MYD-LPC1788 User Manual

Version V1.3

MYIR TECH LIMITED

www.myirtech.com

Version History

Version	Description	Time
Number		
V1.0	Initial Version	2012.09.14
V1.1	Adjust the list of product configuration	2012.11.02
V1.2	Add 7.0-inch screen support)	2013.02.22
V1.3	modify the contact information	2013.03.28

MYIR TECH LIMITED

Directory

Chapter 1 Product Overview	1
1.1 Product Description	1
1.2 Product Preview	1
1.3 Product Features	2
1.4 Product Configuration	4
Chapter 2 Hardware Resource Introduction	5
2.1Hardware Resource Overview	5
2.2 Main Module Introduction	6
2.2.1 Main processor LPC1788	6
2.2.2 SDRAM Module	6
2.2.3 SRAM Module	7
2.2.4 NORFLASH Module	8
2.2.5 SPI FLASH Module	9
2.2.6 EEPROM Module	10
2.2.7Ethernet MAC Module	11
2.2.8Audio Module	12
2.2.9 Touch Controller Module	13
2.2.10User Key and Reset Circuit	14
2.2.11 Buzzer	14
2.2.12 LED	15
2.3 Peripheral Interface introduction	15
2.3.1 UART Interface	15
2.3.2 CAN Interface	16
2.3.3 RS485 Interface	16
2.3.4 SDCARD Interface	17
2.3.5 USB OTG/HOST interface	17
2.3.6 JTAG interface	19

2.3.7 LCD Touch Screen Interface19
2.3.8 User Interface20
2.3.9 ADC and DAC Interface20
2.4Jumper setting21
Chapter 3 MDK Routine23
3.1 Overview23
3.2 Preparation23
3.2.1 Configure and CompileMDK Routine23
3.2.2 DebugMDK Routine26
3.2.3Download programby ULINK228
3.2.4 ISP Download31
3.3MDK source use
3.3.1 Directory structure35
3.3.2 Add function module36
3.3.3Use Printf
3.4 The introduction of MDK routine
3.4.1 ADC_Interrupt
3.4.2 ADC_Polling40
3.4.3 CAN_Test41
3.4.4 Crc_Demo42
3.4.5 Dac_Dma43
3.4.6 Dac_SineWave44
3.4.7 DMA_Flash2Ram44
3.4.8 Eeprom_Demo45
3.4.9 Emc_NorFlashDemo46
3.4.10 Emc_SdramDemo47
3.4.11 Emc_SramDemo47
3.4.12 GPIO_Interrupt48
3.4.13 GPIO_LedBlinky49

3.4.14 Nvic_VectorTableRelocation49
3.4.15 Pwm_SingleEdge50
3.4.16 Pwm_DualEdge50
3.4.17 Pwm_MatchInerrupt51
3.4.18 PWR_Sleep52
3.4.19 PWR_DeepSleep53
3.4.20 Emac_EasyWeb53
3.4.21 Rtc_Alarm54
3.4.22 SSP_Touchscreen55
3.4.23 SSP_Flash56
3.4.24 Systick_100msBase57
3.4.25Timer_MatchInterrupt57
3.4.26 Wdt_Interrupt58
3.4.27 Wdt_Reset59
3.4.28 Lcd_LQ043T3DX0A60
3.4.29Lcd_touch60
3.4.30 Mci_CidCard60
3.4.31 Usb_MassStorage61
3.4.32 Usb_VirtualCom62
3.4.33 I2S_Audio63
3.4.34 I2C_Eeprom64
3.4.35 RS_485-Master&Slave65
Appendix 1sales FAQ and technical support67

Chapter 1 Product Overview

1.1 Product Description

MYIR have latest lunched MYD-LPC1788 board which is based on Cortex-M3 kernel. The Cortex-M3 is a next generation core that offers better performance than ARM7 at the same clock rate, and offers other system enhancements such as modernized debug features and a higher level of support block integration. The processor has 512KB FLASH memory, 96KB on-chip SRAM and 4KB EEPROM. It also has external SD Card interface, USB Host/Device/OTG interface, CAN interface. RS485 interface, Audio input, Ethernet MAC, LCD interface, JTAG interface, function key and so on. It has been widely used in industrial control and medical system.



1.2 Product Preview

Figure 1-1

1.3 Product Features

Based on Cortex-M3 processor, MYD-LPC1788 integrates all the chip functions and

features. The main features are as follows:

Electrical parameters

- ➢ Operating Temperature : -40°C~85°C
- Electrical Specifications: +5V power supply
- Mechanical Dimensions: 115mmx90mm

Processor

- LPC1788(Cortex-M3 kernel), runs at up to 120MHz
- > 96KB on-chip SRAM
- ➢ 512KB on-chip ROM
- ➢ 4KB on-chip EEPROM

Memory

- 32MB SDRAM
- > 1MB SRAM
- > 2MB NORFLASH
- > 256B EEPROM
- > 4MB SPI FLASH

Audio and Video Interface

- An Audio 3.5mm Input Interface
- > A Two-channel Audio 3.5mm Output Interface

LCD Touch-Screen Interface

- 24 True Color
- Resolution: Support up to 1024 x 768

Transmission Interface

- > One serial(select UART0/UART2 by jumper)
- > One high-speed USB HOST interface
- > One mini USB interface

- > One Ethernet MAC.
- Two CAN Interface
- > One RS485 Interface

Input interface

- Standard JTAG Interface
- MicroSD Card Interface

LED indicator

One system power indicator(red)

Applications

- Communications
 - Point-of-sale terminals, Web servers, multi-protocol bridges

Industrial/Medical

 Automation controllers, application control, robotic controls, HVAC, PLC, inverters, circuit breakers, medical scanning, security monitoring, motor drive, video intercom.

Consumer/Appliance

• Audio, MP3 decoders, alarm systems, displays, printers, scanners, small appliances, fitness equipment

Automotive

• Aftermarket, car alarms, GPS/Fleet Monitor

1.4 Product Configuration

NO	Name	Number	Note
1	MYD LPC1788 Development Board	1	
2	1.5 Meters Crossover Cable	1	
3	1.5 Meters high-speed Mini USB 2.0 Cable	1	
4	9Pin to 9Pin serial cable	1	
5	DVD Product	1	Include Schematic (PDF), User Manual, Source Code, etc.
6	4.3/7.0 Inch LCD Touch Screen	1	optional

Table 1-1

Chapter 2 Hardware Resource

Introduction

2.1Hardware Resource Overview

MYD-LPC1788 resources are shown in table2-1:

Item	Feature			
Size	Board size:115mm x 90mm			
CPU	LPC1788(Cortex-M3 Core), Up to 120MHz			
	On-chip: 96KB	SRAM,512KB RO	M, 4KB EEPROM	
Memory	External: 32MB SDR/	AM, 1MB SRAM, 2	MB NORFLASH, 256B	
	EEF	PROM, 4MB SPI F	LASH	
Debug	20 Pin, 2	.54mm JTAG debu	ıg interface	
	Туре	Quantity	Description	
	RS485	1	Support RS485	
	Ethernet	1	100Mbps,DP83848	
	CAN	2	Support CAN	
			Support USB	
	USB	2	HOST/Device 2.0	
5			USBOTG 2.0	
Peripheral	Audio	2	Audio in/out	
	SD interface	1	SD/MMC interface	
	Extension	2	Extension for customer	
	ITA O		Standard 20 pin JTAG	
	JIAG	1	interface	
			Support 4.3/7.0 inch	
	LCD interface 1		touch screen	
Button	User button	3	SW1, SW2, SW3	
Bullon	Reset	1	SW5	

Power

5V/2A

Table 2-1

2.2 Main Module Introduction

2.2.1 Main processor LPC1788

Based on ARM Cortex-M3 processor, LPC1788 is applied in high level of support block integration and low-power embedded product. The processor has LCD controller, 10/100 Mbps Ethernet MAC, high-speed USB Device/Host/OTG controller, CAN controller, SPI, SSP, IIC, IIS and external EMC. It is used in industrial control and medical system specially.

2.2.2 SDRAM Module

SDRAM chooses K4S561632H. Its characteristics are as follows:

- > Auto refresh
- > 64ms refresh cycles

SDRAM circuit is shown in figure 2-1:



SDRAM

Figure 2-1

2.2.3 SRAM Module

SRAM choose IS61LV51216, Its characteristics are as follows:

- High-speed access time: 8,10, and 12ns
- > CMOS low power operation
- Low stand-by power
 - Less than 5mA
- > Fully compatible operation: no clock or refresh required
- > Three state outputs
- > Data control for upper and lower bytes
- > Industrial temperature available

SRAM circuitis show in figure 2-2:



SRAM

Figure 2-2

2.2.4 NORFLASH Module

NORFLASH chooses SST39VF1601. Its characteristics are as follows:

- Superior reliability
 - Endurance:100000 Cycles(Typical)
 - Greater than 100 years Data Retention
- Low Power Consumption(typical values at 5 MHz)
 - Active Current:: 9mA
 - Standby Current: 3µA
- Security-ID Feature
 - SST: 128bit; User: 128bit
- > Fast Read Access Time:70ns, 90ns

NORFLASH circuit is shown in figure 2-3:



NOR FLASH

Figure 2-3

2.2.5 SPI FLASH Module

SPI FLASH chooses AT25DF321A. Its characteristics are as follows:

- > Operating Frequencies: up to 85 MHz
- Flexible program: support Byte/Page program (1~256 Bytes)
- Fast program and erase times
 - 1.0ms typical page program(256 Bytes) Time
 - 50ms typical 4-Kbyte block erase time
 - 250ms typical 32-Kbyte block erase time
 - 400ms typical 64-Kbyte block erase time
- Low power dissipation
- > Endurance: 100000 program/erase cycles

Data retention: 20 years

SPI FLASH circuitis shown in figure 2-4:



SPI FLASH



2.2.6 EEPROM Module

EEPROM chooses AT24C02. Its characteristics are as follows:

- Two-wire Serial Interface
- Bidirectional Data Transfer Protocol
- High-reliability
 - Endurance: 1 Million Write Cycles
 - Data Retention: 100 Years
- > 100 kHz (1.8V) and 400 kHz (2.7V, 5V) Compatibility
- Schmitt Trigger, Filtered Inputs for Noise Suppression

EEPROM circuit is shown in figure2-5:



Figure 2-5

2.2.7Ethernet MAC Module

Ethernet MAC Module chooses DP83848. Its characteristics are as follows:

- ► Low-power 3.3V, 0.18µm CMOS technology
- MII Serial Management Interface
- > IEEE 802.3u Auto-Negotiation and Parallel Detection
- > IEEE 802.3u ENDEC, 10BASE-T transceivers and filters

Ethernet Mac circuit is shown in figure 2-6:



Figure 2-6

2.2.8Audio Module

Audio module chooses UDA1380. Its characteristics are as follows:

- Slave BCK and WS signals
- IIS Bus format
- Multiple format data output interface
- Multiple format data output interface
- ADC front-end features
- DAC features

UDA1380circuitis shown in figure 2-7:



Figure 2-7

2.2.9 Touch Controller Module

Touch controller module chooses TSC2046. Its characteristics are as follows:

- Internal 2.5V reference
- Touch-pressure measurement
- Auto power down

TSC2046circuit is shown in figure 2-8:



Figure 2-8

2.2.10User Key and Reset Circuit

User Key and Reset circuit is shown in figure 2-9 and 2-10:



Figure 2-9



Figure 2-10

2.2.11 Buzzer

Buzzer circuit is shown in figure 2-11:



Figure 2-11

2.2.12 LED

LED circuit is shown in figure 2-12:





2.3 Peripheral Interface introduction

2.3.1 UART Interface

UART circuit is shown in figure 2-13:





2.3.2 CAN Interface

CAN interface chooses TJA1040. Its characteristics are as follows:

- Fully compatible with the ISO 11898 standard
- High speed (up to 1 MBaud)
- Very low ElectroMagnetic Emission
- > Differential receiver with high common-mode range for ElectroMagnetic Immunity (EMI)
- Input levels compatible with 3.3 V and 5 V devices
- > At least 110 nodes can be connected
- Thermally protected

CAN circuit is shown in figure2-14:



Figure 2-14

2.3.3 RS485 Interface

RS485 choose SP3485. Its characteristics are as follows:

- ➢ RS-485 and RS-422 transceivers
- Interoperable with 0.5V logic
- Driver/Receiver enable
- Low power shutdown mode
- > -7V to 12V common-mode input voltage range
- > Allows up to 32 transceiver on the serial bus
- Compatibility with the industry standard 75176pinout
- Driver output short-circuit protection

RS485circuit is shown in figure 2-15:



Figure 2-15

2.3.4 SDCARD Interface

SD card circuit is shown in figure 2-16:





2.3.5 USB OTG/HOST interface

USB OTG chooses ISP1301. Its characteristics are as follows:

> Can transmit and receive serial data at both full-speed (12 Mbit/s) and low-speed

(1.5 Mbit/s) data rates

> Supports various serial data interface protocols; transparent general-purpose buffer mode allows you to control the direction of data transfer

- > Contains Host Negotiation Protocol (HNP) command and status registers
- > Supports serial I2C-bus™ interface for OTG status and command controls
- Supports external charge pump
- > Full industrial grade operation from 40 ℃to 85 ℃

USB OTG circuitis shown in figure 2-17:



Figure 2-17

USB HOST chooses TPS2041. Its characteristics are as follows:

- > 50mA continuous current
- Short-circult and thermal protection with overcurrent logic output
- Undervoltage lockout
- Maximum standby supply current: 1μA (single, Double), or 2μA (Three, Four)
- Bidirectional switch
- ➤ Ambient temperature range: -40°C to 85°C
- ESD protection

USB HOSTcircuitis shown in figure 2-18:



Figure 2-18

2.3.6 JTAG interface

JTAG circuit is shown in figure 2-19:



Figure 2-19

2.3.7 LCD Touch Screen Interface

LCD Interface is shown in figure 2-20:



2.3.8 User Interface

User interface circuit is shown in figure 2-21:



Figure 2-21

2.3.9 ADC and DAC Interface

ADC and DAC interface circuit is shown in figure 2-22:



Figure 2-22

2.4Jumper setting



Figure 2-23

Ne	Function Description			
NO	Connect Disconnect			
JP1 ^[1]				
JP2 ^[1]	Connect 1-2: connect USB OTG, LCDisn't available. Disconnect 2-3: connect LCD, USB OTG isn't available.			
JP3 ^[1]				
JP4	Connect 1-2: serial choose UART0 and output from JP11			
JP5	Connect 2-3: Serial chooses UART2 and output from JP11			
JP6 ^[2]	ISP download module Normal download module			
JP7	Connect 1-2: Enable RS485 to write and read. When use RS485, it needs to be connected			
Connect 2-3: Enable USB Host power detection				

Table 2-2

Note: [1] INT,SCL,SDA in USB OTG share with LCDVD21、LCDVD22、LCDVD23 in LCD,

So at the same time, it can only choose one.

[2] Connect JP6 when download program in ISP module. After download program, reset board, program start to run.

Chapter 3 MDK Routine

3.1 Overview

MDK routines are naked programs without operating system and its development tool is MDK-ARM 4.53. This chapter describes how to use and writetest procedures. The contents include:

(1) MDK development environment to be built and configured;

- (2) MDK sample program debugged, compiled and downloaded;
- (3) The test procedures: functions, usage and phenomenon descriptions.

MDK routines cover a wide range of programs, including DMA, ADC, LCD, Memory, Ethernet MAC and so on. User can make a second development on these examples, which can shorten developmentcycle.

3.2 Preparation

(1) Install MDK-ARM (Version 4.53) development tool and license and then Prepare for MYD-LPC1788 board.

(2) Set serial:Baud Rate: 115200; Data Bits: 8; Parity Bit: None; Stop Bit: 1. Data flow: None.

3.2.1 Configure and CompileMDK Routine

Firstly, find 05-MDK_Source\01-ADC\ADC_Interrupt\Project folder and double click project, then configure project. Steps are as follows (Noted, default project setting can made download successfully,please recheck if program compile or download):

(1) Select"**Option for target FLASH**" or press Alt + F7.The Setting window is shown in figure 3-1:

🛛 Options for Target 'FLASH'								
Device Target	Output List	ing User C/C	C++ Asm	Link	er Debug Ut	ilities	1	
NXP (founded by Philips) LPC1788 Code Generation Operating system: None System-Viewer File (.Sfr):								
Read/Only Me	rmory Areas hip Start	Size	Startup	-Read/V default	/rite Memory Area	as Start	Size	Nolnit
RON	/1:		C		RAM1:			
RON	/12:	- Î	0		RAM2:			
RON	//3:		0		RAM3:			
on-c IRON	hip /1: 0×0 /2:	0x80000	•	▼ □	on-chip IRAM1: 0x100 IRAM2: 0x200	00000	0x10000 0x8000	
		OK			Defaulto	1		Heln

Figure 3-1

(2) Choose LPC1788 in Device. Refer to figure 3-2:

- X 🔣 Options for Target 'FLASH' Device Target Output Listing User C/C++ Asm Linker Debug Utilities Database: Generic CPU Data Base \mathbf{v} Vendor: NXP (founded by Philips) Device: LPC1788 Toolset: ARM D LPC1776 ARM Cortex-M3 processor: . ۰ - CI LPC1777 -running at frequencies of up to 100 MHz. Memory Protection Unit (MPU).
 Nested Vectored Interrupt Controller (NVIC).
 Non-maskable Interrupt (NMI) input.
 Wakeup Interrupt Controller (WIC). 🖾 LPC1778 Ξ 🖾 LPC1785 🖸 LPC1786 - CI LPC1787 - Up to 512 kB on-chip Flash (ÌSP and IAP capabilities). CI 📭 - Up to 96 kB on-chip SRAM 🖾 LPC1810 Ξ - Up to 4 kB on-chip EEPROM. 🛄 LPC1820 External Memory Controller (EMC).
 DMA controller (GPDMA). 🖾 LPC1830 - JTAG interface, Serial Wire Debug, and Serial Wire Trace Port options. 🖾 LPC1850 - Four reduced power modes: Sleep, Deep-sleep, Power-down, Deep power-do LPC2101 LPC2102 Clocks: LPC2103 - On-chip crystal oscillator (operating range of 1 MHz to 25 MHz). CI LPC2104 - 🛄 LPC2104/01 . Cancel Defaults OK Help
 - Figure 3-2
 - (3) Output options (include intermediate file). Refer to figure 3-3:

Options for Target 'FLASH'
Device Target Output Listing User C/C++ Asm Linker Debug Utilities
Select Folder for Objects Name of Executable: ADC_Interrupt
 Create Executable: .\Flash\ADC_Interrupt Debug Information Create Batch File Create HEX File Browse Information
C Create Library: .\Flash\ADC_Interrupt.LIB
OK Cancel Defaults Help

Figure 3-3

(4) Set Linker. Refer to figure 3-4:

😗 Options for	Target 'FLASH'			
Device Targ	et Output Listing User C/C++ A	Asm Linker Debu	ug Utilities	
🗸 Use Mem	ory Layout from Target Dialog			
Make	RW Sections Position Independent	R/O Base:	0x0000000	
Make	RO Sections Position Independent	R/W Base	0×10000000	-
Don't:	Search Standard Libraries t 'might fail' Conditions as Errors	disable Warnings:		
Scatter File				Edit
Misc controls				* •
Linker control string	-cpu Cortex-M3 *.o -library_type=microlib -str -summary_stderr -info summarysizes -map - -info sizes -info totals -info unused -info vene	ict—scatter ".\Flash\ADC -xref—callgraph—symbo eers	∑Interrupt.sct" Is	• •
	OK	Cancel Def	aults	Help

- Figure 3-4
- (5) Choose project->Rebuild all target files project, or click on shortcut icon to compile.

Refer to figure 3-5:

•
•
Alt+F7
F7
17

Figure 3-5

3.2.2 DebugMDK Routine

The following is MDK program configuration and it has a hardware emulator ULink2 in advance. (If need it, please contact the company to purchase it)

(1) After opening project, open setting dialog box and select Debug. Refer to figure

```
3-6:
```

😗 Options for Target 'FLASH'	
Device Target Output Listing User C/C++ Asm	Linker Debug Utilities
C Use Simulator	
Load Application at Startup Initialization File:	✓ Load Application at Startup ☐ Run to main() Initialization File:
Restore Debug Session Settings Breakpoints Watch Windows & Performance Analyzer Memory Display	Restore Debug Session Settings
CPU DLL: Parameter: SARMCM3 DLL -MPU	Driver DLL: Parameter: SARMCM3.DLL -MPU
Dialog DLL: Parameter: DARMP1.DLL -pLPC1788	Dialog DLL: Parameter: TARMP1.DLL -pLPC1788
OK Can	cel Defaults Help

Figure 3-6

(2) Check hardware emulator ULink2

When connecting ULink2 to board, the indicator lights of RUN and COM change blue

and then turn off, while USB indicator lights change red and then remain the same. Thus, it indicates ULink2 is no problem.

(3) Clicking Setting in figure 3-6, there will be connection status of ULink2 (choose SYSRESETREQ in Reset) and development board, as well askernel identification. Refer to figure 3-7:

Cortex-M Target Driver Setup					X
Debug Trace Flash Download					
ULINK USB - JTAG/SW Adapter	-SW Dev	vice			
Serial No: <u>v0000145</u>		IDCODE	Device Name		Move
ULINK Version:	SWDIO	⊙ 0x2BA01477	ARM CoreSight SW-DF	>	Up
Device Family: Cortex-M					Down
Firmware Version: V1.42	Auto	, omatic Detection			
SWJ Port SW 💌	C Mar	nual Configuration	Device Name:		
Max Clock: 1MHz 💌	Add	Delete	Update IR len:		
Debug Connect & Reset Options			- Cache Options	- Download Options -	
Connect Normal Reset:	SYSRES	SETREQ -	Cache Code	Verify Code Dowr	nload
Reset after Connect Stop after Bootloader					
		OK	Cancel		Help

Figure 3-7

(4) Click Ctrl+F5 or shortcut icon, or select Debug->Start/Stop Debug Session to start

debugging. Refer to figure 3-8:

W	C:\Users\Admini	strator\D	esktop\lpc1788\MyirProductCD(Reference)\05-MDK_Source\01-ADC\ADC_Interru	ıpt\ADC_Interrup 🗖 🗖 🔀
File	e Edit View Pro	oject Flas	h Debug Peripherals Tools SVCS Window Help	
) 💕 🛃 🥔 🕹	🗅 🛍 🕒	○ (> <> 🖗 祭 務 務 導 導 川 川 🙆 📃 🔍 💌 💽 () · · 🔗 🍓 🖬 🕶 🖄
RS	7 (1 🛛 🖉 🗐	- {} ⁺ -{}	◈▏ӮҨ҇ҝӻ≡ѽ҉ӝ҇҄ҹ҇ѿҹѿҹѿҹѿҹӀѿҹ҅Ӽ҂҅╡ӟ	
E	Registers	д 🖂	Disassembly 🗣 🔝	Symbols 📮 🖬
Pro	Register	Value	C)0x1FFF0080 F8DF4018 LDR.W r4,[pc,#24] ; @0x1FFF009C	· ·
jec.	R0	0x0000(0x1FFF0088 f8256 LDR r6,[r4,#0x00]	Module / Name Location
Ť	R1	0x00000	VXIFFFUU8A EAUSU606 AND r6,r5,r6	⊕ Virtual Regist
	R3	0x00000. 0x04C1:	Adc_Interrupt.c 🗈 lpc177x_8x_adc.c 🖆 lpc177x_8x_pinsel.c 🔻 🗙	Special Functi
	R4 R5	0x00000 0x00000	077 adc_value = 0;	ADC_Interrupt
	R6	0x10000	078	Adc_In
	-R8	Ox40098 OxFFFFI	080 {	∎ .//Co
		0x10000	081 adc_value = ADC_ChannelGetData(LPC_ADC, B	//Co
	R11	0x00000		■
	R12 R13 (SP)	0x00000 0x1000:	Command 🛛 📮 🛐	* ././Co
		OxFFFFI		₽
	± xPSR	0x01000	Setup(); // Setup for Running	₽
	⊞ Banked System		4	//Co
	Internal		> ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet	
	Mode Privi	Thread Privil(Call Stack + Locals	
	Stack	MSP	Name Location/Value Type	
	Sec	0. 00000		
				•
			ULINK2/ME Co	ortex Debugger t1: 0.00000000 sec

Figure 3-8

3.2.3 Download programby ULINK2

Connect Ulink2 to JTAG (J13) and turn power on.

(1) Open 05 MDK_Source\01- ADC\ADC_Interrupt and configureFLASH Programming Utilities in Project-> Option for target.

Firstly, set Use Target Driver for FLASH Programming option, then select ULINK2/ME Cortex Debugger, and then select Update Target before Debugger Options, and finally click Settings button, thenpop up FLASH Download Setup dialog box.

🖞 Options for Target 'FLASH'	X
Device Target Output Listing User C/C++ Asm Linker Debug Utilities	1
Configure Flash Menu Command STEP1	EP3
ULINK2/ME Cortex Debugger Settings Update Target be	efore Debugging
STEP2 STEP4	
Command:	
Arguments:	
OK Cancel Defaults	Help

Figure 3-9

(2) Configure FLASH Download Setup

Firstly,Configure download options in Download Function as shown in figure 3-10. Then set START: 0x10000000 SIZE: 0x800in RAM for Algorithm option, last configure algorithm, if Programming Algorithm box below has no algorithm file, single hit Add button to select LPC17xxIAP512kB Flash, and finally click OK button, as shown in figure 3-11.

Cortex-M Target Driver Setup
Debug Trace Flash Download
Composition STEP1 STEP2 Composition C Erase Full Chip Image: Program Image: Program Image: C Erase Sectors Image: Verify Start: 0x10000000 Size: 0x0800 C Do not Erase Image: Reset and Run Start: 0x10000000 Size: 0x0800
Programming Algorithm
Description Device Type Device Size Address Range LPC17xx IAP 512kB Flash On-chip Flash 512k 00000000H - 0007FFFFH
STEP3
Start: Size: Size: Add Remove
OK Cancel Help

Description	Device Type	Device Size	
LPC11xx/13xx IAP 16kB Fl	On-chip Flash	16k	
LPC11xx/13xx IAP 24kB Fl	On-chip Flash	24k	
LPC11xx/122x/13xx IAP 32	On-chip Flash	32k	
LPC122x IAP 48kB Flash	On-chip Flash	48k	
LPC122xIAP 64kB Flash	On-chip Flash	64k	
LPC11xx/13xx IAP 8kB Fla	On-chip Flash	8k	
LPC122x IAP 80kB Flash	On-chip Flash	80k	
LPC122x IAP 96kB Flash	On-chip Flash	96k	
LPC17xxIAP 128kB Flash	On-chip Flash	128k	
LPC17xxIAP 256kB Flash	On-chip Flash	256k	
LPC17xx IAP 32kB Flash	On-chip Flash	32k	
LPC17xx IAP 512kB Flash	On-chip Flash	512k	
LPC17xx IAP 64kB Flash	On-chip Flash	64k	·
MB9BF500 256kB Flash	On-chip Flash	256k	
MB9xFxx1 64kB Flash	On-chip Flash	64k	
MB9xFxx2 128kB Flash	On-chip Flash	128k	
		0501	

Figure 3-10



(3) Click download button, download program to FLASH in LPC1788. Refer to figure

3-12.

Image: Second	File Edit View Project Flash Debug Peripherals Tools SVCS Window Help					
Project Image: Construct of the second construction of the second constrel construction of the second construction of the second construc	□ 22 🖬 🗿 🖇 🖻 22 🔹 🕫 🕐 22 🎕 23 23 24 27 28 28 28 28 28 28 28 28 28 28 28 28 28					
Project Image: Construct and Construct	🕸 🕮 🧼 🔜 📴 Flash	🔽 🔊 📥 🔁				
<pre>FLASH</pre>	Project 🗣 🚺	Adc_Interrupt.c 📩 core_cm3.c 📾 abstract.txt 🚵 debug_frmwrk.c 💼 stdint.h 🔍 🔻 🗙				
Image: Core_cminstr.h 060 /************************************	Image: FLASH Image: Startup Image:	<pre>%#include "debug_frmwrk.h" %#include "bsp.h" % %#include "bsp.h" % % % % % % % % % % % % % % % % % % %</pre>				

Figure 3-12

3.2.4 ISP Download

When using ISP software to download program, firstly install FLASH magic (download latest version from http://www.flashmagictool.com), then connect JP6, JP4 (PIN1), JP5 (PIN2) to enable UART0, lastly set dial switch to LOW position and restart board.

Steps:

(1) Open FLASH magic and click"Options", then choose "Advanced Options". Refer to figure 3-13:

🐡 Flash Magic - NON PRODUCTION USE ONLY					
File ISP Options Tools Help					
🖻 🗖 🔍 🖪 Advanced Options					
Step 1 - Co. Disable Hints Update	Step 2 - Erase				
Select LPC1788	Erase block 0 (0x000000-0x000FFF)				
Flash Bank:	Erase block 2 (0x002000-0x002FFF)				
COM Port: COM 1	Erase block 3 (0x003000-0x003EEE) Erase block 4 (0x004000-0x004EEE)				
Baud Rate: 57600 🗸	Erase block 5 (0x005000-0x005FFF)				
Interface: None (ISP)	Erase blocks used by Hex File				
Oscillator (MHz): 12.000000	No. SY				
Step 3 - Hex File Hex File: C\L Isers\Administrator\Deskton\Inc1788\MvirProductCD(Reference) Browse					
Modified: September, 30, 2012, 8:54 more info					
Step 4 - Options Step 5 - Start					
Verify after programming Fill unused Flash					
Did you read the article "Using Flash Memory in Embedded Applications"?					
www.esacademy.com/faq/docs/flash					

Figurec3-13

(2) Choose "Use DTR and RTS to control RST and ISP pin" in "Hardware Config" in "Advanced Options", then click "OK". Refer to figure 3-14:
MYIR Make Your Idea Real





(3) Configure development environment and select LPC1788. COM Port can be saw in manger device. Baud rate choose 57600. Crystal oscillator selects 12M. Selectsend Hex File. Refer to figure 3-15 and 3-16.

🎲 Flash Magic - NON PRODUCTION USE	
File ISP Options Tools Help	
🖻 🖬 🧠 🗿 🐗 🖌 📕 🔈 😻 🔯 🚱 🎖	8
Step 1 - Communications	Step 2 - Erase
Select LPC1788	Erase block 0 (0x000000-0x000FFF)
F 🌧 Device Database	
Image: Constraint of the constraint	▲ UART bootloader 1 flash bank Bank 0: 512KB (0x0007FFFF) F RAM blocks: 0x10000000 -> 0x Signature: 673005383 (0x281D3 Flash erased value: 0xFF Code Read Protection location High speed communcations st
Execute Activate Flash Bank	
Your Training or Consulting Partner: Embedded	Systems Academy
www.esacademy.com	•
	1



Figure 3-15

Figure 3-16

(4) Connect UART to COM and click ISP->Read Device Signature, then Flash Magic

will recognize LPC1788ID. Refer to figure3-17:

MYiR Make Your Idea Real

Device Signature								X
Manufacturer ID: 0x								
Device ID 1: 0x								
Device ID 2: 0x								
Device ID: 0x	281D3F47							
Bootloader Ver:	8.1							
Serial Number:	252642578 1	1397	56137	4 1 3 1 2	624048	6 41104	17920	1
							Close	•

Figure 3-17

(5) Recognizing board and clicking "Start" button, program will be downloaded into

board. Refer to figure 3-18:

🎲 Flash Magic - NON PRODUCTION US	
File ISP Options Tools Help	
🖻 🖬 🔍 🗿 🍏 🖌 📕 🔈 😻 🔯	8
Step 1 - Communications	Step 2 - Erase
Select LPC1788	Erase block 0 (0x000000-0x000FFF)
Flash Bank:	Erase block 2 (0x001000-0x001FFF)
COM Port: COM 1 🗸	Erase block 3 (0x003000-0x003FFF) Erase block 4 (0x004000-0x004FFF)
Baud Rate: 57600 👻	Erase block 5 (0x005000-0x005FFF)
Interface: None (ISP) 🗸	■ Erase all Flash+Code Rd Prot ▼ Erase blocks used by Hex File
Oscillator (MHz): 12.000000	
Step 3 - Hex File	
Hex File: C:\Users\Administrator\Desktop\lpc1	788\MyirProductCD(Reference Browse
Modified: September, 30, 2012, 8:54	more info
Step 4-Options	Step 5 - Start!
Verify after programming	Start
Fill unused Flash	
Gen block checksums	
Activate Flash Bank	
Rotating, fully customizable, remotely updated	nternet links. Embed them in vour
application!	· · · · · · · · · · · · · · · · · · ·
www.embeddedhints.com	
	0

Figure 3-18

(6) After downloading program, disconnect JP3 and reset board, program starts

running. Refer to figure 3-19:

🎲 Flash Ma	💭 Flash Magic - NON PRODUCTION USE ONLY							
File ISP	Options Tools Help							
🖻 🗔 🔍 I	3 🎸 🗸 🎩 🗲 😻 🔯 🚱 🤅	8						
Step 1 - Con	nmunications	Step 2 - Erase						
Select	LPC1788	Erase block 0 (0x000000-0x000FFF)						
Flash Bank;	· · · · · · · · · · · · · · · · · · ·	Erase block 2 (0x002000-0x002FFF)						
COM Port:	СОМ1 🔹	Erase block 3 (0x003000-0x003FFF) Erase block 4 (0x004000-0x004FFF)						
Baud Rate:	57600 🔹	Erase block 5 (0x005000-0x005FFF)						
Interface:	None (ISP)	Erase all Flash+Code Rd Prot Erase blocks used by Hex File						
Oscillator (N	MHz): 12.000000							
Stop 2 - Hoy	File							
Hex File: U	\Users\Administrator\Desktop\lpc17	'88\MyirProductCD(Reference Browse						
	Jamea: September, 30, 2012, 8:54	more mio						
Step 4 - Opti	ons	Step 5 - Start!						
Verify afte	r programming d Electr	Start						
Gen block	checksums							
Execute								
Activate F	lash Bank							
Your Trai	ning or Consulting Partner:	Embedded Svstems						
www.esacad	www.esacademy.com							
Finished		2						

Figure 3-19

3.3MDK source use

3.3.1 Directory structure

MYD-LPC1788 MDK routines is in 05-MDK_Source of CD-ROM directory. Common folder contains MDK routine common code, including start-up, peripheral drivers, core initialization, foreign expansion chip driver. Detailed information is shown in Table 3-1:

Directory	Description			
BoardSupport	External expansion chip driver onboard			

CoreSupport	Macro definition of kernel function		
DeviceSupport	Boot code and system initialization code		
Drivers	Peripheral drivers		

Table 3-1

3.3.2 Add function module

When need to test or use a function module, add .C file in module and then contain module header file in .C files, lastly call module function directly.

For example, if join timer functionin CAN_Test, then select "Add File to Group 'Drivers'" to find lpc177x_8x_timer.c files in \05-MDK_Source\common\Drivers\source and click Add button. Refer tofigure 3-20 and 3-21:



Figure 3-20

MYIR Make Your Idea Real

Add Files to Group 'Drivers'	×
Look in: 📗 source 💽 🗢 🗈 📸 📰 🔻	
Name	Date modified 🔷
Dpc177x_8x_nvic.c	2011-10-24 9:22
Ipc177x_8x_pinsel.c	2012-08-08 13:52
pc177x_8x_pwm.c	2011-10-24 9:22
Dpc177x_8x_qei.c	2011-10-24 9:22
Dpc177x_8x_rtc.c	2011-10-24 9:22
Ipc177x_8x_ssp.c	2012-08-06 14:06
Ipc177x_8x_systick.c	2012-08-08 14:29
pc177x_8x_timer.c	2011-10-24 9:22
pc177x_8x_uart.c	2012-08-01 17:06
Ipc177x_8x_wwdt.c	2011-10-24 9:22
	•
	4
File name: lpc177x_8x_timer.c	Add
Files of type: C Saures file (* a)	
	/

Figure 3-21

Then, .C file of timer function need include header files:

#include "lpc177x_8x_timer.h"

Timer initialization, configuration, features such as delay time can be provided by timer.

3.3.3Use Printf

Using Printf to debug serial to print run-time information is an effective debug means. But print terminal isn't serial portin default, in order to make character flow redirected to serial port, it needs a new definition of fputc function. The specific operation is as follows:

(1) Add retarget.c files to project which defines fputc function. retarget.c file in \05-MDK_Source\common\ CoreSupport directory. User is advised to add retarget.c file to CMSIS-CM3 group, as shown in figure 3-22:

MYIR Make Your Idea Real

Project	ņ	
E-B FLASH		
📄 💼 Startup		
🖕 🖶 CMSIS-CM3		
- 🔚 core_cm3.c		
- 🖾 system_LPC177x_8x.c		
🖾 retarget.c		
📄 🖶 🔄 Drivers		
- 🔚 debug_frmwrk.c		
🚽 🔝 lpc177x_8x_clkpwr.c		
- 🖾 lpc177x_8x_pinsel.c		
🔄 🔝 lpc177x_8x_timer.c		
🖕 🗄 Main		
🖾 🖾 Can_Test.c		
📙 🗄 🖷 🔄 Readme		
🔤 abstract.txt		



(2) Hook Use MicroLIB in engineering configuration, refer to figure 3-23:

V Options for Target 'FLASH'									
Device T	arget 0	utput Listi:	ng User C/	C++ Asm	Link	er Debug	g Utilities	3	
NXP (found	NXP (founded by Philips) LPC1788								
			Xtal (MHz): 12.	0		Cheradon			
Operating	Operating system: None Use Cross-Module Optimization								
System-Vi	ewer File (.Sfr):			Us Us	e MicroLIB	[Big Endian	
-Bead/O	nly Memoi	v Areas — —			ReadM	/rite Memo	rv Areas		
default	off-chip	Start	Size	Startup	default	off-chip	Start	Size	Nolnit
	ROM1:			0		RAM1:			
	ROM2:			0		RAM2:			
	ROM3:			0		RAM3:			
	on-chip			_		on-chip			_
	IROM1:	0×0	0×80000	۲		IRAM1:	0x10000000	0×10000	
	IROM2:			0		IRAM2:	0x20000000	0x8000	
	OK Cancel Defaults Help								

Figure 3-23

(3) .c file need to call printf function contains stdio.h header file.

So, it will be able to use printf to printdata to COM port, and thendisplayby

HyperTerminal.

3.4 The introduction of MDK routine

MDK routines use UART2 port to print debug information, so it needs to set JP4 and JP5 to enable UART2 port (PIN2 connect PIN3 in JP4 and JP5), set baud rate: 115200, 8 data bits, one stop bits, no parity bit, no control flow.

Please note that after download, it needs to disconnect JP6 and then reset board in ISP download, otherwise it may cause abnormal.

3.4.1 ADC_Interrupt

Functional description

This example describes ADC conversion in interrupt mode.

Procedures

After download program, press SW5 to reset board, ADC value displayed in terminal

and changed byturning potentiometer.

Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors ADC INTERRUPT example: - MCU: LPC177x_8x - Core: ARM CORTEX-M3 - Communicate via: UART2 - 115200 bps Use ADC with 12-bit resolution rate of 400KHz, read in INTERRUPT mode To get ADC channel value and display via UART interface Turn the potentiometer to Refer to how ADC value changes ***** ******* ADC value on channel 002 is: 0000000148 ADC value on channel 002 is: 0000000145 ADC value on channel 002 is: 000000146 ADC value on channel 002 is: 0000000144 ADC value on channel 002 is: 000000144 ADC value on channel 002 is: 000000144 ADC value on channel 002 is: 000000144

ADC value on channel 002 is: 0000000146 ADC value on channel 002 is: 0000000146 ADC value on channel 002 is: 0000000146 ADC value on channel 002 is: 0000000144 ADC value on channel 002 is: 0000000146 ADC value on channel 002 is: 0000000146

3.4.2 ADC_Polling

Functional description

This example describes ADC conversion in polling mode.

Procedures

After download program, press SW5 to reset board, ADC value displayed in terminal and changed by turning potentiometer.

Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors ADC POLLING example: - MCU: LPC177x_8x - Core: ARM CORTEX-M3 - Communicate via: UART2 - 115200 bps Use ADC with 12-bit resolution rate of 400KHz, read in POLLING mode To get ADC value and display via UART interface Turn the potentiometer to Refer to ADC value changes ADC value on channel 002 is: 000000148 ADC value on channel 002 is: 0000000145 ADC value on channel 002 is: 0000000149 ADC value on channel 002 is: 0000000149 ADC value on channel 002 is: 000000148 ADC value on channel 002 is: 0000000149 ADC value on channel 002 is: 000000148 ADC value on channel 002 is: 0000000147

ADC value on channel 002 is: 0000000148 ADC value on channel 002 is: 0000000148

3.4.3 CAN_Test

Functional description

This example describes CAN transmit and receive data.

Procedures

Firstly connect CAN1 and CAN2 (Pin1, Pin2 are respectively connected Pin4, Pin5 in J8). After download program and press SW5 to reset board, CAN1 will send a frame data to CAN2. After CAN2's reception, it will be checked and result will be printed in terminal.

Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors							
CAN Self-test example							
- MCU: LPC17xx							
- Core: ARM CORTEX-M3							
- UART Communication: 115200 bps							
Use CAN1 to transmit and CAN2 for receive							

Transmitted buffer:							
Message ID: 0x00001234							
Message length: 0x00000008 BYTES							
Message type: DATA FRAME							
Message format: EXTENDED ID FRAME FORMAT							
Message dataA: 0x12121212							
Message dataB: 0x34343434							
Received buffer:							
Message ID: 0x00001234							
Message length: 0x00000008 BYTES							
Message type: DATA FRAME							
Message format: EXTENDED ID FRAME FORMAT							
Message dataA: 0x12121212							
Message dataB: 0x34343434							
>CAN TEST Successful!!!							

3.4.4 Crc_Demo

Functional description

This example describes CRC engine.

Procedures

After download program, press SW5 to reset board.Program calculates CRC block data firstly and display calculates result after receiving input.

> Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors CRC Demo example: - MCU: LPC177x_8x - Core: ARM CORTEX-M3 - UART Communication: 115200 bps Use CRC engine on LPC177x_8x to calculate CRC for a 8-bit block data You can choose one of three polynomial type: - CRC-CCITT - CRC-16 - CRC-32 Block data: 0x00000000 0x0000001 0x0000002 0x0000003 0x0000004 0x00000005 0x0000006 0x0000007 0x0000008 0x0000009 0x000000A 0x000000B 0x000000C 0x000000E 0x000000D 0x000000F 0x0000010 0x00000011 0x00000012 0x0000013 0x0000014 0x0000015 0x0000017 0x0000018 0x0000016 0x00000019 0x000001A 0x0000001C 0x000001D 0x000001B 0x00000022 0x000001E 0v000001E 0200000020 0v0000021

	0x0000011	0x0000020	070000021	0x0000022
0x00000023	0x00000024	0x00000025	0x00000026	0x0000027
0x00000028	0x00000029	0x0000002A	0x0000002B	0x0000002C
0x0000002D	0x0000002E	0x0000002F	0x0000030	0x0000031
0x00000032	0x0000033	0x00000034	0x0000035	0x0000036
0x00000037	0x0000038	0x0000039	0x000003A	0x000003B
0x000003C	0x000003D	0x000003E	0x000003E	

Choose what polynomial that you want to use, type:

- '1': CRC-CCITT
- '2': CRC-16
- '3': CRC-32

- 'Q': Quit CRC-CCITT Result: 0x0000FD2F Choose what polynomial that you want to use, type: - '1': CRC-CCITT - '2': CRC-16 - '3': CRC-32 - 'Q': Quit CRC-16 Result: 0x00002799 Choose what polynomial that you want to use, type: - '1': CRC-CCITT - '2': CRC-16 - '3': CRC-32 - 'Q': Quit CRC-32 Result: 0x100ECE8C Choose what polynomial that you want to use, type: - '1': CRC-CCITT - '2': CRC-16 - '3': CRC-32 - 'Q': Quit Demo terminated!!!

3.4.5 Dac_Dma

> Functional description

This example describes DMA transfer data to DAC peripheral.

Procedures

After download program, press SW5 to reset board. DMA transfer data to DAC peripheral constantly. DACvalue changes byconstant transferring data.Use multimeter or oscilloscope to detect TP7voltage.

Phenomenon Indicates

Output voltage from 0V to 3.3V, direct jump to 0V, began to rise again in cycle. So there will be saw tooth waveform in oscilloscope. It is observed oscilloscope ("V / price" and "s / lattice" knob settings for 2.00V and 1.00S).Refer to figure 3-24:

MYIR Make Your Idea Real





3.4.6 Dac_SineWave

Functional description

This example describes DMA generatessine wave signal.

Procedures

After download program, press SW5 to reset board.. Oscilloscope probe is connected

to TP7, and thenthere will be sine wave signal.

Phenomenon Indicates

Observe waveform by oscilloscope ("V/price" and "s/lattice" knob settings for 2.00V

and 2.50ms). Refer to figure 3-25:



Figure 3-25

3.4.7 DMA_Flash2Ram

Functional description

This example describes GPDMA function by transferring data from Flash to Ram

memory.

Procedures

After download program, press SW5 to reset board, there will be information in terminal. A block datatransferred byGPDMA from Flash to Ram is checked and the result will be outputted in the terminal.

Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors GPDMA FLASH to RAM example - MCU: LPC177x_8x - Core: ARM CORTEX-M3 - UART Communication: 115200 bps This example used to test GPDMA function by transfer data from Flash to RAM memory Start transfer... Buffer Check success! Demo terminated!

3.4.8 Eeprom_Demo

> Functional description

This example describes tore data in EEPROM memory.

Procedures

After download program, press SW5 to reset board.Program will first write "NXP Semiconductor LPC177x_8x-CortexM3 \n\r\t--- HELLO WORLD!!! ---" into EEPROM, and then reads and displays data from corresponding location in EEPROM. When there will be "NXP Semiconductor LPC177x_8x-CortexM3 \n\r\t--- HELLO WORLD from the terminal!!!," it shows write-in and read-out is normal.

> Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors

EEPROM demo example

- MCU: LPC177x_8x
- Core: ARM CORTEX-M3
- Communicate via: UART2 115200 bps

This example used to demo EEPROM operation on LPC177x_8x.

A 'Hello' sentence will be written into EEPROM memory, then read back and check.

Write data to EEPROM Read data from EEPROM NXP Semiconductor LPC177x_8x-CortexM3 --- HELLO WORLD!!!---Demo is terminated

3.4.9 Emc_NorFlashDemo

Functional description

This example describesEMC read/write NOR FLASH.

Procedures

After download program, press SW5 to reset board. Program firstly check Manufacturer ID and Device ID of NorFlash chip(Model: SST39VF1601). Then entire Flash memory will be erased. 2K block data will be written and read back for verify.

Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors EMC NORFLASH example - MCU: LPC177x_8x - Core: Cortex-M3 - UART Comunication: 115200 bps Write and verify data with on-board NOR FLASH Init NOR Flash... Read NOR Flash ID... Erase entire NOR Flash... Write a block of 2K data to NOR Flash... Verify data... Verifying complete! Testing terminated!

3.4.10 Emc_SdramDemo

Functional description

This example describes EMC read/write SDRAM.

Procedures

After download program, press SW5 to reset board. The process of program: clear SDRAM and write data into SDRAM in 8-bits mode and verify in 32-bits read mode. Then clear SDRAM and write data into SDRAM in 16-bits mode and verify in32-bits read mode.

Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors Test SDRAM K4S561632J with LPC1788 EMC - MCU: LPC177x_8x - Core: Cortex-M3 - UART Comunication: 115200 kbps Write and verify data with on-board SDRAM Clear content of SDRAM... Writing in 8 bits format... Verifying data... Continue writing in 16 bits format... Clear content of SRAM... Writing in 16 bits format... Verifying complete, testing terminated!

3.4.11 Emc_SramDemo

Functional description

This example describes EMC read/write SRAM.

Procedures

After download program, press SW5 to reset board.After clear data and write data into SRAM,program displays memory address.

Phenomenon Indicates

Terminal information:

***** Hello NXP Semiconductors - MCU: LPC177x_8x - Core: Cortex-M3 - UART Comunication: 115200 kbps Clean and write data with on-board SRAM Uartinit finished!!! The value after clearing are: 000000000 000000000 000000000 000000000 000000000 000000000 000000000 000000000 The filling value is:0xABCDDCBA The value after filling are: **0xABCDDCBA 0xABCDDCBA 0xABCDDCBA 0xABCDDCBA 0xABCDDCBA 0xABCDDCBA 0xABCDDCBA 0xABCDDCBA**

3.4.12 GPIO_Interrupt

> Functional description

This example describes GPIO interrupt function.

Procedures

After download program, press SW5 to reset board, program doesn't generate a GPIO interrupt, and D9 flashes. When user pressSW1 to generate a GPIO interrupt, program will enter interrupt process program and D10 flash 10 times, and then return.

Phenomenon Indicates

When program doesn't generate a GPIO interrupt, D10 light and D9 flash. When press SW1, D9 keep origin status and D10 flashes.

3.4.13 GPIO_LedBlinky

> Functional description

This program describes GPIO drives LED.

Procedures

After download program, press SW5 to reset board, D9 flashes.

Phenomenon Indicates

Reset board, andD9 flashes.

3.4.14 Nvic_VectorTableRelocation

Functional description

This example describes relocation vector table.

Procedures

After download program, press SW5 to reset board.Vector Table will be remapped at new address 0x20001000. If remapping is successful, SysTick interrupt can driver D9 flash normally.

Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors Privileged demo - MCU: LPC177x_8x - Core: ARM CORTEX-M3 - UART Communication: 115200 bps This example used to test NVIC Vector Table Relocation function Remapping Vector Table at address: 0x20001000 If Vector Table remapping is successful, LED D9 will blink by using SysTick interrupt

3.4.15 Pwm_SingleEdge

> Functional description

This example describes PWM signal on 6 Channels in single edge mode.

Procedures

Compile program and download it into board. Reset board and observe oscilloscope.

Phenomenon Indicates

Observe pin of PWM0.1 and PWM0.3 by oscilloscope. Refer to figure 3-26 and figure 3-27:



Figure 3-27

3.4.16 Pwm_DualEdge

> Functional description

This example describes generate PWM signal on 3 channels in both edge mode and

single mode.

Procedures

:

After download program, press SW5 to reset board, there will be information in oscilloscope.

Phenomenon Indicates

Observe pwm0.1 waveform byOscilloscope:





3.4.17 Pwm_MatchInerrupt

> Functional description

This example describes PWM Match function in interrupt mode.

Procedures

After download program, press SW5 to reset board, there will be information in oscilloscope.

Phenomenon Indicates

Observe Waveform of PWM0.1, PWM0.3 by oscilloscope. Refer to Figure 3-29 and figure 3-30:

MYIR Make Your Idea Real





Figure 3-30

3.4.18 PWR_Sleep

Functional description

This example describes enter system in sleep mode and wake up by WWDT(Windowed Watchdog Timer) Interrupt

Procedures

After download program, press SW5 to reset board. Receive '1' from serial, system enter sleep mode. Wait 2s to generate a WWDT interrupt and wake up system.

Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors Power Sleep example:

- MCU: LPC177x_8x
- Core: ARM CORTEX-M3

- UART Communication115200 bps This example used to enter system in sleep mode and wake up it by using Watchdog timer interrupt Press '1' to enter system in Deep Sleep mode Enter Deep Sleep mode!

Wait 2s for WDT wake-up system...

System wake-up!

3.4.19 PWR_DeepSleep

> Functional description

This example describes enter system in deep sleep mode and wakeup by external interrupt.

> Procedures

After download program, press SW5 to reset board.Receive '1' from serial, system enter deep sleep mode. Connect JP6(A short time connection) to generate external interrupt ,it will wake up system.

Terminal information:

Hello NXP Semiconductors Power - Deep Sleep example - MCU: LPC177x_8x - Core: ARM CORTEX-M3 - UART Communication115200 bps This example used to enter system in deep sleep mode and wake up it by using external interrupt Press '1' to enter system in deep sleep mode. If you want to wake-up the system, press INT/WAKE-UP button. I'm sleeping... ------ I'm wake up! ------

3.4.20 Emac_EasyWeb

Functional description

This example describes implement an simple web application.

> Procedures

After download program, press SW5 to reset board,connectboard toPCby crosswire.Configure IP address: 192.168.2.100. Open web browser, accessaddress "http://192.168.0.100" to display webservercontent. Turn potentiometer and webdisplaysADCupdate value. Please note, webpagechange a state automatically by 5 seconds.

> Phenomenon Indicates

Refer to figure 3-31:

He1	Hello World!								
This i	is a dynam	ic websi	ite hosted	by the	myir Webse	erver i	ayir.		
Hardwa	ire:								
• Current Board: LPC1788 OEM board rev PA4 mounted on OEM base board rev A • Using IC: LPC1788 (ARM Cortex-M3) with 96MHz, 512KB Flash, 96 KB SRAM • myir EMAC Ethernet Controller									
A/D Converter Input 1:									
OV	0. 5V	1V	1.5V	2V	2. 5V	3V			

Figure 3-31

3.4.21 Rtc_Alarm

> Functional description

This example describes RTC generate interrupt in second and Alarm interrupt.

Procedures

After download program, press SW5 to reset board. Program set initialize time and generate interrupt by second. So alarm interrupt occursafter 10s and alarm sentence will be outputted.

Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors

RTC Alarm Example: - MCU: LPC177x_8x - Core: ARM CORTEX-M3 - UART Communication: 115200 bps A simple RTC example. To generate interrupt in Second Counter Increment Interrupt (1s) and generate Alarm interrupt at 10s Current time set to: 018:045:000 025/003/02011 Second ALARM set to 010s Second: 001 Second: 002 Second: 003 Second: 004 Second: 005 Second: 006 Second: 007 Second: 008 Second: 009 Second: 010 ALARM 10s matched! Second: 011 Second: 012 Second: 013 Second: 014 Second: 015

3.4.22 SSP_Touchscreen

Functional description

The program shows SSP interface read position x and y in touchscreen.

Procedures

After download program, press SW5 to reset board, here will becurrent position X and

Y in terminal.

Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors

SSP Touchscreen Example:

- MCU: LPC177x_8x

- Core: ARM CORTEX-M3
- UART Communication: 115200 bps

A simple ssp-touch example.

When you touch the screen, you will Refer to the X and Y values on the termimal.

Channel X data is:00000 Channel Y data is:04095 Channel X data is:01788 Channel Y data is:02091 Channel X data is:01785 Channel Y data is:02116

3.4.23 SSP Flash

Functional description \geq

This example describes SSP peripheral reads AT25DV321A.

\triangleright Procedures

After download program, press SW5 to reset board, there will be testprocess.

Phenomenon Indicates \triangleright

Terminal information:

********* System Start The SystemCoreClock is 120 MHZ The PeripheralClock is 60 MHZ AT25DF321A Init -ManID: 0x1f -DevID: 0x47 0x01 -ExtStrLen: 0x00 A simple ssp flash write& read example. ***** **********

Testing page num : 16384/16384

3.4.24 Systick_100msBase

> Functional description

This example describes configure System Tick timer to generate interrupt each 100ms.

Procedures

AAfter download program, press SW5 to reset board. The program configures system tick to generate interrupt at each 100ms. Generating interrupt changesD9status each time.

Phenomenon Indicates

D9 flash is at 5Hz.

3.4.25Timer_MatchInterrupt

> Functional description

This example describes Timer generates specific time in interrupt mode.

Procedures

After download program, press SW5 to reset board, terminal print information by second.

Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors

Timer Match Interrupt demo

- MCU: LPC177x_8x

- Core: ARM CORTEX-M3
- UART Communication: 115200 bps

Use timer x toggle MATx.0 at frequency 1Hz

Match interrupt occur...

Match interrupt occur...

Match interrupt occur...

Match interrupt occur... Match interrupt occur... Match interrupt occur... Match interrupt occur... Match interrupt occur... Match interrupt occur...

3.4.26 Wdt_Interrupt

Functional description

This example describes WDT generates timeout interrupt or alarm interrupt.

> Procedures

After download program, press SW5 to reset board.Print options: select "1" display timeout interrupt. When WDT counter is reduced to 0, it will cause interrupt, D13 flashes. Choice "2" display warning interrupt. When WDT counter is close to 0, it will cause interrupt and D13 flashes.

Phenomenon Indicates

Terminal information:

(1) Time out interrupt

Hello NXP Semiconductors

Watch dog timer interrupt (test or debug mode) demo

- MCU: LPC177x_8x
- Core: ARM CORTEX-M3
- UART Communication: 115200 bps

An interrupt will be generated once WWDT is timeout (depend on configuration) or the counter is reached the Warning Value.

After interrupt WDT interrupt is disabled immediately!

BEFORE WDT interrupt!

Press '1' to enable Watchdog timer Interrupt by Timeout only...

Press '2' to enable Watchdog timer Interrupt by Warning ...

Pressed '1' - Working with Normal Timeout Interrupt

The Timer Value causes the Interrupt: 0x00000000

AFTER WDT interrupt

LED is blinking...

(2) Alarm interrupt

Hello NXP Semiconductors Watch dog timer interrupt (test or debug mode) demo - MCU: LPC177x_8x - Core: ARM CORTEX-M3 - UART Communication: 115200 bps An interrupt will be generated once WWDT is timeout (depend on configuration) or the counter is reached the Warning Value. After interrupt WDT interrupt is disabled immediately! BEFORE WDT interrupt! Press '1' to enable Watchdog timer Interrupt by Timeout only... Press '2' to enable Watchdog timer Interrupt by Warning ... Pressed '2' - Working with Warning Interrupt The Timer Value causes the Interrupt: 0x00000268 AFTER WDT interrupt LED is blinking...

3.4.27 Wdt_Reset

Functional description

This example describes WDT generates a reset event.

Procedures

After download program, press SW5 to reset board. After start, WDT counter

decrease until underflow to reset chip. After reset, program will print reset reason.

Phenomenon Indicates

Terminal information:

This Welcome Screen below will executive after reset event Hello NXP Semiconductors Watch dog timer reset when timeout demo - MCU: LPC177x_8x - Core: ARM CORTEX-M3 - UART Communication: 115200 bps Use WDT with Internal RC OSC, reset mode, timeout = 5 seconds To reset MCU when time out. After reset, program will determine what cause of la st reset time (external reset or WDT time-out) The program is currently working in FLASH mode Last MCU reset caused by External! This Welcome Screen below will executive after reset event Hello NXP Semiconductors Watch dog timer reset when timeout demo - MCU: LPC177x_8x - Core: ARM CORTEX-M3 - UART Communication: 115200 bps Use WDT with Internal RC OSC, reset mode, timeout = 5 seconds To reset MCU when time out. After reset, program will determine what cause of la st reset time (external reset or WDT time-out) The program is currently working in FLASH mode

Last MCU reset caused by WDT TimeOut!

3.4.28 Lcd_LQ043T3DX0A

Functional description

This example project describes LCD module displays a static picture.

> Procedures

After download program, press SW5 to reset board, picture is displayed in LCD.

Phenomenon Indicates

After download program, there is picture on LCD.

3.4.29Lcd_touch

Functional description

This example project describes how to use Touch Screen and LCD.

Procedures

After download program, press SW5 to reset board, LCD screen display different colors. Displaydifferent color on LCD screen byclicking the color block.

Phenomenon Indicates

After download program, there is picture on LCD.

3.4.30 Mci_CidCard

Functional description

This example describes Multimedia Card Interface (MCI).

Procedures

After download program, press SW5 to reset board.Insert SD Card and read SD cardinformation and print information in the terminal.

> Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors
MCI CID Card

MCU: LPC177x_8x
Core: ARM CORTEX-M3
UART Communication: 115200 bps

This example is used to test the Multimedia Card Interface (MCI) function. It is able to check, show the CID that retrieved from the card
Currently the SD CARD is being used

Manufacture ID: 0x0000003
OEM/Application ID: 0x00005344
Product Name: 0x5355303247
Product Revision: 0x0000080
Product Serial Number: 0x17915B1F
Manufacturing Date: 0x00000C3

3.4.31 Usb_MassStorage

Functional description

This example describes USB Mass Storage application

Procedures

After download program, press SW5 to reset board.Connect board (J10) to PC by Mini USB. Load "LPC1788" memory automatically and open device, there will be a README.TXT file.

Phenomenon Indicates

Refer to figure 3-32:

MYIR Make Your Idea Real

	3 b		37.2
0:\			
<u>File Edit View T</u> ools	<u>H</u> elp		
Organize 🔻 🕜 Open	▼ Print Burn	New folder	
🔆 Favorites			
🧮 Desktop			
📕 Downloads			
🔚 Recent Places			
	README.TXT		
📜 Libraries			
Computer			
Local Disk (C)			
D (D:)			
E (E:)			
🕞 F (F:)			
M DVD RW Drive (G:) N			
LPC178X (0:)			
🗣 Network			

Figure 3-32

3.4.32 Usb_VirtualCom

> Functional description

This example describes configure USB as virtual COM port.

Procedures

After download program and connect J10 to PC by Mini USB, press SW5 to reset board.

Phenomenon Indicates

After download program, press SW5 to reset board. There appears "new equipment" prompt. Select "install from a list or specified location andlocal project directory. After install driver, "LPC177x_8x USB VCom Port (COMx)" will appear. "X" in the "COMx" is not fixed and is changed with different configuration. Refer to figure 3-33:



Figure 3-33

3.4.33 I2S_Audio

Functional description

This example describes I2S transfers audio data to play a short music.

Procedures

After download program, press SW5 to reset board, when insert microphoneto J5 interface, there will be sound.

Phenomenon Indicates

Terminal information:

Hello NXP Semiconductors USB MassStorage example: - MCU: LPC177x_8x - Core: ARM CORTEX-M3 - Communicate via: UART2 - 115200 bps *******

Init UART2 for debug ... UDA1380 Soft Reset OK! Init UDA1380 registers step 1 OK! Init UDA1380 registers step 2 OK! Init UDA1380 registers step 3 OK! Init UDA1380... Init LPC_I2S... Demo start...

3.4.34 I2C_Eeprom

Functional description

The program shows I2C writes and reads EEPROM.

Procedures

After download program, press SW5 to reset board. Program writes 8 bytes data and verify message. Debug information is outputted in terminal.

Phenomenon Indicates

Terminal information:

****** Hello NXP Semiconductors I2C EEPROM Example: - MCU: LPC177x_8x - Core: ARM CORTEX-M3 - UART Communication: 115200 bps A simple I2C EEPROM example. a page data will write to EEPROM and read out for verification. ***** ******* Write EEPROM OK! Read EEPROM OK! i2c_rx_Buf[0] is 0 i2c_rx_Buf[1] is 1 i2c_rx_Buf[2] is 2 i2c_rx_Buf[3] is 3 i2c_rx_Buf[4] is 4 i2c_rx_Buf[5] is 5 i2c_rx_Buf[6] is 6 i2c rx Buf[7] is 7 I2C EEPROM Test Success!!

MYiR Make Your Idea Real

3.4.35 RS_485-Master&Slave

Functional description

This example describes RS485 communication.

Procedures

This test needs two MYD-LPC1788 boards. Firstly connect PIN1 to PIN2 in JP7, PIN7 to PIN8 in J8, then download program espectively into two development boards. After download program, press SW5 to reset board. Host sends data to slave A and B by turn. When the salve board return ACK (slave address B) after receiving data from host, it shows communication is success.

Phenomenon Indicates

Terminal information:

(1) Master mode:

Hello NXP Semiconductors RS485 demo in Master mode SlvAddr: 65 Dev A have NO reply SlvAddr: 66 ACK SlvAddr: 65 Dev A have NO reply SlvAddr: 66 ACK

(2) Slave mode:

Hello NXP Semiconductors RS485 demo in Slave mode Slave's Receiver is not always enabled - Auto Address Detection is enabled Slave Addr detected! Slave Addr detected! Msg B: Hello NXP BBBBBB Recv a Terminator and Send ACK back Slave Addr detected! Slave Addr detected! Msg B: Hello NXP BBBBBB Recv a Terminator and Send ACK back Slave Addr detected! Slave Addr detected! Msg B: Hello NXP BBBBBB Recv a Terminator and Send ACK back

Appendix 1sales FAQ and technical support

How to buy

We accept paypal payment and bank wire transfer

1.Paypal payment

Please select the products add into shopping cart, the checkout web page will redirect to paypal.com for you payment. Shipment fee will calculated automatically by your locationregion.

2.Bank wire transfer

Pls email or fax us with products list you want, we will send you a pro-invoice with order value total, shipping cost and bank information.

Shipping details

Please select the shipping area catalogue for you location. If you have carrier account to pay the shipment fee, please select "Freight collect" and email us the carrier account. Please visit http://www.myirtech.com/support.asp for more details

Noted

- 1. The shipment will start in 3 biz days by Fedex Express, it usually take 7 days to reach regular cities or regions.
- 2.We will use DHL Express for West asia or middle east countries, it usually take 7 days to reach regular cities or regions.
- 3. The remote regions defined by Fedex/DHL may cause delay, 14 days in generally.
- 4.Some countries have strict import policy, we will help to make shipping invoice with you requirement, like invoice value, trade term, custom statements and H.S code etc. Please contact us with these shipment requirements if your country has strict custom affairs.

Support and maintains

MYIR provides 12 months warranty for hardware products if the defects or failures were notcaused by wrong use.

Return steps for defective products

- 1. Please email or call us get a Return Merchandise Authorization (RMA) by providing purchase details and reasons for return (defective, incorrect etc).
- MYIR will make a shipping invoice (list value total, item description etc) for you return request. China have strict limit on return products, so please use MYIR's shipping invoice to return items to avoid custom delay.

Contact:

Tel:+86-0755-25622735 Fax: +86-0755-2553 2724 Mail to: sales@myirtech.comsupport@myirtech.com Website: www.myirtech.com