This technical note demonstrates how to modify API functions provided with the FT9xx toolchain.

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.
Table of Contents

1 Introduction .................................................................................................................. 2

2 Modification Process ................................................................. 3

   2.1 Create Project ........................................................................................................ 3

   2.2 Copy Source ........................................................................................................... 6

   2.3 Edit the Code .......................................................................................................... 6

   2.4 Rebuild ................................................................................................................... 8

3 Conclusion .................................................................................................................... 9

4 Contact Information ..................................................................................................... 10

Appendix A – References ............................................................................................... 11

   Document References ................................................................................................ 11

   Acronyms and Abbreviations ...................................................................................... 11

Appendix B – List of Tables & Figures ........................................................................ 12

   List of Tables ............................................................................................................... 12

   List of Figures .............................................................................................................. 12

Appendix C – Revision History ...................................................................................... 13
1 Introduction

The FT9xx Peripheral Driver Library is a collection of ‘C’ language based functions that are intended to ease the development of applications running on the FT9xx Microcontroller. See Figure 1.1. Details of the API functions can be found in AN_365 FT9xx API Programmers Manual.

Sometimes the functionality of the API driver functions provided might not meet the needs of the particular operation for a custom design. However, the user has a couple of options:

1. Write your own functions to access the IC at register level. See AN_324 FT900 User Manual for more information.
2. Modify the API driver functions to change the functionality.
3. Reference the API driver source code to create your own functions.

This technical note demonstrates the 2nd option, modifying the API driver functions provided with the FT9xx toolchain.

Figure 1.1 –FT9xx Software Stack
2 Modification Process

To change the way any of the source code in the FT9xx API driver library works, copying the source code of the relevant module into the Eclipse project ignores the version in the library and uses the locally copied version, allowing users to change the code.

An example could be with the I²C master library. At time of writing, the I²C read and write functions only allow for 8-bit register addressing. The API driver functions can be modified to allow for 16-bit register addressing. This example scenario is used in the next sections to help demonstrate the process.

2.1 Create Project

Users can create new projects in Eclipse via File → New Project → C/C++ Project as shown in Figure 2.1.

![Figure 2.1 – New C Project]

Enter the project name, select Executable → Empty Project, and select Bridgetek FT9xx GCC under Toolchains as shown in Figure 2.2.
Click Next and select the MCU types that you require (FT900 / FT930) as shown in Figure 2.3.
Click Next then Finish.

The project will appear in Project Explorer as shown in Figure 2.4.

```
/* Enable the I2C Slave device... */
sys_enable(sys_device_i2c_master);

if defined(__FT900__)  
  #if I2C_CHANNEL == 0
    /* Set GPIDO4 to I2CM_SCL and GPIDO5 to I2CM_SDA... */
    gpio_function(44, pad_i2c0_scl);
    gpio_pull(44, pad_pull_none);
    gpio_function(45, pad_i2c0_sda);
    gpio_pull(45, pad_pull_none);
  #elif I2C_CHANNEL == 1
    /* Set GPIDO4 to I2CM_SCL and GPIDO6 to I2CM_SDA... */
    gpio_function(44, pad_i2c1_scl);
    gpio_pull(44, pad_pull_none);
    gpio_function(46, pad_i2c1_sda);
    gpio_pull(46, pad_pull_none);
  #else
    /* Set the I2C Master pins to channel 1 */
    sys_i2c_swp(1);
  endif /* I2C CHANNEL */

  #elif defined(__FT910__)  
  /* Set GPIO12 to I2CM_SCL and GPIO13 to I2CM_SDA... */
  gpio_function(12, pad_i2cm_scl);
  gpio_pull(12, pad_pull_none);
  gpio_function(13, pad_i2cm_sda);
  gpio_pull(13, pad_pull_none);

  endif /* I2C CHANNEL */

  /* Set up I2C */
  i2cm_init(I2CM_NORMAL_SPEED, 10000);
  i2cm_read(FROM_ADDR, location,eprom_buffer,EPROM_SIZE);
  /* Now keep looping */
  while (1);
  return 0;
```

Figure 2.5 – New Project Code

Some code has been written which utilizes the `i2cm_read` API function as shown in Figure 2.5.
2.2 Copy Source

In order to change the API source code, copy the `i2cm.c` file into the project as shown in Figure 2.6. The API source files can be found here:

C:\Users\Username\Documents\Bridgetek\FT9xx\version\Source\src

The file can be copied at windows explorer level then right-click on the project in Eclipse and select Refresh to see the added file. They can also be copied from Windows explorer and pasted into Eclipse where no refresh is required within Eclipse.

![Figure 2.6 – Insert API source](image)

2.3 Edit the Code

The Project code and API code can now be edited. Figure 2.7 shows some new I²C read code for 16-bit addressing.

Note that the function structure has changed from:

```c
int8_t i2cm_read(const uint8_t addr, const uint8_t cmd, uint8_t *data, uint16_t number_to_read)
```

to

```c
int8_t i2cm_read(const uint8_t addr, const uint16_t reg, uint8_t *data)
```

The `ft900_i2cm.h` header file at the following location needs to be edited to reflect the function attribute change:

C:\Program Files (x86)\Bridgetek\FT9xx Toolchain\Toolchain\hardware\include

**Note**: Administrator privileges are required to edit this file as it resides in the Program Files directory.

**Note**: If the function structure hasn’t changed there is no need to edit the header file.

The project main code should also be updated to reflect this change:

```c
#define I2C_ADDR (0x78)
#define EEPROM_SIZE (1024/8)
uint8_t eeprom_buffer[EEPROM_SIZE] = {0};
const uint8_t addr = I2C_ADDR;
```
i2cm_read(addr, location, eeprom_buffer);

```c
int8_t i2cm_read(const uint8_t addr, const uint16_t reg, uint8_t *data)
{
    int8_t ret = 0;
    uint8_t reg1, regh;

    reg1 = reg & 0xff;
    regh = reg >> 8;

    do
    {
        /**< Write slave address to SA */
        I2CM->I2CM_SLAVE_ADDR = addr;

        /**< Write register high address to BUF */
        I2CM->I2CM_DATA = regh;

        /**< Write command to I2CMCR to send Start and command byte. */
        I2CM->I2CM_CNTL_STATUS = I2C_FLAGS_START | I2C_FLAGS_RUN;
        if(i2c_wait_for())
        { /* An Error Occurred */
            ret = -1;
            break;
        }

        /**< Write register low address to BUF */
        I2CM->I2CM_DATA = reg1;
        I2CM->I2CM_CNTL_STATUS = I2C_FLAGS_RUN | I2C_FLAGS_STOP;
        if(i2c_wait_for())
        { /* An Error Occurred */
            ret = -2;
            break;
        }

        delays1();

        /**< Write slave address to SA with R operation */
        I2CM->I2CM_SLAVE_ADDR = addr | 0x81;

        /**< Receive with a NACK */
        I2CM->I2CM_CNTL_STATUS = I2C_FLAGS_START | I2C_FLAGS_STOP | I2C_FLAGS_RUN;
        if(i2c_wait_for())
        { /* An Error Occurred */
            ret = -4;
            break;
        }

        /**< Read data from device. */
        *data = I2CM->I2CM_DATA;
    } while (0);

    return ret;
}
```

Figure 2.7 – New API Code
2.4 Rebuild

The project can now be rebuilt. This can be done by right-clicking on the project and selecting build, or by clicking on the build icon.

The status is shown in the console window. See Figure 2.8 for a successful build.

![Figure 2.8 - Console Output Success]

Any warnings or errors will be shown in the console windows. Figure 2.9 shows an error case when the header file is not updated to reflect a function type change.

![Figure 2.9 - Console Output Error]
3 Conclusion

The FT9xx API functions can easily be edited to suit your needs, if the default functionality does not suit. Users can also reference this code to create their own functions.
4 Contact Information

Headquarters – 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 1330
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 Troi,
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 (BRTChip) 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 Product Page
FT93x Product Page
AN_324 FT900 User Manual
FT90x Toolchain

Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Terms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>I²C</td>
<td>Inter-Integrated Circuit</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated Circuit</td>
</tr>
</tbody>
</table>
Appendix B – List of Tables & Figures

List of Tables
NA

List of Figures
<table>
<thead>
<tr>
<th>Figure 1.1 – FT9xx Software Stack</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1 – New C Project</td>
<td>3</td>
</tr>
<tr>
<td>Figure 2.2 – New Project Type</td>
<td>4</td>
</tr>
<tr>
<td>Figure 2.3 – New Project Configurations</td>
<td>4</td>
</tr>
<tr>
<td>Figure 2.4 – New Project Structure</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2.5 – New Project Code</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2.6 – Insert API source</td>
<td>6</td>
</tr>
<tr>
<td>Figure 2.7 – New API Code</td>
<td>7</td>
</tr>
<tr>
<td>Figure 2.8 – Console Output Success</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.9 – Console Output Error</td>
<td>8</td>
</tr>
</tbody>
</table>
Appendix C – Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Changes</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Initial Release</td>
<td>2017-07-13</td>
</tr>
</tbody>
</table>