



Application Note

AN_349

FT51A FT800 Spaced Invaders Sample

Version 1.1

Issue Date: 2015-11-26

This document provides a guide for using the FT51A development environment with an attached FT800 to play the Spaced Invaders game.

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

| | |
|---|-----------|
| 1 Introduction | 3 |
| 1.1 Overview..... | 3 |
| 1.2 Features..... | 3 |
| 1.3 Limitations | 3 |
| 1.4 Scope | 3 |
| 2 Firmware Overview | 4 |
| 2.1 FT800 Display..... | 4 |
| 2.2 FT51A Libraries | 4 |
| 3 FT800 Spaced Invaders Firmware | 5 |
| 3.1 Timer | 5 |
| 3.2 Writing to FT800 | 6 |
| 3.3 How the Game Works | 6 |
| 4 Possible Improvements | 7 |
| 5 Contact Information | 8 |
| Appendix A – References | 9 |
| Document References | 9 |
| Acronyms and Abbreviations | 9 |
| Appendix B – FT800 and Joystick Schematic..... | 10 |
| Appendix C – Revision History..... | 11 |

1 Introduction

This application note documents an example firmware project for the FT51A. The source code is available in the "examples\AN_349_FT51A_FT800_Spaced_Invaders_Sample" folder of the FT51A Software Development Kit.

1.1 Overview

The application implements a version of the popular computer game 'Space Invaders'. It will render the game graphics on an FT800 display module. Input to control the game is via GPIO signals.

The hardware modules required to use the application note are a VM800B module (FT800 and display); either a UMFT51-AA or FT51A EVM (FT51A module); A joystick (or simply a left and right button) and a fire button.

The FT800 on the VM800B module is connected to the SPI Master interface of the FT51A on pins DIO_2 (Slave Select 1), DIO_3 (SPI MOSI), DIO_4 (SPI MISO) and DIO_5 (SPI SCLK).

The control signals are DIO_0 (move Left), DIO_7 (move right) and AIO_8 (Start and Fire). A schematic for a connection board between a VM800B and a UMFT51-AA is included in the Appendix. On an FT51A EVM the CN7 connector is used to interface with the VM800B module.

1.2 Features

The FT800 Spaced Invaders example has the following features:

- Open source firmware.
- Reads input from a controller using GPIO.
- Displays graphics output on an FT800 enabled display via an SPI Master Interface.

1.3 Limitations

None specified

1.4 Scope

The guide is intended for developers who are creating applications, extending FTDI provided applications or implementing example applications for the FT51A.

In the reference of the FT51A, an "application" refers to firmware that runs on the FT51A; "libraries" are source code provided by FTDI to help user, access specific hardware features of the chip.

The FT51A Tools are currently only available for Microsoft Windows platform and are tested on Windows 7 and Windows 8.1.

2 Firmware Overview

This section provides details on the libraries used to create the example project.

2.1 FT800 Display

The VM800B module interface for displaying the graphics and text is configured over the SPI Master interface. Commands and data to send are stored in a buffer in RAM and sent in batches to the FT800 device on the VM800B module regular intervals.

The firmware includes an FT800 abstraction layer providing functions which are used to perform actions on the FT800. This layer is in the `ft_gpu` directory of the sample source code. The definition of `FT51A_PLATFORM` and `FT51A_PLATFORM_SPI` in the file `FT_Platform.h` are used to enable the FT51A abstraction in the libraries and use SPI for the communications. The same library can be used for other FTDI microcontrollers.

2.2 FT51A Libraries

The firmware uses the SPI Master library, general config library and the IOMUX library. The IOMUX library is used in the example code to set the output characteristics of the SPI Master interface

The firmware is designed for the FT51A EVM module and may be extended to use the LCD for displaying some information. If so, then it will need the I²C Master library added.

3 FT800 Spaced Invaders Firmware

The firmware included in the example code demonstrates displaying graphics on an FT800 display. All control logic and screen drawing triggers are initiated by timers in the `SpacedInvaders` function in `spaced_invaders.c`. This function contains a while loop that does not exit.

3.1 Timer

A timer is used to provide delays and time measurements for implementing polling intervals.

The first timer `ms_timer` is used to create general purpose delays, for instance implementing a delay function.

There is an array called `updateTimer` which contains several timers which are refreshed each millisecond. These are used for timing the movement of the graphics elements on the screen, i.e. the invaders, bombs, saucers, blasters.

```
void ms_timer_interrupt(const uint8_t flags)
{
    ft_uint8_t timer;
    (void) flags; // Flags not currently used
    if (ms_timer)
    {
        ms_timer--;
    }
    // refresh all update timers ...
    for (timer = 0; timer < REFRESH_TOTAL; timer++)
    {
        if (updateTimer[timer])
        {
            updateTimer[timer]--;
        }
    }
    // Reload the timer
    TH0 = MSB(MICROSECONDS_TO_TIMER_TICKS(1000));
    TL0 = LSB(MICROSECONDS_TO_TIMER_TICKS(1000));
}

void ms_timer_initialise(void)
{
    // Register our own handler for interrupts from Timer 0
    interrupts_register(ms_timer_interrupt, interrupts_timer0);

    // Timer0 is controlled by TMOD bits 0 to 3, and TCON bits 4 to 5.
    TMOD &= 0xF0; // Clear Timer0 bits
    TMOD |= 0x01; // Put Timer0 in mode 1 (16 bit)
    // Set the count-up value so that it rolls over to 0 after 1 millisecond.
    TH0 = MSB(MICROSECONDS_TO_TIMER_TICKS(1000));
    TL0 = LSB(MICROSECONDS_TO_TIMER_TICKS(1000));
    TCON &= 0xCF; // Clear Timer0's Overflow and Run flags
    TCON |= 0x10; // Start Timer0 (set its Run flag)
}
```

3.2 Writing to FT800

Commands to the FT800 can either be sent discretely to registers or to internal RAM in batches when the display is being constantly updated by the firmware.

Both methods will enable the SPI Master to talk to the FT800 with the Slave Select 1 (SS1) line going high before sending commands. The next stage is to send an address of the register or area of RAM to be addressed in the FT800. The data for the register or RAM follows before SS1 is de-asserted to signal the end of the transfer.

A discrete register change is performed using the `FT800_Write08` or `FT800_Write16` functions in the file `ft800_demo_board.c`. These functions call the hardware abstraction layer functions to turn on slave select and send the register commands before turning off slave select.

Batches of commands will be stored in a RAM buffer on the FT51A called `Ft_CmdBuffer` (the size and therefore the end offset of this buffer are held in the variable `Ft_CmdBuffer_Index`). Helper functions `FT800_CmdBuffer_AddCmd` and `FT800_CmdBuffer_AddStr` functions are available to abstract the operation of the buffer.

When the batch is complete the commands are sent to the FT800 RAM using the `FT800_CmdBuffer_Send` function. The size of `Ft_CmdBuffer` is 4 kB, allowing for a maximum display list of 1024 commands (32 bits per command).

Instructions for an entire screen update can be stored in the buffer before being transmitted to the FT800 at a fixed interval. The update is triggered when the `timer_display` value reaches zero at this point all current sensor readings are formatted and graphics created for the screen update.

This document does not deal with the FT800 commands used to render the graphics on the display.

3.3 How the Game Works

The game elements are processed according to a rota each time the timer controlling the element expires. This drives a state machine for each element determining how far it moves, what interaction it has with other elements and what action to take.

Button presses are detected on the GPIO lines and move the player's blaster accordingly.

4 Possible Improvements

The following improvements can be made in order to more closely match the original arcade version of Space Invaders ...

- Award an extra life at 1500 points.
- Change the speed of the last invader depending on direction (arcade version moved 2 pixels at a time when moving to the left and 3 pixels at a time when moving to the right).
- Do not display saucer if there are eight or less invaders left alive.
- Add sound for shots being fired, explosions, the saucer and finally the sound of the invaders marching.
- Add animation on the start screen (there are videos available of the original arcade Space Invaders).

5 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
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) 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
E-Mail (Support) us.support@ftdichip.com
E-Mail (General Enquiries) 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
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) 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
E-mail (Support) cn.support@ftdichip.com
E-mail (General Enquiries) 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

FTDI MCU web page: <http://www.ftdichip.com/MCU.html>

USB Test and Measurement Class specification:
http://www.usb.org/developers/docs/devclass_docs/USBTMC_1_006a.zip

IVI Foundation: <http://www.ivifoundation.org/>

SCPI specification: <http://www.ivifoundation.org/docs/scpi-99.pdf>

USB Device Firmware Update Class specification:
http://www.usb.org/developers/docs/devclass_docs/DFU_1.1.pdf

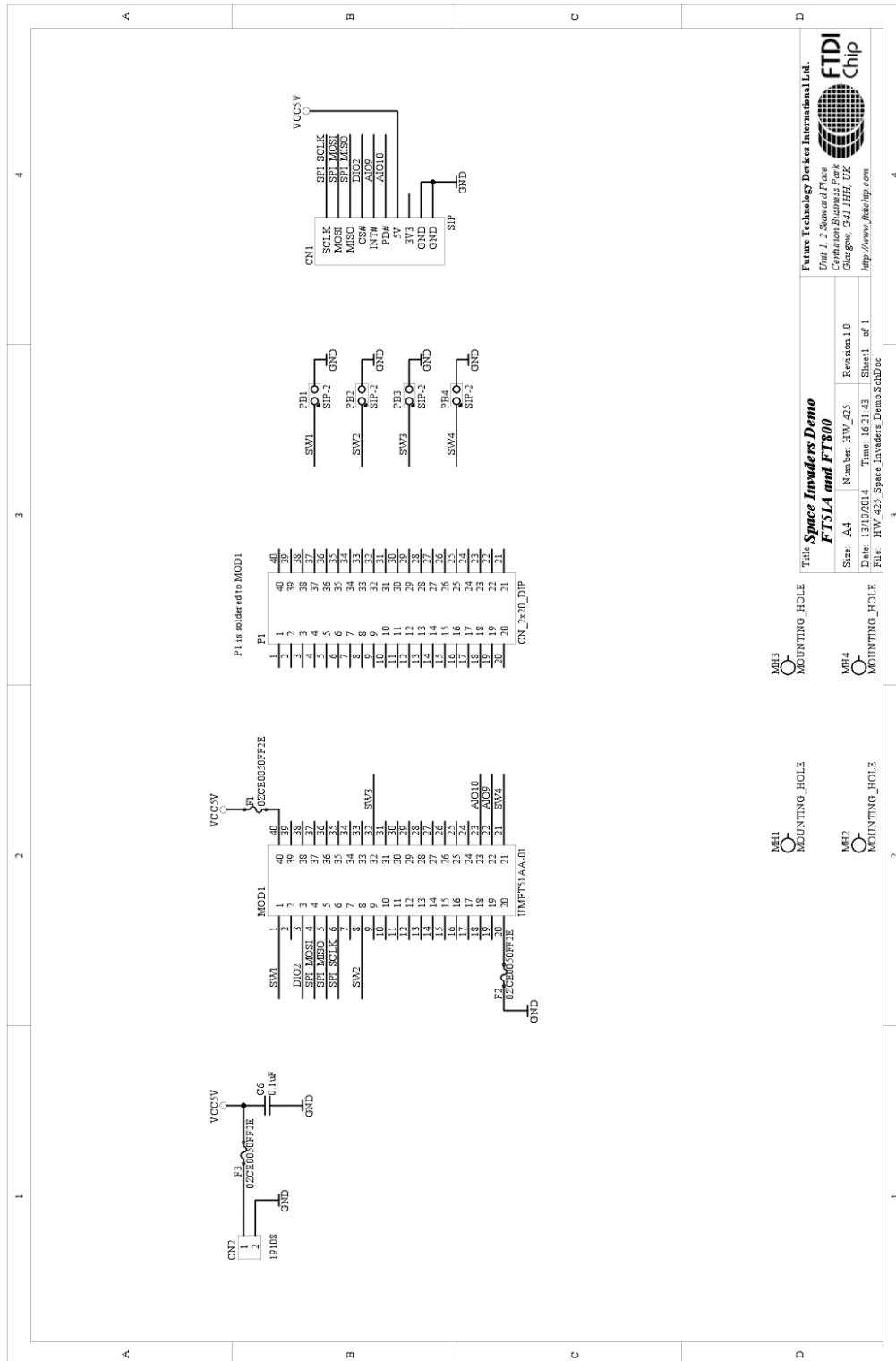
[FT800 datasheet](#)

[FT800 Series Programmers Guide](#)

Acronyms and Abbreviations

| Terms | Description |
|--------|--|
| HID | Human Interface Device |
| MTP | Multiple Time Program – non-volatile memory used to store program code on the FT51A. |
| SPI | Serial Peripheral Interface |
| USB | Universal Serial Bus |
| USB-IF | USB Implementers Forum |

Appendix B – FT800 and Joystick Schematic



Appendix C – Revision History

Document Title: AN_349 FT51A FT800 Spaced Invaders Sample
Document Reference No.: FT_001128
Clearance No.: FTDI# 430
Product Page: <http://www.ftdichip.com/FTPProducts.htm>
Document Feedback: [Send Feedback](#)

| Revision | Changes | Date |
|----------|---------------------------------|------------|
| 1.0 | Initial Release | 2014-12-12 |
| 1.1 | Update FT51 references to FT51A | 2015-11-26 |