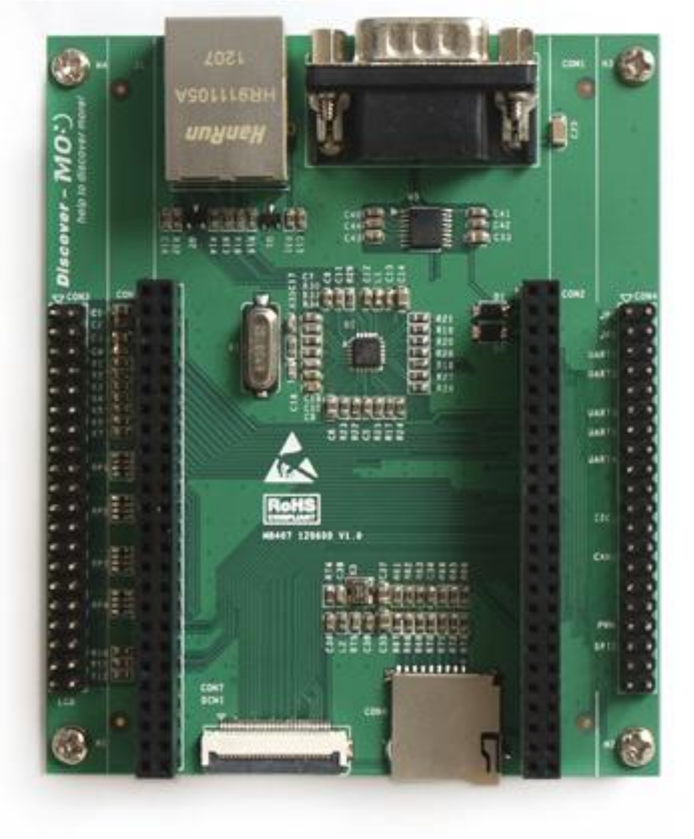
STM32F4DIS-BB User Manual

Discover-more

help to discover more!

STM32F4DIS-BB Base Board

- Base Board for STM32F4DISCOVERY High-Performance Discovery Board
- Extended peripherals including UART, Ethernet, CAN, Camera, LCD, TF, SPI, I2C
- Supports optional 3.5'' LCD module and 1.3 Megapixel digital camera module
- Supports FatFs_vR0.08a File System (Used for TF card)
- Supports LwIP_v1.3.2 Protocol Stack
- Supports Micrium µC/OS-II_v2.91



Declaration

element14/Embest and ST have launched the STM32F4DIS-BB, a low cost development platform based on STM32F4DISCOVERY.The platform also supports two modules STM32F4DIS-LCD, STM32F4DIS-CAM.

Glossary

Term	**Meaning**
STM32F4DIS-BB, DM-STF4BB	the Base Board for the STM32F4DISCOVERY
STM32F4DIS-LCD,DM-LCD35RT	the LCD module for the STM32F4DIS-BB
STM32F4DIS-CAM,DM-CAM130	the Camera module for the STM32F4DIS-BB
Devkit407	STM32F4DIS-BB and STM32F4DISCOVERY

STM32F4DIS-BB,STM32F4DIS-LCD,STM32F4DIS-CAM are the ST part numbers.

DM-STF4BB,DM-LCD35RT,DM-CAM130 are the Embest part numbers.

COPYRIGHT

- ,Discover-more,Devkit407,DM-STF4BB,DM-LCD35RT,DM-CAM130,are trademarks of Embest Technology Co., LTD.

- ,STM32F407,STM32F4DISCOVERY,STM32F4DIS-BB,STM32F4DIS-LCD,STM32F4DIS-CAM are trademarks of STMicroelectronics.

- is trademark of Element14.

- Microsoft, MS-DOS, Windows XP are trademarks of Microsoft Corporation.

Important Notice

Embest has the ownership and rights to the use of this document. Information in the document is within the protection of copyright. Unless specifically allowed, no part of this document should be modified, issued or copied in any manner or form without prior written approval of Embest Technology Co., LTD.

Version of update records:

Rev	Date	Description
V1.0	28-Dec-2012	Initial version

Contact:

If you want to order products from Embest, please contact the Marketing Department:

Tel: +86-755-25635656 / 25636285

Fax: +86-755-25616057

E-mail: market@embedinfo.com

If you need technical assistance from Embest, please contact the Technical Assistance Department:

Tel: +86-755-25503401

E-mail: support@embedinfo.com

URL: http://www.armkits.com

Address: Room 509, Luohu Science &Technology Building, #85 Taining Road, Shenzhen, Guangdong, China (518020)

Contents

STM32F4DIS-BB User Manual 1

Chapter 1 Overview 10

1.1 Product Introduction 10

1.2 How to Form DevKit407 11

1.3 Interface of STM32F4DIS-BB 11

1.4 Diagram of DevKit407 12

1.5 Hardware Features..13

1.6 ESD Precautions and Proper Handling Procedures 15

1.7 Software Features 15

Chapter 2 Hardware System 17

2.1 CPU 17

2.2 Hardware Interface 18

2.2.1 Serial Ports..18

2.2.2 Ethernet Interface...18

2.2.3 MicroSD Card Interface 19

2.2.4 TFT_LCD Interface 20

2.2.5 I/O Interface...21

2.2.6 Camera Interface 23

2.3 Pin Multiplexing 24

2.4 Hardware Dimensions 27

Chapter 3 Development Environment 28

3.1 Hardware Environment..28

3.2 Software Environment 28

3.3 PC Environment 29

3.3.1 HyperTerminal Connection...29

3.3.2 PC Network Settings 32

Chapter 4 Peripherals Examples 36

4.1 USART Example 36

4.2 SDIO Example....37

4.2.1 FatFs 37

4.2.2 uSDCard....39

4.3 LCD Example 40

4.3.1 LCD_35T 40

4.3.2 LCD_Touch....41

4.4 DCMI Example 42

4.5 ETH_LwIP Example 44

4.5.1 Standalone\httpserver 45

4.5.2 Standalone\tcp_echo_client 48

4.5.3 Standalone\tcp_echo_server 51

4.5.4 Standalone\udp_echo_client....54

4.5.5 Standalone\udp_echo_server 57

4.5.6 FreeRTOS\httpserver_netconn....60

4.5.7 FreeRTOS\httpserver_socket 63

4.5.8 FreeRTOS\udptcp_echo_server_netconn 63

4.6 USB Example 64

4.6.1 USB_Device_Examples\DFU 65

4.6.2 USB_Device_Examples\MSC....71

4.6.3 USB_Device_Examples\VCP 73

4.6.4 USB_Host_Examples\HID 78

4.6.5 USB_Host_Examples\MSC 81

Chapter 5 Applications Examples....85

5.1 STM32F4xx_uCOSII_Example 85

5.2 STM32F4xx_FPU_FFT_Example....86

Chapter 6 Other Test Scenarios 87

6.1 USART Testing 87

6.2 SDIO Testing 87

6.3 LCD Testing 87

6.4 DCMI Testing 87

6.5 Ethernet Testing..87

6.6 USB Testing 87

Technical Support & Warranty Service 88

Technical support service 88

Maintenance Service Clause...89

Basic Notice for Protecting LCD Screen 90

Value Added Services...90

Chapter 1 Overview

1.1 Product Introduction

The STM32F4DIS-BB Base Board from Embest is an expansion board designed especially for STMicroelectronics' STM32F4DISCOVERY High-Performance Discovery Board which is based on the STM32F407VGT6 ARM Cortex-M4 microcontroller and includes an ST-LINK/V2 embedded debug tool, two ST MEMS sensors, digital accelerometer and digital microphone, one audio DAC with integrated class D speaker driver, LEDs, push buttons and an USB OTG micro-AB connector.

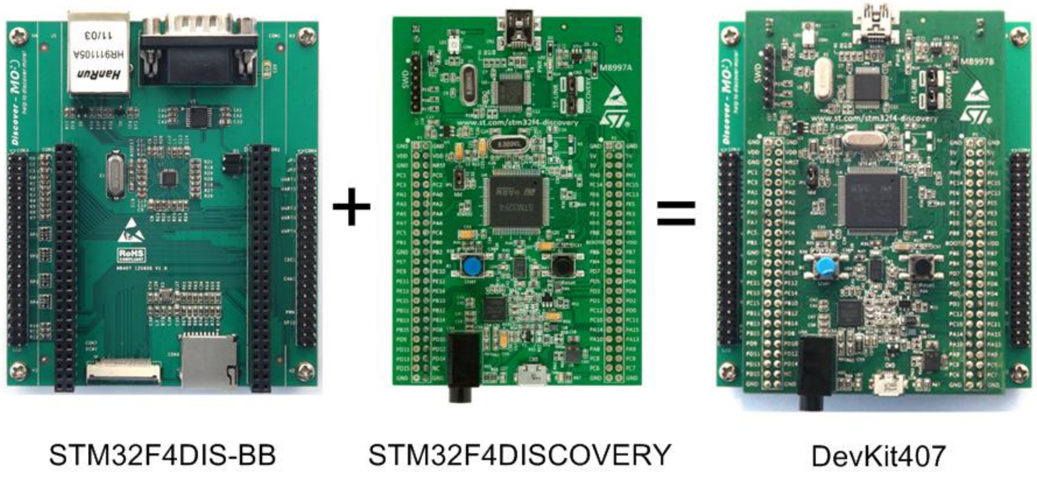
The STM32F4DIS-BB Base Board enables STM32F4DISCOVERY board users to discover more features of the STM32F4. It brings additional functionality to the STF32F4DISCOVERY, including serial ports, USB, Ethernet, CAN, SPI, I2C, GPIO, Camera, TF Card, LCD and touch screen interfaces on board.

The combination of the STM32F4DIS-BB and the STM32F4DISCOVERY forms the evaluation board Devkit407 for evaluating the STM32F4xx series ARM Cortex-M4 microcontrollers and allows easy prototyping of third party solutions with STM32F4DISCOVERY board or STM32F4xx series microcontrollers.

Embest has ported Micrium µC/OS-II to the DevKit407 board and the software also features LwIP_v1.3.2 protocol support. Embest also offers rich software examples for DevKit407 to complement those provided by the original STM32F4DISCOVERY kit. Included in the package is a CD containing the µC/OS-II BSP and plenty of software examples, board schematic and user manual to help you better understand this board and develop your own applications.

To expand the performance of the STM32F4DISCOVERY board, Embest also offers the STM32F4DIS-CAM 1.3 Mega pixel camera and STM32F4DIS-LCD 3.5" LCD modules which interface easily with the STM32F4-BB base board.

1.2 How to Form DevKit407



Mount your STM32F4DISCOVERY board to your STM32F4DIS-BB board through CON1 and CON2 to form DevKit407.

1.3 Interface of STM32F4DIS-BB

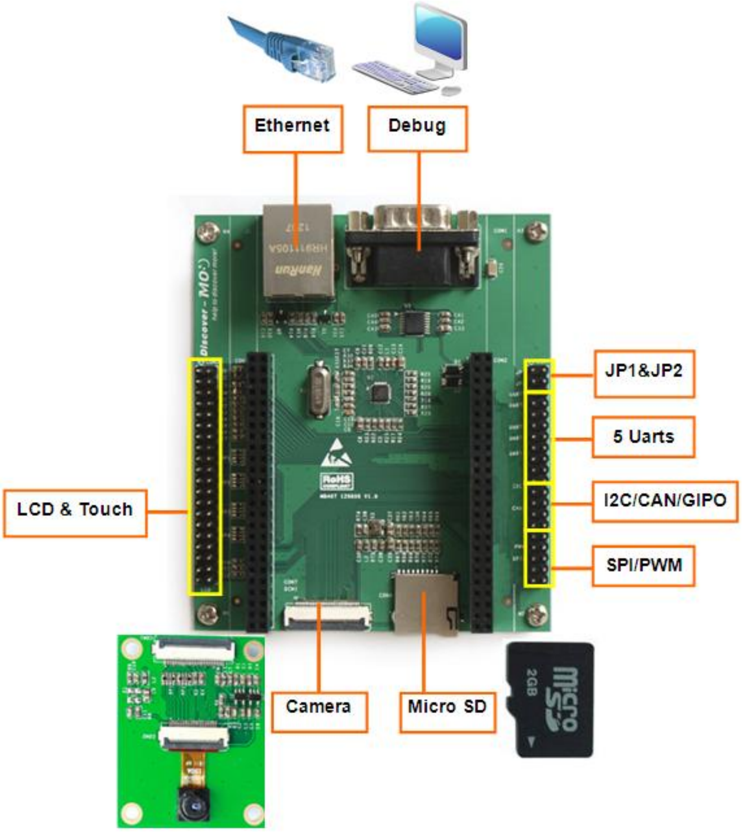
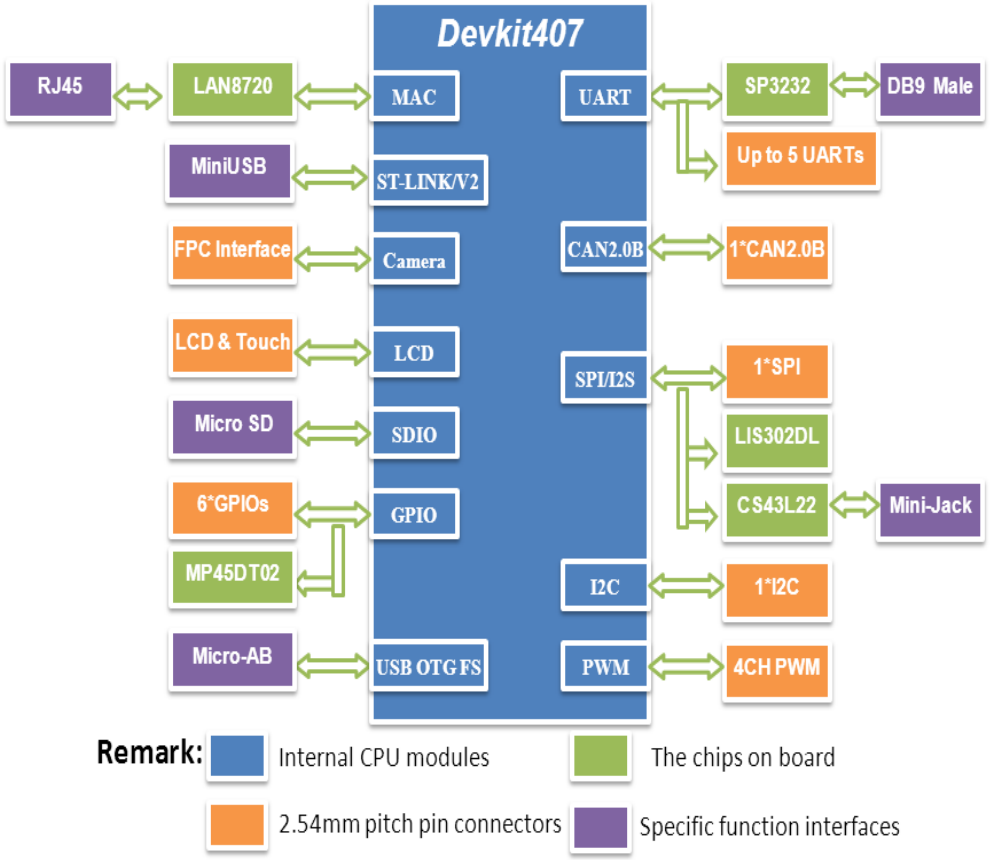


Figure 1-1 Interface of STM32F4DIS-BB

1.4 Diagram of DevKit407



1.5 Hardware Features

Processor

- STMicroelectronics STM32F407VGT6 Flash Microcontroller
  - ARM 32-bit Cortex-M4 CPU with ART accelerator, frequency up to 168 MHz
  - On chip 1MB of Flash memory and 192+4KB of SRAM
  - Flexible static memory controller that supports Compact Flash, SRAM, PSRAM, NOR and NAND memories
  - LCD parallel interface, 8080/6800 modes
  - USB 2.0 High-Speed/Full-Speed Device/Host/OTG
  - 10/100 Ethernet MAC, supports IEEE 1588v2 hardware, MII/RMII
  - 2 CAN 2.0B interfaces: up to 4 USARTs and 2 UARTs, 3 SPI (30Mbit/s), 2 with mux'ed I2S
  - 8- to 14-bit parallel camera interface (up to 48MB/s)
  - 1-/4-/8-bit SD/MMC/SDIO interface, supports up to 32GB storage
  - Up to 140 I/O ports up to 84 MHz
  - Up to 17 timers (two 32-bit timers), up to 168 MHz
  - 3 x 12-bit A/D converters, 2 x 12-bit D/A converters
  - Analog true random number generator
  - Low power supports Sleep, Stop and Standby modes
  - Supports booting from Flash, System memory or SRAM
  - Supports ISP and IAP programming

External Memory

- Micro SD card slot

MEMS accelerometer

- LIS302DL, ST MEMS motion sensor, 3-axis digital output accelerometer

Audio interfaces

- I2S Audio DAC, stereo audio jack for headset

- MP45DT02, ST MEMS audio sensor, omnidirectional digital microphone

Camera Interface

- 1 x camera interface

LCD/Touch Screen

- 3.5 inch TFT color LCD (240 x 320-pixel RGB resolution, 262000 colors, 16-bit 8080 parallel interface, brightness control via PWM)
- 4-wire resistive touch screen

Data Transfer Interfaces

- 1 x 5-wire RS232 Serial Port
- 1 x USB2.0 OTG/Device/Host, Full-speed, up to12Mbit/s
- 1 x 10/100 Ethernet with IEE 1588v2 (RJ45 connector)

Programming and Debugging Interface

- On-board ST-LINK/V2 with selection mode switch to use the kit as a standalone ST-LINK/V2 (with SWD connector for programming and debugging)

I/O Interface

- 5 x 3- wire TTL UART
- 1 x 4 channel PWM
- 1 x SPI, I2C, CAN
- 6 GPIOs

All the I/Os are extended by extension connectors.

Board Power Supply

- Through USB bus or from an external 5 V supply voltage
- External application power supply: 3 V and 5 V

Mechanical Parameters

- Power consumption: 180 mA @ 5 V (without LCD module)
- Storage and Working Temp.: 0°C ~ 45°C
- Humidity Range: 20% ~ 90%
- CE and RoHS compliant

1.6 ESD Precautions and Proper Handling Procedures

Please note that the board comes without any case/box and all components are exposed. Therefore, extra attention must be paid to ESD (electrostatic discharge) precautions. To effectively prevent electrostatic damage, please follow the steps below:

- Avoid carpets in cool, dry areas. Leave development kits in their anti-static packaging until ready to be installed.
- Dissipate static electricity before handling any system components (development kits) by touching a grounded metal object, such as the system unit unpainted metal chassis.
- If possible, use antistatic devices, such as wrist straps and floor mats.
- Always hold an evaluation board by its edges. Avoid touching the contacts and components on the board.
- Take care when connecting or disconnecting cables. A damaged cable can cause a short in the electrical circuit.
- Prevent damage to the connectors by aligning connector pins before you connect the cable. Misaligned connector pins can cause damage to system components at power-on.
- When disconnecting a cable, always pull on the cable connector or strain-relief loop, not on the cable itself.

Warning：

This is a class A product. In a domestic this product may cause radio interference in which case the user may be required to take adequate measures.

1.7 Software Features

Features

The DevKit407 software features:

- Support for Micrium µC/OS-II_v2.91 operating system

- Supports FatFs_vR0.08a file system
- Supports LWIP _v1.3.2 protocol stack

Integrated Development Environment Support

- IAR EWARM

  All drivers support IAR EWARM, the EWARM version should be V6.40 or above.

  µC /OS-II Demos do not support IAR EWARM at present.

- Keil MDK-ARM

  All drivers and applications support Keil MDK-ARM, the MDK-ARM version should be V4.22a or above.

Debug Tools Support

- ULINK2

  Best used with Keil MDK-ARM.

- JLINK-V8

  Best used with IAR EWARM.

- ST-LINK/V2

  Can be used with either Keil MDK-ARM or IAR EWARM.

Chapter 2 Hardware System

2.1 CPU

CPU Instruction

This ARM Cortex-M4 32-bit MCU with FPU has 210 DMIPS, up to 1 MB Flash/192+4 KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm. interfaces and a camera.

Features

- 168 MHz/210 DMIPS Cortex-M4 with single cycle DSP MAC and floating point unit providing:
  - Boosted execution of control algorithms
  - More features possible for your applications
  - Ease of use
  - Better code efficiency
  - Faster time to market
  - Elimination of scaling and saturation
  - Easier support for meta-language tools
- Designed for high performance and ultra-fast data transfers: ART Accelerator, 32-bit, 7-layer AHB bus matrix with 7 masters and 8 slaves including 2 blocks of SRAM, Multi DMA controllers: 2 general purpose, 1 for USB HS, 1 for Ethernet, One SRAM block dedicated to the core, providing performance equivalent to 0-wait execution from Flash concurrent execution and data transfers and simplified resource allocation
- Outstanding power efficiency: Ultra-low dynamic power, RTC < 1 μA typical in VBAT mode, 3.6 V down to 1.7 V VDD, Voltage regulator with power scaling capability, providing extra flexibility to reduce power consumption for applications requiring both high processing and low power performance when running at low voltage or on a rechargeable battery

- Maximum integration: Up to 1 MB of on-chip Flash memory, 192 KB of SRAM, reset circuit, internal RCs, PLLs, WLCSP package available, providing more features in space constrained applications
- Superior and innovative peripherals providing new possibilities to connect and communicate high speed data at greater precision due to high resolution
- Extensive tools and software solutions providing a wide choice within the STM32 ecosystem to develop your applications.

2.2 Hardware Interface

2.2.1 Serial Ports

Table 2-1 Serial Ports Interface

COM1		
Pin	**Signal**	**Description**
1	NC	NC
2	RXD	Receive data
3	TXD	Transit data
4	NC	NC
5	GND	GND
6	DSR	Data Set Ready
7	NC	NC
8	CTS	Clear To Send
9	NC	NC

2.2.2 Ethernet Interface

Table 2-2 Ethernet Interface

J1		
Pin	**Signal**	**Description**

1	TX+	TX+ output
2	TX-	TX- output
3	RX+	RX+ input
4	CT	CT
5	CT	CT
6	RX-	RX- input
7	NC	NC
8	SHIELD	Shield
9	LED1	LINK LED
10	VDD3V3	3.3 V Power for LED
11	LED2	SPEED LED
12	VDD3V3	3.3 V Power for LED
13	4&5	Connect to shield
14	7&8	Connect to shield
15	NC	NC
16	NC	NC

2.2.3 MicroSD Card Interface

Table 2-3 MicroSD Card Interface

CON6		
Pin	**Signal**	**Description**
1	DAT2	Card data 2
2	DAT3	Card data 3
3	CMD	Command Signal
4	VDD	VDD
5	CLK	Clock
6	VSS	VSS
7	DAT0	Card data 0

8	DAT1	Card data 1
9	CD	Card detect

2.2.4 TFT_LCD Interface

Table 2-4 TFT_LCD Interface

CON3		
Pin	**Signal**	**Description**
1	VDD5	+5 V
2	VDD5	+5 V
3	GND	GND
4	GND	GND
5	VDD33	+3.3 V
6	VDD33	+3.3 V
7	LCD_PWM	LED Dimming Control by PWM Signal
8	I2C_SCL	I2C master serial clock
9	I2C_SDA	I2C serial bidirectional data
10	TC_INT	Touch screen interrupt
11	LCD_RST	LCD reset
12	LCD_cs	LCD chip select
13	GND	GND
14	GND	GND
15	GND	GND
16	D0	16-bit 8080 parallel interface, Data bit 0
17	D1	16-bit 8080 parallel interface, Data bit 1
18	D2	16-bit 8080 parallel interface, Data bit 2
19	D3	16-bit 8080 parallel interface, Data bit 3
20	D4	16-bit 8080 parallel interface, Data bit 4
21	D5	16-bit 8080 parallel interface, Data bit 5

22	GND	GND
23	D6	16-bit 8080 parallel interface, Data bit 6
24	D7	16-bit 8080 parallel interface, Data bit 7
25	GND	GND
26	D8	16-bit 8080 parallel interface, Data bit 8
27	D9	16-bit 8080 parallel interface, Data bit 9
28	D10	16-bit 8080 parallel interface, Data bit 10
29	D11	16-bit 8080 parallel interface, Data bit 11
30	D12	16-bit 8080 parallel interface, Data bit 12
31	D13	16-bit 8080 parallel interface, Data bit 13
32	D14	16-bit 8080 parallel interface, Data bit 14
33	D15	16-bit 8080 parallel interface, Data bit 15
34	GND	GND
35	GND	GND
36	GND	GND
37	LCD_DC	LCD Parallel Interface
38	LCD_RD	Read signal
39	LCD_WR	Write signal
40	GND	GND

2.2.5 I/O Interface

Table 2-5 I/O Interface

CON4		
Pin	**Signal**	**Description**
1	UART6_TXD	UART6_TXD
2	JP1	UART6_TX
3	UART6_RXD	UART6_RXD
4	JP2	UART6_RX

5	UART1_TXD	UART1_TXD
6	UART2_TXD	UART2_TXD
7	UART1_RXD	UART1_RXD
8	UART2_RXD	UART2_RXD
9	GND1	GND
10	GND2	GND
11	UART3_TXD	UART3_TXD
12	UART5_TXD	UART5_TXD
13	UART3_RXD	UART3_RXD
14	UART5_RXD	UART5_RXD
15	UART4_TXD	UART4_TXD
16	GPIO1	IO
17	UART4_RXD	UART4_RXD
18	GPIO2	IO
19	GND3	GND
20	GND4	GND
21	I2C1_SCL	I2C1_SCL
22	GPIO3	IO
23	I2C_SDA	I2C_SDA
24	GPIO4	IO
25	CAN1_RX	CAN1_RX
26	GPIO5	IO
27	CAN1_TX	CAN1_TX
28	GPIO6	IO
29	GND5	GND
30	GND6	GND
31	TIM3_CH1	TIM3_CH1
32	SPI1_NSS	SPI1_NSS

33	TIM3_CH1	TIM3_CH1
34	SPI1_SCK	SPI1_SCK
35	TIM3_CH1	TIM3_CH1
36	SPI1_MISO	SPI1_MISO
37	TIM3_CH1	TIM3_CH1
38	SPI1_MOSI	SPI1_MOSI
39	VDD5V	+5 V
40	VDD3V	+3 V

2.2.6 Camera Interface

Table 2-6 Camera Interface

CON7		
Pin	**Signal**	**Description**
1	GND1	GND
2	D0	NC
3	D1	NC
4	D2	Digital image data bit 0
5	D3	Digital image data bit 1
6	D4	Digital image data bit 2
7	D5	Digital image data bit 3
8	D6	Digital image data bit 4
9	D7	Digital image data bit 5
10	D8	Digital image data bit 6
11	D9	Digital image data bit 7
12	D10	NC
13	D11	NC
14	GND2	GND
15	PCLK	Pixel clock

16	GND3	GND
17	HS	Horizontal synchronization
18	VDD50	NC
19	VS	Vertical synchronization
20	VDD33	+3.3 V
21	XCLKA	Clock output a
22	XCLKB	NC
23	GND4	GND
24	FLD	NC
25	PWR_EN	Power Enable
26	RST	Reset the camera
27	SDA	I2C master serial clock
28	SCL	I2C serial bidirectional data
29	GND5	GND
30	VDDIO	+3.3 V

2.3 Pin Multiplexing

There is pin multiplexing between the STM32F4Discovery and STM32F4DIS-BB.

Some functions therefore cannot work simultaneously as shown below:

1) OTG_FS

OTG_FS and LCD, UART2 cannot work simultaneously.

2) Audio

Audio and LCD, Camera, UART1/4/5/6, SPI2 cannot work simultaneously.

3) LEDs

LED3~LED6 and Camera, LCD cannot work simultaneously.

4) MEMS

MEMS and Ethernet, Camera cannot work simultaneously.

5) CAN

CAN and LCD cannot work simultaneously.

6) UART1/6

UART1/6 and Camera cannot work simultaneously.

7) UART2

UART2 and Camera, LCD cannot work simultaneously.

8) UART3

UART3 and LCD cannot work simultaneously.

9) UART4/5

UART4/5 and MicroSD cannot work simultaneously.

For more detail, please refer to Table 2-7 Pin multiplexing of STM32F4Discovery and STM32F4DIS-BB and Table 2-8 Pin multiplexing of CON4 extension port and STM32F4DIS-BB.

Table 2-7 Pin multiplexing of STM32F4Discovery and STM32F4DIS-BB

Discovery	Multiplexing Function 1	GPIO	Multiplexing Function2	STM32F4DIS-BB
OTG_FS	OTG_FS_ID	PA10	PA10	GPIO2
	OTG_FS_Overcurrent	PD5	LCD_WR	LCD
			UART2_TX	UART2
Audio	Audio_SCL	PB6	DCMI_D5	Camera
			UART1_TX	UART1
	I2S3_MCK	PC7	DCMI_D1	Camera
			UART6_RX	UART6
	I2S3_SCK	PC10	SDIO_D2	MicroSD
			UART4_TX	UART4
	I2S3_SD	PC12	SDIO_CK	MicroSD
			UART5_TX	UART5
	I2S3_WS	PA4	DCMI_HSYNC	Camera
	Audio_RST	PD4	LCD_RD	LCD

	PDM_OUT	PC3	SPI2_MOSI	SPI2
LED	LED4	PD12	Camera_RST	Camera
	LED3	PD13	LCD_PWM	LCD
	LED5	PD14	LCD_D0	
	LED6	PD15	LCD_D1	
MEMS	SPI1_SCK	PA5	PA5	GPIO6
	SPI1_CS	PE3	ETH_RST	Ethernet
	SPI1_MOSI	PA7	ETH_RMII_CRS_DV	
	SPI1_MISO	PA6	DCMI_PIXCLK	Camera
	MEMS_INT1	PE0	DCMI_D2	
	MEMS_INT2	PE1	DCMI_D3	

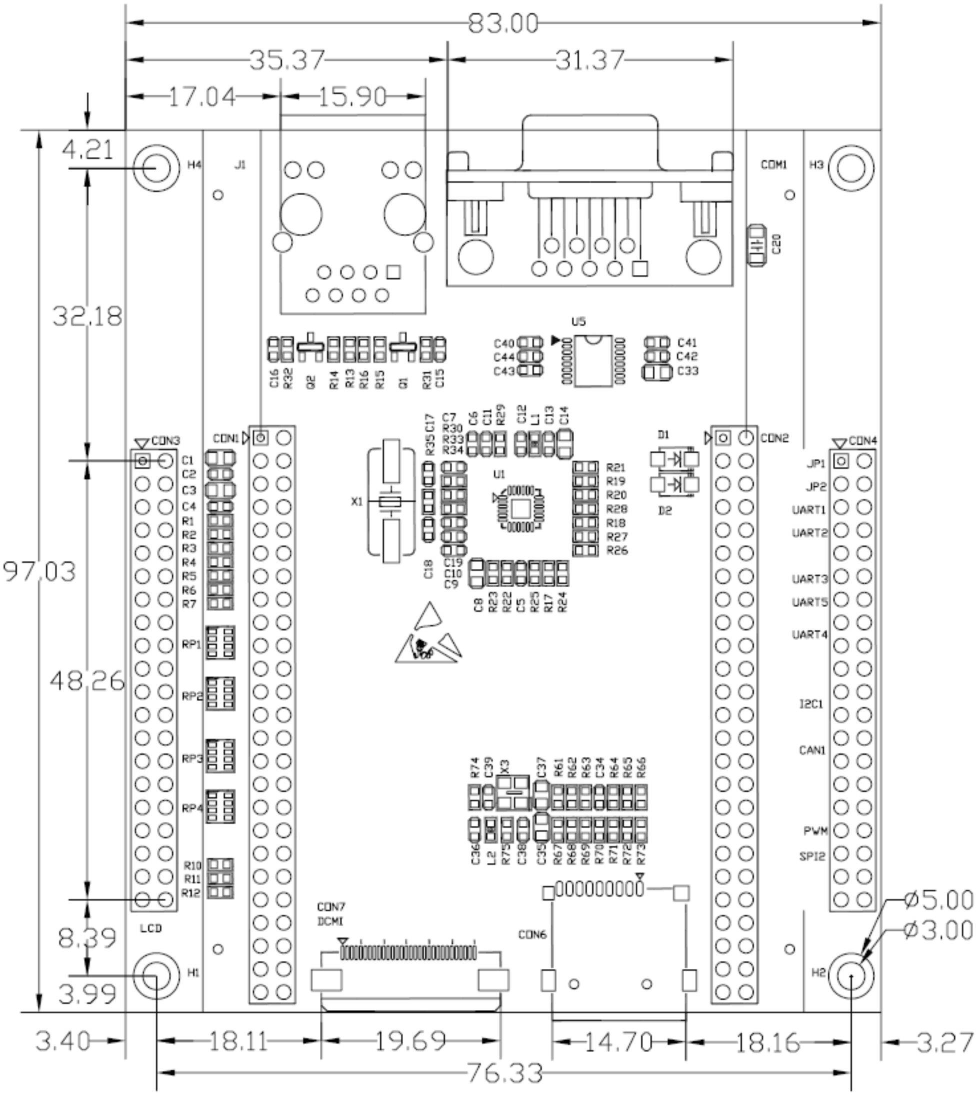
Table 2-8 Pin multiplexing of CON4 extension port and STM32F4DIS-BB

Discovery	Multiplexing Function 1	GPIO	Multiplexing Function2	STM32F4DIS-BB
CAN1	CAN1_RX	PD0	LCD_D2	LCD
	CAN1_TX	PD1	LCD_D3	
UART1	TXD1	PB6	DCMI_D5	Camera
	RXD1	PB7	DCMI_VSYNC	
UART2	TXD2	PD5	LCD WR	LCD
	RXD2	PD6	Camera_PWR_EN	Camera
UART3	TXD3	PD8	LCD_D13	LCD
	RXD3	PD9	LCD_D14	
UART4	TXD4	PC10	SDIO_D2	MicroSD
	RXD4	PC11	SDIO_D3	
UART5	TXD5	PC12	SDIO_CK	
	RXD5	PD2	SDIO_CMD	

UART6	TXD6	PC6	DCMI_D0	Camera
	RXD6	PC7	DCMI_D1	

2.4 Hardware Dimensions

The hardware dimensions of STM32F4DIS-BB (Unit: mm):



Top Layer Component Height$_{MAX}$: 13.5 mm

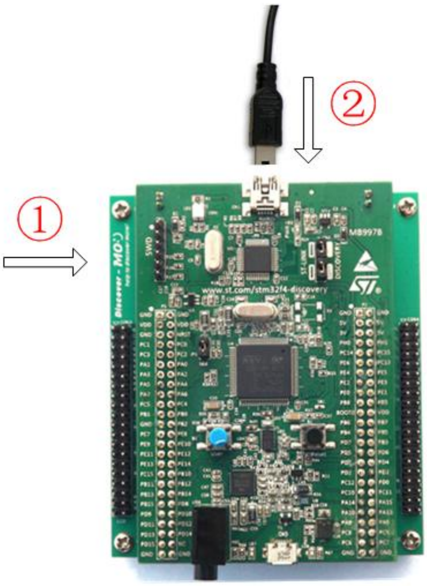
Board Thickness: 1.6 mm

Chapter 3 Development Environment

3.1 Hardware Environment

Setup DevKit407 hardware environment following these steps:

1) Mount STM32F4DISCOVERY board to STM32F4DIS-BB board through CON1 and CON2 to form DevKit407.
2) Connect the DevKit407 board to a PC with a 'USB type A (Male) to Mini-B (Male)' cable through USB connector CN1 to power the board.



3.2 Software Environment

DevKit407 supports two development environments: Keil MDK-ARM and IAR EWARM. Keil MDK-ARM requires V4.22a or above. IAR EWARM requires V6.40.2 or above. The instructions given in ***Getting Started with Software and Firmware Environments for the STM32F4DISCOVERY Kit.pdf*** describes how to use development toolchains to build, debug and run your project. Please refer to the document for more details.

3.3 PC Environment

3.3.1 HyperTerminal Connection

In order to use HyperTerminal on a PC, please follow the below process to setup Hyper Terminal connection (for example, on Windows XP OS):

1) Select *Start* -> *All Programs* -> *Accessories* -> *Communication* -> *Hyper Terminal*. Find the HyperTerminal, as shown below:

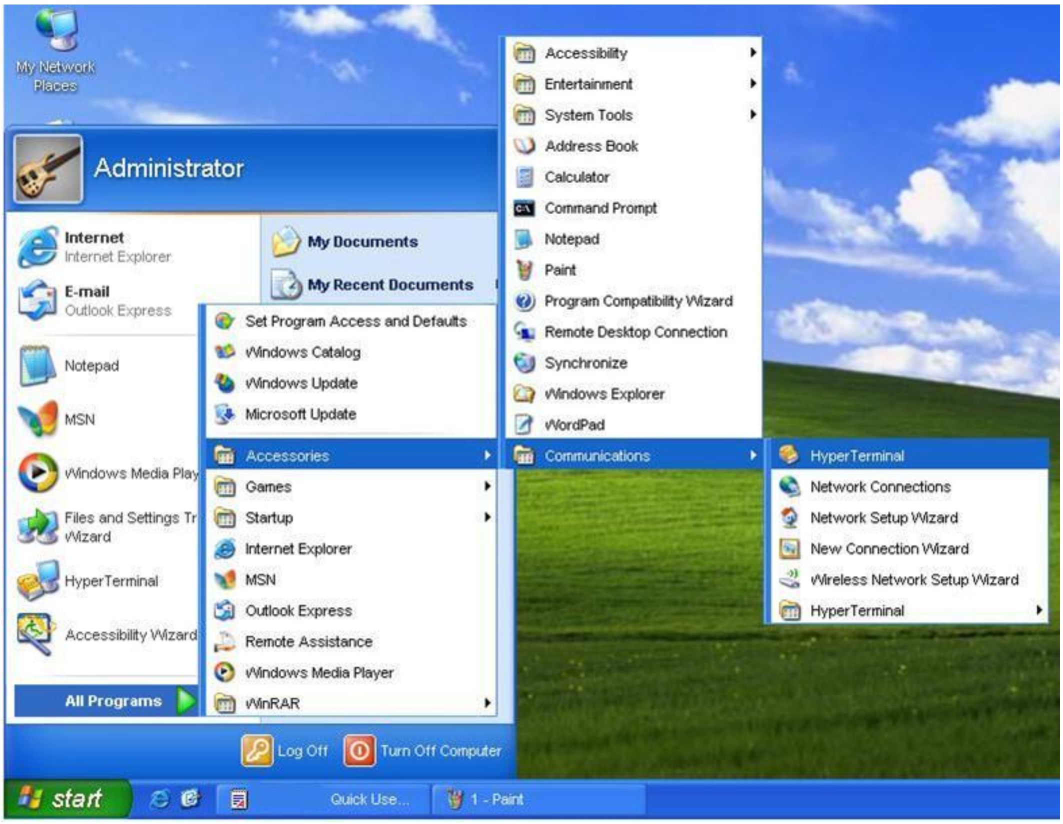
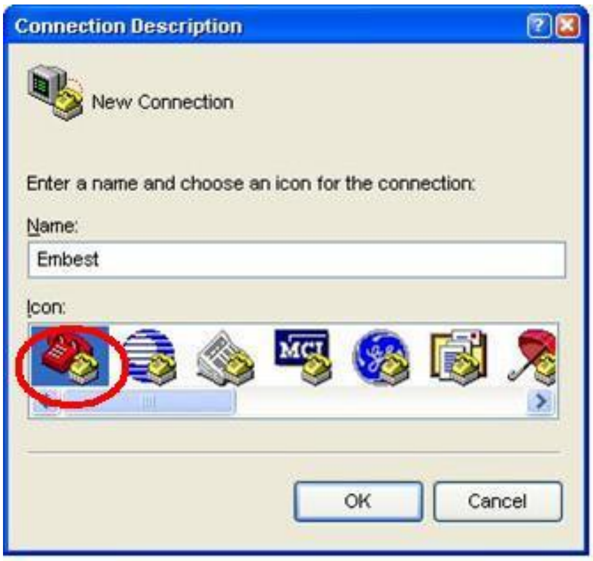


Figure 3-1 Create a new HyperTerminal

2) Create a new HyperTerminal connection, enter a name and choose an icon.



Connection Description

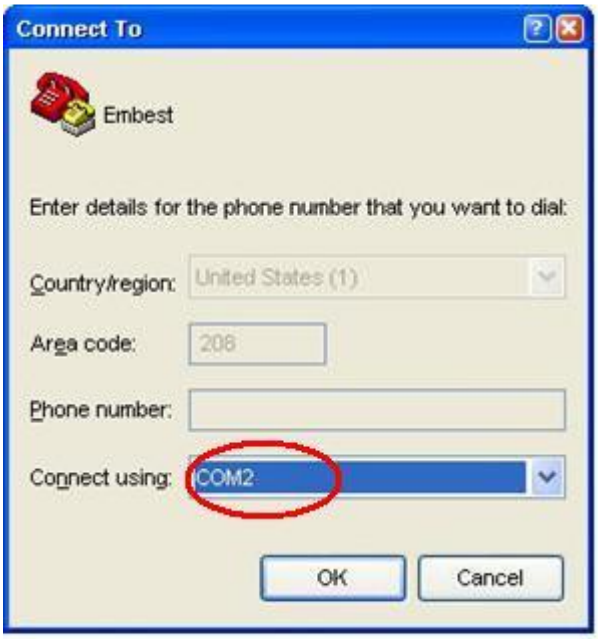
New Connection

Enter a name and choose an icon for the connection:

Name: Embest

Figure 3-2 Description for new connection

3) Select the specific serial port from the list as per your computer COM port configuration:



Connect To

Embest

Enter details for the phone number that you want to dial:

Country/region: United States (1)

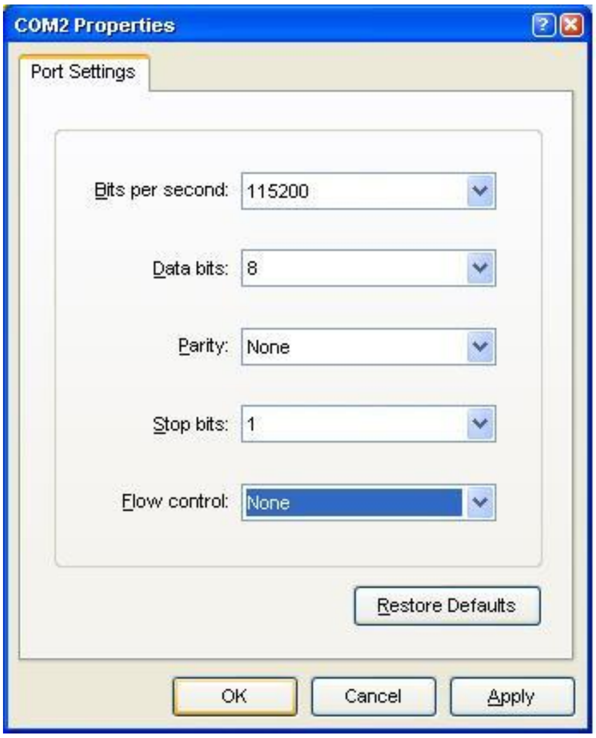
Area code: 208

Phone number:

Connect using: COM2

Figure 3-3 Select a COM port for connection

4) Set parameters for serial port connection as follows:



COM2 Properties

Port Settings

Bits per second: 115200

Data bits: 8

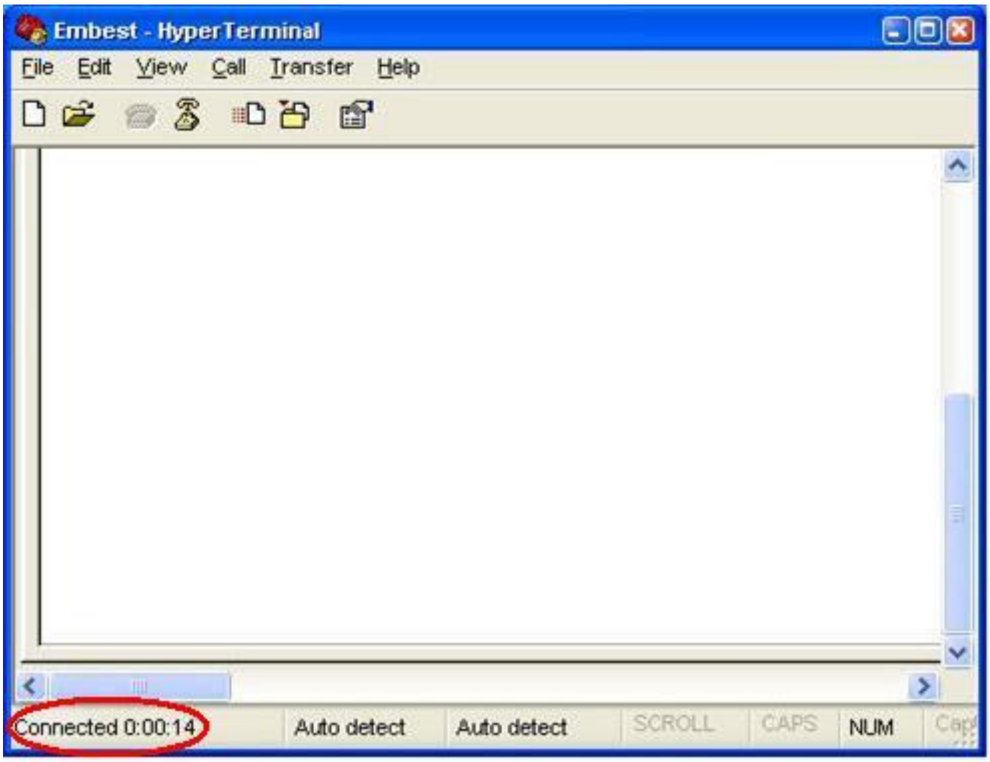
Parity: None

Stop bits: 1

Flow control: None

Figure 3-4 Settings for the selected port

5) A Hyper Terminal connection with PC serial port will have been established as shown below:



Embest - HyperTerminal

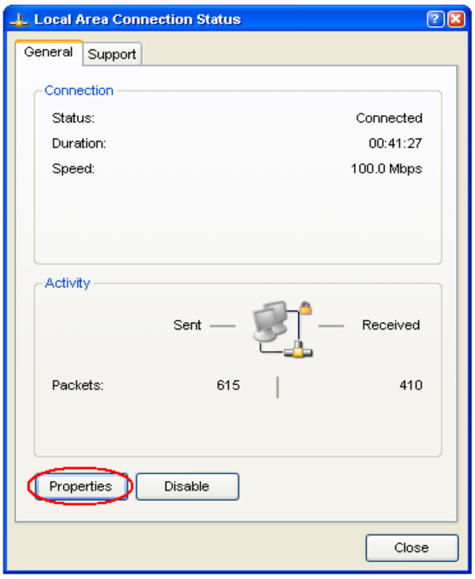
Connected 0:00:14 Auto detect Auto detect SCROLL CAPS NUM

Figure 3-5 Hyper Terminal successfully built

3.3.2 PC Network Settings

In order to test the Ethernet examples, first set up the PC network environment. Make sure that the PC's IP address and the DevKit407 board's IP address are on the same network.

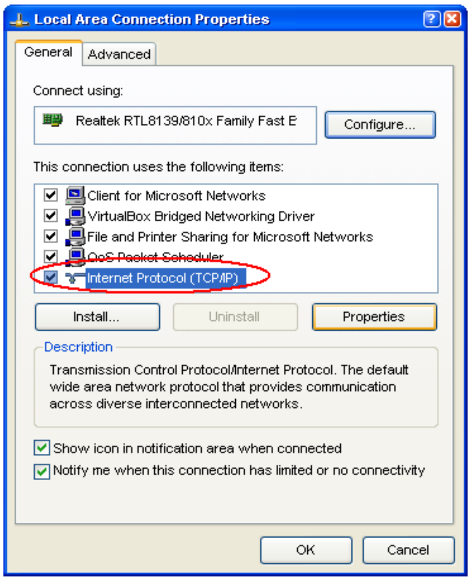
1) On PC, select *Start-> Control Panel-> Network connections-> Local Area Connection-> Properties*, as shown below:



Status: Connected
Duration: 00:41:27
Speed: 100.0 Mbps
Packets: 615 | 410

Figure 3-6 Local Area Connection

2) Click ‘Properties’, this will open the window of Local Area Connection Properties, as shown below:



Connect using:

Realtek RTL8139/810x Family Fast E

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ VirtualBox Bridged Networking Driver
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol (TCP/IP)

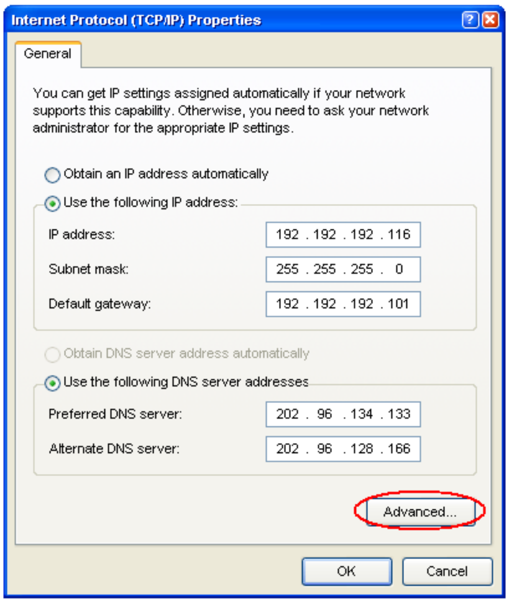
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

☑ Show icon in notification area when connected

☑ Notify me when this connection has limited or no connectivity

Figure 3-7 Local Area Connection Properties

3) Double click ‘TCP / IP Options’, opens a window for TCP / IP Properties, as shown below:



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address:	192 . 192 . 192 . 116
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	192 . 192 . 192 . 101

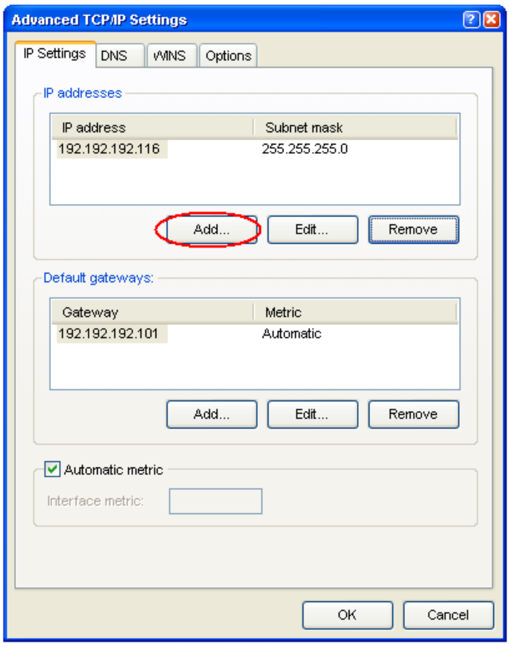
○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server:	202 . 96 . 134 . 133
Alternate DNS server:	202 . 96 . 128 . 166

Figure 3-8 TCP / IP Options

4) Click the ‘Advanced’ option, open a window for “Advanced TCP / IP settings”, as shown below:



IP address	Subnet mask
192.192.192.116	255.255.255.0

Gateway	Metric
192.192.192.101	Automatic

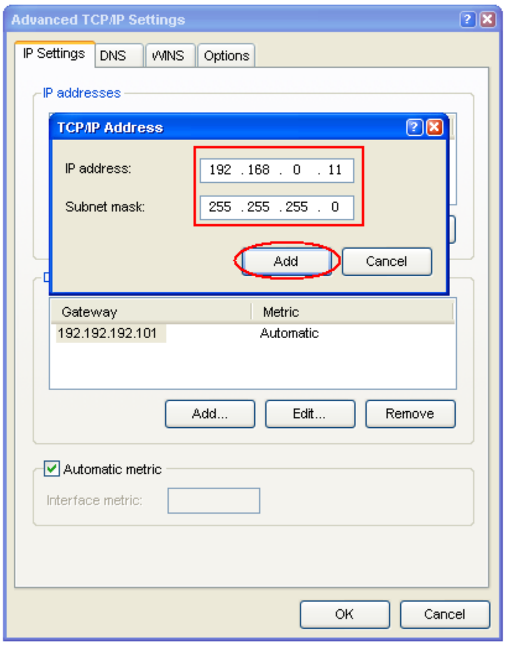
☑ Automatic metric

Interface metric:

Figure 3-9 Advanced TCP / IP settings

5) Click the ‘Add’ option, open a window for "Add TCP / IP".

Enter the IP address and subnet mask, and then click "Add”, as shown below:



IP address: 192 . 168 . 0 . 11

Subnet mask: 255 . 255 . 255 . 0

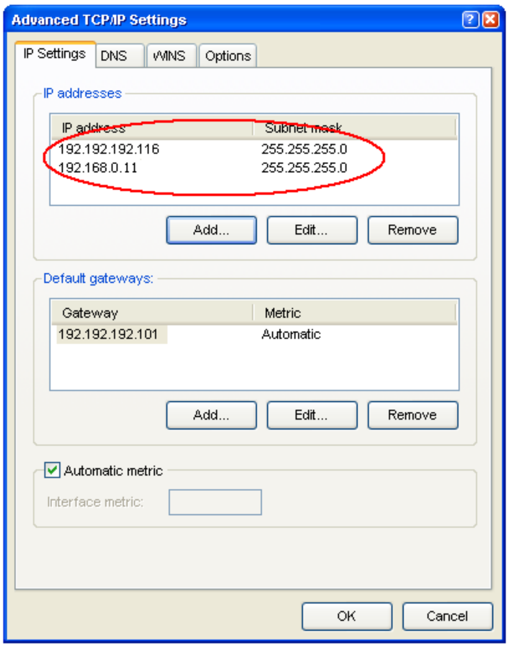
Gateway	Metric
192.192.192.101	Automatic

☑ Automatic metric

Interface metric:

Figure 3-10 Add TCP/IP address

6) Click “OK” to finish network setup.



Advanced TCP/IP Settings

IP Settings | DNS | WINS | Options

IP addresses

IP address	Subnet mask
192.192.192.116	255.255.255.0
192.168.0.11	255.255.255.0

Default gateways:

Gateway	Metric
192.192.192.101	Automatic

☑ Automatic metric

Interface metric:

Figure 3-11 TCP/IP address successfully setup

Chapter 4 Peripherals Examples

4.1 USART Example

The STM32F4xx_USART_Example folder contains one example:

- USART_Printf

USART_Printf

1. Purpose

This example shows how to retarget the C library printf function to the USART. The implementation output the printf message on the HyperTerminal using USART6.

2. Description

The example is located in the following folder:

```
\Codes\STM32F4xx_USART_Example\Project\USART_Printf
```

3. Software Configuration

HyperTerminal is used in this example. Please refer to 3.3.1 HyperTerminal Connection for more details.

4. Steps to run

1) Connect a null-modem female/female RS232 cable between the DB9 connector COM1 (USART6) and PC serial port.
2) Make sure that jumpers JP1 and JP2 are fitted.
3) Open hyperterminal on PC.
4) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.
5) Open the project, rebuild all files, load project image and then run program.
6) Testing actions and results:
   - After reset, then see the serial terminal, it should be like this:

   ```
   USART Printf Example: retarget the C library printf function to the USART
   ```

- Type a character on the keyboard, then the HyperTerminal displays the received character on the screen.

4.2 SDIO Example

The STM32F4xx_SDIO_Example folder contains two examples:

- FatFs
- uSDCard

FatFs example provides an example of how to use the SDIO firmware library and an associate driver to implement Fatfs on the SD Card memory.

uSDCard example provides a basic example of how to use the SDIO firmware library and an associate driver to perform read/write operations on the SD Card memory (SD Card V1.0, V1.1, V2.0 and SDHC (High Capacity) protocol)that could be mounted on the Devkit407 board.

4.2.1 FatFs

1. Description

The example is located in the following folder:

```
\Codes\STM32F4xx_SDIO_Example\Project\FatFs
```

2. Hardware Configuration

A MicroSD card is needed in this example and *Kingston 1GB/2GB* or *SanDisk 2GB* MicroSD Card is recommended. The card should be formatted before used.

Note: *Kingston 1GB/2GB and SanDisk 2GB MicroSD Card have been tested on DevKit407. It's not guaranteed that all kind of MicroSD card work well on the board.*

3. Software Configuration

HyperTerminal is used in this example. Please refer 3.3.1 HyperTerminal Connection to for more details.

4. Steps to Run

1) Connect a null-modem female/female RS232 cable between the DB9 connector COM1 (USART6) and PC serial port.

2) Make sure that jumpers JP1 and JP2 are fitted.

3) Open hyperterminal on PC.

4) Copy *message.txt* in the project folder to a MicroSD card.

5) Insert the MicroSD card into MicroSD slot CON6.

6) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.

7) Open the project, rebuild all files, load project image and then run program.

8) Testing actions and results:

   ➢ After reset, then see the serial terminal, it should be like this:

```
Debug Module Init
Open a test file (message.txt)
Type the file content
hello world!
Close the file
Create a new file (hello.txt)
Write a text data. (hello.txt)
14 bytes written
Close the file
read the file (hello.txt)
Type the file content(hello.txt)
Hello world!
Close the file (hello.txt)
Open root directory
Directory listing...
        14   HELLO.TXT
        12   MESSAGE.TXT
```

```
Test completed
```

4.2.2 uSDCard

1. Description

The example is located in the following folder:

```
\Codes\STM32F4xx_SDIO_Example\Project\uSDCard
```

2. Hardware Configuration

A MicroSD card is needed in this example and *Kingston 1GB/2GB* or *SanDisk 2GB* MicroSD Card is recommended. The card should be formatted before used.

Note: *Kingston 1GB/2GB and SanDisk 2GB MicroSD Card have been tested on DevKit407. It's not guaranteed that all kind of MicroSD card work well on the board.*

3. Steps to Run

1) Insert the MicroSD card into MicroSD slot CON6.
2) Connect the DevKit407 board to a PC with a 'USB type A (Male) to Mini-B (Male)' cable through USB connector CN1 to power the board.
3) Open the project, rebuild all files, load project image and then run program.
4) Testing actions and results:
    - If the Erase operation is PASSED then LED3 ON else the LED6 is ON and LED3 is OFF
    - If the Single Block Write/Read operation is PASSED then LED4 ON else the LED6 is ON and LED4 is OFF
    - If the Multi Blocks Write/Read operation is PASSED then LED5 ON else the LED6 is ON and LED5 is OFF
    - Any SD Card operation including the SD Initialization error is signaled by LED6 ON.

4.3 LCD Example

The STM32F4xx_LCD_Example folder contains two examples:

- LCD_35T
- LCD_Touch

LCD_35T example project describes how to test LCD module STM32F4DIS-LCD on Devkit407.

LCD_Touch example describes how to precede LCD touch screen calibration. In this example, four points on the corner of touch screen need to be touch to complete calibration.

4.3.1 LCD_35T

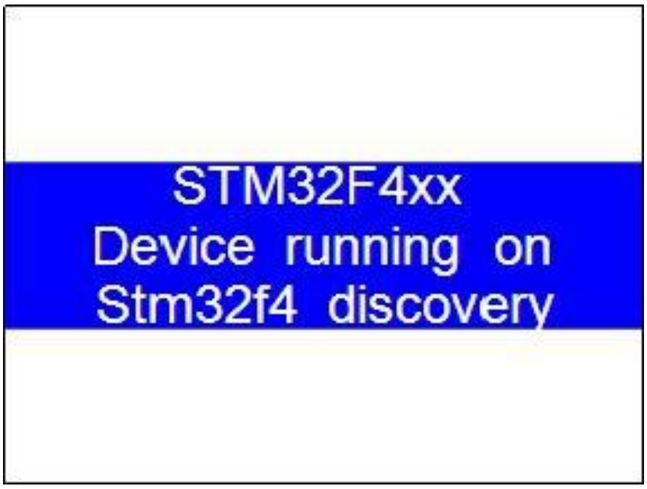
1. Description

The example is located in the following folder:

\Codes\STM32F4xx_LCD_Example\Project\LCD_35T

2. Steps to Run

1) Connect LCD module STM32F4DIS-LCD to DevKit407 CON3 via LCD cable. The red line onside indicates the first pin of LCD cable. The triangle onside indicates the first pin of LCD connector.
2) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.
3) Open the project, rebuild all files, load project image and then run program.
4) Testing actions and results:
   - A message display on the LCD



STM32F4xx
Device running on
Stm32f4 discovery

Figure 4-1 LCD display message

- LCD displays three color bars, red, green and blue.

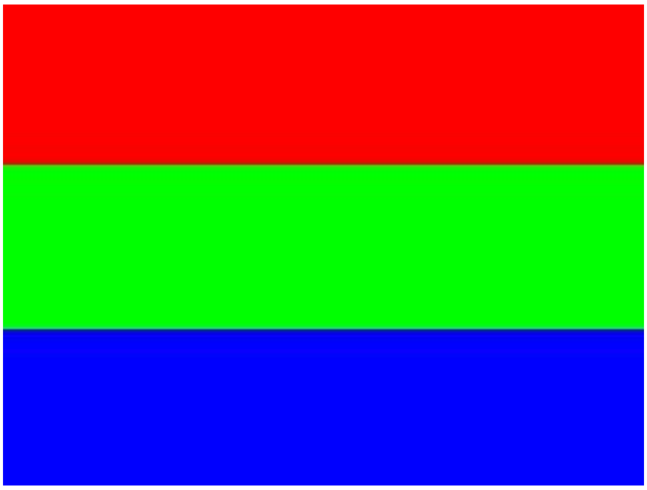


Figure 4-2 Three color bars

4.3.2 LCD_Touch

1. Description

The example is located in the following folder:

```
\Codes\STM32F4xx_LCD_Example\Project\LCD_Touch
```

2. Steps to Run

1) Connect LCD module STM32F4DIS-LCD to DevKit407 CON3 via LCD cable. The red line onside indicates the first pin of LCD cable. The triangle onside indicates the first pin of LCD connector.

2) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’

cable through USB connector CN1 to power the board.

3) Open the project, rebuild all files, load project image and then run program.

4) Testing actions and results:

5) Click calibration points accurately using a touch pen.

6) LCD will show a message if the calibration is OK. If calibration is OK, then MCU will enter into Calibration_Test_Dispose function.

7) In this function LCD will display the value of points touched by the pen. Both ADC values and coordinate values are displayed.

4.4 DCMI Example

The STM32F4xx_Camera_Example folder contains one example:

- OV9655_Camera

1. Purpose

OV9655_Camera example shows how to use the DCMI to control the OV9655 Camera module (STM32F4DIS-CAM) connected with Devkit407 board.

2. Description

The example is located in the following folder:

```
\Codes\STM32F4xx_Camera_Example\Project\OV9655_Camera
```

3. Hardware Configuration

A Camera module, STM32F4DIS-CAM, is needed in this example (options module for DevKit407).

A MicroSD card is needed in this example and *Kingston 1GB/2GB* or *SanDisk 2GB* MicroSD Card is recommended. The card should be formatted before used.

***Note:** Kingston 1GB/2GB and SanDisk 2GB MicroSD Card have been tested on DevKit407. It's not guaranteed that all kind of MicroSD card work well on the board.*

4. Steps to Run

1) Connect STM32F4DIS-CAM CON1 to DevKit407 CON7 via FFC soft cable when the

power is turned off. Please make sure that STM32F4DIS-CAM module is mounted correctly.

Note: *Make sure that JP1 and JP2 are not fitted.*

2) Insert a MicroSD card into MicroSD slot CON6.
3) Connect LCD module STM32F4DIS-LCD to DevKit407 board through CON3
4) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.
5) Open the MDK project, rebuild all files, load project image and then run program.
6) When the program is running, images collected by the module is shown on LCD. Press User button to take a photograph. The photograph will be stored into MICROSD card automatically and named as “PICxx.BMP”. ”xx” is picture’s number; “BMP” is picture’s format.

4.5 ETH_LwIP Example

STM32F4xx_Ethernet_Example folder contains Standalone and FreeRTOS subfolders:

For Standalone demonstrations LwIP v1.3.2 is used as the TCP/IP stack.

For FreeRTOS demonstrations, LwIP v1.3.2 is used as the TCP/IP stack and FreeRTOS v6.1.0 is used as the Real Time Kernel.

Standalone Examples

There are five examples in Standalone subfolder:

- httpserver
- tcp_echo_client
- tcp_echo_server
- udp_echo_client
- udp_echo_server

Httpserver example shows how to implement a web server application for STM32F4x7 devices.

Tcp_echo_client example shows how to implement a TCP echo client demonstration for STM32F4x7 devices.

Tcp_echo_server example shows how to implement a TCP echo server demonstration for STM32F4x7 devices.

Udp_echo_client example shows how to implement a UDP echo client demonstration for STM32F4x7 devices.

Udp_echo_server example shows how to implement a UDP echo server demonstration for STM32F4x7 devices.

4.5.1 Standalone\httpserver

1. Description

The HTTP server demo shows an implementation of a web server with the following features:

- URL parsing
- support of CGI (Common Gateway Interface)
- support of SSI (Server Side Includes)
- dynamic Header generation
- support of HTTP Post request

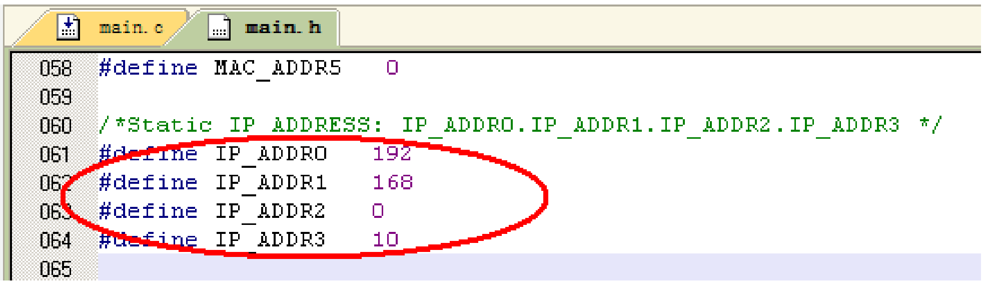
2. Software Configuration

Ethernet is used in this example. Please reference at 3.3.2 PC Network Settings for more details.

Try to make an external 10K, 3 pins vary-resistor, 1 terminal connects to 3V, the other connects to GND, and the rest (middle pin) connects to ADC1_IN3 (PA3, CON4.26).

3. Steps to Run

1) Configure IP address (The default Static IP address) of evaluation board. Modify the relevant macro in main.h file as per your requirement, as shown below.



```
#define MAC_ADDR5    0

/*Static IP ADDRESS: IP_ADDR0.IP_ADDR1.IP_ADDR2.IP_ADDR3 */
#define IP_ADDR0    192
#define IP_ADDR1    168
#define IP_ADDR2    0
#define IP_ADDR3    10
```

Figure 4-3 Configure IP address of DevKit407

You can also uncomment option “USE_DHCP” to enable the DHCP to assign IP addresses dynamically.

2) Connect the external 10K, 3 pins vary-resistor following the above instructions.

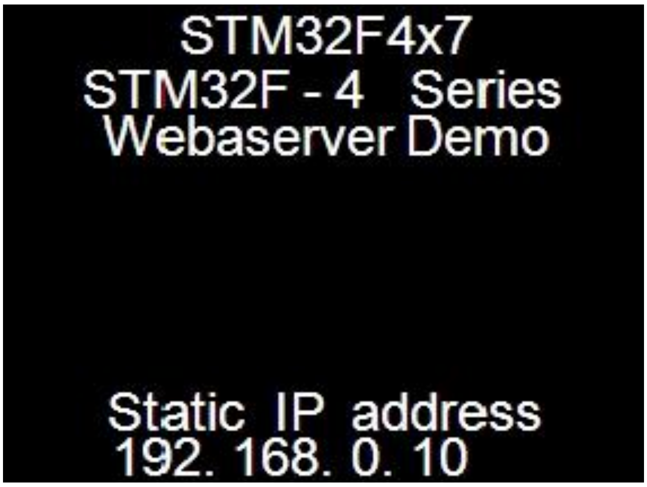
3) Connect LCD module STM32F4DIS-LCD to DevKit407 board through CON3

4) Connect the DevKit407 board to a PC with a crossover Ethernet cable through RJ45 connector J1.

5) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.

6) Rebuild all files, load project image and then run program.

7) After reset, LCD displays the IP address of the DevKit407 board.

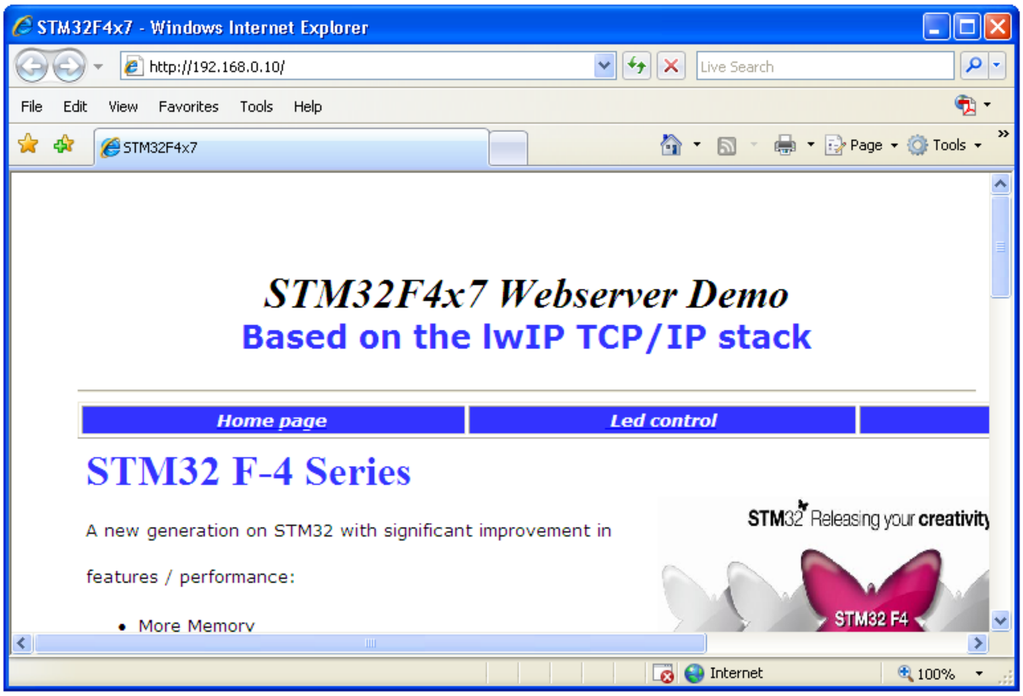


STM32F4x7
STM32F - 4 Series
Webaserver Demo

Static IP address
192. 168. 0. 10

Figure 4-4 LCD display message

8) On the remote PC, open a web client (Mozilla Firefox or Internet Explorer) and type the board’s IP address in a web browser. The default IP address is 192.168.0.10.



STM32F4x7 Webserver Demo

Based on the lwIP TCP/IP stack

Home page | Led control

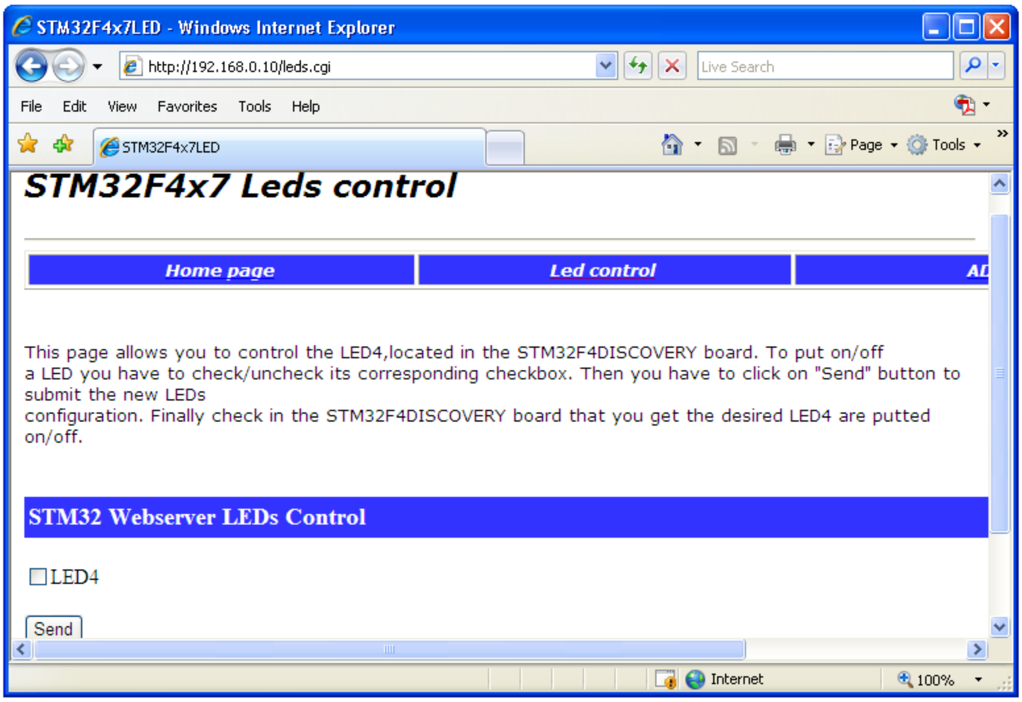
STM32 F-4 Series

A new generation on STM32 with significant improvement in features / performance:

- More Memory

Figure 4-5 Home page of the HTTP server demo

- Click “LED control” to get into LED control interface, select or cancel LED4 and press “Send”, the LED4 on the board will work accordingly.


STM32F4x7LED - Windows Internet Explorer
http://192.168.0.10/leds.cgi
Live Search
File Edit View Favorites Tools Help
STM32F4x7LED
Page Tools
Home page
Led control
AD
Internet
100%

STM32F4x7 Leds control

This page allows you to control the LED4,located in the STM32F4DISCOVERY board. To put on/off
a LED you have to check/uncheck its corresponding checkbox. Then you have to click on "Send" button to submit the new LEDs
configuration. Finally check in the STM32F4DISCOVERY board that you get the desired LED4 are putted on/off.

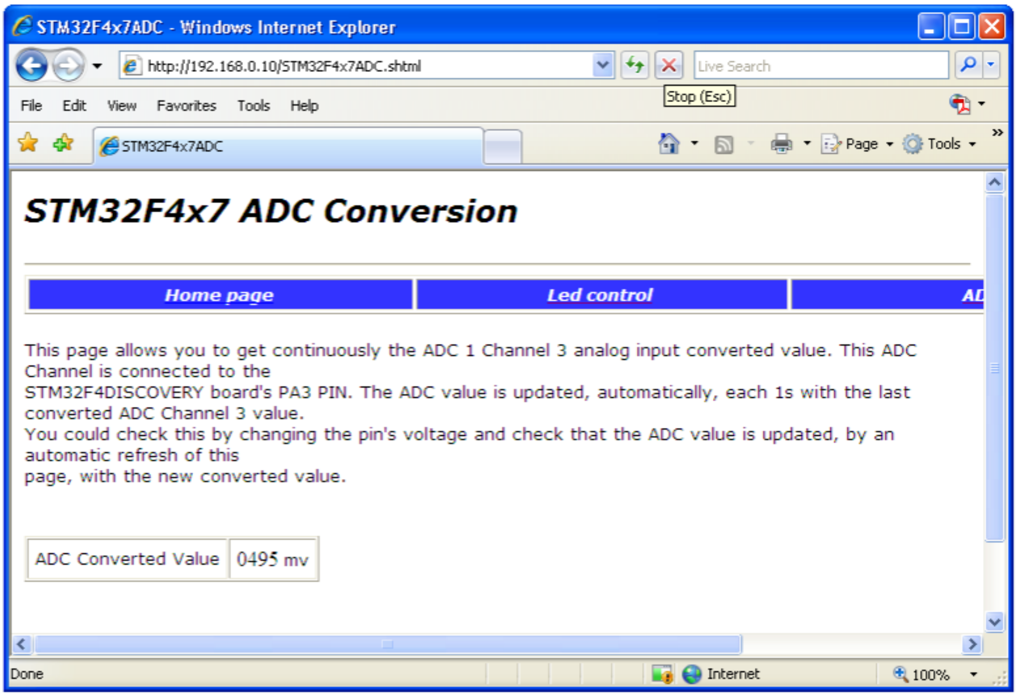
STM32 Webserver LEDs Control

□LED4

Send

Figure 4-6 Led control page of the HTTP server demo

- Click “ADC status bar” to get the voltage value of potentiometer.


STM32F4x7ADC - Windows Internet Explorer
http://192.168.0.10/STM32F4x7ADC.shtml
Live Search
Stop (Esc)
File Edit View Favorites Tools Help
STM32F4x7ADC
Page Tools
Home page
Led control
AD
Done
Internet
100%

STM32F4x7 ADC Conversion

This page allows you to get continuously the ADC 1 Channel 3 analog input converted value. This ADC Channel is connected to the
STM32F4DISCOVERY board's PA3 PIN. The ADC value is updated, automatically, each 1s with the last converted ADC Channel 3 value.
You could check this by changing the pin's voltage and check that the ADC value is updated, by an automatic refresh of this
page, with the new converted value.

ADC Converted Value	0495 mv

Figure 4-7 ADC status bar

4.5.2 Standalone\tcp_echo_client

1. Description

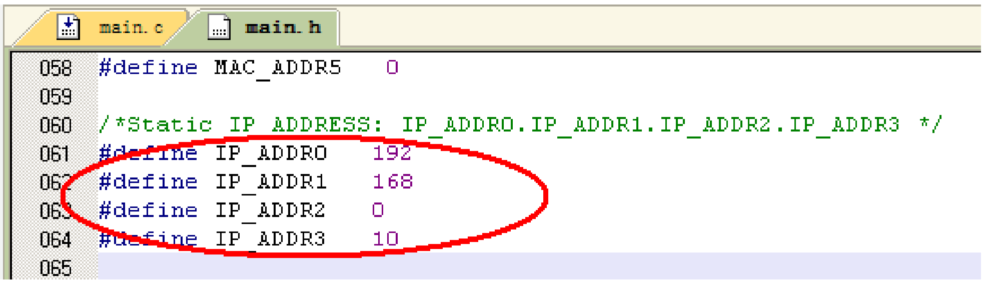
This demo is used to test a basic TCP connection. In this demo, the STM32F4 acts as a TCP client that connects to the TCP server. The client sends a string and the server echoes back the same string to the client.

2. Software Configuration

Ethernet is used in this example. Please reference at 3.3.2 PC Network Settings for more details.

3. Steps to Run

1) Configure IP address (The default Static IP address) of evaluation board. Modify the relevant macro in main.h file as per your requirement, as shown below.

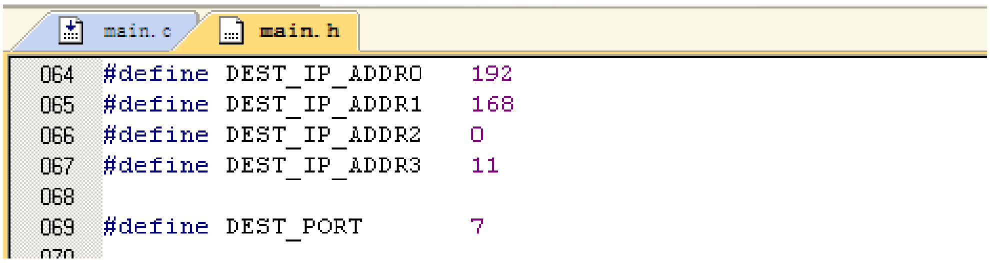


```
#define MAC_ADDR5   0

/*Static IP ADDRESS: IP_ADDR0.IP_ADDR1.IP_ADDR2.IP_ADDR3 */
#define IP_ADDR0   192
#define IP_ADDR1   168
#define IP_ADDR2   0
#define IP_ADDR3   10

```

Figure 4-8 Configure IP address of DevKit407

You can also uncomment option “USE_DHCP” to enable the DHCP to assign IP addresses dynamically.

2) Configure IP address (The default Static IP address) of remote PC. Modify the relevant macro in main.h depending on your needs, as shown below.



```
#define DEST_IP_ADDR0   192
#define DEST_IP_ADDR1   168
#define DEST_IP_ADDR2   0
#define DEST_IP_ADDR3   11

#define DEST_PORT       7
```

Figure 4-9 Configure IP address of remote PC

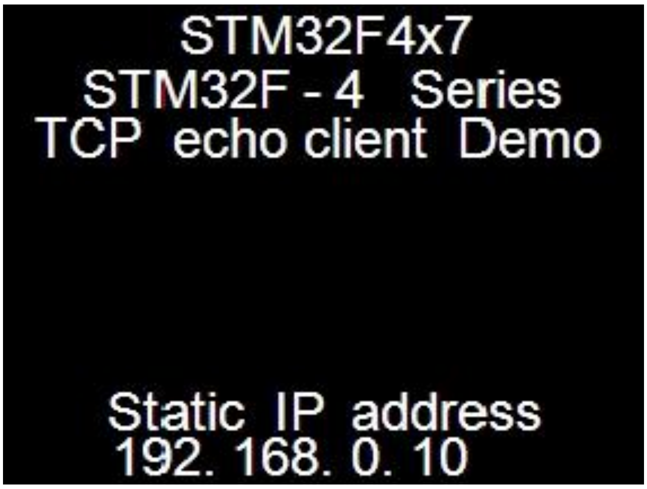
3) Connect LCD module STM32F4DIS-LCD to DevKit407 board through CON3

4) Connect the DevKit407 board to a PC with a crossover Ethernet cable through RJ45 connector J1.

5) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.

6) Rebuild all files, load project image and then run program.

7) After reset, LCD displays the IP address of the DevKit407 board.



STM32F4x7
STM32F - 4 Series
TCP echo client Demo

Static IP address
192. 168. 0. 10

Figure 4-10 LCD display message

8) On the remote PC, copy the echotool software to C root directory.

   The echotool software is located in the folder of CD-ROM:

   \Codes\STM32F4xx_Ethernet_Example\Utilities\PC_Software

9) On the PC, open a command prompt window. (In Windows, select ***Start > All Programs > Accessories > Command Prompt.***)

10) At the command prompt, enter:

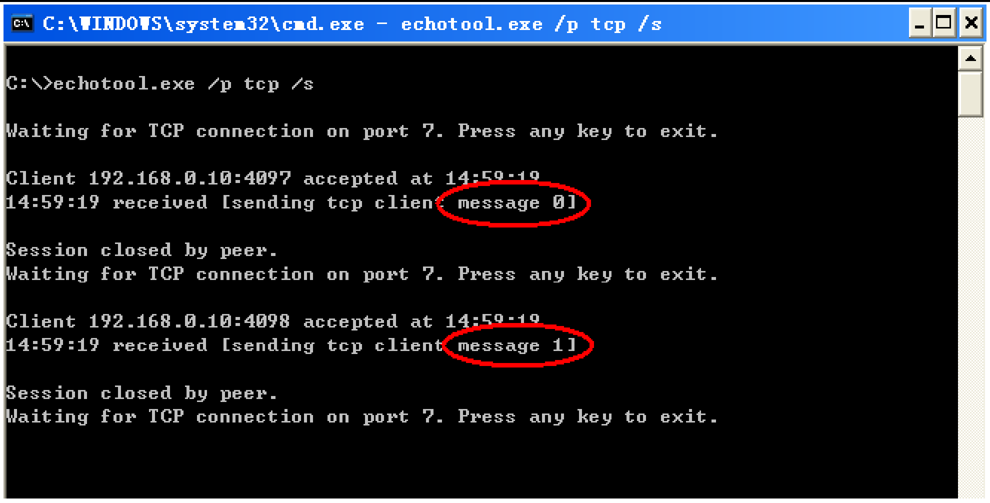
    C:\>echotool /p tcp /s

    Where:

    – /p tcp is the TCP protocol (TCP protocol)

    – /s is the actual mode of connection (Server mode)

11) When the USER1 button on the DevKit407 board is pressed, the client sends a string and the server echoes back the same string to the client. The below screenshot shows an example of the command string and the module’s response.



```
C:\>echotool.exe /p tcp /s

Waiting for TCP connection on port 7. Press any key to exit.

Client 192.168.0.10:4097 accepted at 14:59:19
14:59:19 received [sending tcp client message 0]

Session closed by peer.
Waiting for TCP connection on port 7. Press any key to exit.

Client 192.168.0.10:4098 accepted at 14:59:19
14:59:19 received [sending tcp client message 1]

Session closed by peer.
Waiting for TCP connection on port 7. Press any key to exit.
```

Figure 4-11 TCP echo client demo

4.5.3 Standalone\tcp_echo_server

1. Description

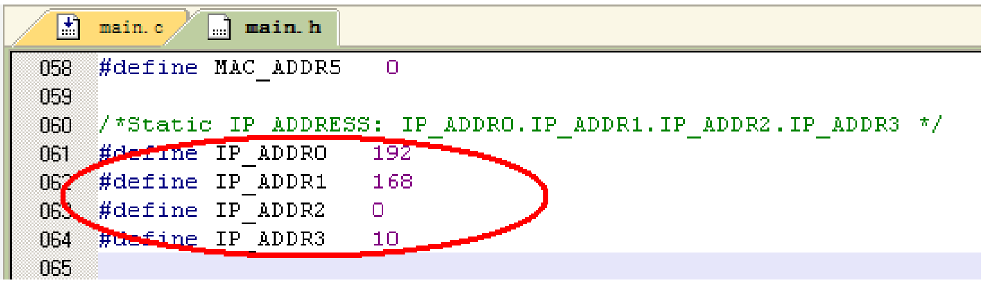
This demo is used to test a basic TCP connection. In this demo, the STM32F4 acts as a TCP server that waits for client requests. It simply echoes back whatever is sent.

2. Software Configuration

Ethernet is used in this example. Please reference at 3.3.2 PC Network Settings for more details.

3. Steps to Run

1) Configure IP address (The default Static IP address) of evaluation board. Modify the relevant macro in main.h file as per your requirement, as shown below.



```
#define MAC_ADDR5   0

/*Static IP ADDRESS: IP_ADDR0.IP_ADDR1.IP_ADDR2.IP_ADDR3 */
#define IP_ADDR0   192
#define IP_ADDR1   168
#define IP_ADDR2   0
#define IP_ADDR3   10
```

Figure 4-12 Configure IP address of DevKit407

You can also uncomment option “USE_DHCP” to enable the DHCP to assign IP addresses dynamically.

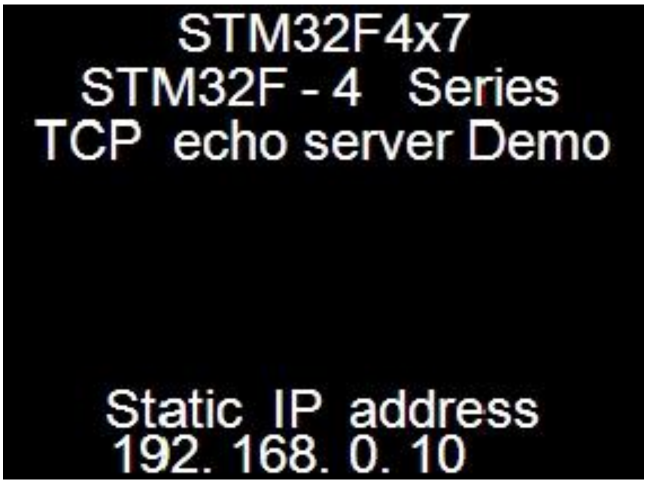
2) Connect LCD module STM32F4DIS-LCD to DevKit407 board through CON3

3) Connect the DevKit407 board to a PC with a crossover Ethernet cable through RJ45 connector J1.

4) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.

5) Rebuild all files, load project image and then run program.

6) After reset, LCD displays the IP address of the DevKit407 board.



STM32F4x7
STM32F - 4 Series
TCP echo server Demo

Static IP address
192. 168. 0. 10

Figure 4-13 LCD display message

7) On the PC, copy the echotool software to C root directory.

The echotool software is located in the folder of CD-ROM:

\Codes\STM32F4xx_Ethernet_Example\Utilities\PC_Software

8) On the remote PC, open a command prompt window. (In Windows, select ***Start > All Programs > Accessories > Command Prompt.***)

C:\>echotool.exe IP_address /p tcp /r 7 /n 15 /t 2 /d Testing LwIP TCP echo server

where;

– IP_address is the actual board's IP address;

By default the following static IP address is used: 192.168.0.10

– /p tcp is the protocol (TCP protocol)

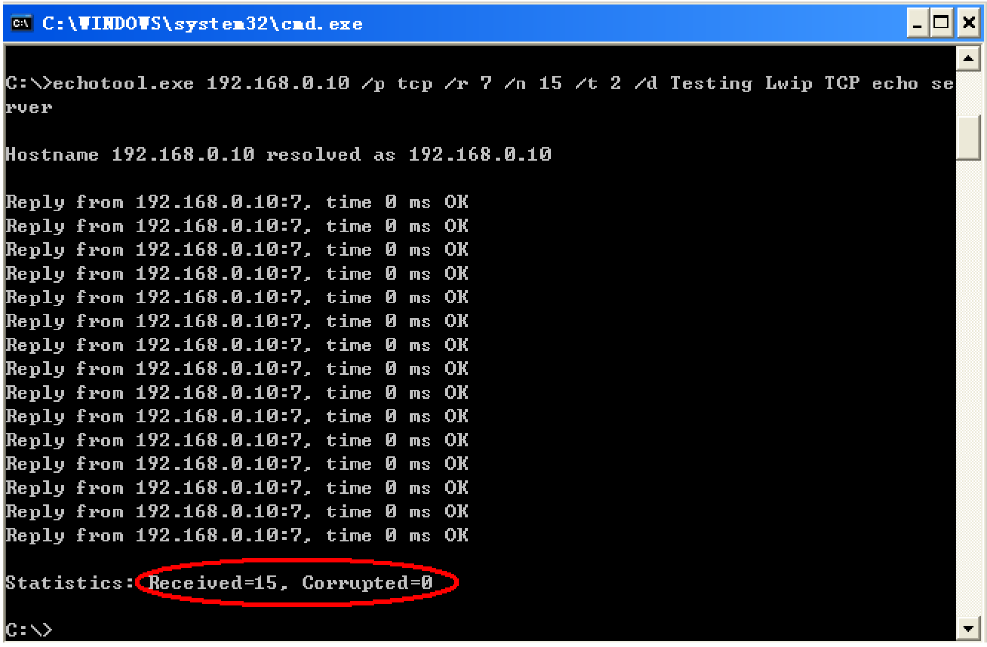
– /r is the actual remote port on the echo server (echo port)

– /n is the number of echo requests

– /t is the connection timeout in seconds

– /d is the message to be sent for echo

9) The below screenshot shows an example of this command string and the module response.



```
C:\>echotool.exe 192.168.0.10 /p tcp /r 7 /n 15 /t 2 /d Testing Lwip TCP echo se
rver

Hostname 192.168.0.10 resolved as 192.168.0.10

Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK

Statistics: Received=15, Corrupted=0

C:\>
```

Figure 4-14 TCP echo server demo

4.5.4 Standalone\udp_echo_client

1. Description

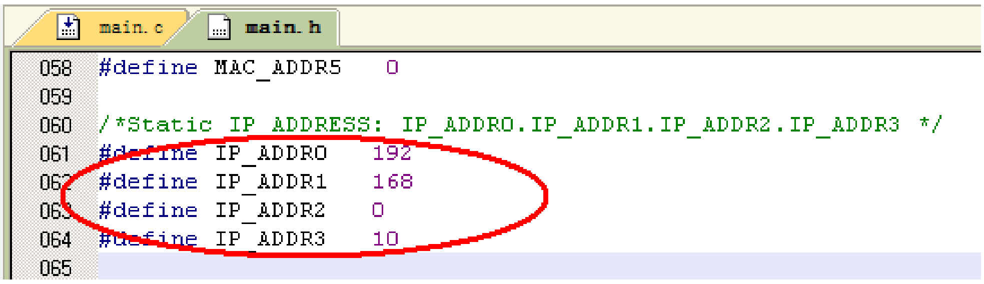
This demo is used to test a basic UDP echo connection. In this demo the STM32 acts as a UDP client that connects to a UDP server.

2. Software Configuration

Ethernet is used in this example. Please reference at 3.3.2 PC Network Settings for more details.

3. Steps to Run

1) Configure IP address (The default Static IP address) of evaluation board. Modify the relevant macro in main.h file as per your requirement, as shown below.

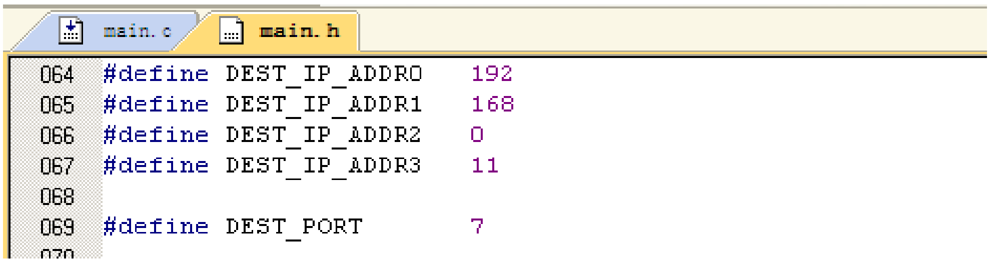


```
#define MAC_ADDR5    0

/*Static IP ADDRESS: IP_ADDR0.IP_ADDR1.IP_ADDR2.IP_ADDR3 */
#define IP_ADDR0   192
#define IP_ADDR1   168
#define IP_ADDR2   0
#define IP_ADDR3   10

```

Figure 4-15 Configure IP address of DevKit407

You can also uncomment option “USE_DHCP” to enable the DHCP to assign IP addresses dynamically.

2) Configure IP address (The default Static IP address) of remote PC. Modify the relevant macro in main.h depending on your needs, as shown below.



```
#define DEST_IP_ADDR0   192
#define DEST_IP_ADDR1   168
#define DEST_IP_ADDR2   0
#define DEST_IP_ADDR3   11

#define DEST_PORT       7
```

Figure 4-16 Configure IP address of remote PC

3) Connect LCD module STM32F4DIS-LCD to DevKit407 board through CON3

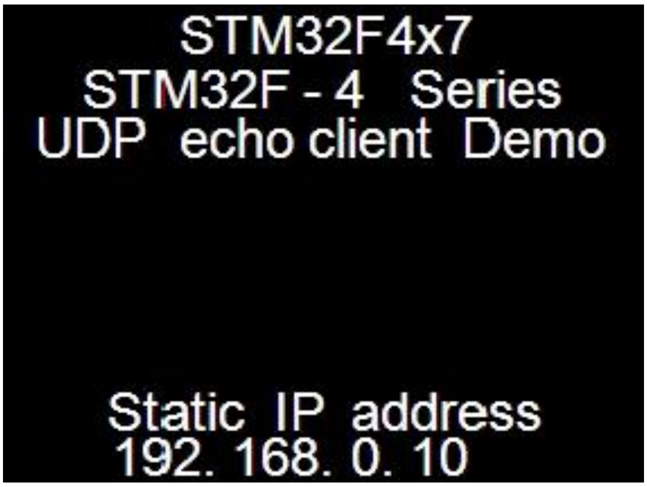
4) Connect the DevKit407 board to a PC with a crossover Ethernet cable through RJ45

connector J1.

5) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.

6) Rebuild all files, load project image and then run program.

7) After reset, LCD displays the IP address of the DevKit407 board.



```
STM32F4x7
STM32F - 4   Series
UDP   echo client   Demo

Static  IP  address
192. 168. 0. 10
```

Figure 4-17 LCD display message

8) On the PC, copy the echotool software to C root directory.

The echotool software is located in the folder of CD-ROM:

\Codes\STM32F4xx_Ethernet_Example\Utilities\PC_Software

9) On the remote PC, open a command prompt window. (In Windows, select ***Start > All Programs > Accessories > Command Prompt.***)

10) At the command prompt, enter:

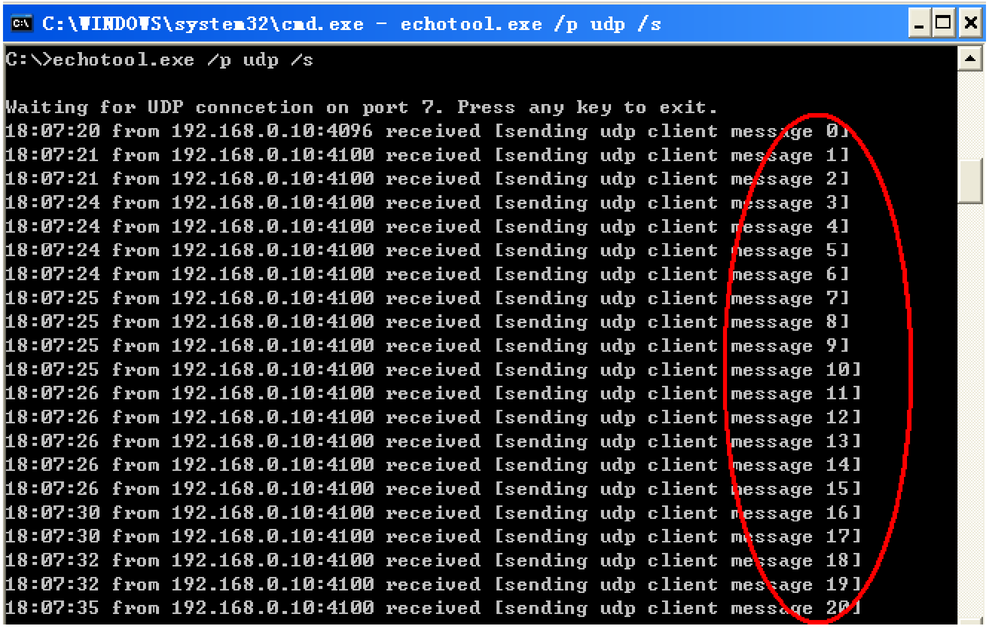
C:\>echotool /p udp /s

where;

– /p udp is the protocol (UDP protocol)

– /s is the actual mode of connection (Server mode)

11) When the USER1 button on the DevKit407 board is pressed, the client sends a string and the server echoes back the same string to the client. The follow figure shows an example of this command string and the module’s response.



```
C:\>echotool.exe /p udp /s

Waiting for UDP conncetion on port 7. Press any key to exit.
18:07:20 from 192.168.0.10:4096 received [sending udp client message 0]
18:07:21 from 192.168.0.10:4100 received [sending udp client message 1]
18:07:21 from 192.168.0.10:4100 received [sending udp client message 2]
18:07:24 from 192.168.0.10:4100 received [sending udp client message 3]
18:07:24 from 192.168.0.10:4100 received [sending udp client message 4]
18:07:24 from 192.168.0.10:4100 received [sending udp client message 5]
18:07:24 from 192.168.0.10:4100 received [sending udp client message 6]
18:07:25 from 192.168.0.10:4100 received [sending udp client message 7]
18:07:25 from 192.168.0.10:4100 received [sending udp client message 8]
18:07:25 from 192.168.0.10:4100 received [sending udp client message 9]
18:07:25 from 192.168.0.10:4100 received [sending udp client message 10]
18:07:26 from 192.168.0.10:4100 received [sending udp client message 11]
18:07:26 from 192.168.0.10:4100 received [sending udp client message 12]
18:07:26 from 192.168.0.10:4100 received [sending udp client message 13]
18:07:26 from 192.168.0.10:4100 received [sending udp client message 14]
18:07:26 from 192.168.0.10:4100 received [sending udp client message 15]
18:07:30 from 192.168.0.10:4100 received [sending udp client message 16]
18:07:30 from 192.168.0.10:4100 received [sending udp client message 17]
18:07:32 from 192.168.0.10:4100 received [sending udp client message 18]
18:07:32 from 192.168.0.10:4100 received [sending udp client message 19]
18:07:35 from 192.168.0.10:4100 received [sending udp client message 20]
```

Figure 4-18 UDP echo client demo

4.5.5 Standalone\udp_echo_server

1. Description

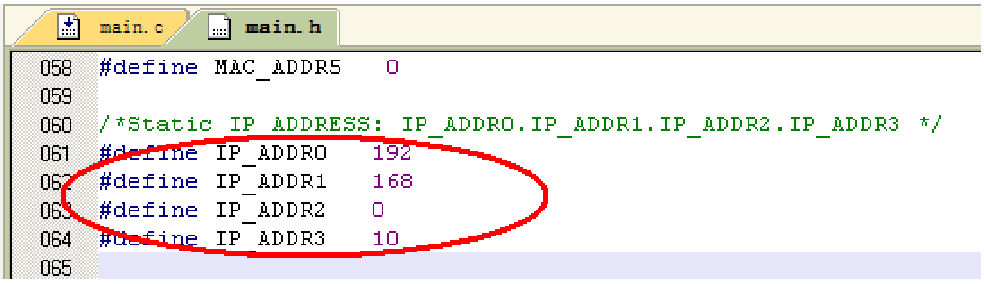
This demo is used to test a basic UDP connection. In this demo, the STM32 acts as a UDP server that waits for client requests.

2. Software Configuration

Ethernet is used in this example. Please reference at 3.3.2 PC Network Settings for more details.

3. Steps to Run

1) Configure IP address (The default Static IP address) of evaluation board. Modify the relevant macro in main.h file as per your requirement, as shown below.



```
#define MAC_ADDR5   0

/*Static IP ADDRESS: IP_ADDR0.IP_ADDR1.IP_ADDR2.IP_ADDR3 */
#define IP_ADDR0   192
#define IP_ADDR1   168
#define IP_ADDR2   0
#define IP_ADDR3   10

```

Figure 4-19 Configure IP address of DevKit407

You can also uncomment option “USE_DHCP” to enable the DHCP to assign IP addresses dynamically.

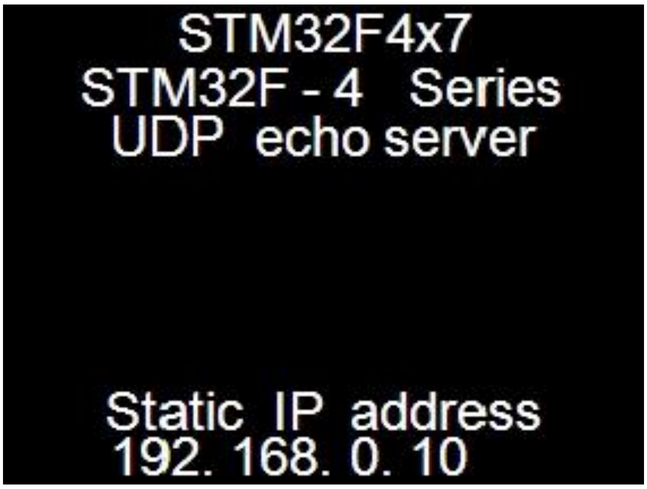
2) Connect LCD module STM32F4DIS-LCD to DevKit407 board through CON3

3) Connect the DevKit407 board to a PC with a crossover Ethernet cable through RJ45 connector J1.

4) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.

5) Rebuild all files, load project image and then run program.

6) After reset, LCD displays the IP address of the DevKit407 board.



STM32F4x7
STM32F - 4 Series
UDP echo server

Static IP address
192. 168. 0. 10

Figure 4-20 LCD display message

7) On the PC, copy the echotool software to C root directory.

The echotool software is located in the folder of CD-ROM:

\Codes\STM32F4xx_Ethernet_Example\Utilities\PC_Software

8) On the PC, open a command prompt window. (In Windows, select ***Start > All Programs > Accessories > Command Prompt.***)

9) At the command prompt, enter:

C:\>echotool.exe IP_address /p udp /r 7 /l 7 /n 15 /t 2 /d Testing LwIP UDP echo server

where;

– IP_address is the actual board's IP address;

By default the following static IP address is used: 192.168.0.10

– /p udp is the protocol (UDP protocol)

– /r is the actual remote port on the echo server (echo port)

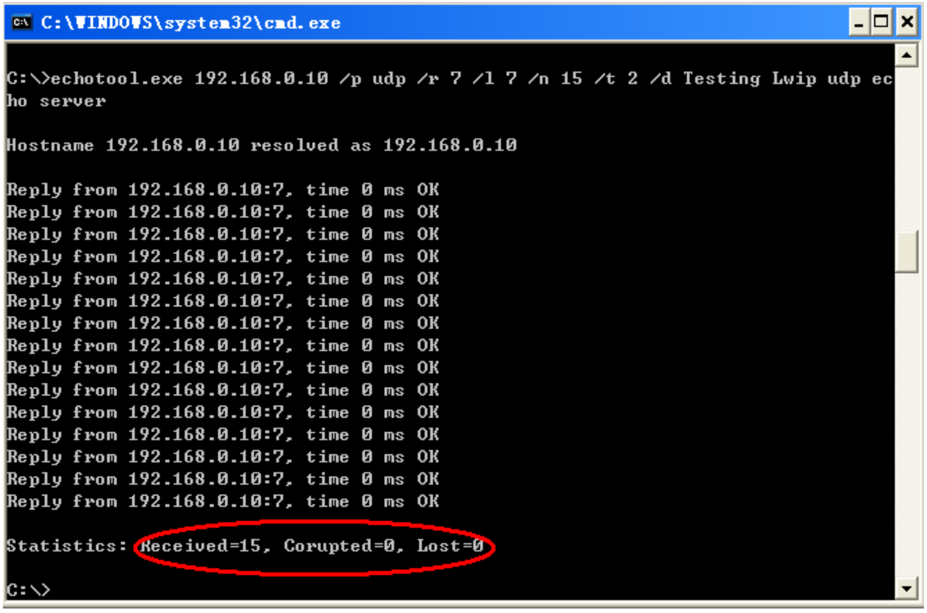
– /l is the actual local for the client (echo port)

– /n is the number of echo requests

– /t is the connection timeout in seconds

– /d is the message to be sent for echo

10) The below screenshot shows an example of this command string and the module's response.



```
C:\>echotool.exe 192.168.0.10 /p udp /r 7 /l 7 /n 15 /t 2 /d Testing Lwip udp ec
ho server

Hostname 192.168.0.10 resolved as 192.168.0.10

Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK
Reply from 192.168.0.10:7, time 0 ms OK

Statistics: Received=15, Corupted=0, Lost=0

C:\>
```

Figure 4-21 UDP echo server demo

FreeRTOS Examples

There are three examples in FreeRTOS subfolder:

- httpserver_netconn
- httpserver_socket
- udptcp_echo_server_netconn

Httpserver_netconn example shows how to implement a web serverapplication, based on the netconn API, for STM32F4x7 devices.

Httpserver_socket example shows how to implement a web server application, based on the socket API, for STM32F4x7 devices.

Udptcp_echo_server_netconn example shows how to implement a UDP-TCP echo server demonstration for STM32F4x7 devices.

4.5.6 FreeRTOS\httpserver_netconn

1. Description

This demo is used to connect the DevKit407 board with a web browser and to load HTML pages.

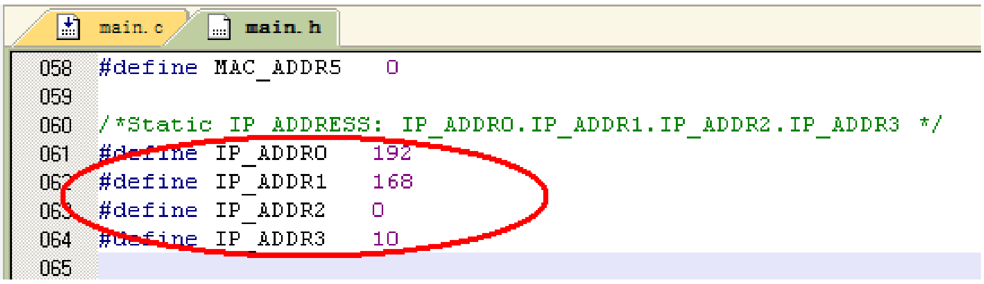
This demo has two HTML pages. The first one contains general information about STM32F4x7 microcontrollers, the demonstration package and the stack LwIP. The second one contains the list of running tasks and their status. This page is automatically updated every second.

2. Software Configuration

Ethernet is used in this example. Please reference at 3.3.2 PC Network Settings for more details.

3. Steps to Run

1) Configure IP address (The default Static IP address) of evaluation board. Modify the relevant macro in main.h file as per your requirement, as shown below.



```
#define MAC_ADDR5    0

/*Static IP ADDRESS: IP_ADDR0.IP_ADDR1.IP_ADDR2.IP_ADDR3 */
#define IP_ADDR0    192
#define IP_ADDR1    168
#define IP_ADDR2    0
#define IP_ADDR3    10
```

Figure 4-22 Configure IP address of DevKit407

You can also uncomment option “USE_DHCP” to enable the DHCP to assign IP addresses dynamically.

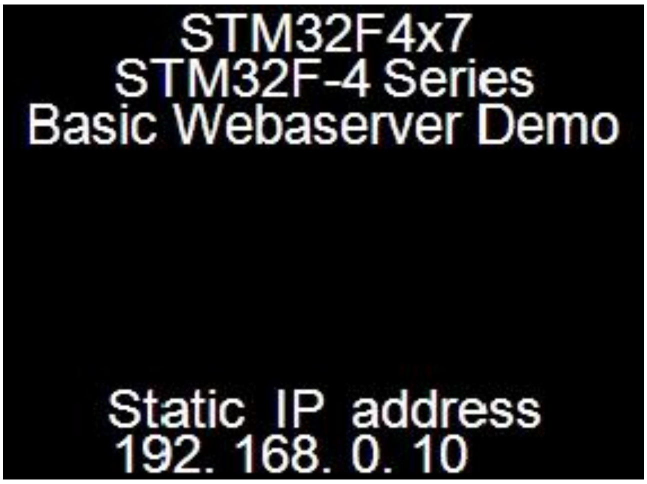
2) Connect LCD module STM32F4DIS-LCD to DevKit407 board through CON3

3) Connect the DevKit407 board to a PC with a crossover Ethernet cable through RJ45 connector J1.

4) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.

5) Rebuild all files, load project image and then run program.

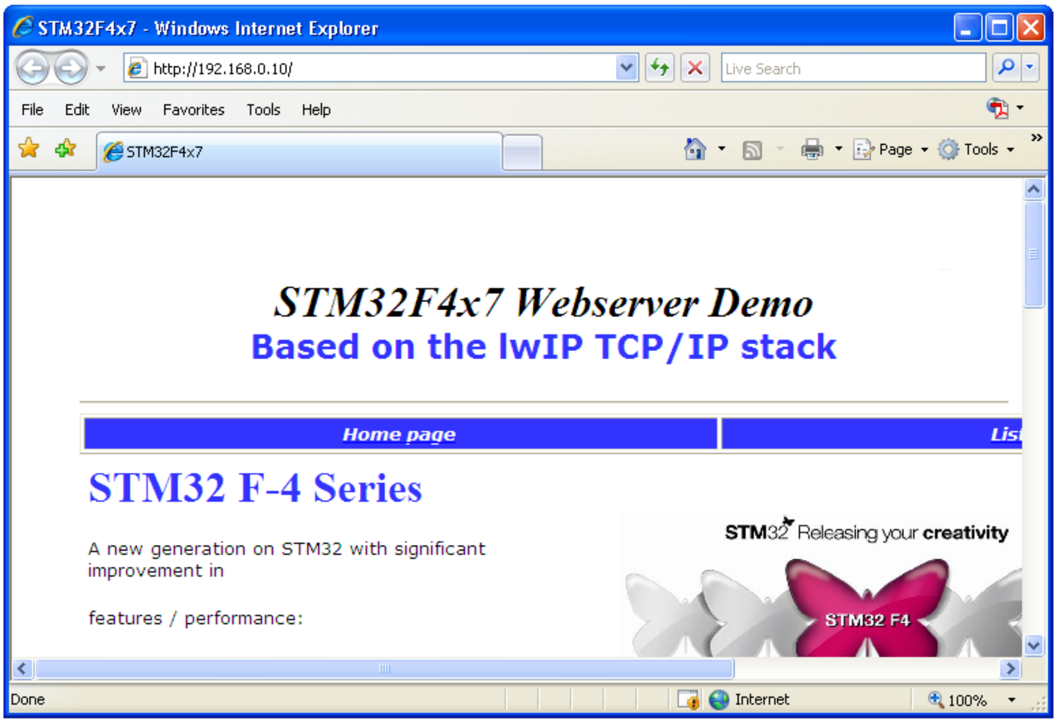
6) After reset, LCD displays the IP address of the DevKit407 board.



```
STM32F4x7
STM32F-4 Series
Basic Webaserver Demo

Static  IP  address
192. 168. 0. 10
```

Figure 4-23 LCD display message

7) On the remote PC, open a web client (Mozilla Firefox or Internet Explorer) and type the board’s IP address in a web browser. The default IP address is 192.168.0.10.



STM32F4x7 - Windows Internet Explorer
http://192.168.0.10/
Live Search
File Edit View Favorites Tools Help
STM32F4x7
Page Tools
Home page
Lis
STM32 Releasing your creativity
STM32 F4
Done
Internet
100%

STM32F4x7 Webserver Demo

Based on the lwIP TCP/IP stack

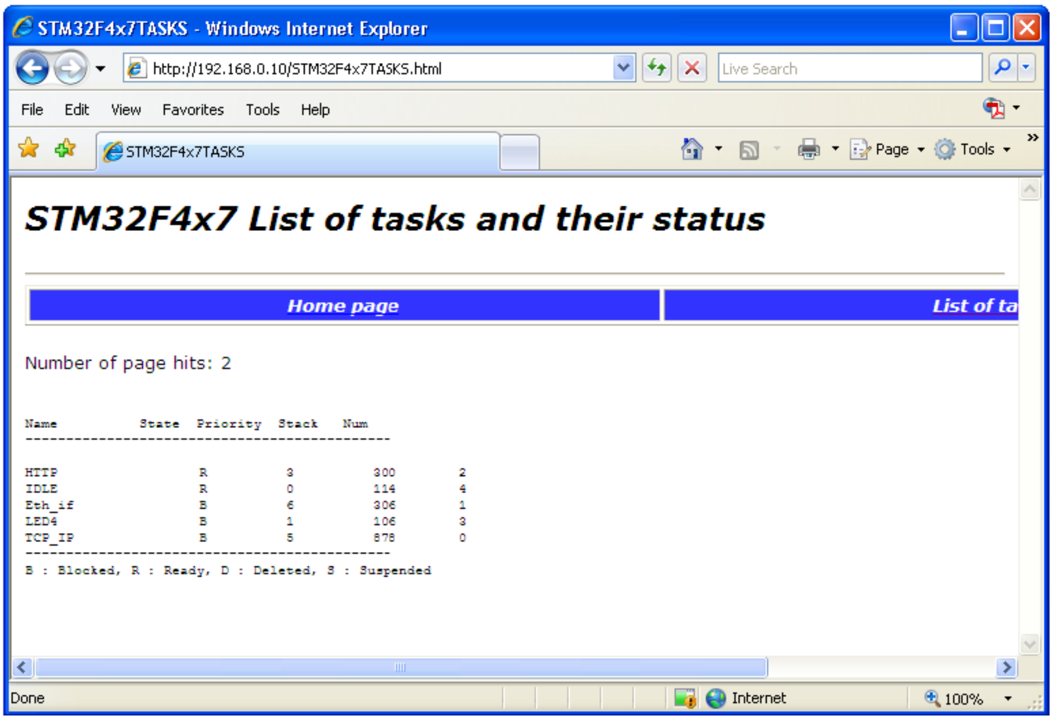
STM32 F-4 Series

A new generation on STM32 with significant improvement in

features / performance:

Figure 4-24 Home page of the HTTP server netconn demo

- Click the "List of tasks" into task status monitor page of FreeRTOS real-time system. As shown below:



STM32F4x7TASKS - Windows Internet Explorer
http://192.168.0.10/STM32F4x7TASKS.html
Live Search
File Edit View Favorites Tools Help
STM32F4x7TASKS
Page Tools
Home page
List of ta
Done
Internet
100%

STM32F4x7 List of tasks and their status

Number of page hits: 2

```
Name            State  Priority  Stack   Num
---------------------------------------------


HTTP                 R         3         300       2
IDLE                 R         0         114       4
Eth_if               B         6         306       1
LED4                 B         1         106       3
TCP_IP               B         5         878       0
---------------------------------------------
B : Blocked, R : Ready, D : Deleted, S : Suspended
```

Figure 4-25 List of tasks page of the HTTP server netconn demo

4.5.7 FreeRTOS\httpserver_socket

The HTTP server socket demo shows an implementation of web server application based on the socket API.

To test this demo, please refer to the 4.5.6 FreeRTOS\httpserver_netconn.

4.5.8 FreeRTOS\udptcp_echo_server_netconn

This demo provides the echo service application on both TCP and UDP protocols:

To test the UDP TCP echo server netconn demo in TCP server mode, please refer to the 4.5.3 Standalone\tcp_echo_server.

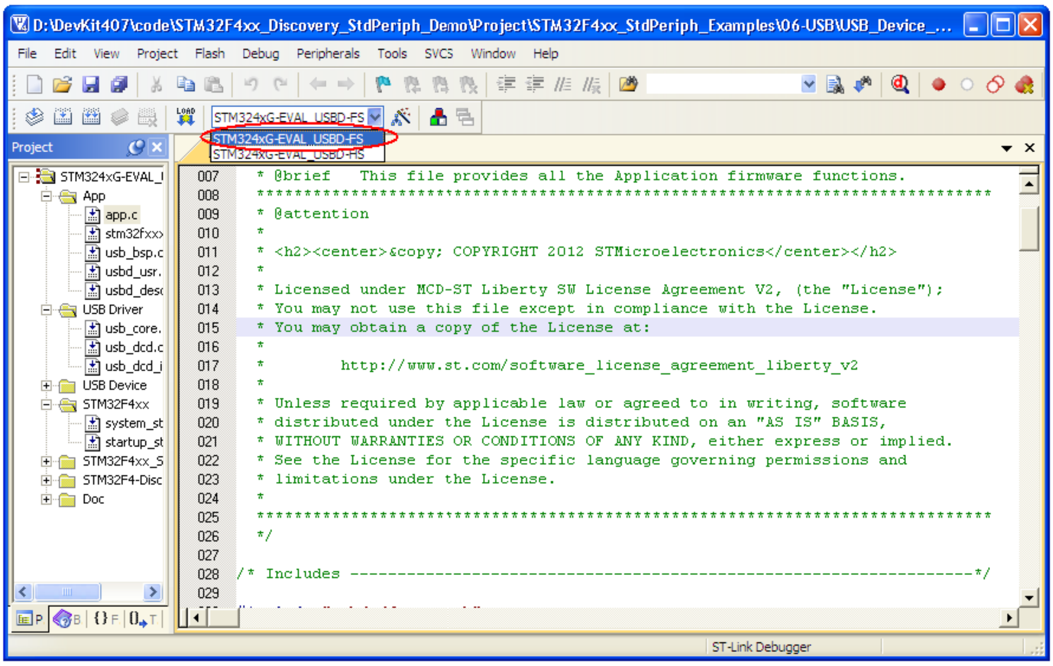
To test the UDP TCP echo server netconn demo in UDP server mode, please refer to the 4.5.5 Standalone\udp_echo_server.

4.6 USB Example

STM32F4xx_USB_Example folder contains USB_Device_Examples and USB_Host_Examples subfolders.

Please refer to *STM32F4xx USB On-The-Go host and device library.pdf* for more information.

Note: *USB High Speed is not available on DevKit407.Plesae select macro in IDE : STM324xG-EVAL_USBD_FS*



```
  * @brief   This file provides all the Application firmware functions.
  ******************************************************************************
  * @attention
  *
  * <h2><center>&copy; COPYRIGHT 2012 STMicroelectronics</center></h2>
  *
  * Licensed under MCD-ST Liberty SW License Agreement V2, (the "License");
  * You may not use this file except in compliance with the License.
  * You may obtain a copy of the License at:
  *
  *        http://www.st.com/software_license_agreement_liberty_v2
  *
  * Unless required by applicable law or agreed to in writing, software
  * distributed under the License is distributed on an "AS IS" BASIS,
  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  * See the License for the specific language governing permissions and
  * limitations under the License.
  *
  ******************************************************************************
  */

/* Includes ------------------------------------------------------------------*/
```

Figure 4-26 select macro STM324xG-EVAL_USBD_FS in IDE

USB_Device_Examples

There are three examples in USB_Device_Examples subfolder:

- DFU
- MSC
- VCP

DFU example presents the implementation of a device firmware upgrade (DFU) capability in the STM32F2xx, STM32F4xx and STM32F105/7 devices.

MSC example gives a typical example of how to use the STM32F2xx, STM32F4xx and STM32F105/7 USB OTG Device peripheral to communicate with a PC Host using the bulk transfer while the MicroSD card is used as storage media.

VCP example presents the implementation of a Virtual Com Port (VCP) capability in the STM32F2xx, STM32F4xx and STM32F105/7 devices.

4.6.1 USB_Device_Examples\DFU

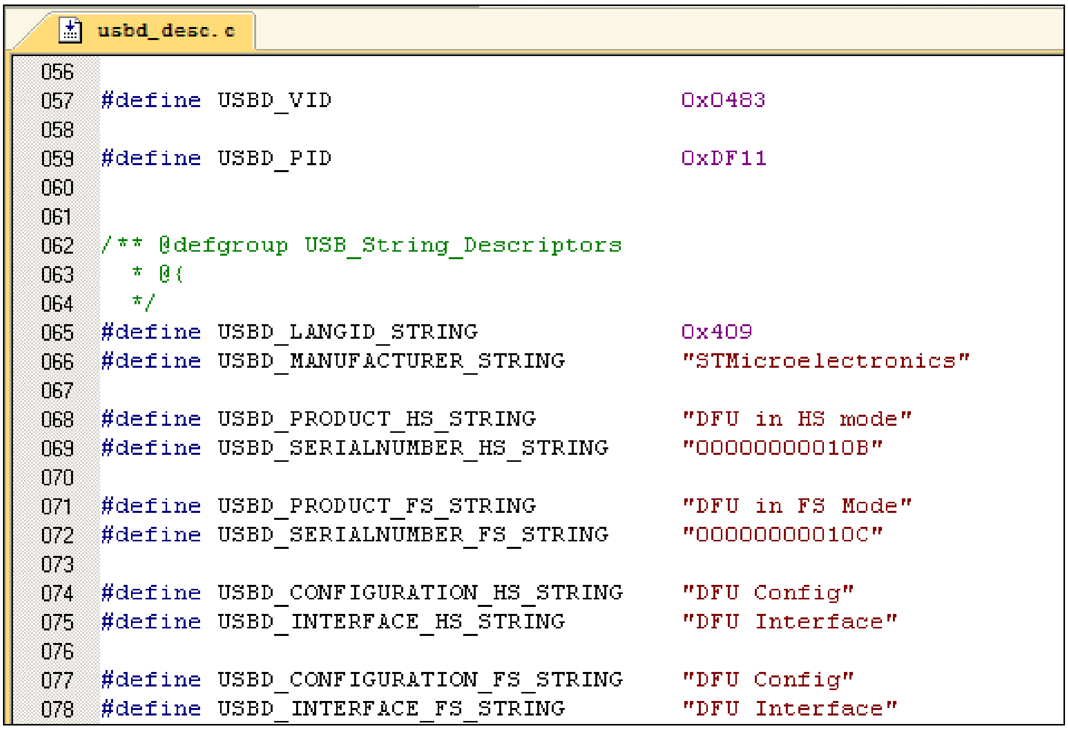
1. Description

The DFU(Device Firmware Upgrade)example allows a device firmware upgrade using the DFU drivers.

The supported memories for this example are:

- Internal Flash memory for STM32F105/7, STM32F2xx and STM32F4xx devices
- OTP memory for STM32F2xx and STM32F2xx devices.

DFU device information is located in usbd_desc.c, as shown below:



```

#define USBD_VID                          0x0483

#define USBD_PID                          0xDF11


/** @defgroup USB_String_Descriptors
  * @{
  */
#define USBD_LANGID_STRING                0x409
#define USBD_MANUFACTURER_STRING          "STMicroelectronics"

#define USBD_PRODUCT_HS_STRING            "DFU in HS mode"
#define USBD_SERIALNUMBER_HS_STRING       "00000000010B"

#define USBD_PRODUCT_FS_STRING            "DFU in FS Mode"
#define USBD_SERIALNUMBER_FS_STRING       "00000000010C"

#define USBD_CONFIGURATION_HS_STRING      "DFU Config"
#define USBD_INTERFACE_HS_STRING          "DFU Interface"

#define USBD_CONFIGURATION_FS_STRING      "DFU Config"
#define USBD_INTERFACE_FS_STRING          "DFU Interface"
```

Figure 4-27 USB DFU device information

2. Hardware Configuration

A USB type A (Male) to Micro AB (Male) cable is needed in this example.

3. Steps to Run

1) Install DfuSe_Demo_V3.0.2 software on the PC. The software is located in the folder at CD-ROM:

   \Codes\STM32F4xx_USB_Example\Utilities\PC_Software\DfuSe_Demo_V3.0.2

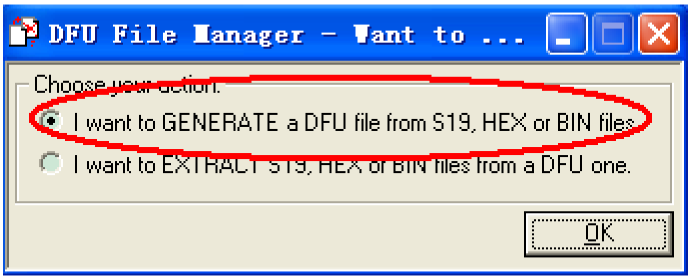
   If your PC is 64-bit, please install *DfuSe Demo V3.0.2_Setup_amd64.exe*.

2) Generate DFU upgrade file on the PC(Optional)

   Note: *There is a DFU file for testing the USB DFU example. User can skip this step. The DFU file is located in following folder:*

   \Codes\STM32F4xx_USB_Example\Project\USB_Device_Examples\DFU\binary_template\MDK-ARM

   ➢ In Installation directory of DfuSe_Demo_V3.0.2, open *BIN* folder, this opens a *DfuFileMgr* software, as shown below:



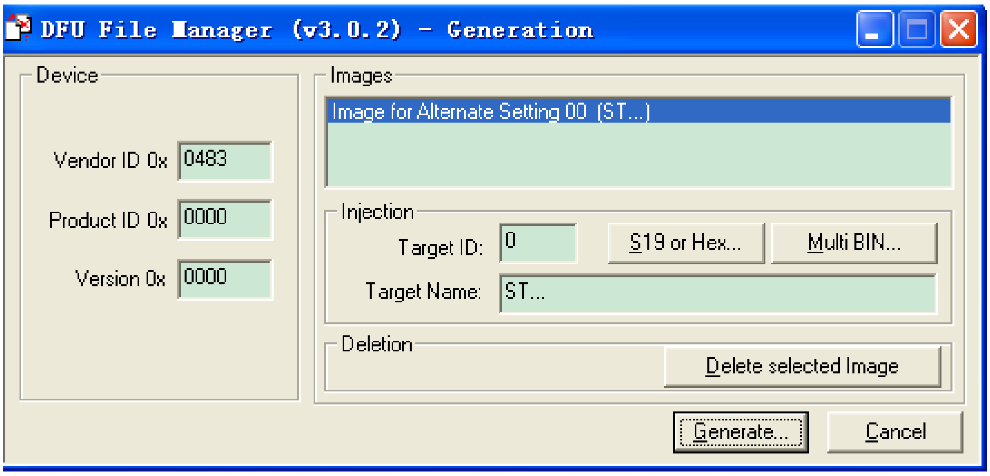
Choose your action:

● I want to GENERATE a DFU file from S19, HEX or BIN files

○ I want to EXTRACT S19, HEX or BIN files from a DFU one.

Figure 4-28 DFU file manage

- Click “OK”, this opens a window as shown below:



Vendor ID 0x 0483

Product ID 0x 0000

Version 0x 0000

Image for Alternate Setting 00 (ST...)

Target ID: 0

Target Name: ST...

Figure 4-29 Generate DFU file

- Click “S19 or Hex” button, select the file to be upgraded, then click “generate” button to generate DFU file.

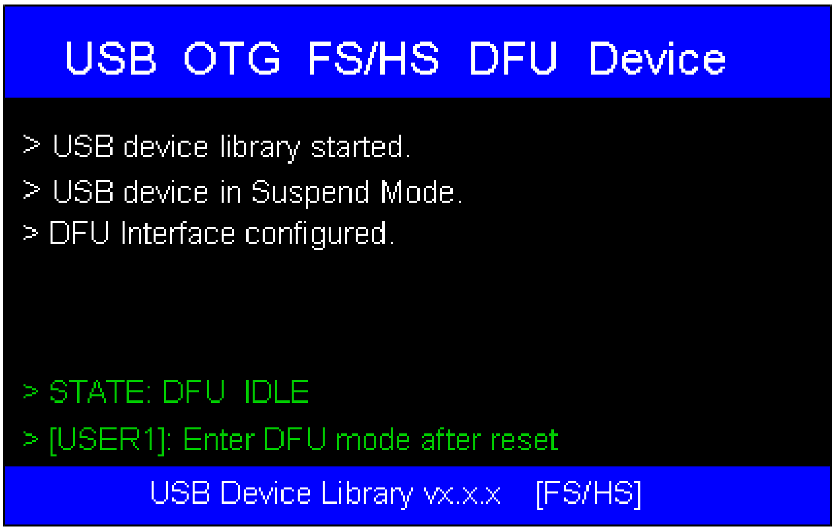
3) Connect LCD module STM32F4DIS-LCD to DevKit407 CON3 via LCD cable.

4) Connect the DevKit407 board to a PC with a 'USB type A (Male) to Micro AB (Male)' cable through USB connector CN5.

5) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.

6) Open the project, rebuild all files, load project image and then run program.

7) After reset, the LCD displays the following messages:

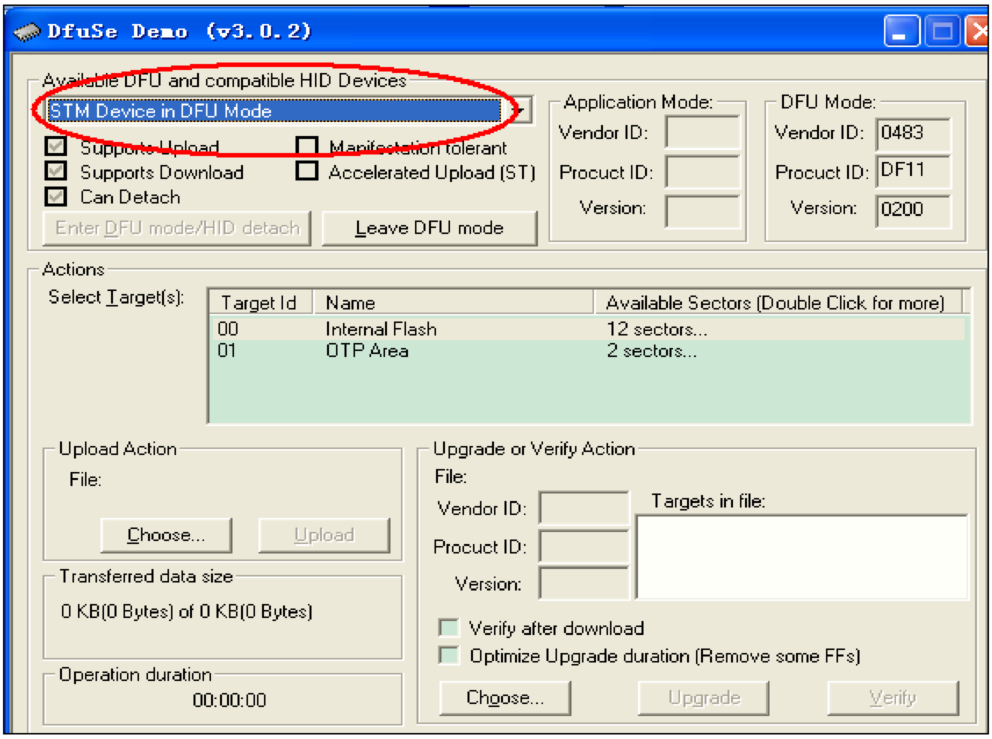


```
USB OTG FS/HS DFU Device

> USB device library started.
> USB device in Suspend Mode.
> DFU Interface configured.


> STATE: DFU IDLE
> [USER1]: Enter DFU mode after reset
USB Device Library vx.x.x   [FS/HS]
```

Figure 4-30 USB device firmware upgrade cable connected display message

8) Run DfuSe DEMO software on PC.

If PC identifies the DFU device (DevKit407 board), below window will be displayed, which means board is ready for USB DFU test.



STM Device in DFU Mode

☑ Supports Upload
☑ Supports Download
☑ Can Detach
□ Manifestation tolerant
□ Accelerated Upload (ST)

Application Mode: Vendor ID: Procuct ID: Version:

DFU Mode: Vendor ID: 0483 Procuct ID: DF11 Version: 0200

Target Id	Name	Available Sectors (Double Click for more)
00	Internal Flash	12 sectors...
01	OTP Area	2 sectors...

Upload Action File:

Transferred data size 0 KB(0 Bytes) of 0 KB(0 Bytes)

Operation duration 00:00:00

Upgrade or Verify Action File: Vendor ID: Procuct ID: Version:

□ Verify after download
□ Optimize Upgrade duration (Remove some FFs)

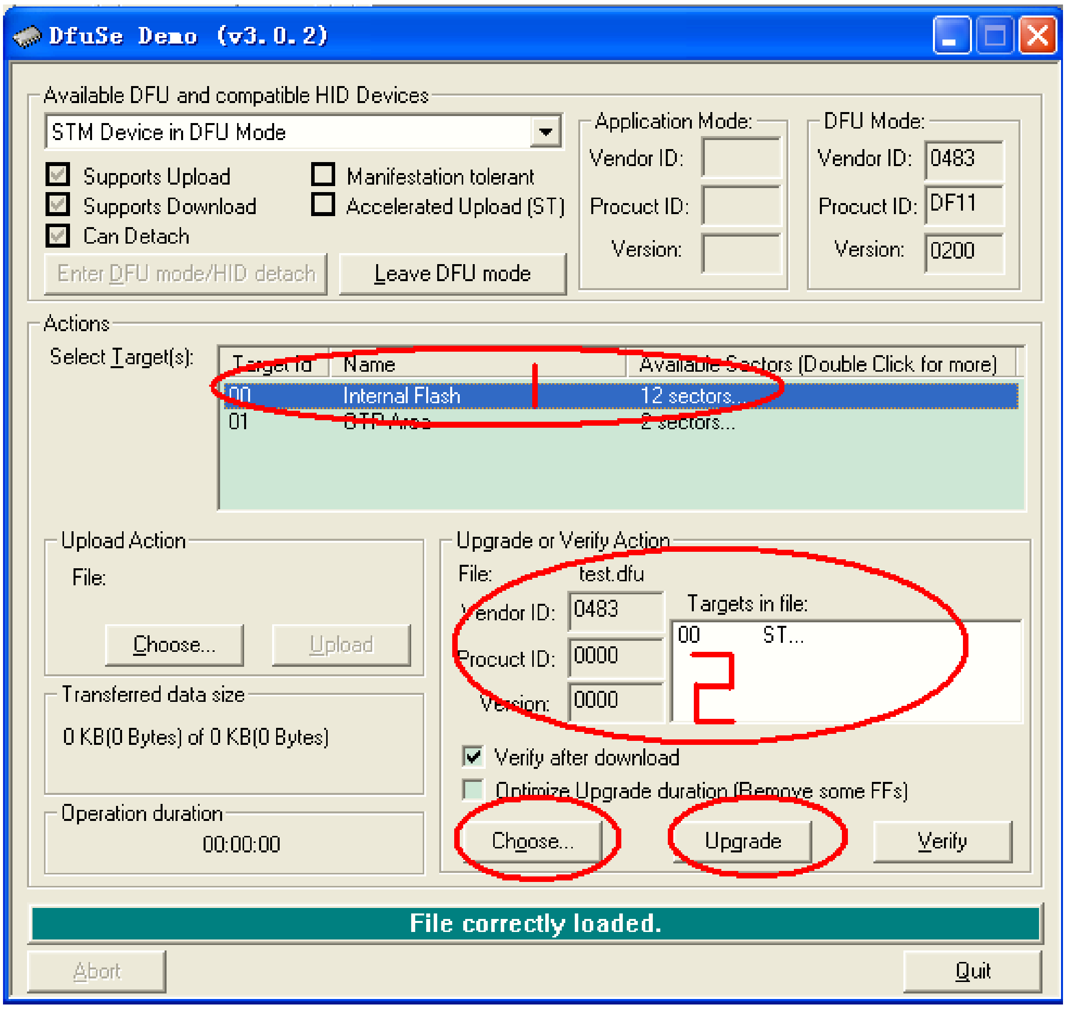
Figure 4-31 STM Device in DFU mode

9) Select the target area to be programmed, as shown in below figure with number 1.

10) Select the DFU file to be programmed. Click “Choose” button select the DFU to be upgraded, as shown in below figure with number 2.

There is a DFU file for USB DFU testing purpose at the folder location:

\Codes\STM32F4xx_USB_Example\Project\USB_Device_Examples\DFU\binary_template\MDK-ARM



DfuSe Demo (v3.0.2)

Available DFU and compatible HID Devices: STM Device in DFU Mode

☑ Supports Upload
☑ Supports Download
☑ Can Detach
□ Manifestation tolerant
□ Accelerated Upload (ST)

Enter DFU mode/HID detach | Leave DFU mode

Application Mode: Vendor ID: Procuct ID: Version:

DFU Mode: Vendor ID: 0483 Procuct ID: DF11 Version: 0200

Actions

Select Target(s):

Target Id	Name	Available Sectors (Double Click for more)
00	Internal Flash	12 sectors...
01	OTP Area	2 sectors...

Upload Action — File: Choose... | Upload

Transferred data size: 0 KB(0 Bytes) of 0 KB(0 Bytes)

Operation duration: 00:00:00

Upgrade or Verify Action — File: test.dfu

Vendor ID: 0483 Procuct ID: 0000 Version: 0000

Targets in file: 00 ST...

☑ Verify after download
□ Optimize Upgrade duration (Remove some FFs)

Choose... | Upgrade | Verify

File correctly loaded.

Abort | Quit

Figure 4-32 Upgrade DFU file

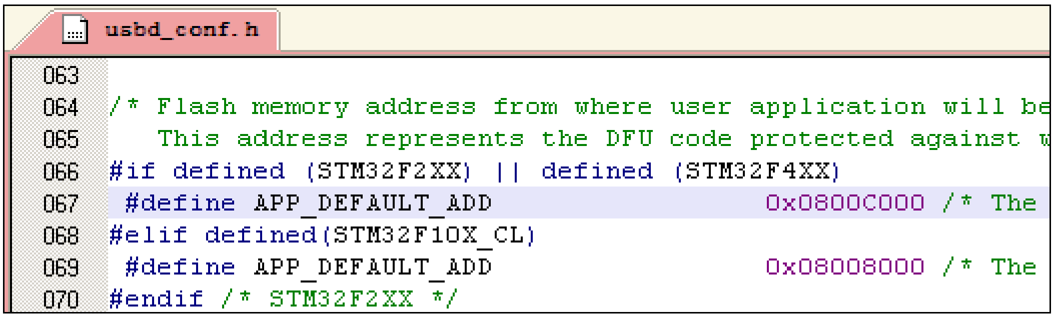
11) In order to update the firmware click “Upgrade” button to start the firmware update. Once completed a message will appear to indicate upgrade is successful or not.

12) After reset, MCU run in the new firmware.

Reset MCU while the User button is pushed, the MCU run the example that downloaded in step 11).

To go back to the DFU example, you have to reset the device (using RESET button or software reset).

Note: *In the DFU DEMO, the application start address is set to 0x0800C000, as* *shown below. This address represents the DFU code protected against write and erase operations. You can modify this address in usbd_conf.h, but you must make sure that there enough space for DFU code (0x08000000 ～ application start address).*



```

/* Flash memory address from where user application will be
   This address represents the DFU code protected against w
#if defined (STM32F2XX) || defined (STM32F4XX)
 #define APP_DEFAULT_ADD                    0x0800C000 /* The
#elif defined(STM32F10X_CL)
 #define APP_DEFAULT_ADD                    0x08008000 /* The
#endif /* STM32F2XX */
```

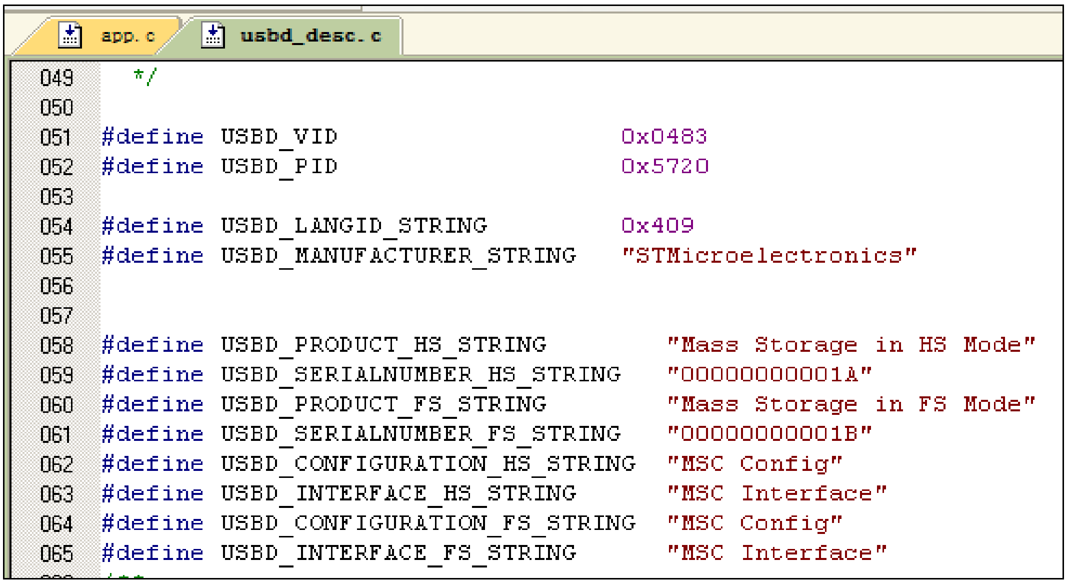
Figure 4-33 Configure start address of application

4.6.2 USB_Device_Examples\MSC

1. Description

The MSC (Mass Storage) example gives a typical example of how to use the STM32F4xx USB OTG Device peripheral to communicate with a PC Host using the bulk transfer while the MicroSD card is used as storage media. On PC, user can open, close, create, delete, copy and paste the files stored in the SD card.

MSC device information is located in usbd_desc.c, as shown below:



```
  */

#define USBD_VID                     0x0483
#define USBD_PID                     0x5720

#define USBD_LANGID_STRING           0x409
#define USBD_MANUFACTURER_STRING     "STMicroelectronics"


#define USBD_PRODUCT_HS_STRING          "Mass Storage in HS Mode"
#define USBD_SERIALNUMBER_HS_STRING     "00000000001A"
#define USBD_PRODUCT_FS_STRING          "Mass Storage in FS Mode"
#define USBD_SERIALNUMBER_FS_STRING     "00000000001B"
#define USBD_CONFIGURATION_HS_STRING    "MSC Config"
#define USBD_INTERFACE_HS_STRING        "MSC Interface"
#define USBD_CONFIGURATION_FS_STRING    "MSC Config"
#define USBD_INTERFACE_FS_STRING        "MSC Interface"
/**
```

Figure 4-34 USB MSC device information

2. Hardware Configuration

A USB A type (Male) to Micro AB (Male) cable is needed in this example.

A MicroSD card is needed in this example and *Kingston 1GB/2GB* or *SanDisk 2GB* MicroSD Card is recommended. The card should be formatted before used.

Note: *Kingston 1GB/2GB and SanDisk 2GB MicroSD Card have been tested on DevKit407. It's not guaranteed that all kind of MicroSD card work well on the board.*

3. Steps to Run

1) Insert the MicroSD card into MicroSD slot CON6.

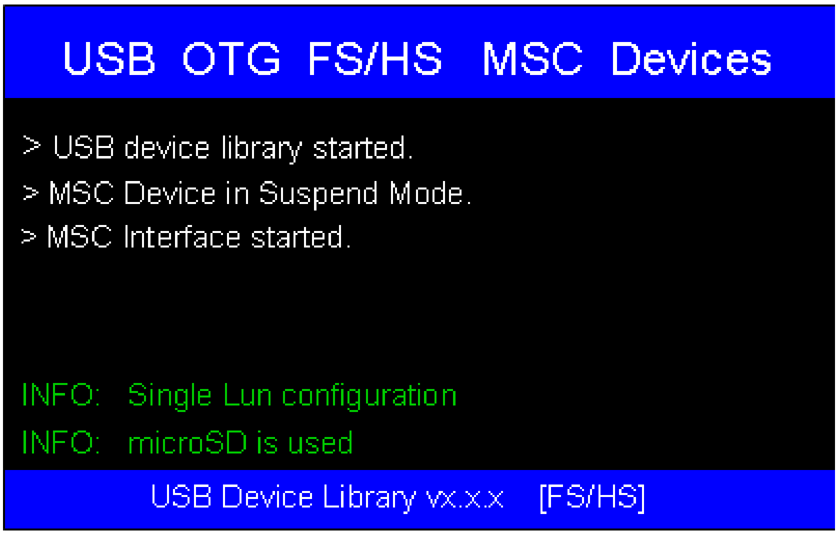
2) Connect LCD module STM32F4DIS-LCD to DevKit407 CON3 via LCD cable.

3) Connect the DevKit407 board to a PC with a 'USB type A (Male) to Micro AB (Male)' cable through USB connector CN5.

4) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.

5) Open the project, rebuild all files, load project image and then run program.

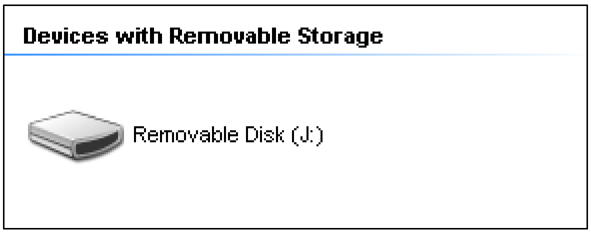
6) After reset, the LCD displays the following messages:



```
USB OTG FS/HS MSC Devices
> USB device library started.
> MSC Device in Suspend Mode.
> MSC Interface started.

INFO:   Single Lun configuration
INFO:   microSD is used
        USB Device Library vx.x.x    [FS/HS]
```

Figure 4-35 Cable connected display message

7) PC will identify the removable disk automatically. Users can use it the same as an U-disk, as shown below:



Devices with Removable Storage

Removable Disk (J:)

Figure 4-36 MSC device displayed on PC

4.6.3 USB_Device_Examples\VCP

1. Description

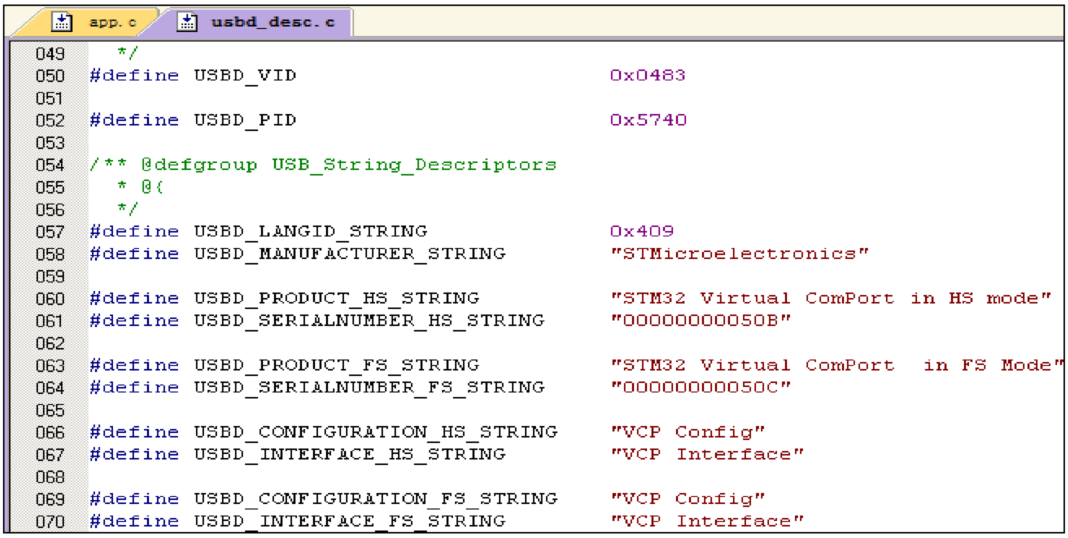
The VCP example illustrates an implementation of the CDC class following the PSTN subprotocol.

The VCP example allows the STM32 device to behave as a USB-to-RS232 bridge.

- On one side, the STM32 communicates with host (PC) through USB interface in Device mode.
- On the other side, the STM32 communicates with other devices (same host, other host, other devices…) through the USART interface (RS232).

The support of the VCP interface is managed through the ST Virtual Com Port drive

VCP device information is located in usbd_desc.c, as shown below:



```
  */
#define USBD_VID                             0x0483

#define USBD_PID                             0x5740

/** @defgroup USB_String_Descriptors
  * @{
  */
#define USBD_LANGID_STRING                   0x409
#define USBD_MANUFACTURER_STRING             "STMicroelectronics"

#define USBD_PRODUCT_HS_STRING               "STM32 Virtual ComPort in HS mode"
#define USBD_SERIALNUMBER_HS_STRING          "00000000050B"

#define USBD_PRODUCT_FS_STRING               "STM32 Virtual ComPort  in FS Mode"
#define USBD_SERIALNUMBER_FS_STRING          "00000000050C"

#define USBD_CONFIGURATION_HS_STRING         "VCP Config"
#define USBD_INTERFACE_HS_STRING             "VCP Interface"

#define USBD_CONFIGURATION_FS_STRING         "VCP Config"
#define USBD_INTERFACE_FS_STRING             "VCP Interface"
```

Figure 4-37 USB VCP device information

In order to facilitate testing, a PC plays as two host of VCP.

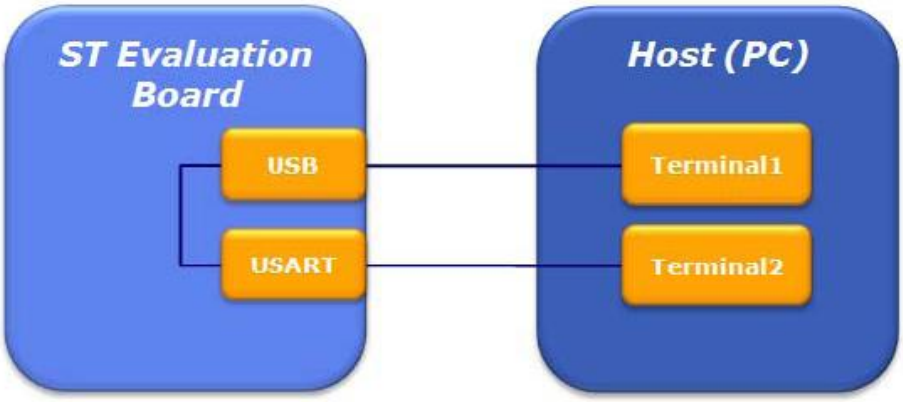


Figure 4-38 One single Host for USB and USART

2. Hardware Configuration

A USB type A (Male) to Micro AB (Male) cable is needed in this example.

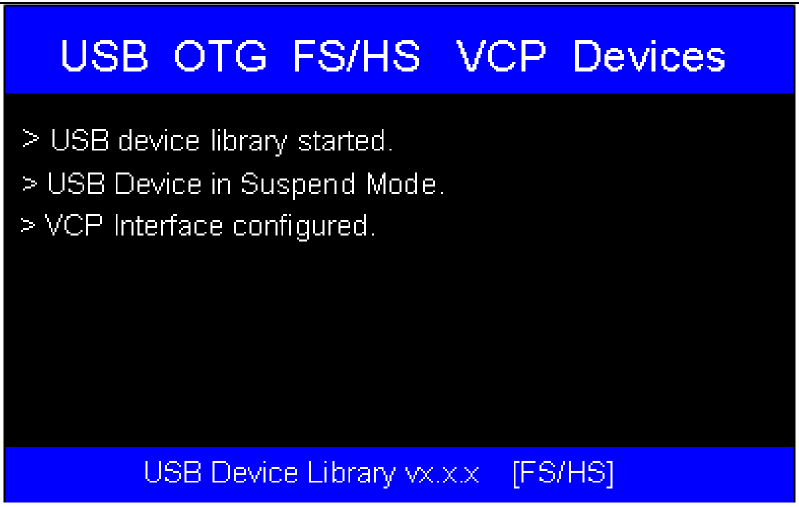
3. Steps to Run

1) Install VCP_V1.3.1_Setup.exe on the PC. The software is located CD-ROM at the following location:

   \Codes\STM32F4xx_USB_Example\Utilities\PC_Software\stm32_vcp

   If your PC is 64-bit, please install *VCP_V1.3.1_Setup_x64.exe*.

2) Connect LCD module STM32F4DIS-LCD to DevKit407 CON3 via LCD cable.
3) Connect the DevKit407 board to a PC with a 'USB type A (Male) to Micro AB (Male)' cable through USB connector CN5.
4) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.
5) Open the project, rebuild all files, load project image and then run program.
6) After reset, the LCD displays the following messages:



USB OTG FS/HS VCP Devices

> USB device library started.

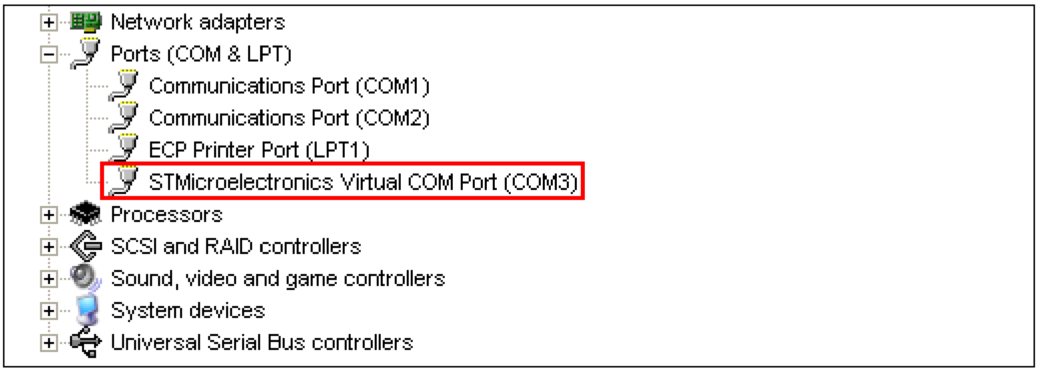
> USB Device in Suspend Mode.

> VCP Interface configured.

USB Device Library vx.x.x [FS/HS]

Figure 4-39 USB VCP device cable connected display message

7) USB device (DevKit407) is enumerated as serial communication port



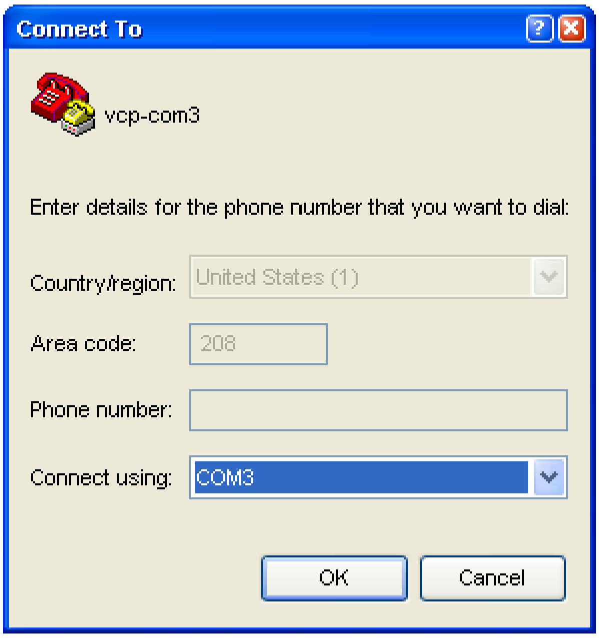
Network adapters
Ports (COM & LPT)
Communications Port (COM1)
Communications Port (COM2)
ECP Printer Port (LPT1)
STMicroelectronics Virtual COM Port (COM3)
Processors
SCSI and RAID controllers
Sound, video and game controllers
System devices
Universal Serial Bus controllers

Figure 4-40 DevKit407 have been enumerated as VCP device

8) Configure the virtual com port as below.

Start HyperTerminal by clicking on ***Start -> Programs -> Accessories -> Communications ->HyperTerminal***.

The 'Connect To' dialog box appears. Ignore the first three boxes – these are used with dial-up modem services. In the last box 'Connect using' select the COM port that you will be using and press 'OK'.



Connect To

vcp-com3

Enter details for the phone number that you want to dial:

Country/region: United States (1)

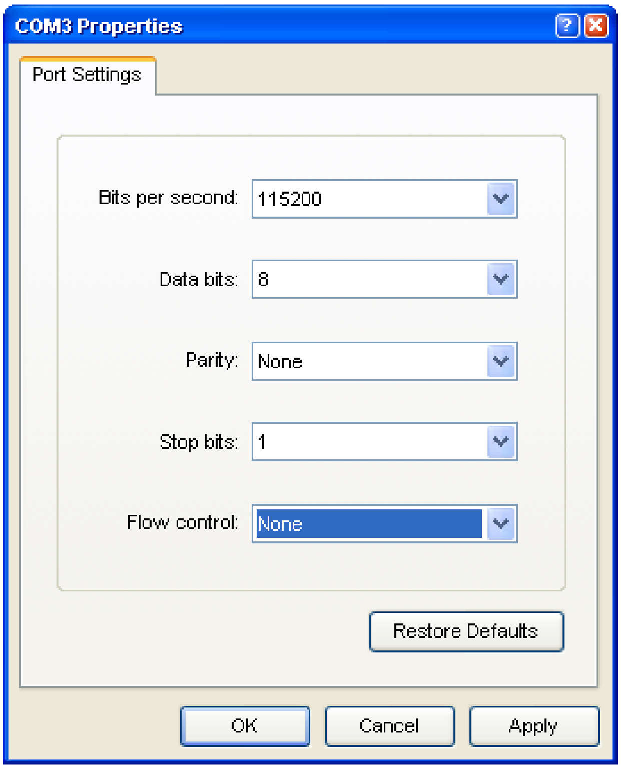
Area code: 208

Phone number:

Connect using: COM3

Figure 4-41 Create HyperTerminal for the virtual com port

In the following ‘COM properties’ dialog box you can set up the communication parameters for the COM port. Set for 115200 bits per second, 8 data bits, no parity, 1 stop bit and no flow control. Press ‘OK’ when done.



COM3 Properties

Port Settings

Bits per second: 115200

Data bits: 8

Parity: None

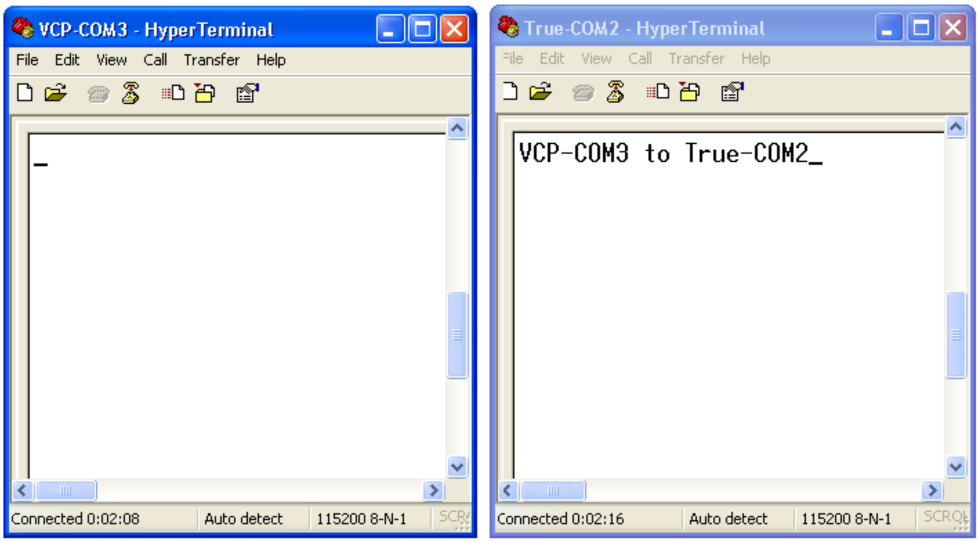
Stop bits: 1

Flow control: None

Figure 4-42 VCP port settings

9) Configure com port that connected to DevKit407 board in the same way.

10) Communication test. Try sending some characters with the HyperTerminal of virtual serial port, the other HyperTerminal (COM3) will receive these characters.

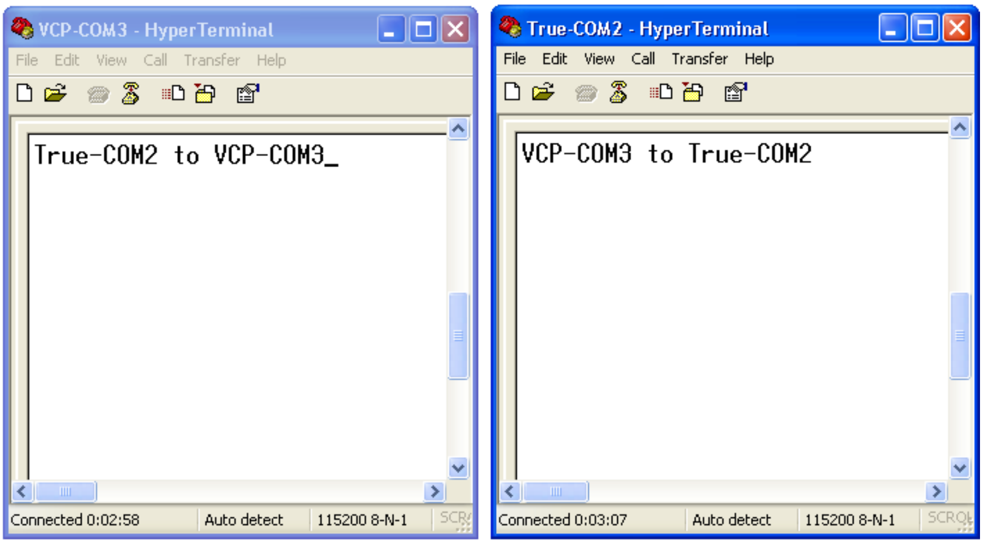


```
_
```

```
VCP-COM3 to True-COM2_
```

Figure 4-43 Message from VCP COM to True COM

Both the two HyperTerminals can send or receive data. As shown below:



```
True-COM2 to VCP-COM3_
```

```
VCP-COM3 to True-COM2
```

Figure 4-44 Message from True COM to VCP COM

USB_Host_Examples

There are two examples in USB_Host_Examples subfolder:

- HID
- MSC

HID example provides a description of how to use the USB OTG host peripheral on the STM32F2xx, STM32F4xx and STM32F105/7 devices.

MSC example provides a description of how to use the USB OTG host peripheral on the STM32F2xx, STM32F4xx and STM32F105/7 devices.

4.6.4 USB_Host_Examples\HID

1. Description

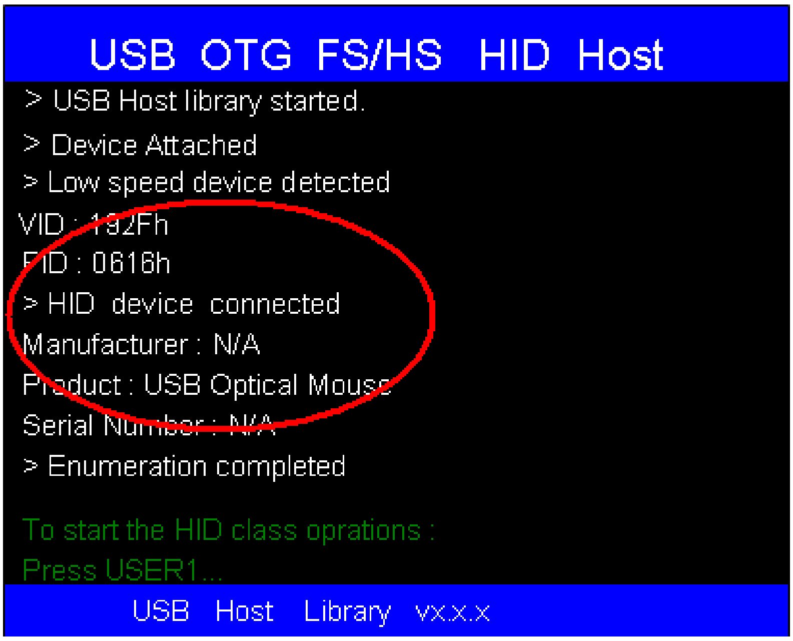
When an USB Device is attached to the Host port, the device is enumerated and checked whether it can support HID device or not, if the attached device supports HID, upon pressing the User button, the mouse or the keyboard application will be launched.

2. Hardware Configuration

A mouse and a USB type A (Female) to Micro AB (Male) cable are needed in this example.

3. Steps to Run

1) Connect LCD module STM32F4DIS-LCD to DevKit407 CON3 via LCD cable.
2) Connect the DevKit407 board to a mouse with a 'USB type A (Female) to Micro AB (Male)' cable through USB connector CN5.
3) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.
4) Open the project, rebuild all files, load project image and then run program.
5) After reset, the LCD displays the following messages:



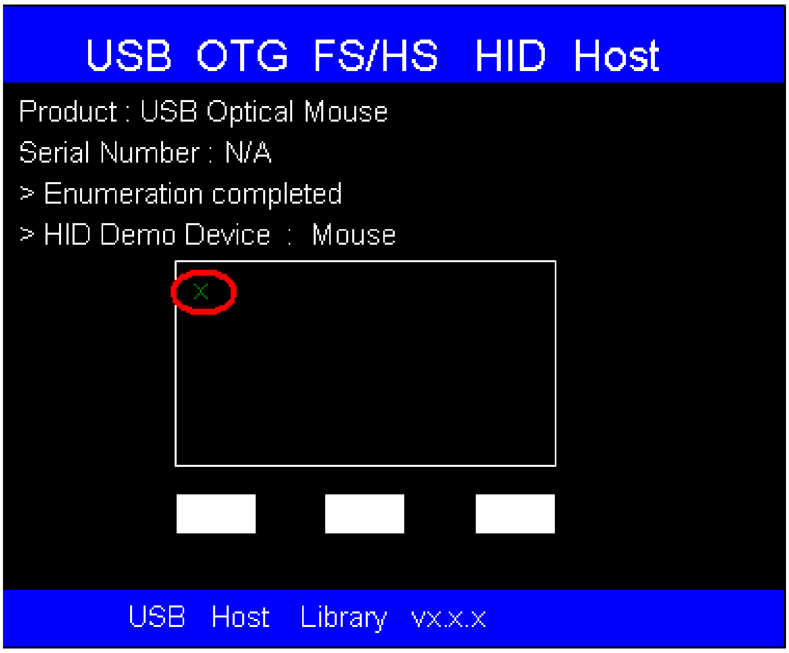
USB OTG FS/HS HID Host

```
> USB Host library started.
> Device Attached
> Low speed device detected
VID : 192Fh
PID : 0616h
> HID device connected
Manufacturer : N/A
Product : USB Optical Mouse
Serial Number : N/A
> Enumeration completed

To start the HID class oprations :
Press USER1...
```

USB Host Library vx.x.x

Figure 4-45 USB HID Host connected display message

6) When user presses the USER1 button, the application displays the mouse pointer and buttons.



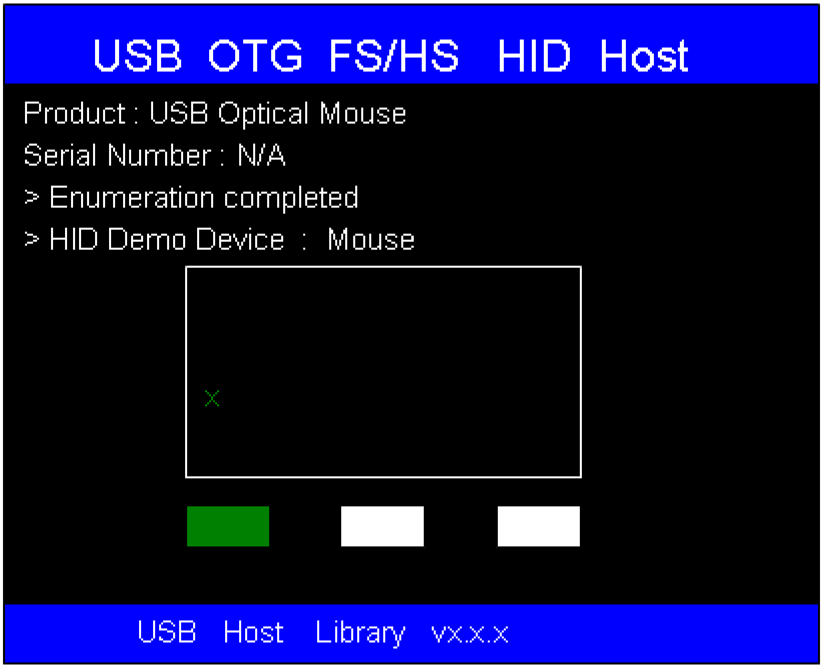
USB OTG FS/HS HID Host

```
Product : USB Optical Mouse
Serial Number : N/A
> Enumeration completed
> HID Demo Device : Mouse
```

USB Host Library vx.x.x

Figure 4-46 USB HID Host user key message

Moving the mouse will move the pointer in the display rectangle and if a button is pressed, the corresponding rectangle will be highlighted in green.



USB OTG FS/HS HID Host

Product : USB Optical Mouse

Serial Number : N/A

> Enumeration completed

> HID Demo Device : Mouse

Figure 4-47 USB HID Host user key pressed

4.6.5 USB_Host_Examples\MSC

1. Description

The STM32F4 behave as a mass storage Host that can enumerate, show content and display the supported BMP image in the attached USB flash disk.

2. Hardware Configuration

A USB Flash Disk and a USB type A (Female) to Micro AB (Male) cable are needed in this example.

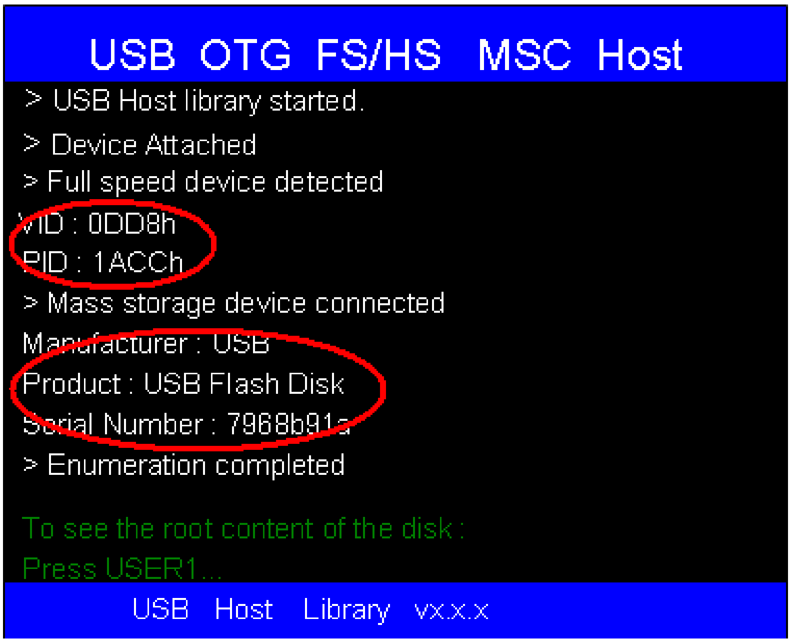
Note: *Kingston 1GB/2GB and SanDisk 4GB USB Flash Disk have been tested on DevKit407. It's not guaranteed that all kind of USB Flash Disk work well on the board.*

3. Steps to Run

1) There are some BMP files for testing purpose located in the following location:

   \Codes\STM32F4xx_USB_Example\Utilities\Binary\Media

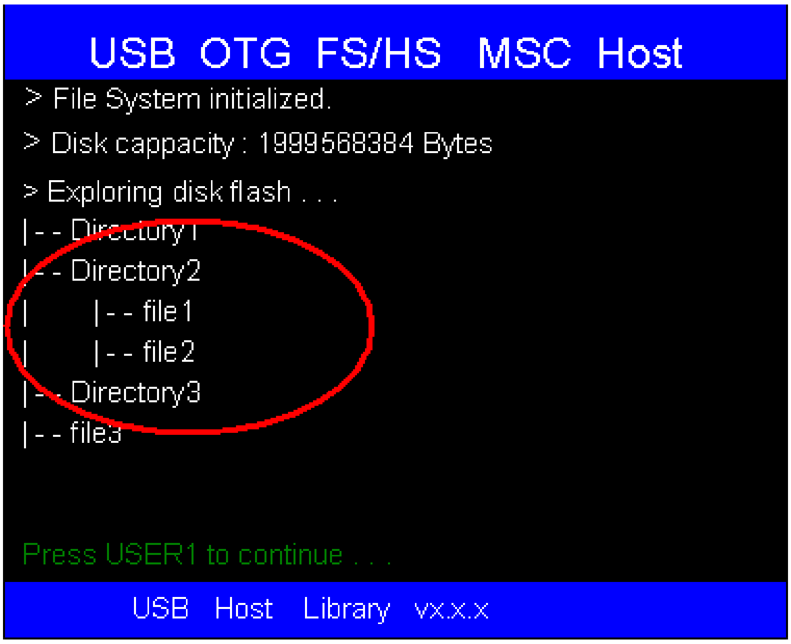
   Copy these files to the root of the USB flash disk
2) Connect the DevKit407 board to the USB flash disk with a 'USB type A (Female) to Micro AB (Male)' cable through USB connector CN5.
3) Connect LCD module STM32F4DIS-LCDto DevKit407 CON3 via LCD cable.
4) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.
5) Open the project, rebuild all files, load project image and then run program.
6) After reset, the LCD displays the following messages:



```
USB  OTG  FS/HS   MSC  Host
> USB Host library started.
> Device Attached
> Full speed device detected
VID : 0DD8h
PID : 1ACCh
> Mass storage device connected
Manufacturer : USB
Product : USB Flash Disk
Serial Number : 7968b91a
> Enumeration completed

To see the root content of the disk :
Press USER1...
          USB   Host   Library   vx.x.x
```

Figure 4-48 USB mass storage host display message

7) When the user press the User button, the application explore the USB flash disk content and the LCD displays the following messages:

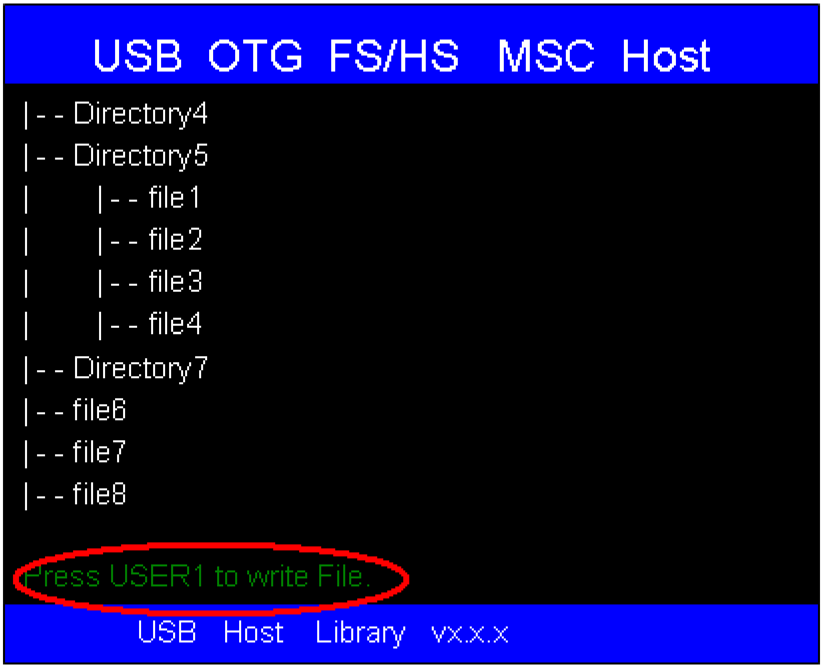


```
USB  OTG  FS/HS   MSC  Host
> File System initialized.
> Disk cappacity : 1999568384 Bytes
> Exploring disk flash . . .
|- - Directory1
|- - Directory2
|        |- - file1
|        |- - file2
|- - Directory3
|- - file3


Press USER1 to continue . . .
          USB   Host   Library   vx.x.x
```

Figure 4-49 USB mass storage explorer display message

Note: *The contents circled by red color depend on the USB device that plugged in.*

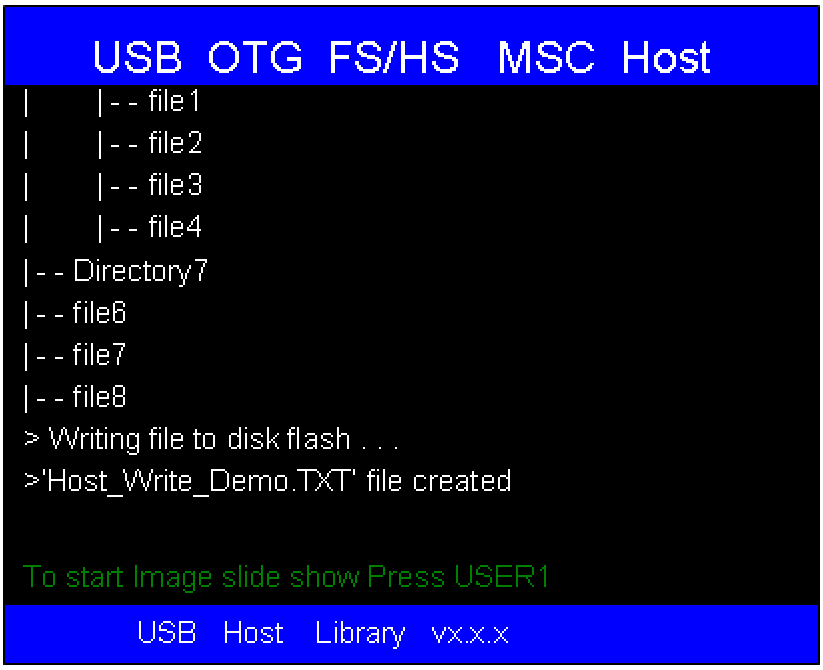
8) User has to press the User button to display the whole disk (recursion level 2). Below is a screenshot when the entire flash disk is shown:



```
USB  OTG  FS/HS   MSC  Host
|-- Directory4
|-- Directory5
|       |-- file1
|       |-- file2
|       |-- file3
|       |-- file4
|-- Directory7
|-- file6
|-- file7
|-- file8

Press USER1 to write File.
USB   Host   Library   vx.x.x
```

Figure 4-50 USB mass storage explorer display message (last screen)

9) The user has to press the User button to write a small file, e.g. Host_Write_Demo.txt (less to 1 KB) on the disk.



```
USB  OTG  FS/HS   MSC  Host
|       |-- file1
|       |-- file2
|       |-- file3
|       |-- file4
|-- Directory7
|-- file6
|-- file7
|-- file8
> Writing file to disk flash . . .
>'Host_Write_Demo.TXT' file created

To start Image slide show Press USER1
USB   Host   Library   vx.x.x
```

Figure 4-51 USB mass storage write file display message

10) After writing the file to the disk, user can press the USER1 button to start the Image slide show. Only the BMP files with the following format are supported :

- Width: 320
- Height: 240

- BPP: 16
- Compression: RGB bitmap with RGB masks

Press the User button to start the Image (BMP file for testing) slide show:

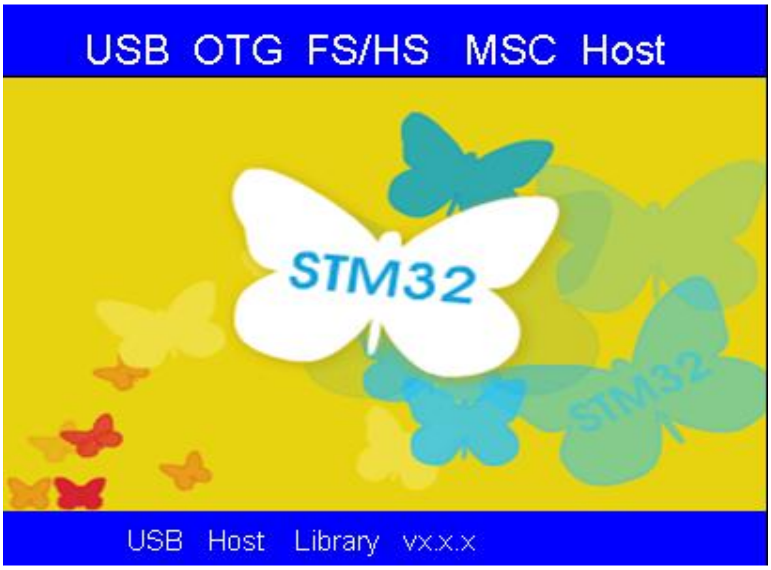


Figure 4-52 USB mass storage slideshow example

Note: *BMP files should be located in the USB Disk root.*

Chapter 5 Applications Examples

5.1 STM32F4xx_uCOSII_Example

1. Description

This example provides a description of how to use the uC/OS-II on the STM32F4xx devices.

Note: *This example just supplies port files. You should download uC/OS-II-v2.91 source code from the Micrium website: http://micrium.com/page/downloads/source_code. Copy uC/OS-II-v2.91 source from the Micrium folder(Micrium\Software\uCOS-II\Source) to Project folder(STM32F4xx_uCOSII_Example\uCOS-II\Source).*

Two tasks work in this example:

- App_TaskStar--Control the LED blinking
- App_TaskKbd--Control the LED blinking frequency by User button pressing

The MDK project of this example is located in the following folder:

```
\Codes\STM32F4xx_uCOSII_Example\Project\MDK-ARM
```

2. Steps to Run

1) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.
2) Open the project, rebuild all files, load project image and then run program.
3) Testing actions and results:
   - LEDs turn on and then turn off in an order: LED4 -> LED6 -> LED5 -> LED3.
   - Long press on User button, then the blinking speed of the LEDs will be lower than what we saw at the beginning.

5.2 STM32F4xx_FPU_FFT_Example

1. Description

This example shows how to use the Cortex-M4's FPU unit.

This example demonstrates the calculation of the maximum energy bin in the frequency domain of the input signal with the use of Complex FFT, Complex Magnitude, and Maximum functions.

The MDK project of this example is located in the following folder:

```
\Codes\STM32F4xx_FPU_FFT_Example\Project\FPU_FFT_Example\MDK-ARM
```

2. Steps to Run

1) Connect the DevKit407 board to a PC with a ‘USB type A (Male) to Mini-B (Male)’ cable through USB connector CN1 to power the board.
2) Open the project, rebuild all files, load project image and then run program.
3) Testing actions and results:
   - ➢ If the demon run ok, the LED6 will turn on, otherwise the LED3 Will turn on.

Chapter 6 Other Test Scenarios

This Chapter gives a short description about how to test the DevKit407 peripheral

6.1 USART Testing

Please reference at 4.1 USART Example.

6.2 SDIO Testing

Please reference at 4.2 SDIO Example.

6.3 LCD Testing

Please reference at 4.3 LCD Example.

6.4 DCMI Testing

Please reference at 4.4 DCMI Example.

6.5 Ethernet Testing

Please reference at 4.5 ETH_LwIP Example.

6.6 USB Testing

Please reference at 4.6 USB Example

Technical Support & Warranty Service

Embest Technology Co.,LTD., established in March of 2000, is a global provider of embedded hardware and software. Embest aims to help customers to reduce time to market with improved quality by providing the most effective total solutions for the embedded industry. In the rapidly growing market of high end embedded systems, Embest provides comprehensive services to specify develop and produce products and help customers to implement innovative technology and product features. Progressing from prototyping to the final product within a short time frame and thus shorten the time to market, and to achieve the lowest production costs possible. Embest insists on a simple business model to offer customers high-performance, low-cost products with the best quality and service. The content below is important information for our products technical support and warranty service:

Technical support service

Embest provides one year free technical support service for all products. Technical support service covers:

- Embest embedded platform products software/hardware materials
- Assist customers in compiling and running the source code we offer.
- Assist in troubleshooting on our embedded software/hardware platforms given that the customer has followed the documentation that we offer.
- Judge whether the product failure exists.

The services listed below are not included in the range of our free technical support service, and Embest will handle each situation with discretion:

- Software/Hardware issues a user meets during the development process

- Issues which occur when users compile/run the embedded OS which has been modified by users themselves.
- User's own applications.
- Problems which occur during the modification of our software source code

Maintenance Service Clause

1) The products except LCD, which are not used properly, will take the warranty since the day of the sale:

   PCB: Provide 12 months free maintenance service.

2) The situations listed below are not included in the range of our free maintenance service, Embest will charge the service fees with discretion:

   a) Can't provide valid Proof-of-Purchase, the identification label is torn up or illegible, the identification label is altered or doesn't accord with the actual products;

   b) Don't follow the instruction of the manual in order to damage the product;

   c) Due to the natural disasters (unexpected matters), or natural attrition of the components, or unexpected matters leads to the defects of appearance/function;

   d) Due to the power supply, bump, leaking of the roof, pets, moisture, impurities into the boards, all those reasons which lead the defects of appearance/function;

   e) User unauthorized weld or dismantle parts leads the product's bad condition, or let other people or institution which are not authorized by Embest to dismantle, repair, change the product leads the product bad connection or defects of appearance/function;

   f) User unauthorized install the software, system or incorrect configuration or computer virus leads the defects;

   g) Purchase the products through unauthorized channel;

h) Those commitments which is committed by other institutions should be responsible by the institutions, Embest has nothing to do with that;

3) During the warranty period, the delivery fee which delivery to Embest should be covered by user, Embest will pay for the return delivery fee to users when the product is repaired. If the warranty period is expired, all the delivery fees will be charged by users.

4) When the board needs repair, please contact technical support department.

Note: *Those products are returned without the permission of our technician, we will not take any responsibility for them.*

Basic Notice for Protecting LCD Screen

1) Do not use finger nails or hard/sharp objects to touch the surface of the LCD, otherwise the above services will be void.
2) Embest recommend user to purchase a screen wipe to wipe the LCD after long time use, please avoid cleaning the surface with fingers or hands to leave fingerprint.
3) Do not clean the surface of the screen with chemicals, otherwise user can not enjoy above service.

Note: *Embest do not supply maintenance service to LCD. We suggest the customer first check the LCD after getting the goods. In case the LCD cannot run or show display, customer should inform Embest within 7 business days from the moment of getting the goods.*

Value Added Services

We will provide following value added services:

- Provide services of driver development based on Embest's embedded platforms, like serial port, USB interface devices, LCD screen.
- Provide the services of control system transplant, BSP driver development, API software development.
- Other value added services like power adapter, LCD parts.
- Other OEM/ODM services.
- Technical training.

Please contact Embest to get technical support:

- Support Tel:+86-755-25503401
- Fax:+86-755-25616057
- Pre-Sale consultation: market@embedinfo.com
- After-Sale consultation: support@embedinfo.com