



Application Note

AN_419

FT90x I2C EEPROM Utility

Version 1.0

Issue Date: 2019-06-14

This application note describes the I²C EEPROM Utility implemented on the FT900. The network configuration is stored on the external I²C EEPROM. There is additional functionality allowing for reading and writing blocks of the I²C EEPROM memory.

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

Table of Contents

1 Introduction	3
1.1 Overview	3
1.2 Scope	3
1.2.1 Features	4
1.2.2 Enhancement	4
2 Project Overview	5
2.1 Main Program	5
2.2 Network Code.....	5
2.2.1 Network Abstraction	5
2.2.2 lwIP Library.....	5
2.3 Persistent Storage.....	5
2.4 Web Pages	7
2.5 Other Features	8
3 Code Structure.....	9
4 Configuration.....	11
4.1 Default Configuration	12
4.1.1 Network.....	12
4.1.2 UART.....	12
4.2 Modifying the Configuration	12
4.2.1 Network Configuration	13
4.2.2 Read EEPROM Memory	14
4.2.3 Write EEPROM Memory	15
5 Importing into the FT9xx Toolchain.....	16
5.1 Changing the Application Software	16
6 Contact Information	17
Appendix A – References	18
Document References	18
Acronyms and Abbreviations.....	18

Appendix B – List of Tables & Figures	19
List of Tables.....	19
List of Figures	19
Appendix C – Revision History	20

1 Introduction

The FT900/FT901 and FT905/FT906 devices include an Ethernet interface. This can be used with a suitable networking stack to allow the device to be networked. This application note shows how the Ethernet interface can be used to bridge I²C EEPROM to a configuration webpage. In addition, the network information such the MAC and IP address are retrieved and stored on the onboard external I²C EEPROM, giving the advantage that subsequent application software updates will not overwrite this information.

1.1 Overview

Any network configuration changes are stored onto the EEPROM via I²C. The user can also control the EEPROM via I²C to perform reads and writes.

The network interface supports DHCP to obtain configuration information or it can be configured using a web page served by the FT90x firmware. EEPROM reads and writes are also performed using the same web page.

The configuration is stored either on FT90x Flash or external EEPROM, so that they are not lost after power off:

- FT90x FlashROM: Hostname, I²C Address
- External EEPROM: MAC address, Alternate MAC address, IP/ gateway/mask address, DHCP setting

If the EEPROM key is not valid, the software enables DHCP and sets other configuration information to the default values.

A pre-programmed permanently protected MAC address resides at the last 6 bytes of memory in the EEPROM. This was preprogrammed by Microchip. This is read by the application code and used for the network configuration. An alternative MAC address can be used which is stored in a writable area of the EEPROM memory. The application checks at startup if the alternative MAC address should be used.

Third-party open source code is used to implement a TCP/IP stack in this application note:

- TCP/IP - lwIP (LightWeight IP) library which has been ported to the FT90x.
- Printf – tinyprintf.

Links to resources for these libraries are in 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.

1.2 Scope

I²C can be used to control lots of things like EEPROM memory, MCUs, Data converters, LED controllers, Accelerometers, etc.

This simple example demonstrates how to control I²C EEPROM from a web interface where the PC and FT90x exist on the same private network.

The external I²C EEPROM used in this example is the [24AA02E48T-I/OT](#) which is a 2K serial EEPROM device. This memory is present on the [MM900EVxA](#) , excluding the MM900EV-LITE. Note that the first 1K is writable and the last 1k is write protected since the preprogrammed MAC address resides there.

1.2.1 Features

The application note highlights the use of lwIP to provide a TCP/IP stack. The "arch" folder of lwIP in the source code for the application note contains the architecture-specific parts of lwIP required for the FT90x.

1.2.2 Enhancement

Enhancements to this application might include:

- Provide a more general I²C interface to control other I²C device types
- Allow only permitted IP addresses to connect to the device.
- Reduce the overall size of the application code.

This I²C EEPROM utility can be used to update the EEPROM. Full source code is provided allowing users to build and modify if required:

<http://brtchip.com/ft90x/>

See section 5 Importing into the FT9xx Toolchain for more information.

2 Project Overview

The project files for this application note are divided into the following folders.

Folder	Description
Source	Application source code and abstraction files.
httpdfs	File system for web server.
lib	Library files.
lib\tinyprintf	tinyprintf library.
lib\lwip	lwIP library.

Table 2.1 Project Files Overview

The application source code is contained within the "Sources" folder. The main() function and the high-level functions of the application are in the "main.c" file. The network interface is within "net.c".

2.1 Main Program

The main program is responsible for handling configuration and control by the web interface. The code generating data for the simple server side includes (SSI) functions in the lwIP httpd server. Data is returned through HTTP GET requests to update the configurations. The httpd server uses callbacks for the server size and includes the HTTP GET handler.

2.2 Network Code

The networking layer is provided by the lwIP library which has been ported to support the FT90x. The lwIP code is available separately from FTDI/Bridgetek with the FT90x architecture extensions.

2.2.1 Network Abstraction

The network interface is abstracted from the main project in the file "net.c". All lwIP features which refer to an interface (netif) are passed through this layer.

The lwIP network interface is initialized and configured using data supplied from the main application. The function which processes incoming packets for the network interface within lwIP is called from the net_tick() function.

2.2.2 lwIP Library

The library is available with a BSD-style license. Version 2.0.0 of the code is used. No changes have been made to the code from the lwIP project page.

The FT90x architecture code is responsible for obtaining packets from the Ethernet interface and transmitting packets to the Ethernet interface.

The application makes extensive use of the callback methods in lwIP to permit sending and receiving of data.

The httpd app is used to provide an interface for configuration. This makes use of server side includes (SSI) and CGI scripts (HTTP GET requests) to present configuration data to the user and change it when requested to do so.

2.3 Persistent Storage

Networking parameters such as MAC address, IP/ gateway/mask address and DHCP setting are stored on external EEPROM via I²C reads and writes.

The following variables in the application code show the location of each of the parameters which reside in the I²C EEPROM:

```

struct eeprom_net_config {
    uint16_t key;
    uint8_t dhcp;
    uint8_t alt_mac;
    ip_addr_t ip;
    ip_addr_t gw;
    ip_addr_t mask;
};
    
```

Figure 2.1 shows an example of this through an EEPROM block read.

I2C EEPROM Memory Block (32 bytes):

Byte [0:7]:	45	45	00	00	01	FF	AC	10
Byte [8:15]:	03	23	AC	10	02	05	FF	FF
Byte [16:23]:	FC	00	FF	FF	FF	FF	FF	FF
Byte [24:31]:	FF							

Figure 2.1 I²C EEPROM Configuration Location 0x0

Translation of the memory dump:

```

key = 0x4545 = EEPROM_VALID_KEY
dhcp = 0x01 = DHCP Enabled,
alt_mac = 0x00 = Alternative MAC disabled
ip = 0xAC 10 03 1B = 172.16.3.27
gw= 0xAC 10 02 05 = 172.16.2.5
mask = 0xFF FF FC 00 = 255.255.252.0
    
```

The alternative MAC address is stored at offset 0x7A in the EEPROM as shown in Figure 2.2.

```

altmac = 0x00 04 A3 65 9B 9A = 00:04:A3:65:9B:9A (previously programmed)
    
```

I2C EEPROM Memory Block (32 bytes):

Byte [0:7]:	FF							
Byte [8:15]:	FF							
Byte [16:23]:	FF							
Byte [24:31]:	FF	FF	00	04	A3	65	9B	9A

Figure 2.2 I²C EEPROM Configuration Alternative MAC Address

The permanent MAC address is stored in the last 6 bytes of the EEPROM memory (0xFA to 0xFF) as shown in Figure 2.3. Note that this is permanently protected and cannot be overwritten. The alternative MAC address is similar and resides in location 0x7A to 0x7F which is in a writable area of EEPROM so it can be updated if required.

I2C EEPROM Memory Block (32 bytes):

Byte [0:7]:	FF							
Byte [8:15]:	FF							
Byte [16:23]:	FF							
Byte [24:31]:	FF	FF	00	04	A3	65	9B	9E

Figure 2.3 I²C EEPROM Configuration Location 0xE0

Figure 2.4 shows the readable format of the data shown in Figure 2.1 and Figure 2.3.

Network Configuration:

Hostname:	<input type="text" value="I2CBridge"/>
MAC Address:	<input type="text" value="00:04:A3:65:9B:9E"/>
Alt MAC:	<input type="checkbox"/>
Alt MAC Address:	<input type="text" value="00:00:00:00:00:00"/>
Enable DHCP:	<input checked="" type="checkbox"/>
IP address:	<input type="text" value="172.16.3.27"/>
Gateway:	<input type="text" value="172.16.2.5"/>
Netmask:	<input type="text" value="255.255.252.0"/>
	<input type="button" value="Apply"/>

Figure 2.4 I²C EEPROM Configuration

When the FlashROM is reprogrammed by the FT90x Programming Utility this data will not be overwritten since it resides in external memory, with exception of the hostname.

2.4 Web Pages

The configuration web pages are stored in a folder called httpdfs on the top level of the directory structure for the project. The raw html and image files are compiled into a single C file (fsdata.c) which is included in the source code of the application. An internal library in the lwIP httpd code reads this data and extracts files for the web service.

A batch file (makefs.bat) and a compilation utility (makefsdata.exe) Windows platforms are included to perform the compilation and place the fsdata.c file in the correct place. Changes to the web pages have to be compiled before the application is recompiled.

To make changes to the webpage, edit the htm or html files using a text editor. Knowledge of HTML code is required. The pages must be rebuilt by browsing to the project's top level directory in a windows console and typing makefs.bat as shown in Figure 2.5.

```
makefsdata - HTML to C source converter
  by Jim Pettinato           - circa 2003
  extended by Simon Goldschmidt - 2009

Writing to file "lib\lwip\apps\httpd\fsdata.c"
HTTP 1.0 header will be statically included.
  Processing all files in directory httpdfs and subdirectories...

processing subdirectory /img/...
processing /img/1.gif...
processing /img/2.gif...
processing /img/3.gif...
processing /img/off.png...
processing /img/on.png...
processing /404.html...
processing /index.shtm...
processing /restart.shtm...
processing /update.html...

Creating target file...

Processed 9 files - done.

httpd file system done.
```

Figure 2.5 Makefs.bat Output

Further details on how lwIP performs Server Side Include (SSI) and acts on the HTTP GET requests used to update the configuration are to be found in the [lwIP documentation](#).

2.5 Other Features

The DFU-C facility is enabled for this application. This is called from the macro `STARTUP_DFU()` in the `main()` function. It will briefly enable the USB device on the FT90x and allow a DFU utility to update the application code. This can be removed entirely or configured to alter the number of milliseconds it will wait before closing the USB device and continuing with the application.

3 Code Structure

Data sent and received from the network are handled by the lwIP library. Simple FT90x API functions used in the HTTPD SSI and CGI functions:

- `i2cm_read`. Reads a specified number of bytes from an I²C device.
- `i2cm_write`. Writes a specified number of bytes to an I²C device.

More information on these FT90x API functions can be found in [AN_365 FT9xx API Programmers Manual](#).

Data received from the network like the I²C address and start address are used in each of the functions above and is handled in the HTTPD SSI and CGI functions.

The main function "`i2c_bridge()`" shows the main network setup and the lwIP library (Figure 3.1).

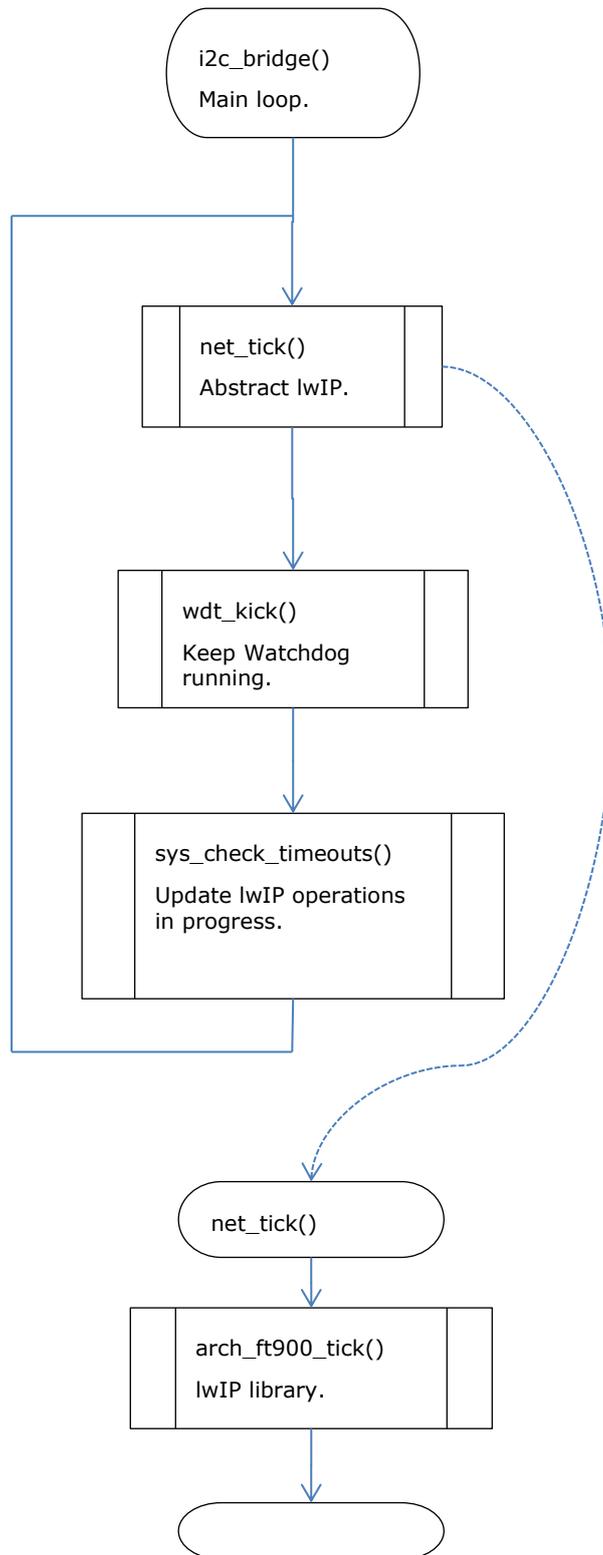
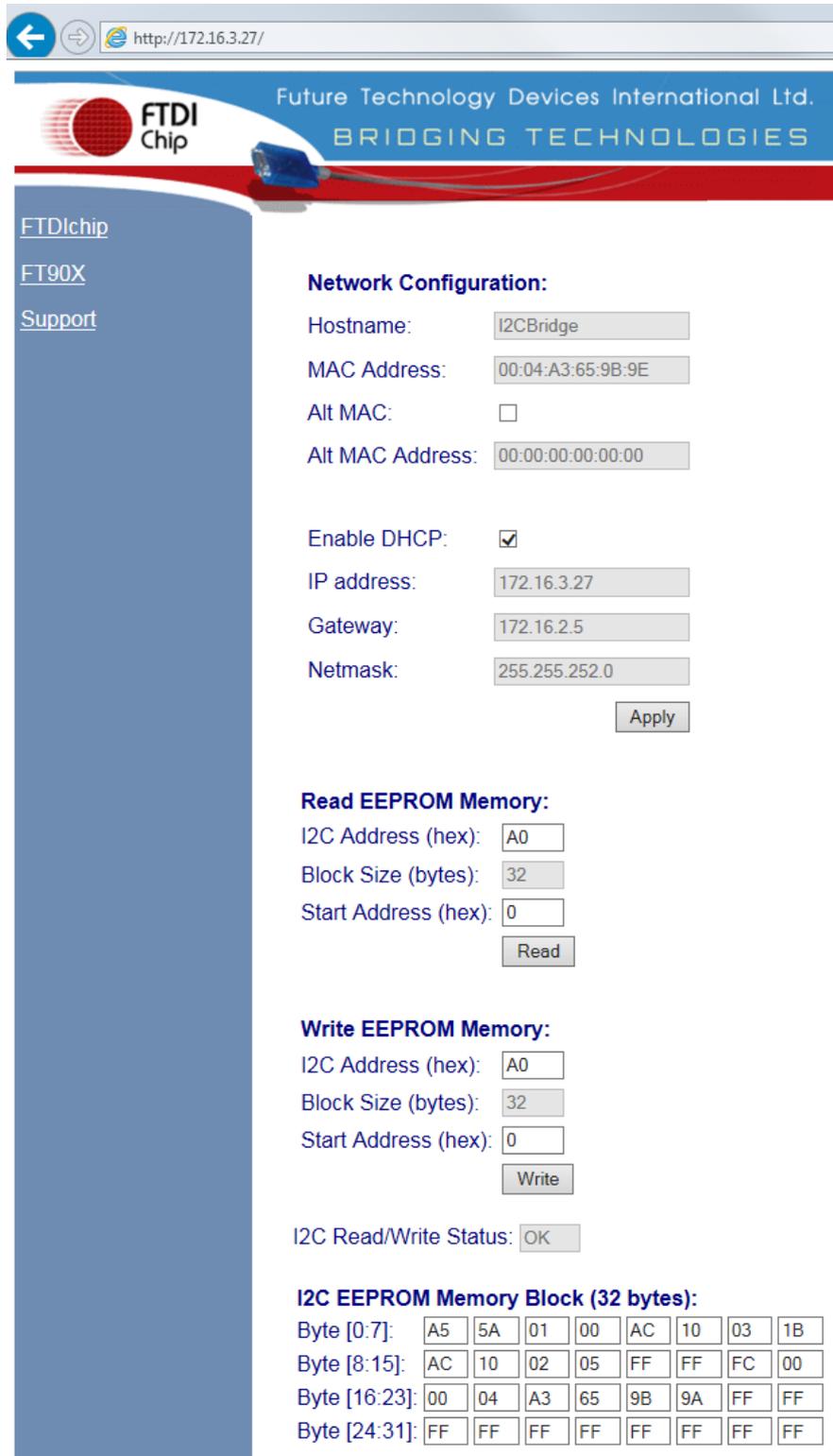


Figure 3.1 Main Function Flowchart

4 Configuration

This section describes the default configuration used by the application and how to change the configuration. Figure 4.1 shows a screen capture of the default configuration.



Browser address bar: http://172.16.3.27/

Page Header: FTDI Chip Future Technology Devices International Ltd. BRIDGING TECHNOLOGIES

Left Sidebar: [FTDIchip](#), [FT90X](#), [Support](#)

Network Configuration:

Hostname:

MAC Address:

Alt MAC:

Alt MAC Address:

Enable DHCP:

IP address:

Gateway:

Netmask:

Read EEPROM Memory:

I2C Address (hex):

Block Size (bytes):

Start Address (hex):

Write EEPROM Memory:

I2C Address (hex):

Block Size (bytes):

Start Address (hex):

I2C Read/Write Status:

I2C EEPROM Memory Block (32 bytes):

Byte [0:7]:	A5	5A	01	00	AC	10	03	1B
Byte [8:15]:	AC	10	02	05	FF	FF	FC	00
Byte [16:23]:	00	04	A3	65	9B	9A	FF	FF
Byte [24:31]:	FF							

Figure 4.1 Web Configuration and Control Interface

A web configuration page is accessible when the device connects to the network. This allows both networking settings and I²C EEPROM control to be altered.

In order to serve the configuration web page the device must be connected to the network and have obtained (or have been configured with) an IP address. DHCP is enabled by default if the EEPROM is blank allowing the application to request an IP address when it is first started.

In many cases it is possible to query the DHCP server on a network to find out the IP address allocated to a particular device. In this application the device will send the DHCP server the hostname "I2CBridge" which some DHCP servers may make available to allow users to identify the allocated address more easily.

Another way to obtain the IP address to access the web configuration page is to check the UART debug text, while NET_DEBUG is defined in net.c, for example:

```
IP config:
ipaddr = 172.16.3.125
gwaddr = 172.16.2.5
netmask = 255.255.252.0
```

4.1 Default Configuration

The MAC address is obtained from the onboard 24AA02E48T-I/OT EEPROM during initialization (see 'net_init' in net.c). This memory is present on the [MM900EVxA](#), excluding the MM900EVLITE. The pre-programmed permanently protected MAC address resides at the last 6 bytes of memory in the EEPROM.

Note: The code should be changed if you are using hardware that does not have this external EEPROM.

4.1.1 Network

DHCP is turned on by default unless the EEPROM config is valid and DHCP is not set. The application will ask for an IP address, gateway and netmask from any available DHCP server and use these values to configure the network interface. It will also provide a hostname of "I2CBridge" to the DHCP server.

4.1.2 UART

The following settings are used by default for the UART which displays debug messages:

- 19200 baud
- 8 bits data
- 1 stop bit
- no parity

4.2 Modifying the Configuration

The configuration web page is split into 3 sections:

- a 'Network Configuration' section.
- a 'Read EEPROM Memory' section.
- a 'Write EEPROM Memory' section.
- an 'I²C EEPROM Memory Block (32 bytes)' section

There are also shortcut links on the left to visit the FTDI web site, the FT90x section and the support information within the FTDI website.

4.2.1 Network Configuration

The network configuration section has 2 sub-sections.

The first is the MAC address and hostname. As has been mentioned earlier the MAC address is required by Ethernet to be unique on a network. This is the hardware address used for all low-level communications on Ethernet. A network will act in an unspecified manner if this is duplicated.

It is recommended that bit 2 in the most significant byte (first octet transmitted) be set to 1 to indicate that the MAC address is locally administered.

The hostname field is fixed in the code and is sent to the DHCP server during the DHCP negotiation and may be used by the server to identify a device. In some DHCP setups it can be passed onto DNS to allow the hostname to be used seamlessly within a local network.

The third network option is to enable or disable DHCP. When enabled, this will instruct the application to request an IP address, gateway address and netmask from a DHCP server. Therefore the fields that configure these values are disabled. The values obtained from the server are shown in the fields for information only.

If DHCP is disabled then the 3 fields are enabled allowing specific addresses to be configured and are subsequently read from the external EEPROM.

There is no error checking that the values in the form are correctly formatted. The IP address, gateway and netmask are 4 decimal values separated by dots.

4.2.1.1 Programming the Network Configuration

To program the Network Configuration via the configuration webpage, make edits as required then click on the 'Apply' button, then click on 'Home' on the next page. Note that the FT90x IC will be rebooted. You may also have to reset the hardware.

Network Configuration:

Hostname:	<input type="text" value="I2CBridge"/>
MAC Address:	<input type="text" value="00:04:A3:65:9B:9E"/>
Alt MAC:	<input checked="" type="checkbox"/>
Alt MAC Address:	<input type="text" value="00:04:A3:65:9B:9A"/>
Enable DHCP:	<input type="checkbox"/>
IP address:	<input type="text" value="172.16.3.27"/>
Gateway:	<input type="text" value="172.16.2.5"/>
Netmask:	<input type="text" value="255.255.252.0"/>
	<input type="button" value="Apply"/>

Figure 4.2 Network Configuration

4.2.2 Read EEPROM Memory

The default I²C address in this example is 0xA0 to allow read of the onboard EEPROM for the MAC address and other network configuration parameters if they exist. This can be changed in the code or on the webpage to access different I²C EEPROMs. Note that the I²C address will not be retained after a power cycle and will default to 0xA0.

To perform an I²C read, enter the I²C address and EEPROM start address as shown in Figure 4.3. The size of the memory is fixed to 32 bytes in the application code. Click on 'Read'.

Read EEPROM Memory:

I2C Address (hex):

Block Size (bytes):

Start Address (hex):

Figure 4.3 Read EEPROM Memory

For example, if you want to read the last 32 bytes of the EEPROM memory, enter the start address as E0 then click 'Read'.

The resulting memory read is shown in the 'I²C EEPROM Memory Block' at the bottom of the webpage. Figure 4.4 shows a 32 byte read of the EEPROM from location 0 which has some network parameters already programmed.

I2C EEPROM Memory Block (32 bytes):

Byte [0:7]:	45	45	00	00	01	FF	AC	10
Byte [8:15]:	03	23	AC	10	02	05	FF	FF
Byte [16:23]:	FC	00	FF	FF	FF	FF	FF	FF
Byte [24:31]:	FF							

Figure 4.4 Read EEPROM Memory Dump

Note: reading a 32 byte block which goes beyond the last location of the EEPROM memory will result in a mirroring effect where the start of the EEPROM memory data will also be returned. The FT90x API function `i2cm_read` is used to read from the I²C EEPROM. This is a fairly simple function to use and is integrated within the HTTPD SSI and CGI functions in the application code. More information on this FT90x API functions can be found in [AN_365_FT9xx API Programmers Manual](#).

4.2.3 Write EEPROM Memory

The default I²C address in this example is 0xA0 to allow read of the onboard EEPROM for the MAC address and other network configuration parameters if they exist. This can be changed in the code or on the webpage to access different I²C EEPROMs. Note that the I²C address will not be retained after a power cycle and will default to 0xA0.

To perform an I²C write, edit the data in the 'I²C EEPROM Memory Block' at the bottom of the page with hexadecimal numbers as shown in Figure 4.5. It may be wise to perform an I²C read first.

I2C EEPROM Memory Block (32 bytes):

Byte [0:7]:	45	45	00	00	01	FF	AC	10
Byte [8:15]:	03	23	AC	10	02	05	FF	FF
Byte [16:23]:	FC	00	FF	FF	FF	FF	FF	FF
Byte [24:31]:	FF							

Figure 4.5 Write EEPROM Memory Block

Enter the I²C address and EEPROM start address as shown in Figure 4.6. The size of the memory is fixed to 32 bytes in the application code.

Write EEPROM Memory:

I2C Address (hex):

Block Size (bytes):

Start Address (hex):

Figure 4.6 Write EEPROM Memory

The resulting memory write is shown in the 'I²C EEPROM Memory Block' at the bottom of the webpage when the webpage refreshes.

Note: writing a 32 byte block which goes beyond the last location of the EEPROM memory will result in a mirroring effect where the start of the EEPROM memory data will also be written.

The FT90x API function `i2cm_write` is used to write to the I²C EEPROM. This is a fairly simple function to use and is integrated within the HTTPD SSI and CGI functions in the application code.

More information on this FT90x API functions can be found in [AN_365 FT9xx API Programmers Manual](#).

5 Importing into the FT9xx Toolchain

The AN_419 Firmware found at the following link can be easily imported into the [FT9xx Toolchain](http://brtchip.com/ft90x/):
<http://brtchip.com/ft90x/>

Once installed, select File --> Import --> General --> Existing Projects into Eclipse, and point to the downloaded and extracted project directory.

The project will appear in Eclipse Project Explorer as shown in Figure 5.1.

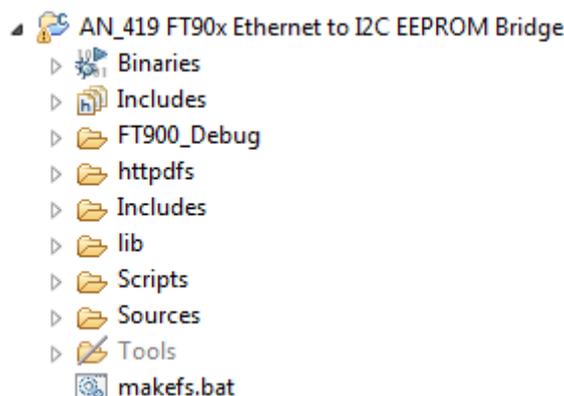


Figure 5.1 Eclipse Project Structure

5.1 Changing the Application Software

The application software provided can be altered and changed if required. The [FT9xx Toolchain](#) is a free tool to enable code development and debug for the FT90x series and is based on plug-ins for the free popular IDE using the GCC compiler.

For example, the following definitions can be changed to alter the I²C address and network hostname:

```
uint8_t i2cadd_var = 0xA0;  
char hostname[64] = {'I', '2', 'C', 'B', 'r', 'i', 'd', 'g', 'e', 0};
```

With each software change, the project should be rebuilt and reprogrammed into the FT90x IC. Please refer to [AN_325 FT9xx Toolchain Installation Guide](#) for further information.

6 Contact Information

Head Quarters – Singapore

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

E-mail (Sales) sales.apac@brtchip.com
E-mail (Support) support.apac@brtchip.com

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

E-mail (Sales) sales.apac@brtchip.com
E-mail (Support) 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) sales.emea@brtchip.com
E-mail (Support) support.emea@brtchip.com

Branch Office – Vietnam

Bridgetek VietNam Company Limited
Lutaco Tower Building, 5th Floor, 173A Nguyen Van
Troï,
Ward 11, Phu Nhuan District,
Ho Chi Minh City, Vietnam
Tel : 08 38453222
Fax : 08 38455222

E-mail (Sales) sales.apac@brtchip.com
E-mail (Support) support.apac@brtchip.com

Web Site

<http://brtchip.com/>

Distributor and Sales Representatives

Please visit the Sales Network page of the [Bridgetek 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 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

FT90x Datasheet: <http://brtchip.com/m-ft9/>

lwIP <http://savannah.nongnu.org/projects/lwip/>

tinyprintf <http://www.sparetimelabs.com/tinyprintf/tinyprintf.php>

[AN_325 FT9xx Toolchain Installation Guide](#)

[AN_365 FT9xx API Programmers Manual](#)

DFU [AN_380 FT900 Bootloader DFU Usage](#)

[TN_157 Ethernet Explained](#)

[FT90x Toolchain](#)

AN_419 Firmware:

<http://brtchip.com/ft90x/>

24AA02E48T-I/OT EEPROM <http://www.microchip.com/wwwproducts/en/24AA02E48>

MM900EVxA <http://brtchip.com/ft9xx-development-modules-landing/>

Acronyms and Abbreviations

Terms	Description
CGI	Common Gateway Interface
DFU	Device Firmware Update
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
MAC	Media Access Controller
SSI	Server Side Include
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver and Transmitter (Serial Port)
I ² C	Inter-Integrated Circuit
EEPROM	Electrically Erasable Programmable Read Only Memory

Appendix B – List of Tables & Figures

List of Tables

Table 2.1 Project Files Overview	5
--	---

List of Figures

Figure 2.1 I ² C EEPROM Configuration Location 0x0.....	6
Figure 2.2 I ² C EEPROM Configuration Alternative MAC Address	6
Figure 2.3 I ² C EEPROM Configuration Location 0xE0	7
Figure 2.4 I ² C EEPROM Configuration	7
Figure 2.5 Makefs.bat Output	8
Figure 3.1 Main Function Flowchart	10
Figure 4.1 Web Configuration and Control Interface	11
Figure 4.2 Network Configuration.....	13
Figure 4.3 Read EEPROM Memory	14
Figure 4.4 Read EEPROM Memory Dump	14
Figure 4.5 Write EEPROM Memory Block	15
Figure 4.6 Write EEPROM Memory	15
Figure 5.1 Eclipse Project Structure.....	16

Appendix C – Revision History

Document Title: AN_419 FT90x I2C EEPROM Utility
Document Reference No.: BRT_000132
Clearance No.: BRT#140
Product Page: <http://brtchip.com/m-ft9/>
Document Feedback: [Send Feedback](#)

Revision	Changes	Date
1.0	Initial Release	2017-03-27