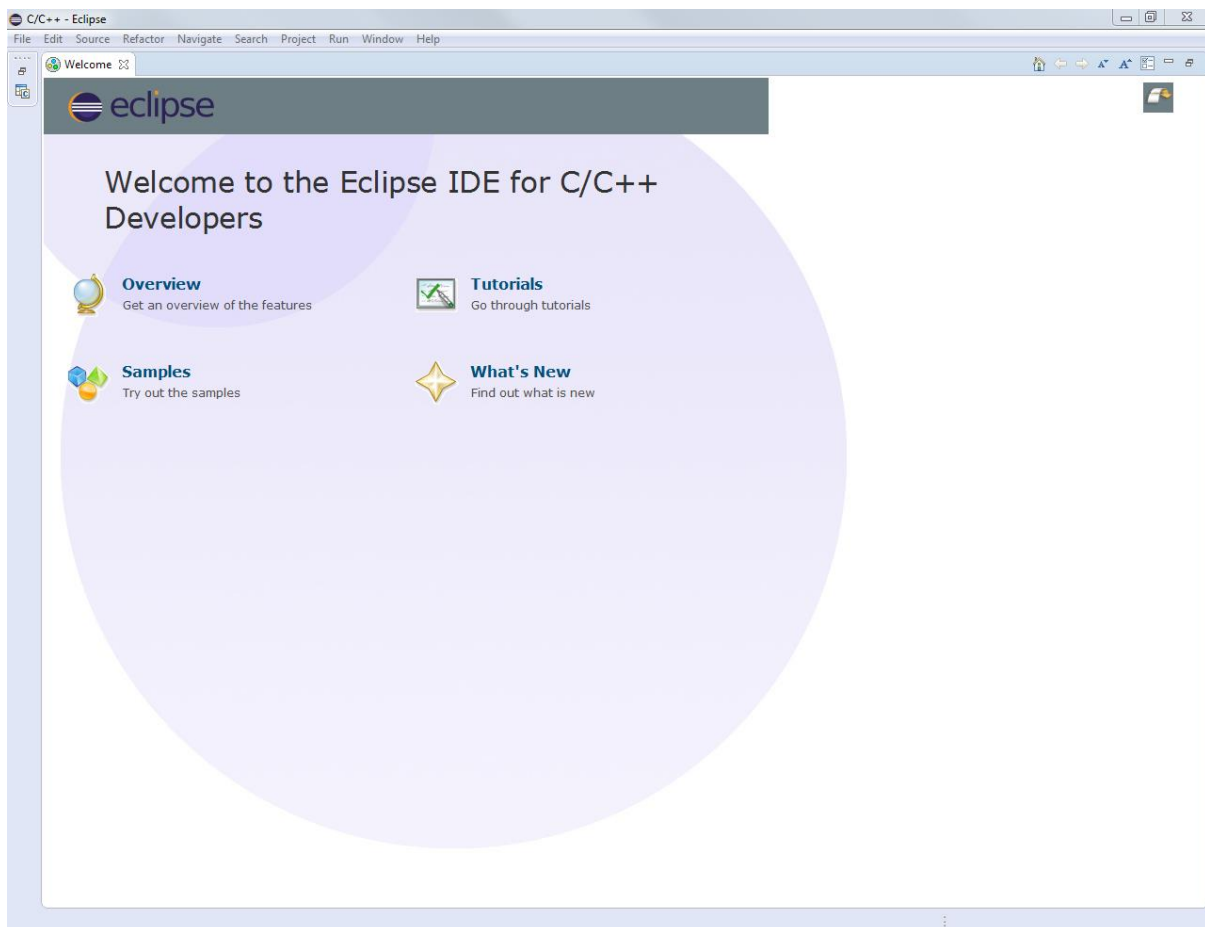


Figure 2.2 Initial Eclipse Startup

When the user is finished with the initial welcome screen, click on the restore icon as shown in Figure 2.3 to start the normal Eclipse IDE environment. Note that the welcome screen can be accessed at any point from the Help → Welcome menu.

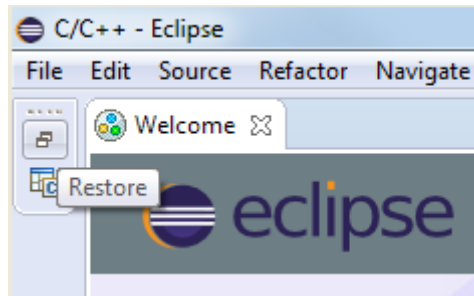


Figure 2.3 Restore Eclipse Icon

Figure 2.4 shows the Project Explorer, editor and tool icons, although there is not much shown until a project is created or imported.

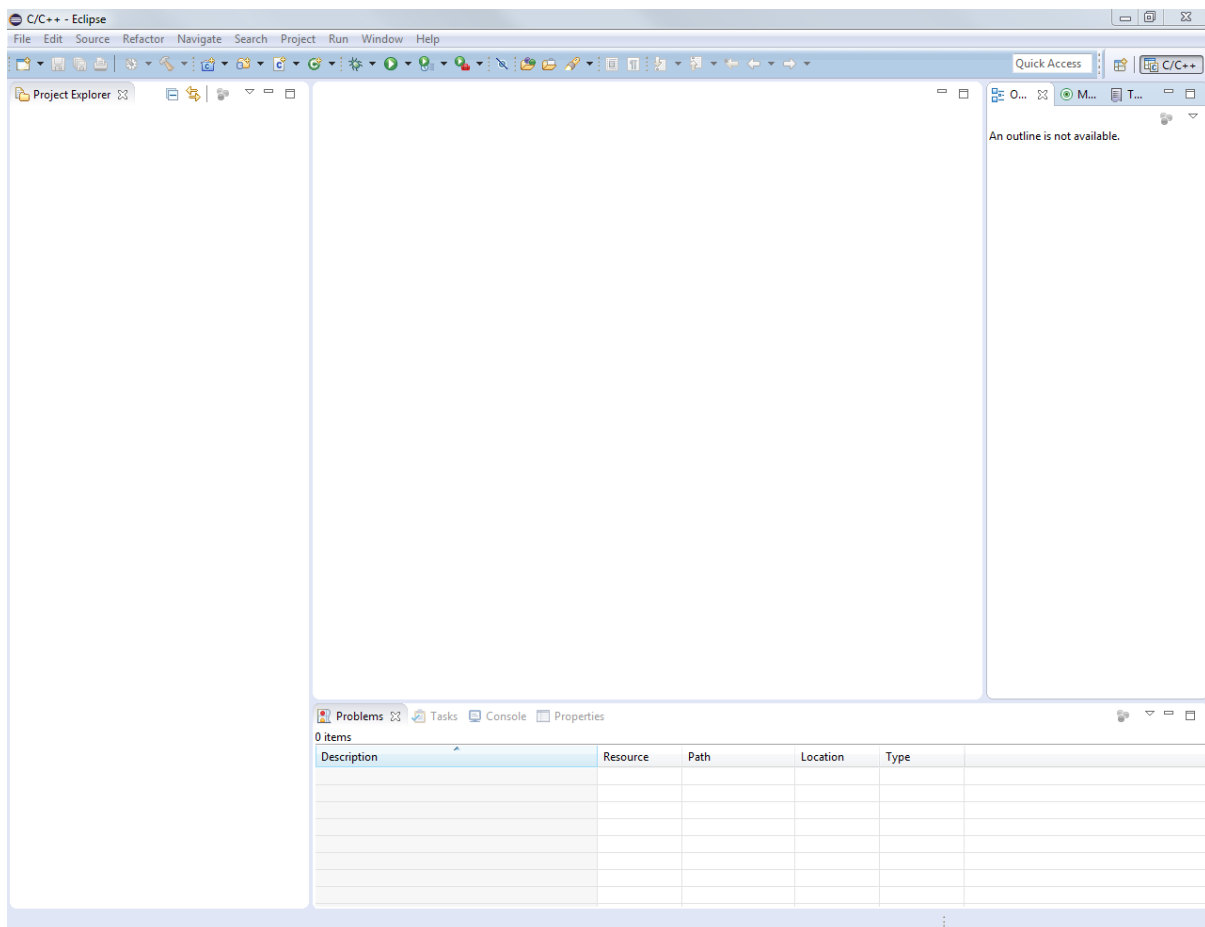


Figure 2.4 Standard Eclipse IDE Screen

## 3 MCU Specific Installation Requirements

Please see the following App Notes for details on additional tools required to use the Eclipse IDE with FTDI MCUs:

- FT51A (8 bit MCU): [AN 352 FT51 Installation Guide](#)
- FT90x (32 bit MCU): [AN 325 FT900 Tool Chain Installation Guide](#)

In summary the following installation options are available for each FTDI MCU product:

### 3.1 FT51A

1. Install the Software Development Kit (SDK) from FTDI: installs tools, libraries, sample code, documentation and source code. Options available:
  - FT51A Programmers Guide
  - FT51A Firmware Libraries & Examples
  - FT51A Tools
    - External Tools
      - SDCC
      - GnuWin32 Make & CoreUtils
      - Java JRE
      - Eclipse
    - Install FT51A Tools in SDCC bin directory
    - FT51A Tools Source Code
  - FT51A Accessories
    - FT51A Plugins for Eclipse
    - libusb-win32 and WinUSB Driver for FT51A DFU

### 3.2 FT90x

1. Install the FT90x Toolchain Setup Executable. This installs tools, libraries, example code and documentation. Options available:
  - Eclipse IDE for FT90x
  - FT90x Driver Library
  - FT90x Example Applications
  - External Utilities
    - GnuWin32 Make & CoreUtils
    - Python
  - Update PATH variable
  - Create icons on desktop

Install Eclipse plugins via Help → Install New software menu if you are using your own version of Eclipse.

---



## 4 Eclipse Projects

There are a couple of options for Eclipse projects which are detailed in this section:

- Importing Example Projects
- Creating New Projects

### 4.1 Importing Example Code

FTDI provide example software projects with full source code in order to help users learn about our products and on-chip modules and how to use them.

Within Eclipse, perform the following actions:

- File → Import Projects → General → Existing Projects into Workspace
- Browse to the location of the example code provided as part of the FTDI installer:
  - FT51A: C:\Users\Username\Documents\FTDI\FT51A\_SDK\Version\examples
  - FT90x: C:\Users\Username\Documents\FTDI\FT90x\Examples
- The user can decide to select the 'Copy projects into workspace' option or to use the existing location of the code.

The project should now appear in 'Project Manager'.

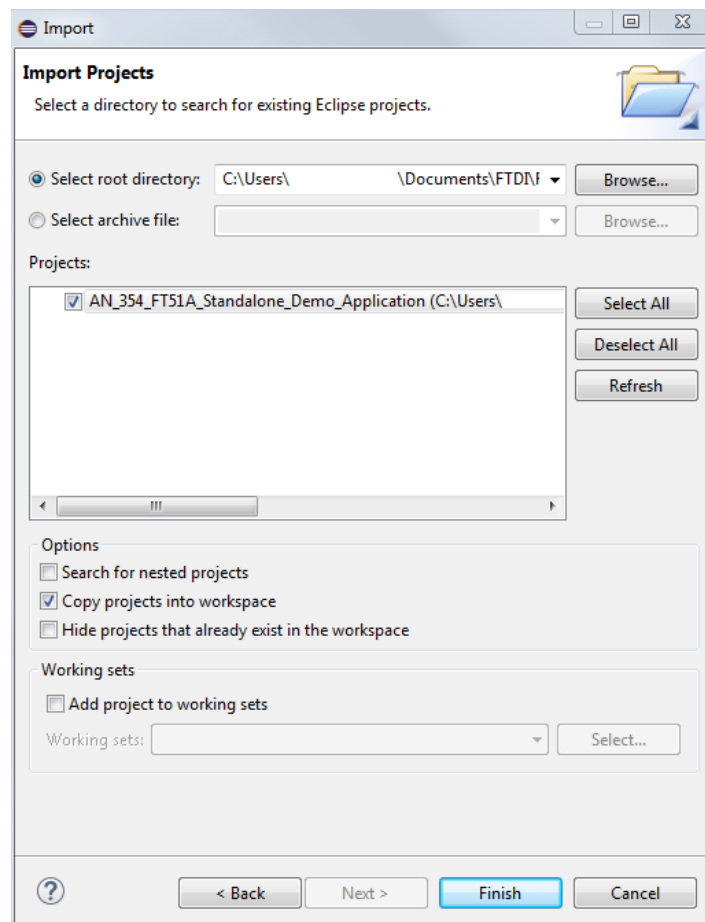


Figure 4.1 Import Project

## 4.2 Creating New Projects

The process for creating a new project is detailed in this section.

### 4.2.1 FT51A

Within Eclipse, perform the following actions:

- File → New → C Project
- Enter Project Name. Note: avoid names with spaces.
- In Project Type, select 'Empty Project' or 'Basic Project for SDCC'. Note that other options exist.
- In Toolchains, select 'SDCC Tool Chain'

The project should now appear in 'Project Manager'.

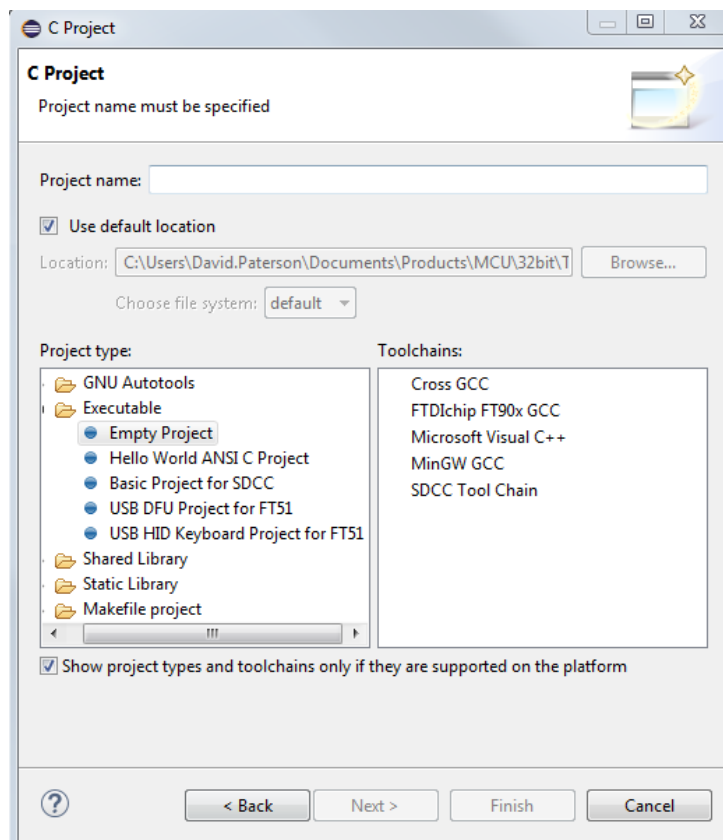


Figure 4.2 FT51A New Project

### 4.2.2 FT90x

Within Eclipse, perform the following actions:

- File → New → C Project
- Enter Project Name. Note: avoid names with spaces.
- In Project Type, select 'Empty Project'.
- In Toolchains, select 'FTDIchip FT90x GCC'

The project should now appear in 'Project Manager'.

## 5 Adding Files to Projects

To add files to a project there are a few methods detailed here.

### 5.1 Add New Files

Right-click on the project within Project Explorer and select 'New' then select Source File or Header File for example.

### 5.2 Add Existing files

Drag existing files from windows explorer into the project in Eclipse. Note that Eclipse gives the option to either copy the file to the current project or link the file to the project from the existing location.

The user can also manage files at windows explorer level, then right-click on project within Project Explorer and select Refresh (F5).

### 5.3 Adding Include Directories

In order to add include directories like libraries, right-click on Project within Project Explorer and select Properties. Navigate to C/C++ General → Paths and Symbols. Note the options available which should be selected depending on your needs.

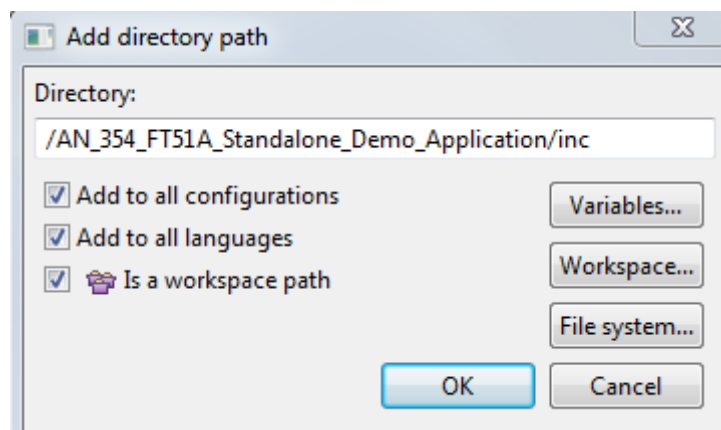


Figure 5.1 Add Directory Path

## 6 Building Projects

Building a project is the process to create an executable which can be programmed onto the MCU. Additional outputs are debugging and file mapping or memory information.


Firstly, you should select the type of project that you would like to build. This can be done through 'Project → Build Configurations → Set Active' menu.

The options are:

- Debug
- Release

Unique directories with executable are created for each option.

There are a few options available to build a project:

- Right-click on the project within Project Explorer and select 'Build Project'
- Select the project in Project Explorer. Click the build icon 
- Press CTRL + B to build all projects in Project Manager

The output of the build process will appear in the 'Console' window.

Note that you should save files before building. There is no warning about unsaved files. Unsaved files have '\*' before the filename within the editor.

Sometimes a project clean may be required. This can be done by:


- Right-clicking on the project within Project Explorer and select 'Clean Project'

## 7 Programming devices

The process for programming FT51A and FT90x devices are different. See the sub-sections below for further information.

### 7.1 FT51A

FTDI provide an external tool as part of the SDK to program the device, and this is run from within Eclipse. This needs to be setup initially, but can be run via a button thereafter.

- Select Run → External Tools → External Tools Configuration.
- Click on Program, then click the 'New Launch Configuration' icon 
- Enter the details as shown in Figure 7.1 as an example.
- Click Apply and Run.

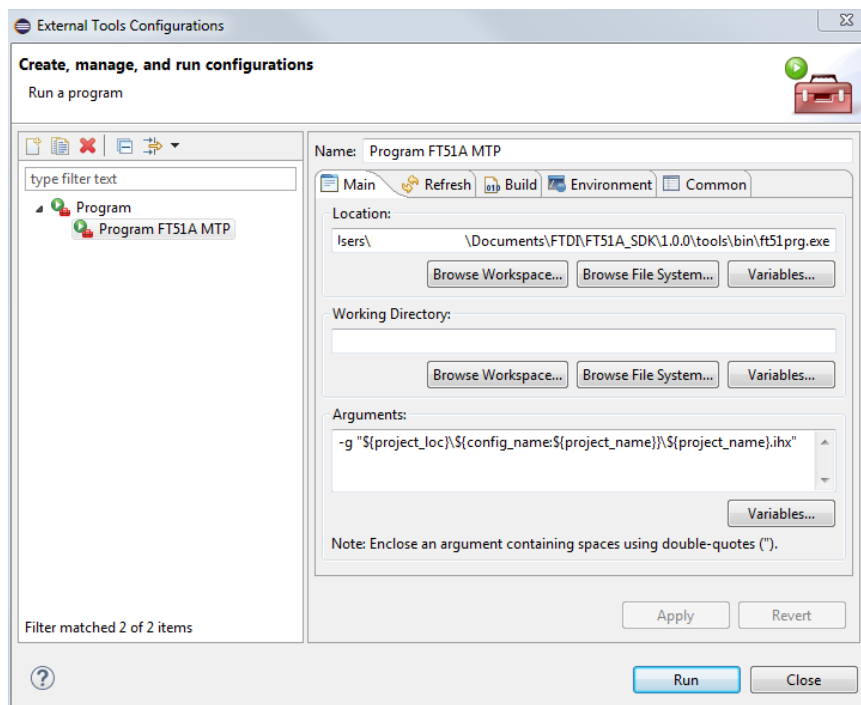


Figure 7.1 FT51A External Tools Configuration

The console window shows the progress. A successful program should be like:

*Available devices are:*


*Description: "FT230X Basic UART". Serial Number: DBXOPYIY.*

*Opened debugger port: FT230X Basic UART*

*Programming ShadowRAM... Done.*

*Committing...Done.*


*Running.*

The next programming action can be performed by the  button.

Note that there are other options available like programming Shadow RAM, and the FT51A can be programmed over USB via DFU (Device Firmware Upgrade). Please see [AN\\_352 FT51 Installation Guide](#) for further information.

## 7.2 FT90x

FTDI provide an external tool as part of the SDK to program the device, and this is run from within Eclipse. This needs to be setup initially, but can be run via a button thereafter.

- Select Run → External Tools → External Tools Configuration.
- Click on Program, then click the 'New Launch Configuration' icon 
- Enter the details as shown in Figure 7.2 as an example.
- Click Apply and Run.

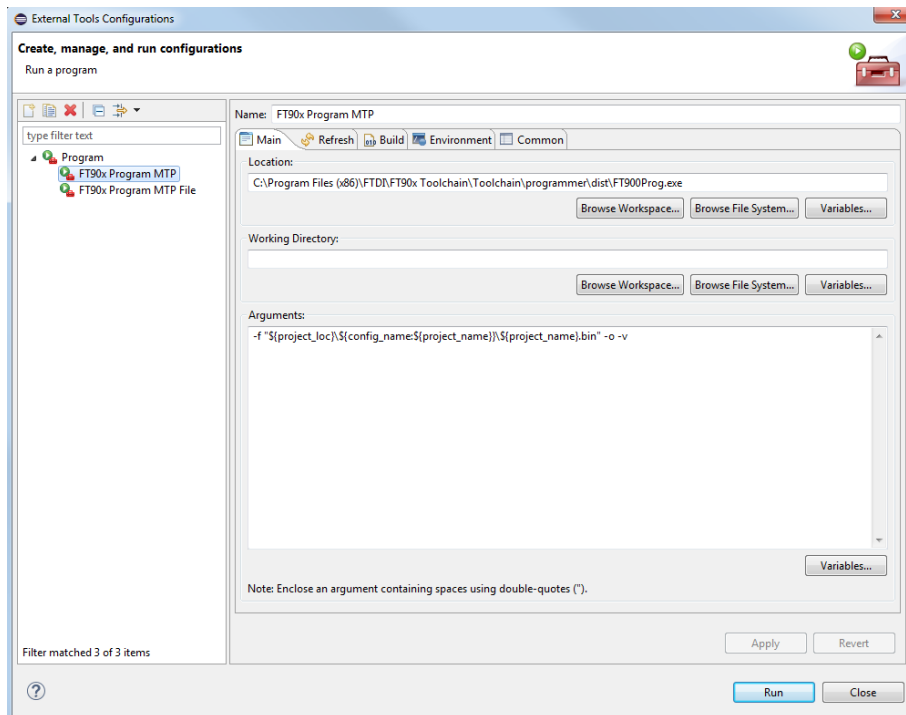



Figure 7.2 FT900 External Tool Configuration

The console window shows the progress. A successful program should be like:

```

-----
!!!Welcome to FT900 Programming Utility!!!
-----
Initializing...
.
Flash programming progress:
.
[=====] 100%
  
```

The next programming action can be performed by the  button.

Note that there is an external GUI version available which can be used to program the FT90x device. The shortcut is created on your desktop and is available in Eclipse 'FTDI Utilities' menu as part of the installation. Please see [AN 325 FT900 Tool Chain Installation Guide](#) for further information.

## 8 Debug

The Eclipse IDE provides a highly configurable debug environment as shown in Figure 8.1 including:

- Debug control buttons (step, run, halt, stop, etc)
- Source Code and project files
- Variables and breakpoints
- Memory windows (requires range setup)

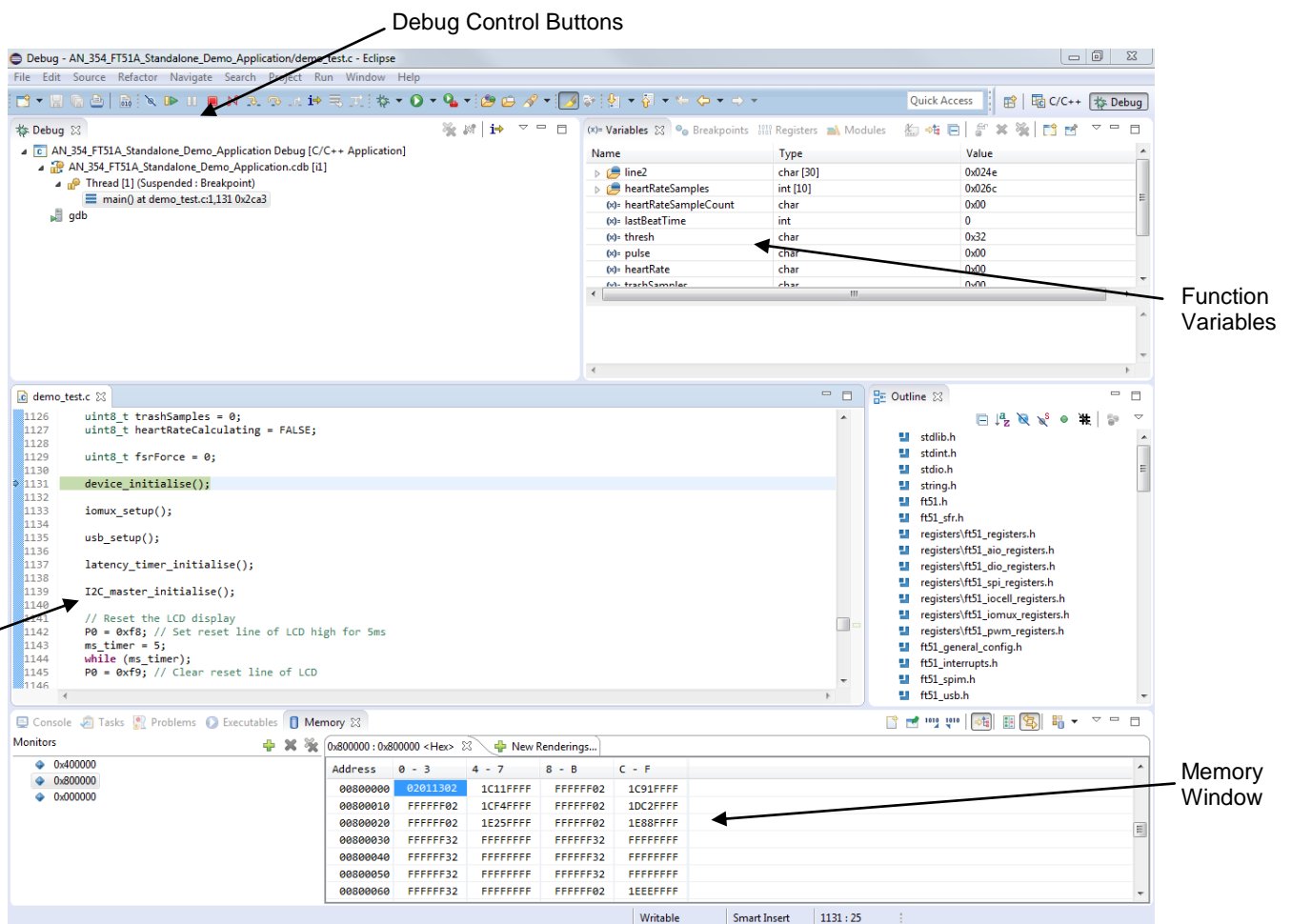
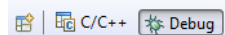


Figure 8.1 Debug Environment

The project must be compiled before debug can be setup.


Note: switch between Development and Debug perspectives at top right hand corner




## 8.1 FT51A

FTDI provide a tool called 'ft51gdb.exe' which is executed through Eclipse. This is based on the SDCC debugger but implements an "MI" interface similar to GDB.

Unique debug configurations are required to be set up for each project as shown in Figure 8.2.

- Select Run → Debug Configurations.
- Click on 'C/C++ Application' and then click the 'New Launch Configuration' icon 
- Choose the .cdb file for your project.
- Modify the Debugger tab with 'ft51gdb.exe'.
- Click Apply and Run.

The next debugging action can be performed by the  button.

Please see [AN\\_352\\_FT51\\_Installation\\_Guide](#) for further information.

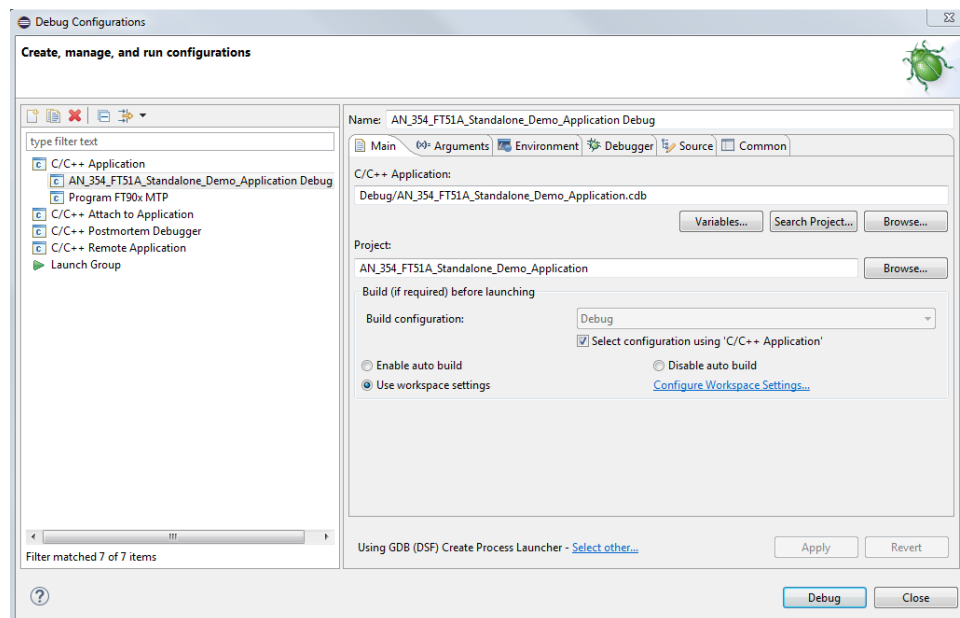


Figure 8.2 FT51A Debug Configuration





## 8.2 FT90x

FT90x provides a GDB based debugger (ft32-elf-gdb) as part of the installer which is executed through Eclipse.

'FT90x GDB Bridge' desktop shortcut or Eclipse 'FTDI Utilities' menu is provided for converting GDB commands into the debugger module commands. This must be run before debug can start.

Unique debug configurations are required to be set up for each project as shown in Figure 8.3.

- Select Run → Debug Configurations.
- Click on 'C/C++ Remote Application' and then click the 'New Launch Configuration' icon 
- Choose the .elf file for your project.
- Select 'GDB (DSF) Manual Remote Debugging Launcher'
- Modify the Debugger tab with 'ft32-elf-gdb.exe'.
- Set the Connection to Port number 9998.
- Click Apply and Run.

The next debugging action can be performed by the  button.

Please see [AN\\_325\\_FT900\\_Tool\\_Chain\\_Installation\\_Guide](#) for further information.

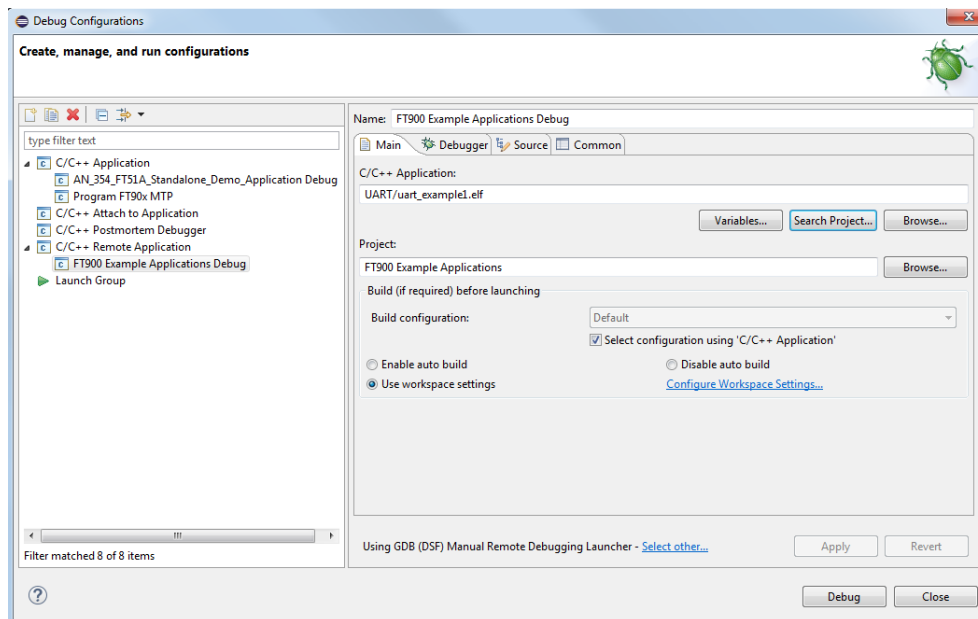


Figure 8.3 FT90x Debug Configuration

## 9 Tips and Tricks

This section details some tips and tricks when using Eclipse IDE.

### 9.1 Editor

Warning / Error symbols appear on Left Hand Side of text editor which provides quick view to newly typed code for example. Mouse over for details eg 'symbol could not be resolved' and missing ';'.

To find where a function resides, select a function, right-click and select 'Open Declaration' and the editor should take you to that function code even if it exists in another file.

The yellow arrows provide ease of navigation though code



### 9.2 Build Automatically

The Workbench will automatically perform incremental builds of resources modified since the last build. Whenever a resource is modified, another incremental build occurs. Select the option via:

Project → Build Automatically

To avoid all projects building before programming FT51A memory, uncheck the following option:

Window → Preferences → Run/Debug → Launching → Build (if required) before launching

### 9.3 Debug

There are 3 hardware breakpoints available on FTDI MCU products. When initiating debug, a temporary breakpoint is set at the start of the main function.

When in the debug window, right-click on the blue bar at left hand side of the source code window and select Toggle breakpoint to set/reset breakpoints.

There is also a Breakpoint window visible via Window → Show View → Breakpoints.

To restart debug quickly, Right-click on the debug session in the debug tab and select 'Terminate and Relaunch'.

Note that you must kill a debug session before device can be programmed as the same hardware is used for both program and debug.

### 9.4 Eclipse Specifics

A wide range of Eclipse plug-ins are available. See Help → Eclipse Marketplace. For example, a 'terminal' plug-in is available to allow external commands to be run within Eclipse, particularly useful when running FT90x debug.

CTRL + SHIFT + L gives a list of all Eclipse shortcuts which can be configured via Windows → Preferences → General → Keys.

Quick Access box or CTRL + 3 allows users to quickly go to views, perspectives and more.

See Help → Tips and Tricks for more!

## 10 Troubleshooting

### 10.1 Access is Denied

If you sometimes see the following error in Eclipse when trying to build projects, it is caused by AVG or similar antivirus software because a batch file is run in a kind of sandbox, in order to protect the PC:

```
make: *** Access is denied.  
. Stop.  
make: *** Waiting for unfinished jobs....
```

The solution is to retry, or temporarily disable antivirus software, or try adding an exception to the antivirus software (exclipse.exe).

### 10.2 Breakpoints

There are only 3 breakpoints available on FT51A and FT90x. If you try to set more than 3 breakpoints in Eclipse, the program may become unstable and debugging is no longer allowed.

There are no warnings that more breakpoints have been used than are available.

To overcome this you must terminate and restart the debugger. You may also have to disconnect and reconnect the debugger.

### 10.3 Java Error

If you see the following error after launching Eclipse, this is caused by a different Java and Oracle Java Development Kit (JDK/JRE) versions:

```
Java was started but returned exit code=13
```

This can happen after updating your PC's Java (java.com) version. For example, installed Java Version 8 Update 25 from java.com, but current Oracle JDK/JRE version was Java SE Development Kit 7.

The solution is to download Java SE Development Kit from Oracle to match the Java.com version.

### 10.4 Build Error

If you see errors during build of a project that relies on library code or other include code/files, Eclipse may not know where the files exist. For example:

```
../kbd.c:9:18: fatal error: ft51.h: No such file or directory  
compilation terminated.  
make: *** [kbd.rel] Error 1  
make: Target `main-build' not remade because of errors.
```

To add include files and directories, see section '5.3 Adding Include Directories'.

### 10.5 Viewing Definitions

In Eclipse, right click on a function or variable name and select 'Open Declaration' to see the source or definition. If you see an error saying that it cannot be found, then you may need to add the include directory. See section '5.3 Adding Include Directories'.

Alternatively try Project → C/C++ Index → Rebuild, and then Project → C/C++ Index → Freshen all files to try to get Eclipse to recognize the libraries.

---

## 10.6 Directory Names

When trying to debug your application code and receive an error like the following:

*Error in final launch sequence*

*Failed to execute MI command:*

*-environment-cd  
Stuff/eclipse/workspace/FT51A\_UART"*

*"C:/Users/username/Documents/MCU*

*Error message from debugger back end:*

*Cannot CD to directory*

*Cannot CD to directory*

This is because there is a space in the directory name ("MCU Stuff").

Always choose a workspace, or project directory, with a continuous directory name.

## 10.7 Environment Variables

When installing the tools, you need to add some data to the PATH environment variables.

These tips should ensure no issues:

1. Ensure that there are no spaces before or after the ';' for each entry in the environment variable
2. Close all windows file and internet explorer windows for the changes to take effect. If in doubt, restart the PC.

## 10.8 FT51A Build Error 1

If you see the following error when trying to build an FT51A project:

*make -k all*

*Cannot run program "make": Launching failed*

*Error: Program "make" not found in PATH*

Then you need to make sure the GnuWin32 bin directory path the environment variable.

Note that you need to close all windows explorer applications (file and internet) for any environment variable changes to come into effect.

## 10.9 FT51A Build Error 2

If you see the following error when trying to build an FT51A project:

*\*\*\*\**

*make all*

*make: \*\*\* No rule to make target `all'. Stop.*

This can be due to incorrect importing of projects into Eclipse.

If you want to import the example projects, you must do the following:

*File --> Import --> General --> Existing Projects into Workspace*

Do not choose *C/C++ --> Existing Code as a Makefile Project* as FT51A projects do not use makefiles.

Note that if you incorrectly import a project, this changes the project settings, which means that it may become corrupt. The solution is to use a fresh version of the project that needs to be imported.

---

Also, when you right-click on the project and the 'Build Project' is greyed out, then you must first left-click on the project, then right-click. Alternatively use the build icon.

## 10.10 FT51A Programming Error 1

If you see the following error:

*ERROR: no filename specified*

Then either the external programmer has not been setup correctly, or you simply need to click on the project in Eclipse before running the external programmer.

Other things to try:

1. Put quotation marks around the file/location data:

```
-s -g "${project_loc}\${config_name:${project_name}}\${project_name}.ihx"
```

2. Use an absolute path (for testing purposes):

```
-s -g "C:\Users\Username\Documents\Products\MCU\8bit\FT51A_UART\Debug\FT51A_UART.ihx"
```

3. Check that the file physically exists at the location specified in the External Build Tool configuration:

```
"C:\Users\Username\Documents\FTDI\FT51A_SDK\Version\tools\bin\ft51prg.exe"
```

## 10.11 FT51A Programming Error 2

When trying to program the FT51A device and see the following error:

*Available devices are:*

*Description: "FT230X Basic UART". Serial Number: DBX0PYIY*

*Cannot open debugger " "*

Then you need to change the Arguments in the Run-->External Tools-->External Tools Configuration in Eclipse to include the debugger

Original argument:

```
-g "${project_loc}\${config_name:${project_name}}\${project_name}.ihx"
```

New argument:

```
-p "FT230X Basic UART" -g  
"${project_loc}\${config_name:${project_name}}\${project_name}.ihx"
```

## 11 Contact Information

### Head Office – Glasgow, UK

Future Technology Devices International Limited  
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) [sales1@ftdichip.com](mailto:sales1@ftdichip.com)  
E-mail (Support) [support1@ftdichip.com](mailto:support1@ftdichip.com)  
E-mail (General Enquiries) [admin1@ftdichip.com](mailto:admin1@ftdichip.com)

### Branch Office – Taipei, Taiwan

Future Technology Devices International Limited  
(Taiwan)  
2F, No. 516, Sec. 1, NeiHu Road  
Taipei 114  
Taiwan, R.O.C.  
Tel: +886 (0) 2 8791 3570  
Fax: +886 (0) 2 8791 3576

E-mail (Sales) [tw.sales1@ftdichip.com](mailto:tw.sales1@ftdichip.com)  
E-mail (Support) [tw.support1@ftdichip.com](mailto:tw.support1@ftdichip.com)  
E-mail (General Enquiries) [tw.admin1@ftdichip.com](mailto:tw.admin1@ftdichip.com)

### Branch Office – Tigard, Oregon, USA

Future Technology Devices International Limited  
(USA)  
7130 SW Fir Loop  
Tigard, OR 97223-8160  
USA  
Tel: +1 (503) 547 0988  
Fax: +1 (503) 547 0987

E-Mail (Sales) [us.sales@ftdichip.com](mailto:us.sales@ftdichip.com)  
E-Mail (Support) [us.support@ftdichip.com](mailto:us.support@ftdichip.com)  
E-Mail (General Enquiries) [us.admin@ftdichip.com](mailto:us.admin@ftdichip.com)

### Branch Office – Shanghai, China

Future Technology Devices International Limited  
(China)  
Room 1103, No. 666 West Huaihai Road,  
Shanghai, 200052  
China  
Tel: +86 21 62351596  
Fax: +86 21 62351595

E-mail (Sales) [cn.sales@ftdichip.com](mailto:cn.sales@ftdichip.com)  
E-mail (Support) [cn.support@ftdichip.com](mailto:cn.support@ftdichip.com)  
E-mail (General Enquiries) [cn.admin@ftdichip.com](mailto:cn.admin@ftdichip.com)

### Web Site

<http://ftdichip.com>

## Distributor and Sales Representatives

Please visit the Sales Network page of the [FTDI Web site](#) 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 Future Technology Devices International Ltd (FTDI) 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 FTDI devices and other materials) is provided for reference only. While FTDI has taken care to assure it is accurate, this information is subject to customer confirmation, and FTDI disclaims all liability for system designs and for any applications assistance provided by FTDI. Use of FTDI 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 FTDI 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. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH, United Kingdom. Scotland Registered Company Number: SC136640

## Appendix A – References

### Document References

Eclipse: <https://eclipse.org/>

MikroElektronika: <http://www.mikroe.com/>

<http://www.ftdichip.com/Support/Documents/AppNotes.htm>

AN\_352 FT51\_Installation\_Guide

AN\_325 FT900\_Tool\_Chain\_Installation\_Guide

### Acronyms and Abbreviations

Terms	Description
USB	Universal Serial Bus
IDE	Integrated Development Environment
MCU	Micro Controller Unit
PC	Personal Computer
JDK	Java Developer Kit
SDCC	Small Device C Compiler
PATH	Environment Variable
SDK	Software Development Kit
DFU	Device Firmware Upgrade
GDB	GNU Debugger
C/C++	Programming Language

## Appendix B – List of Tables & Figures

### List of Figures

Figure 2.1 Eclipse Workspace Launcher .....	5
Figure 2.2 Initial Eclipse Startup .....	6
Figure 2.3 Restore Eclipse Icon .....	7
Figure 2.4 Standard Eclipse IDE Screen .....	7
Figure 4.1 Import Project .....	9
Figure 4.2 FT51A New Project .....	10
Figure 5.1 Add Directory Path .....	11
Figure 7.1 FT51A External Tools Configuration .....	13
Figure 7.2 FT900 External Tool Configuration .....	14
Figure 8.1 Debug Environment .....	15
Figure 8.2 FT51A Debug Configuration .....	16
Figure 8.3 FT90x Debug Configuration .....	17



## Appendix C – Revision History

Document Title: TN\_160 Eclipse Projects  
Document Reference No.: FT\_001175  
Clearance No.: FTDI# 475  
Product Page: <http://www.ftdichip.com/FTProducts.htm>  
Document Feedback: [Send Feedback](#)

Revision	Changes	Date
1.0	Initial Release	2015-10-13