



# Application Note

## AN\_353

# FT32 GNU Toolchain Quick Start Guide

**Version 1.0**

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

This document provide an overview of FT32 GNU toolchain and its quick usage, it also provide solutions for FT90x security feature and chip configuration function. At the end of this document, resource for further study is provided.

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)**

Unit 1, 2 Seaward Place, Glasgow G41 1HH, United Kingdom

Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758

Web Site: <http://ftdichip.com>

Copyright © 2015 Future Technology Devices International Limited

## **Table of Contents**

<b>1 Introduction .....</b>	<b>2</b>
<b>1.1 GCC .....</b>	<b>2</b>
<b>1.2 GNU Binutils .....</b>	<b>2</b>
<b>2 Basic Tool Chain Usage .....</b>	<b>4</b>
<b>3 FT90x Security Feature Implementation.....</b>	<b>5</b>
<b>4 FT90x Chip Configuration Function Implementation..</b>	<b>6</b>
<b>5 Further Study .....</b>	<b>7</b>
<b>6 Contact Information .....</b>	<b>8</b>
<b>Appendix A – References .....</b>	<b>9</b>
Document References .....	9
Acronyms and Abbreviations.....	9
<b>Appendix B – Revision History .....</b>	<b>10</b>

## 1 Introduction

The GNU based software development tool set is the first available tool chain for FT32 processor core powered SoC, such as FT90x.

The GNU tool chain for FT32 includes the following tools:

### 1.1 GCC

GCC stands for GNU Compiler Collection. GCC is highly flexible compiler system. It has different compiler front-ends for different languages. It has many back-ends that generate assembly code for many different processors and host operating systems. They all share a common "middle-end", containing the generic parts of the compiler, including a lot of optimizations.

GCC is different from most other compilers. GCC focuses on translating a high-level language to the target assembly only. FT32 GCC has two available compilers for the FT32: C language and C++. The compiler itself does not assemble or link the final code.

GCC is also known as a "driver" program, in that it knows about, and drives other programs seamlessly to create the final output. The assembler and the linker are part of another open source project called GNU Binutils. GCC knows how to drive the GNU assembler (gas) to assemble the output of the compiler. GCC knows how to drive the GNU linker (ld) to link all of the object modules into a final executable.

When GCC is built for the FT32 target, the actual program names are prefixed with "ft32-elf-". So the actual executable name for FT32 GCC is: ft32-elf-gcc

### 1.2 GNU Binutils

The name GNU Binutils stands for "Binary Utilities". It contains the GNU assembler (GAS), and the GNU linker (ld), but also contains many other utilities that work with binary files that are created as part of the software development toolchain.

Again, when these tools are built for the FT32 target, the actual program names are prefixed with "ft32-elf-". For example, the assembler program name, for a native assembler is "as" (even though in documentation the GNU assembler is commonly referred to as "gas"). But when built for an FT32 target, it becomes "ft32-elf-as". Below is a list of the programs that are included in Binutils:

ft32-elf-as

The Assembler.

ft32-elf-ld

The Linker.

ft32-elf-ar

Create, modify, and extract from libraries (archives).

**ft32-elf-ranlib**

Generate index to library (archive) contents.

**ft32-elf-objcopy**

Copy and translate object files to different formats.

**ft32-elf-objdump**

Display information from object files including disassembly.

**ft32-elf-size**

List section sizes and total size.

**ft32-elf-nm**

List symbols from object files.

**ft32-elf-strings**

List printable strings from files.

**ft32-elf-strip**

Discard symbols from files.

**ft32-elf-readelf**

Display the contents of ELF format files.

**ft32-elf-addr2line**

Convert addresses to file and line.

## 2 Basic Tool Chain Usage

As the FT32 tool chain is GNU based, all the commands employ GNU tool chain's interface and functions. To have a quick guide of all the command usage, you can issue the command with "--help" parameter under command prompt and the tool will print out its general usage.

### 3 FT90x Security Feature Implementation

FT90x has a special security feature which enables the invisibility of all or part of the firmware in flash memory. This will prevent anyone from finding any information by reverse engineering of the firmware binary.

The segments in FT90x built-in flash memory which need to be protected are programmed by FT90x eFuse module. However, it is the firmware developer's responsibility to put the code that needs to be protected to the correct location.

In this section, one of the possible methods to put the certain function(s) to a designated memory zone using GNU tool chain is provided.

For those functions that need to be relocated to certain memory zone must have a function prototype declaration like following:

```
RETURN_TYPE    function_name    (PARAMETER_LIST)    __attribute__((section  
("SECTION_NAME")));
```

Where RETURN\_TYPE is function's return type, function\_name is the name of the function, PARAMETER\_LIST is the function's parameters and SECTION\_NAME is the section name that you will use later.

If you want to put several functions into one memory bar, be sure to put them under the same section name.

To enable this feature, you have to use the customized linker script instead of the default one, of course, you could start from modifying the default linker script.

From the default linker script, add a session after the .text output session, this assumes that the hidden session will be put after the normal code segment. In this new output session's description, add the input object file name and the session name that we defined above as input section.

In the Makefile, under linker command, i.e. ft32-elf-ld, add the "--section-start" option with the SECTION\_NAME as above and assign the appropriate offset to it.

## 4 FT90x Chip Configuration Function Implementation

Sometimes, there is some configuration information needed for firmware, such as MAC address for Ethernet, USB peripheral PID, VID etc. This kind of information may be different for each device; however, it is not practical if we have to build different firmware for each device.

To solve this problem, we need to find a solution which could keep the firmware intact while changing the configuration information. One the possible method by the GNU tool chain is discussed as follows:

In C code, define the data structure which defines configuration information, then define a variable of this data structure such as following:

```
struct ft900_config
{
    uint8_t eth_mac_addr[6];
    uint8_t CAN_identifier[4];
};

struct ft900_config __flash__ stConfig __attribute__((section(
"chip_config"))) =
{
    {0x01, 0x02, 0x03, 0x04, 0x05, 0x06},
    {0x11, 0x12, 0x13, 0x14}
};
```

This is to create the variable which contains the default value of the chip configuration information and belongs to the special section.

Then in the Makefile, when linker is executed, apply the “—section-start” with section name above and appropriate offset value. After that, we can either directly change the content of firmware binary file at the offset specified or we can modify the firmware downloading tools to include the updated configuration information and the data that firmware read will be changed without the need or rebuild firmware.

## 5 Further Study

For further study of FT32 GNU based tool chain, we can refer to following webpages.

For GCC: <https://gcc.gnu.org/onlinedocs/>

For binutils: <https://sourceware.org/binutils/docs-2.24/>



## 6 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 – 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 – 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 – 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

<http://www.ftdichip.com/Products/ICs/FT90x.html>

### Acronyms and Abbreviations

Terms	Description
FT32	FTDI Proprietary 32-bit Core
GAS	GNU Assembler
GCC	GNU Compiler Collection
GNU	GNU (Gnu's Not Unix) Operating System
MAC	Media Access Control (Ethernet)
PID	Product ID
SoC	System on Chip
USB	Universal Serial Bus
VID	Vendor ID

## Appendix B – Revision History

Document Title: AN\_353 FT32 GNU Toolchain Quick Start Guide  
Document Reference No.: FT\_001136  
Clearance No.: FTDI# 473  
Product Page: <http://www.ftdichip.com/FTProducts.htm>  
Document Feedback: [Send Feedback](#)

Revision	Changes	Date
1.0	Initial Release	2015-10-13