

# 面向嵌入式工程师的 MPLAB<sup>®</sup> XC8 用户指南

# 面向嵌入式工程师的MPLAB<sup>®</sup> XC8用户指南

简介

本文档提供了5个适用于8位器件和MPLAB XC8 C编译器的代码示例。读者需要掌握一些单片机和C编程语言的相关知识。

- 1. 点亮或熄灭LED
- 2. 使用 delay()函数使LED闪烁
- 3. 使用中断作为延时在LED上递增计数
- 4. 使用A/D在LED上显示电位器值
- 5. 在LED上显示EEPROM数据值
- A 在MPLAB X IDE中运行代码
- B 获取软件和硬件

# 1. 点亮或熄灭LED

```
本示例将在带PIC16F1719单片机(MCU)的Explorer 8板上交替点亮各LED。更多
信息,请参见第B节"获取软件和硬件"。
#include <xc.h> ◀─── 请参见第1.1节
// PIC16F1719 Configuration Bit Settings
// For more on Configuration Bits,
                                               - 请参见第1.2节
// consult your device data sheet
// CONFIG1
#pragma config FOSC = ECH // External Clock, 4-20 MHz
#pragma config WDTE = OFF // Watchdog Timer (WDT) disabled
#pragma config PWRTE = OFF // Power-up Timer disabled
#pragma config MCLRE = ON // MCLR/VPP pin function is MCLR
#pragma config CP = OFF // Flash Memory Code Protection off
#pragma config BOREN = ON // Brown-out Reset enabled
#pragma config CLKOUTEN = OFF // Clock Out disabled.
#pragma config IESO = ON // Internal/External Switchover on
#pragma config FCMEN = ON
                               // Fail-Safe Clock Monitor enabled
// CONFIG2
#pragma config WRT = OFF
                               // Flash Memory Self-Write Protect off
#pragma config PPS1WAY = ON // PPS one-way control enabled
#pragma config ZCDDIS = ON // Zero-cross detect disabled
#pragma config PLLEN = OFF // Phase Lock Loop disable
#pragma config STVREN = ON // Stack Over/Underflow Reset enabled
#pragma config BORV = LO // Brown-out Reset low trip point
#pragma config LPBOR = OFF // Low-Power Brown Out Reset disabled
#pragma config LVP = OFF // Low-Voltage Programming disabled
#define LEDS_ON_OFF 0x55 ◀─── 请参见第1.3节
void main(void) {
    ANSELD = 0x0; // set to digital I/O (not analog)
    TRISD = 0x0; // set all port bits to be output
    LATD = LEDS_ON_OFF; // write to port latch - RD[0:3] = LED[0:3]
    // Port B access
    ANSELB = 0x0; // set to digital I/O (not analog)
    TRISB = 0x0; // set all port bits to be output
    LATB = LEDS ON OFF; // write to port latch - RB[0:3] = LED[4:7]
    return;
}
```

### 1.1 头文件<xc.h>

本头文件允许源文件中的代码访问编译器或器件特定的功能。可在MPLAB XC8 安装目录的include子目录下找到本头文件和其他头文件。

编译器将根据您选择的器件设置相应的宏,以使xc.h指向正确的器件特定头文件。请 勿将器件特定的头文件包含在您的代码中,否则将导致代码不可移植。

### 1.2 配置位

Microchip器件具有配置寄存器,其中的位可用于使能和/或设置器件功能。

注: 如果未正确设置配置位,器件将无法运行,或至少不按预期运行。

要设置的配置位

请特别注意以下几项:

- **振荡器选择**——该项必须与硬件的振荡器电路匹配。如果设置有误, *器件时钟可能* 无法运行。通常情况下,开发板使用高速晶振。示例中的相关代码如下: #pragma config FOSC = ECH
- 看门狗定时器——建议在必要时禁止该定时器。这样可防止*意外复位*。示例中的相关代码如下:
   #pragma config WDTE = OFF

 代码保护——在必要时关闭代码保护。这样可确保器件存储器可完全访问。示例中的相关代码如下: #pragma config CP = OFF

使用其他8位器件(而非本示例中使用的PIC16F1719MCU)时,可能需要设置不同的 配置位。请参见具体器件数据手册了解相应配置位的名称和功能。可使用部件编号在 http://www.microchip.com上搜索相应的数据手册。

有关每款器件可用配置位的更多信息,请参见MPLAB XC8 安装路径下的如下文件: MPLAB XC8 Installation Directory/docs/chips

设置配置位的方法

在MPLAB X IDE中,可使用Configuration Bits(配置位)窗口查看和设置这些位。请 选择*Window>PIC Memory Views>Configuration Bits*(窗口>PIC存储器视图>配置位) 打开该窗口。

utj	out C	Configuration	Bits %				
	Addres	ss Name	Value	Field	Option	Category	Setting
	8007	CONFIG1	FFE7	FOSC	ECH 🚽	Oscillator Selection Bits	ECH, Externa
				WDTE	OFF	Watchdog Timer Enable	WDT disabled
				PWRTE	OFF	Power-up Timer Enable	PWRT disable
				MCLRE	ON	MCLR Pin Function Select	MCLR/VPP pin
				CP	OFF	Flash Program Memory Code Protection	Program memo
				BOREN	ON	Brown-out Reset Enable	Brown-out Re
				CLKOUTEN	OFF	Clock Out Enable	CLKOUT funct
				IESO	ON	Internal/External Switchover Mode	Internal/Ext
				FCMEN	ON	Fail-Safe Clock Monitor Enable	Fail-Safe Cl
	8008	CONFIG2	DEFF	WRT	OFF	Flash Memory Self-Write Protection	Write protec
	•			111			۰.
٩e	mory	onfiguration Bits	•]	Format Re	ad/Write	<ul> <li>Generate Source Code to Output</li> </ul>	1

完成所需设置后,请单击**Generate Source Code to Output**(生成源代码并输出), 然后将pragma伪指令从**Output**(输出)窗口复制到您的代码中,如示例代码中所示。

图1: 配置窗口

### 1.3 为LED值定义宏

如下一节所述,要写入LED的值已赋值给描述性宏(LEDS\_ON\_OFF),即LED D1、 D3、D5和D7将点亮,LED D2、D4、D6和D8将熄灭。有关电路板原理图(**第B.4节 "获取并设置Explorer 8板"**)的信息,请参见*Explorer 8 Development Board User's Guide*(DS40001812)。

### 1.4 端口访问

器件数字I/O引脚可与外设I/O引脚复用。为确保当前仅使用数字I/O, 需禁止其他外设。为此, 可使用代表外设寄存器及其位的预定义C变量。这些变量列于编译器 include 目录下的器件特定头文件中。关于哪些外设共用哪些引脚的信息, 请参见具体器件的数据手册。

对于本节中的示例,端口D和端口B引脚与默认禁止的外设复用。惟一的问题在于这些引脚默认设置为模拟I/O,因此需要将它们设置为数字I/O。对于端口D:

ANSELD = 0x0; // set to digital I/O (not analog)

器件引脚连接至器件的数字I/O端口(PORT)或锁存器(LAT)寄存器。本示例中使用 LATD和LATB。宏LEDS ON OFF赋值给两个锁存器。对于端口D:

LATD = LEDS ON OFF; // write to port latch - RD[0:3] = LED[0:3]

此外,还有一个寄存器用于指定引脚的方向是输入还是输出,它称为TRIS寄存器。本节的示例中使用TRISD和TRISB。将位设置为0可将引脚设为输出,将位设置为1可将引脚设为输入。对于端口D:

TRISD = 0x0; // set all port bits to be output

### 2. 使用\_delay()函数使LED闪烁

```
本示例在上一示例代码的基础上进行修改。本代码不仅点亮LED,还将使LED交替
闪烁。
#include <xc.h>
// PIC16F1719 Configuration Bit Settings
// For more on Configuration Bits, consult your device data sheet
// CONFIG1
#pragma config FOSC = ECH // External Clock, 4-20 MHz
#pragma config WDTE = OFF
                           // Watchdog Timer (WDT) disabled
#pragma config PWRTE = OFF // Power-up Timer disabled
#pragma config MCLRE = ON // MCLR/VPP pin function is MCLR
#pragma config CP = OFF
                           // Flash Memory Code Protection off
#pragma config BOREN = ON // Brown-out Reset enabled
#pragma config CLKOUTEN = OFF // Clock Out disabled.
#pragma config IESO = ON // Internal/External Switchover on
                            // Fail-Safe Clock Monitor enabled
#pragma config FCMEN = ON
// CONFIG2
#pragma config WRT = OFF
                            // Flash Memory Self-Write Protect off
#pragma config PPS1WAY = ON // PPS one-way control enabled
#pragma config ZCDDIS = ON // Zero-cross detect disabled
#pragma config PLLEN = OFF // Phase Lock Loop disable
#pragma config STVREN = ON // Stack Over/Underflow Reset enabled
#pragma config BORV = LO
                           // Brown-out Reset low trip point
#pragma config LPBOR = OFF // Low-Power Brown Out Reset disabled
#pragma config LVP = OFF
                           // Low-Voltage Programming disabled
#define LEDS_ON_OFF 0x05
#define LEDS OFF ON 0x0A
#define INSTR CYCLE DELAY 25000
void main(void) {
   // Port D access
   ANSELD = 0x0; // set to digital I/O (not analog)
   TRISD = 0x0; // set all port bits to be output
   // Port B access
   ANSELB = 0x0; // set to digital I/O (not analog)
   TRISB = 0x0; // set all port bits to be output
   while(1) { 一 请参见第2.1节
       LATD = LEDS_ON_OFF; // RD[0:3] = LED[0:3]
       LATB = LEDS ON OFF; // \text{RB}[0:3] = \text{LED}[4:7]
       // delay value change ◀─── 请参见第2.2节
       delay(INSTR CYCLE DELAY); // delay in instruction cycles
       LATD = LEDS OFF ON; // RD[0:3] = LED[0:3]
       LATB = LEDS OFF_ON; // RB[0:3] = LED[4:7]
       delay(INSTR CYCLE DELAY); // delay in instruction cycles
   return;
}
```

### 2.1 while() 循环和变量值

要使端口D和端口B上的LED发生变化,需在循环的第一部分为端口赋值宏 LEDS\_ON\_OFF,然后在循环的第二部分为端口赋值互补宏LEDS\_OFF\_ON。使用了 while(1) { }执行循环。

### 2.2 \_delay()函数

由于执行速度在大多数情况下都会导致LED的闪烁速度超出人眼的识别能力,因此需要降低执行速度。\_delay()是编译器的内置函数。

有关内置延时函数的更多详细信息,请参见《MPLAB<sup>®</sup> XC8 C编译器用户指南》 (DS50002053D\_CN)。

### 3. 使用中断作为延时在LED上递增计数

```
本示例在上一示例代码的基础上进行修改。尽管上一示例中的延时循环对于降低循
环执行速度很有用,但是会在程序中引入停滞时间。为避免这一问题,可使用定时器
中断。
#include <xc.h>
// PIC16F1719 Configuration Bit Settings
// For more on Configuration Bits, consult your device data sheet
// CONFIG1
#pragma config FOSC = ECH
                            // External Clock, 4-20 MHz
#pragma config WDTE = OFF // Watchdog Timer (WDT) disabled
#pragma config PWRTE = OFF // Power-up Timer disabled
#pragma config MCLRE = ON // MCLR/VPP pin function is MCLR
#pragma config CP = OFF
                             // Flash Memory Code Protection off
#pragma config BOREN = ON // Brown-out Reset enabled
#pragma config CLKOUTEN = OFF // Clock Out disabled.
#pragma config IESO = ON // Internal/External Switchover on
#pragma config FCMEN = ON
                             // Fail-Safe Clock Monitor enabled
// CONFIG2
                             // Flash Memory Self-Write Protect off
#pragma config WRT = OFF
#pragma config PPS1WAY = ON // PPS one-way control enabled
#pragma config ZCDDIS = ON // Zero-cross detect disabled
#pragma config PLLEN = OFF
                            // Phase Lock Loop disable
#pragma config STVREN = ON
                            // Stack Over/Underflow Reset enabled
                             // Brown-out Reset low trip point
#pragma config BORV = LO
#pragma config LPBOR = OFF
                             // Low-Power Brown Out Reset disabled
#pragma config LVP = OFF
                             // Low-Voltage Programming disabled
// Interrupt function <
                             - 请参见<mark>第3.1节</mark>
void interrupt isr(void) {
   // only process Timer0-triggered interrupts
   if(INTCONDits.TMR0IE && INTCONDits.TMR0IF) {
            \ensuremath{{//}} static variable for permanent storage duration
            static unsigned char portValue;
            // write to port latches
            LATD = ++portValue;
                                 // RD[0:3] = LED[0:3]
            LATB = (portValue >> 4); // RB[0:3] = LED[4:7]
            // clear this interrupt condition
            INTCONDITS.TMR0IF = 0;
   }
}
void main(void) {
    // Port D access
    ANSELD = 0x0; // set to digital I/O (not analog)
    TRISD = 0x0; // set all port bits to be output
    // Port B access
    ANSELB = 0x0; // set to digital I/O (not analog)
    TRISB = 0x0; // set all port bits to be output
```

#### 

```
OPTION_REG = 0xD7; // timer 0 internal clock, prescaler 1:256
INTCONDits.TMR0IE = 1; // enable interrupts for timer 0
ei(); // enable all interrupts
```

while(1);

return;

}

### 3.1 中断函数isr()

可使用interrupt说明符使函数成为中断函数。由于这一中断函数可能必须处理多个中断源,因此添加了相关代码以确保计数器portValue仅在Timer0产生中断时递增。

### 3.2 Timer0 设置

由于变量值更改在中断服务程序中执行,因此还需要向主程序中添加相关代码以使能 和设置定时器、允许定时器中断以及更改锁存器赋值。

要允许所有中断,请使用xc.h中定义的ei()。

### 4. 使用A/D在LED上显示电位器值

本示例与上一示例使用相同的器件以及端口B和端口D上的LED。但在本示例中,将使用演示板上电位器的值通过端口A提供A/D输入,然后进行转换并显示在LED上。

代码无需手动编写,而是使用MPLAB代码配置器(MPLAB Code Configurator, MCC) 生成。MCC 是一款插件,可在MPLAB X IDE 菜单 <u>Tools>Plugins</u>(工具>插件)的 Available Plugins(可用插件)选项卡下安装。有关如何安装插件的更多信息,请参 见MPLAB X IDE 帮助。

有关MCC的信息(包括《MPLAB<sup>®</sup>代码配置器用户指南》(DS40001725B\_CN)), 请访问MPLAB代码配置器网页:

http://www.microchip.com/mplab/mplab-code-configurator

本示例中的MCC GUI设置如以下各图所示。

<u>图2: ADC项目系统资源配置</u>

	Projects	Files Cl	asses	88	-	
	Projec	t Resourc	es	Generate		
	▼ System	m			_	
	Sys	tem Module				
	Pin	Module				
	Inte	errupt Module	e			
Start Page 🛛 🗱 💓 MPL	AB X Store 🛛 🗱 🕅	PLAB® Code Co	nfigurator	86		
System Module			2			
Easy Setup	aisters 🔥 Notificat	tions:0				
▼ INTERNAL OSCILLAT	OR					
Current System clock 8	3 MHz					
Oscillator Select	ECH, External Clock,	High Power Mo	ode (4-32	MHz): devic	e clock s	upplied to CLKIN pin 🔹
System Clock Select	FOSC 🔫					
Internal Clock	500KHz_MF		LL Capabl	e Frequency	,	
External Clock	8 MHz					
	Software PLL Enal	bled				
PLL Enabled						
PLL Enabled						
PLL Enabled	ramming Enable					
PLL Enabled     Low-voltage progr      WDT	ramming Enable					
PLL Enabled     Low-voltage progi     WDT     Watchdog Timer Enable	e WDT disabled					•

# 图3: ADC项目ADC资源配置

Device Resource	s		Projects Fil	es Class	ses	26
Documents			Project I	Resources	Gene	rate
PIC16F1719 Datas	heet		▼ System			
Peripherals			U Syster	n Module		
🔻 🔦 ADC			Pin M	odule		
ADC	Double Click Here	e .	Interru	upt Module		
CCP			▼ Peripher	als		
🕨 🦫 CLC			А 🕀 🔺	DC		
🕨 🗠 COG			~			
DC	gisters 🔥 Notifications	:1	_			
ardware Settings						
Enable ADC						
ADC Clock			Descrit Allegenerat	1-4		_
			Result Alignment	leit		
Clock Source	FOSC/2 💌		Positive Reference	VDD		*
1 TAD	250.0 ns		Negative reference	VSS		-
Sampling Frequency	347.8261 kHz		Auto-conversion Trig	ger no auto	trigger	-
Conversion Time	= 11.5 * TAD = 2.875	us				
Enable ADC Interrup	ot		]			
Selected Sharenels						
Selected Channels						
Pin	Channel			Custom Nar	ne	
Internal Channel	DAC2_Output	channe	I_DAC2_Output			
Internal Channel	DAC1_Output	channe	I_DAC1_Output			
Internal Channel	FVRBuffer1	channe	I_FVRBuffer1			
Internal Channel	Temp	channe	I_Temp			
RAO	ANO	channe	I_AN0			
< C						

Package:	PDIP40	-		Pin No:		2	3	4	5	6	7	14	13	33	34	35	36	37	38	39	4
		_						-	Port	A V	-	-	-		-	-	Port	В▼		_	
Module	Fu	nctio	n	Direction	n	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	•
	ANx		ir	nput		â	î.	î	î.		î			î.	î.	î.	ĵ.	ĵ.	î.		Γ
ADC 🔻	VREF+		ir	nput 🦯	7				ĵ,												
	VREF-		ir	nput				Ĵ.													
	CLKIN		ir	nput									â								
055	CLKOL	л	0	utput								î.									
OSC V	OSC1		ir	nput									î.								
	OSC2		ir	nput								ĵ.									
Die Markela 🖛	GPIO		ir	nput	1	ì	î.	î.	î.	î	î.	î.	î.	î.	1						
Pin Wodule V	GPIO		o	utput	1	è	ĵ.	î	î	î	î	î	î.	î.	î	î.	ĵ,	ĵ,	î.	ì	1
RESET	MCLR		ir	nput																	

# 图4: ADC项目ADC引脚资源——网格

# 面向嵌入式工程师的MPLAB® XC8用户指南

		F	Projects File	es Cla	sses	× -	]			
			Project R	lesource	s Ge	enerate				
		Γ	▼ System							
			System	Module						
			Pin Mo	odule						
			Interru	pt Module			1			
			<ul> <li>Periphera</li> </ul>	als						
			AD	с		×	l			
			-				'			
		L								
		L								
Start Page	8 🋒 MPLAB X Str	ore 🛚 🖾 MPLA	B® Code Configura	tor %					< >	•
Start Page <b>Pin Modu</b>	88 🛒 MPLAB X Sto	ore 🛚 🕅 MPLA	B® Code Configura	tor 🛛			_		<>	•
Start Page Pin Modu {ট্টু Easy Set	8 🛒 MPLAB X Sto le up 📄 Registers	ore 8 MB MPLA	B® Code Configura	tor 🕺						•
Start Page <b>Pin Modu</b> Selected Pack: Bin Name	MPLAB X Sto le up E Registers age : PDIP40	ore 8 MBB MPLA	B® Code Configura	tor 🔉	Analog	Output		0.0		
Start Page <b>Pin Modu</b> Selected Packa <b>Pin Name</b> RA0	MPLAB X Str le up Registers age : PDIP40 Module ADC	ore 80 MBB MPLA	B® Code Configura	tor 28	Analog	Output	WPU	OD		
Start Page Pin Modu Selected Pack Pin Name RA0 BB0	MPLAB X Sta  MPLAB X Sta  Pup Registers  age : PDIP40  Module  ADC  Pin Module	Notification	B® Code Configura Is : 1 Custom Name Channel_AN0 To ano	tor 28 Start High	Analog V	Output	WPU	OD	IOC none	
Start Page Pin Modu Selected Packa Pin Name RAO RBO PP1	MPLAB X Str    MPLAB X Str      MPLAB X Str	Pre 80 MBB MPLA Notification Function AN0 GPIO	B® Code Configura Ins : 1 Custom Name Channel_AN0 IO_RB0	tor 28	Analog	Output	WPU	OD	IOC none none	
Start Page Pin Modu Selected Pack Pin Name RA0 RB0 RB1 P22	MPLAB X Str    MPLAB X Str     Module   Pin Module Pin Module Pi	ANO GPIO CIECO	B® Code Configura Is : 1 Custom Name Channel_AN0 IO_RB0 IO_RB1	tor %	Analog	Output	WPU	OD	IOC none none none	
Start Page Pin Modu Selected Pack Pin Name RA0 RB0 RB1 RB2 PD2	MPLAB X Str    MPLAB X Str      Module	AND AND GPIO GPIO GPIO	B® Code Configura Is : 1 Custom Name channel_ANO 10_RB0 10_RB1 10_RB2	tor 🔉	Analog	Output	WPU	OD	IOC none none none none	
Start Page Pin Modu Selected Pack Pin Name RA0 RB0 RB1 RB2 RB3	<ul> <li>MPLAB X Strain</li> <li>MPLAB X Strain</li> <li>PDIP40</li> <li>Module</li> <li>Pin Module</li> <li>Pin Module</li> <li>Pin Module</li> <li>Pin Module</li> <li>Pin Module</li> </ul>	ANO GPIO GPIO GPIO GPIO GPIO	B® Code Configura Is : 1 Custom Name Channel_ANO IO_RB0 IO_RB1 IO_RB2 IO_RB3	tor %	Analog	Output       ✓       ✓       ✓       ✓       ✓       ✓	WPU	OD	IOC none none none none none	
Start Page Pin Modu Selected Pack Pin Name RA0 RB0 RB1 RB2 RB3 RD0	<ul> <li>MPLAB X Str</li> <li>Registers</li> <li>PDIP40</li> <li>Module</li> <li>ADC</li> <li>Pin Module</li> </ul>	AND AND GPIO GPIO GPIO GPIO GPIO GPIO GPIO	B® Code Configura Ins : 1 Custom Name Channel_ANO IO_RB0 IO_RB1 IO_RB2 IO_RB3 IO_RD0	tor 28	Analog	Output           Image: Contract of the second seco	WPU	OD	IOC none none none none none	
Start Page Pin Modu Selected Pack Pin Name RA0 RB0 RB1 RB2 RB3 RD0 RD1	<ul> <li>MPLAB X Strain</li> <li>PDIP40</li> <li>Module</li> <li>Pin Module</li> </ul>	Notification    Notification	B® Code Configura Is : 1 Custom Name Channel_ANO IO_RB0 IO_RB1 IO_RB2 IO_RB3 IO_RD0 IO_RD1	tor Start High	Analog	Output           Image: Constraint of the second of	WPU	OD	IOC none none none none none	
Start Page Pin Modu Selected Pack Pin Name RA0 RB0 RB1 RB2 RB3 RD0 RD1 RD2	<ul> <li>MPLAB X Str</li> <li>Registers</li> <li>PDIP40</li> <li>Module</li> <li>ADC</li> <li>Pin Module</li> </ul>	ANO ANO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO	B® Code Configura Custom Name Channel_ANO 10_RB0 10_RB1 10_RB2 10_RB3 10_RD0 10_RD1 10_RD2	tor 28	Analog	Output           Image: Construction of the second	WPU	OD	IOC none none none none none	
Start Page Pin Modu Selected Pack Pin Name RA0 RB0 RB1 RB2 RB3 RD0 RD1 RD2 RD3	<ul> <li>MPLAB X Strategy</li> <li>Registers</li> <li>PDIP40</li> <li>Module</li> <li>Pin Module</li> </ul>	Notification Notification Punction AN0 GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO GPIO	B® Code Configura Is : 1 Custom Name Channel_ANO IO_RB0 IO_RB1 IO_RB2 IO_RB3 IO_RD3 IO_RD3	tor %	Analog	Output           ✓	WPU	OD	IOC none none none none	

# 面向嵌入式工程师的MPLAB<sup>®</sup> XC8用户指南

Output - MPL/	AB® (	ode (	Confi	oura	tor	Pin	Mar	ade	er: Gi	id []	исс	*											
Package:	PD	P40	-	]	Pin	No:		2	3	4	5	6	7	14	13	33	34	35	36	37	38	39	40
-	-		_	-		_		_			Por	t A 🖲	,				-		Po	rt B	,		-
Module		Fu	nctio	n	Dire	ction		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
	_	ANx			input			â	î.	î.	Ъ		î.			ษ	Ъ	î.	n,	î.	°.		
ADC 🔻		VREF+			input						î.												1
		VREF-			input					Ъ			1	$\square$						+	1		1
		CLKIN			input										â								
		CLKO	JT		output									ì									
OSC 🔻		OSC1			input						$\square$	$\square$	1		Ъ					+			$\top$
		OSC2			input						$\square$	$\square$		î.						+			$\top$
	_	GPIO			input			î.	î.	î.	Ъ	ĵ.	î.	î.	Ъ	î.	Ъ	î.	n.	î.	î.	î	P
Pin Module	•	CRIO						2	2	2	2	2	2	2	2	0	Δ		0	2	2	2	2
		GFIO			output			÷										1		1			
RESET		MCLR			input													2					
RESET Output - MPLAE Package:	3® Coo PDIP40		figura	itor P	Pin M	anage 15	er: G	rid [1	MCC]	× 23	24	25	26 1	9 20	21	22	27	28	29	30	3 9	10	-
RESET Output - MPLAE Package:	3® Coc PDIP40		figura	itor P	Pin M	anage 15	er: G 16	rid [1	MCC] 18 Port	23 C▼	24	25	26 1	9 20	21	22 Port	27 D V	28	29	30	B 9 Por	10 t E ▼	-
RESET Output - MPLAE Package: Module	3® Coo PDIP40	MCLR	figura	tor P Di	Pin M in No:	anage 15 0	er: G 16	rid [17	MCC] 18 Port 3	23 C▼ 4	24	25	26 1	9 20	21	22 Port 3	27 D▼ 4	28	29 3	30 8	<ul> <li>9</li> <li>Por</li> <li>1</li> </ul>	10 t E ▼ 2	= 1 3
RESET Output - MPLAE Package: Module	3® Coo PDIP40	MCLR	figura	tor P Di input	Pin M in No:	anage 15 0	er: G 16	rid [1 17 2	MCC] 18 Port 3	<ul> <li>≈</li> <li>23</li> <li>C ▼</li> <li>4</li> <li>1</li> <li>1</li> </ul>	24	25 2 6	26 1 7 1	9 20 0 1	21	22 Port 3	27 D V 4	28	29 :	30 a 7 0	9 Por 1	10 t E ▼ 2 10	<b>1</b>
RESET	3® Coc PDIP40 AN VR	MCLR	figura	itor P Di input	Pin M in No:	anage 15 0	er: G 16	rid [1 17	MCC] 18 Port 3	≈ 23 C ▼ 4 23	24	6 6 1 1 1 1 1 1 1 1 1 1 1 1 1	26 1 7 1	9 20 0 1 9 1	21	22 Port 3	27 D V 4	28	29 : 6	30 a 7 0	9 Por 1 2	10 t ∈ ▼ 2 10	1
RESET	3® Coo PDIP40 AN VR VR	Functi x EF+	figura	Di input input	Pin M in No:	anage 15 0	er: G 16	rid [1 17 2	MCCC] 18 Port 3 18	23 C▼ 4	24	25 6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		9 20	21	22 Port 3 1 2	27 D▼ 4 1 27	28 5 1 1 28	29 : 6 <b>1</b>		9 Por 1	10 t E ▼ 2 10	<b>1</b>
RESET	Be Coo PDIP40 AN VR CL	Functi x EF+ EF- CIN	figura	Di input input input	Pin M in No: rection	anage 15 0	er: G 16	ea rid [1 17 2 2 2	MCCC] 18 Port 3 C	23 C ▼ 4	24	6 1 1 1 1 1 1 1 1 1 1 1 1 1		9 20 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	21	22 Port 3	27 D V 4	28	29 : 6 1		B           9           Por           1           B	10 ± E ▼ 2 2 10 10 10 10 10 10 10 10 10 10	1
RESET Output - MPLAE Package: Module ADC ▼	38 Coo PDIP4( AN VR VR VR CL CL OS	He Cont MCLR Functi x EF+ EF- CIN	figura	tor P Di input input input outp input	Pin M in No: rection	anago 15 0	16	2 2	MCC] 18 Port 3 2	8 23 C ▼ 4 1 1 1	24	6 6 1 1 1 1 1 1 1 1 1 1 1 1 1		9 200 1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1	21	22 Port 3	27 D▼ 4 1 27	28 5 6			9 Por 1 1 1 9	10 t E ▼ 2 2	1
RESET Output - MPLAE Package: Module ADC ▼	AN VR VR CL CL OS OS	MCLR MCLR Function Function K K K K K K K K K K K K K	figura	Di input input input input input	Pin M in No: rection	15 0 0	er: G 16	rid [1 17 2 2	MCCC] 18 Port 3 2 2	23 C▼ 4 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	24	6 6 1 1 1 1 1 1 1 1 1 1 1 1 1			21	22 Port 3 2	27 D ¥ 4	228	29 : 6 1 1 1		<ul> <li>B</li> <li>9</li> <li>Por</li> <li>1</li> <li>B</li> <li>9</li> </ul>		1
RESET	38 Coc PDIP4( AN VR CL CL CL CS OS	Functi           x           FF+           EF+           GIN           COUT           C1           C2           IO	on	Di input input input input input input	Pin M in No: rection	anago 15 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	er: G 16	E (1)	MCCCJ 18 Port 3 19	23 C ▼ 4 1 23						22 Port 3 1 2	27 D▼ 4 1 1 27				8           9           Por           1	10 t E ▼ 2 7 10	1
RESET Output - MPLAE Package: Module ADC ▼ OSC ▼ Pin Module ▼	AN VR CL CL OS OS	Be Con           MCLR           Be Con           Functi           X           EF+           EF+           EF-           CI           C2           IO           IO	on	Di input input input input input input outp	Pin M in No: rection	anage 15 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	er: G 16 1		MCCCJ 18 Port 3 Cu	23 C ▼ 4 1 23 C ▼ 4 1 23 0 23 0 23 0 23 0 23 0 23 0 23 0 23						22 Port 3 6	27 D▼ 4 1 1 2 3				8         9           Por         1           1         10           1         10           1         10           1         10           1         10           1         10           1         10           1         10           1         10		

# 图7: ADC项目GPIO引脚资源——封装



# 面向嵌入式工程师的MPLAB<sup>®</sup> XC8 用户指南

按照上述各图配置完代码后,请单击"Project Resources"(项目资源)窗口上的 Generate(生成)按钮。通过MCC生成的代码是模块化的。因此,主程序代码、系统 代码和外设代码均位于单独的文件中。此外,每个外设都有自己的头文件。

向程序中添加功能时始终需要编辑main.c。请查看生成的文件以找到您的代码中可能 需要的任何函数或宏。

Projects <sup>88</sup> Files Services Classes	-
Example4	
🖨 ि Header Files	
🖮 📠 MCC Generated Files	
🖭 adc.h	
- 🖭 mcc.h	
🖭 pin_manager.h	
🖨 🕞 Important Files	
MCC Configuration	
🛱 庙 Source Files	
🔁 main.c	
🗄 🖓 🔂 MCC Generated Files	
🖭 adc.c	
🖭 mcc.c	
pin_manager.c	
🕞 Libraries	
Loadables	

图8: 通过MCC生成的代码的ADC项目树

### 4.1 main.c修改后的代码

```
main.c模板文件已经过编辑,如下所示。部分注释用< >括起,表示已删除。新添加
到main()中的代码以红色显示。
/**
 Generated Main Source File
<See generated main.c file for file information.>
*/
/*
 (c) 2016 Microchip Technology Inc. and its subsidiaries. You may use
 this software and any derivatives exclusively with Microchip
 products.
<See generated main.c file for additional copyright information.>
*/
#include "mcc_generated_files/mcc.h"
/*
                    Main application
*/
void main(void) {
   // initialize the device
   SYSTEM Initialize();
   // <No interrupts used - see generated main.c file for code.>
   while (1) {
      ADC StartConversion(channel AN0);
      while(!ADC IsConversionDone());
      LATD = ADRESH; // RD[0:3] = LED[0:3]
      LATB = (ADRESH >> 4); // RB[0:3] = LED[4:7]
   }
}
/**
End of File
*/
```

### 4.2 启动A/D转换

使用adc.c模块中的如下函数: void ADC\_StartConversion(adc\_channel\_t channel) 变量channel属于adc.h中定义的**typedef**adc\_channel\_t。对于本示例,电位器输 入位于**RA0**上,因此选择channel AN0。

### 4.3 等待ADC完成

使用adc.c模块中的如下函数:

bool ADC\_IsConversionDone()

该函数会返回ADCON0bits.GO\_nDONE位(在器件头文件中定义)的取反值。但是, main while循环中需要该位的实际值,因此需要将返回值再次取反。

# 4.4 写入端口锁存器

由于只有8个LED,因此仅显示ADRESH的值。低4位通过LATD显示在LED0至LED3 上,高4位会先进行移位,以便能够通过LATB显示在LED4至LED7上。

# 5. 在LED上显示EEPROM数据值

本示例使用另一款Microchip器件PIC16F1939 MCU来演示如何读写EEPROM数据 (EEData)。读取的值显示在端口D和端口BLED上。

再次使用MPLAB代码配置器(MCC)来生成大部分代码。欲了解如何安装MCC和获取MCC用户指南的信息,请参见:

第4节"使用A/D在LED上显示电位器值"。

本示例中的MCC GUI设置如以下各图所示。

### 图9: EEDATA项目系统资源配置

	Projects Files Classes % 🖃	
	Project Resources Generate	
	▼ System	
	System Module	
	Pin Module	
	Interrupt Module	
Start Page 🛛 🗱 🛒 MPL	AB X Store III MPLAB® Code Configurator III	
System Module		
දිටු Easy Setup	gisters 🔥 Notifications : 0	
▼ INTERNAL OSCILLAT	DR	
Current System clock 8	MHz	
Oscillator Select	ECH, External Clock, High Power Mode (4-32 MHz): device clock supplied to CLKIN	l pin 👻
System Clock Select	FOSC -	
Internal Clock	500KHz_MF	
External Clock	8 MHz	
PLL Enabled	Software PLL Enabled	
Low-voltage prog	amming Enable	
▼ WDT		
Watchdog Timer Enabl	e WDT disabled	

# 图10: EEDATA项目存储器资源配置

Memory     MEMORY     MEMORY     SRLatch     SRLatch     System     System Module   Pin Module   Interrupt Module   Peripherals     MEMORY     IN     Interrupt Module     Peripherals     MEMORY     Interrupt Module     Peripherals     MEMORY     Interrupt Module     Peripherals     MEMORY     Interrupt Module     Peripherals     MEMORY     Iterrupt Module     Peripherals     MEMORY     Iterrupt Module     Peripherals     Iterrupt Module     Peripherals     Iterrupt Melde X Store     MEMORY     Iterrupt Module     Peripherals     Iterrupt Melde X Store     Iterrupt Melde X Store	Memory   MEMORY   Double Click Here   SRLatch   Timer  braries   Bootloader Generator   IN   Ethernet     MEMORY     Memory     Project Resources   System   System   System   System   System   System   System   System   System   Notifications : 0     Flash Routines are included   Image   Add DataEE Routines	vice Resources	Projects	Files	Classe	s %
MEMORY Double Click Here SRLatch Timer aries Bootloader Generator LIN Ethernet MEMORY MPLAB X Store 18 1000 MPLAB® Code Configurator 18 10 10 10 MEMORY Basy Setup Registers A Notifications : 0 Flash Routines are included Add DataEE Routines	MEMORY Double Click Here SRLatch Timer aries Bootloader Generator LIN Ethernet MEMORY Easy Setup Registers Notifications : 0 Flash Routines are included Add DataEE Routines	Memory	Projection	ct Resou	irces	Generate
SRLatch Timer ies Bootloader Generator LIN Ethernet MEMORY Ethernet MEMORY Easy Setup Registers Notifications : 0 Flash Routines are included Add DataEE Routines	SRLatch Timer ies Bootloader Generator LIN Ethernet MEMORY Easy Setup Registers Notifications : 0 Flash Routines are included Add DataEE Routines	MEMORY Double Click Here	▼ Syste	em		
Pin Module Interrupt Module Peripherals MEMORY ernet MEMORY EASY Setup Registers Notifications : 0 ash Routines are included Add DataEE Routines	eret Pin Module Interrupt Module Peripherals MEMORY ernet MEMORY Easy Setup Registers Notifications : 0 ash Routines are included Add DataEE Routines	Latch	Sys	stem Modu	ile	
Interrupt Module Peripherals MEMORY	Interrupt Module Peripherals MEMORY MPLAB X Store IN MPLAB® Code Configurator IN CONFIGURATION MEMORY Easy Setup Registers Notifications : 0 Flash Routines are included Add DataEE Routines	Timer	Pin	n Module		
Bootloader Generator LIN Ethernet age MPLAB X Store MPLAB® Code Configurator Memory MEMORY Easy Setup Registers Notifications : 0 Flash Routines are included Add DataEE Routines	Bootloader Generator LIN Ethernet	es	Int	errupt Mo	dule	
N hernet age MPLAB X Store  MPLAB® Code Configurator  MEMORY MEMORY Easy Setup Registers Notifications : 0 Hash Routines are included Add DataEE Routines	N hernet MEMORY age MPLAB X Store N MPLAB® Code Configurator N I I I I MEMORY Easy Setup Registers Notifications : 0 Hash Routines are included Add DataEE Routines	ootloader Generator	▼ Perip	herals		
thernet	thernet	IN		MEMORY	,	
age MPLAB X Store     MEMORY     Image     Registers     Notifications : 0     Flash Routines are included     Image     Add DataEE Routines	age MPLAB X Store   MEMORY     Image     Registers     Notifications:0     Flash Routines are included     Image     Add DataEE Routines	Ethernet	<u>_</u>			
		Add DataEE Routines				

# 面向嵌入式工程师的MPLAB® XC8用户指南

#### 图11: EEDATA项目引脚资源配置

Pin Module

Pin Name 🔺

Pin Module

GPIO

GPIO

GPIO

GPIO

GPIO

GPIO

GPIO

RBO

RB1

RB2

RB3

RDO

RD1

RD2

RD3



none

none

none

Ŧ

Ŧ

Ŧ

 $\checkmark$ 

 $\checkmark$ 

 $\checkmark$ 

 $\checkmark$ 

 $\checkmark$ 

 $\checkmark$ 

 $\checkmark$ 

当图12中选中引脚RB0:3和RD0:3后,这些引脚将出现在上面的窗口中。 在窗口中显示后,可为每个引脚选择引脚配置。

IO\_RB1

IO\_RB2

IO\_RB3

IO\_RD0

IO\_RD1

IO\_RD2

IO\_RD3

# 图12: EEDATA项目GPIO引脚资源——网格

par	Pin	mana	iger:	Grid	INC										-											
Packa	age:	PDIP	40	*		Pin I	No:		2	3	4	5	6	7	14	1	3	33	34	35	36	37	1 3	38	39	40
					_							Рог	t A V	,							Por	t B '	•			
Mo	dule		Fur	ictio	n	Direc	tion		0	1	2	3	4	5	6		7	0	1	2	3	4		5	6	7
		0	LKIN			input										1	â									
		C		т		output									î											
OSC	C▼	C	DSC1			input								Γ		1						Τ	Τ			
		c	DSC2			input									î.								1			
		0	SPIO			input		'	è	ĵ	î	ĥ	ĥ	î.	î	1		Ĵ.	Ĵ.	î.	î	î.	ı î		î.	ĥ
Pin Mo	dule N		SPIO			output		'	è i	ĥ	î.	î.	î.	î.	n n	1		â	ô	â	â		1		î.	î.
		N	NCLR			input												1000			-	-				
BECE																										
NEDI		١	/CAP			input		'	à					ì	ì	I										
utput Packao	Pin M	anage	/CAP r: Grid	[MC	C] %	input	15	16	17	18	23	24	25	26	19	20	21	22	27	28	29	30	8	9	10	1
utput Packag	Pin Ma ge:	anage	/CAP r: Grid	[MC	C] ※	input in No:	15	16	17	18 Por	23 t C ▼	24	25	26	19	20	21	22 Port	27 t D ▼	28	29	30	8	9 Por	10 t E ▼	1
utput Packag Modu	Pin M. ge: F	anage PDIP40	/CAP r: Grid	[MC	C] % F Di	input Pin No:	15	16	17	18 Por 3	23 t C ▼ 4	24	25	26	19	20	21	22 Port	27 t D ▼ 4	28	29	30	8	9 Por	10 t E ▼ 2	1
utput Packag Modu	Pin M ge: F	anage PDIP40	r: Grid	[MC	C] % F Di inpu	in put 'in No: irection	15	16	<b>1</b> 7	18 Por 3	23 t C ▼ 4	24	25	26	19 0	20	21	22 Port	27 t D ▼ 4	28	29	30	8	9 Por 1	10 t E ▼ 2	1
ncol Packag Modu	Pin M. ge: F ule	anage PDIP40	/CAP	[MC	C] % F Di inpu outp	input in No: irection t ut	15 0	16	17 2	18 Por 3	23 t C ▼ 4	24	25	26	19 0	20	21	22 Port	27 D▼ 4	28	29	30	8	9 Por 1	10 t E ▼ 2	1
Modu OSC	Pin M. ge: F ule	anage PDIP40	r: Grid	[MC	C] % F Di inpu outp	input in No: rection t ut	15	16	17 2	18 Por	23 t C ▼ 4	24	25	26	19 0	20	21	22 Port	27 t D ▼ 4	28	29	30	8	9 Por 1	10 t E ▼ 2	3
utput Packag Modu	Pin M. ge: F ule	anage PDIP40 CLK1 CLK2 OSC	Function IN OUT	[MC	C] % F Di inpu outp inpu inpu	input in No: irection t ut t	15	16	2	18 Port	23 t C ▼ 4	24	25	26	0	20	21	22 Port	27 ± D ▼ 4	28	29 6	30	8	9 Por 1	10 t E ▼ 2	1
Modu	Pin M. ge: F ule	anage popP40 CLKX CLKX OSC OSC	r: Grid	[MC	C] % F Di inpu inpu inpu inpu	input in No: irection t ut t t	15	16		18 Port 3	23 t C ▼ 4	24	25	26	0	20	21 2	22 Port	27 27 4 4	28	29	30 7	8 0	9 Por 1	10 t E ▼ 2	3
Modu Packag OSC 1	Pin M. ge: F ule	Anage PDIP40 I CLK2 CLK2 CLK2 CLK2 CLK2 CLK2 CLK2 CLK2	r: Grid Function IN OUT 1 2 D D	[MC	C] % F Di inpu inpu inpu inpu outp	input in No: irection t ut t t t ut	15 0 1 1 1 1 1 1 1 1 1			18 Port 3 2 2 2 2 2 2 2 2 2 2 2 2 2	23 t C ▼ 4 1 1 1	24	25 6 18	26 7	19 0	20 1	21 2	22 Port 3	27 t D ▼ 4	28	29 6 19	30 7	8 0	9 Por 1	10 t E ▼ 2 10	1
Modu Pin Modu	Pin M. ge: F ule	Anage PDIP40	r: Grid Functic IN OUT 1 2 D D R	[MC	C] X F Di inpu inpu inpu inpu outp inpu	input in No: rection t ut t t t t t	15 0 1 1 0	16		18 Port 3 2	23 t C ▼ 4 7 10 7 10	24	25 6 9	26 7	19 0	20	21 2	22 Port	27 ± D ▼ 4 1 1 1 1 1 1 1 1 1 1 1 1 1	28	29 6 9	30 7	8 0	9 Por 1	10 t E ▼ 2 6 7 8	

# 图13: EEDATA项目GPIO引脚资源——封装



# 面向嵌入式工程师的MPLAB<sup>®</sup> XC8 用户指南

按照上述各图配置完代码后,请单击"Project Resources"(项目资源)窗口上的 Generate(生成)按钮。通过MCC生成的代码是模块化的。因此,主程序代码、系统 代码和外设代码均位于单独的文件中。此外,每个外设都有自己的头文件。

向程序中添加功能时始终需要编辑main.c。请查看生成的文件以找到您的代码中可能 需要的任何函数或宏。



图 14: 通过 MCC 生成的代码的 EEDATA 项目树

### 5.1 main.c修改后的代码

```
main.c模板文件已经过编辑,如下所示。部分注释用< >括起,表示已删除。新添加
的代码以红色显示。
/**
 Generated Main Source File
<See generated main.c file for file information.>
*/
/*
 (c) 2016 Microchip Technology Inc. and its subsidiaries. You may use
 this software and any derivatives exclusively with Microchip
 products.
<See generated main.c file for additional copyright information.>
*/
#include "mcc_generated_files/mcc.h"
#define NUM EE VALUES 64
#define INSTR CYCLE DELAY 25000
/*
                       Main application
*/
void main(void) {
   // initialize the device
   SYSTEM Initialize();
   // <No interrupts used - see generated main.c file for code.>
   // Declare RAM array, loop variable 🚽 —— 请参见第5.2节
   volatile unsigned char RAMArray[NUM EE VALUES];
   unsigned char i;
   PIR2bits.EEIF = 0x0; // clear write flag
   for(i=0; i<NUM EE VALUES; i++) {</pre>
       DATAEE WriteByte ( EEADRL EEADRL POSN + i, i);
       while(!PIR2bits.EEIF); // check for write finished
       PIR2bits.EEIF = 0x0;
   }
   while(1){
       // Read from EEPROM and display 🚽 —— 请参见第5.4节
       for(i=0; i<NUM_EE_VALUES; i++) {</pre>
           RAMArray[i] = DATAEE ReadByte( EEADRL EEADRL POSN + i);
          LATD = RAMArray[i]; // RD[0:3] = LED[0:3]
          LATB = (RAMArray[i] >> 4); // RB[0:3] = LED[4:7]
           delay(INSTR CYCLE DELAY); // delay value change
       3
```

```
// Write to EEPROM in reverse order
for(i=0; i<NUM_EE_VALUES; i++) {
    DATAEE_WriteByte(_EEADRL_EEADRL_POSN +
        (NUM_EE_VALUES - 1) - i, RAMArray[i]);
    while(!PIR2bits.EEIF); // check for write finished
        PIR2bits.EEIF = 0x0;
    }
  };
}
}
/**
End of File
*/
```

### 5.2 EEData 相关变量

用于存储EEData读/写数据的变量必须与读/写函数原型中指定的类型匹配,函数原型 由mcc.h引用且可在memory.h中找到:

```
void DATAEE_WriteByte(uint8_t bAdd, uint8_t bData);
uint8_t DATAEE_ReadByte(uint8_t bAdd);
```

从stdint.h引用时, uint8 t与unsigned char相同。

### 5.3 写入EEData

本示例中两次写入EEData: 第一次用于初始化EEData存储器中的值,第二次用于更改数据以实现动态显示。

写入EEData 需要多个周期,因此使用写完成标志来确定写操作何时完成 (PIR2bits.EEIF)。该标志最初为清零状态,并且会在每次写操作完成时再次清 零。(该标志必须用软件清零。)

#### 5.4 读取 EEData

写入EEData后,存储器值会被读入RAM数组,然后显示在端口D和端口BLED上。该 写循环中使用RAM数组中的值更改EEData存储器中的值。

由于执行速度在大多数情况下都会导致LED的闪烁速度超出人眼的识别能力,因此需要再次使用\_delay()函数(同例2)来降低执行速度。

# A. 在MPLAB X IDE 中运行代码

按如下步骤创建一个项目:

- 1. 启动MPLAB X IDE。
- 2. 从IDE中启动新建项目向导(*File>New Project*(文件>新建项目))。
- 3. 按照屏幕提示创建一个新项目:
  - a) **Choose Project (选择项目):** 选择 "Microchip Embedded" (Microchip 嵌入式), 然后选择 "Standalone Project" (独立项目)。
  - b) Select Device (选择器件): 选择示例器件。
  - c) Select Header (选择调试头):无。
  - d) Select Tool(选择工具):选择硬件调试工具,SNxxxxx。如果调试工具名称下未显示序列号(Serial Number,SN),请确保调试工具已正确安装。 有关详细信息,请参见调试工具文档。
  - e) Select Plugin Board (选择接插板):无。
  - f) Select Compiler (选择编译器):选择XC8(最新版本号)[bin文件夹位置]。如果XC8下未显示编译器,请确保编译器已正确安装且MPLAB X IDE已获知(<u>Tools>Options</u>(工具>选项),Embedded(已安装工具)按钮,Build Tools(编译工具)选项卡)。有关详细信息,请参见MPLAB XC8和MPLAB X IDE文档。

g) Select Project Name and Folder(选择项目名和文件夹):为项目命名。

创建完项目后,根据您使用的示例执行以下操作之一:

- 1. 对于示例1、2和3,创建一个文件来保存示例代码:
  - a) 右键单击Projects(项目)窗口中的项目名称。选择<u>New>Empty\_File</u>(新建>空白文件)。随即将显示New Empty File(新建空白文件)对话框。
  - b) 在 "File name" (文件名) 下输入名称。
  - c) 单击Finish (完成)。
  - d) 从本用户指南中剪切示例代码并粘贴到空白编辑器窗口中,然后选择 *File>Save*(文件>保存)。
- 2. 对于示例4和5,请按照每个部分中的说明使用MCC生成代码,然后用所显示的 代码编辑main.c文件。

最后,编译代码并下载到器件中,然后选择Debug Run(调试运行)执行代码。演示 板上的LED将交替点亮。单击Halt(暂停)结束执行。



Debug Main Project	Finish Debugger Session (Shift+F5)
调试运行	暂停

# B. 获取软件和硬件

对于本文档中的MPLAB XC8项目,装有PIC16F1719或PIC16F1939 MCU的Explorer 8板由9V外部电源供电,并使用标准(ICSP™)通信。使用MPLAB X IDE进行开发。

### B.1 获取MPLAB X IDE和MPLAB XC8 C编译器

可从以下网址找到MPLAB X IDE v3.35及以上版本: http://www.microchip.com/mplab/mplab-x-ide 可从以下网址找到MPLAB XC8 C编译器v1.38及以上版本:

http://www.microchip.com/mplab/compilers

### B.2 获取MPLAB代码配置器(MCC)

可从以下网址找到MCC v3.15及以上版本: http://www.microchip.com/mplab/mplab-code-configurator

# B.3 获取PIC<sup>®</sup> MCU

可从以下网址找到本示例中使用的PIC MCU: http://www.microchip.com/PIC16F1719 http://www.microchip.com/PIC16F1939

### B.4 获取并设置Explorer 8板

可从以下网址找到Explorer 8开发工具包(DM160228): http://www.microchip.com/DM160228

下表列出了跳线设置。

#### <u>表1-1: 项目的跳线选择</u>

跳线	选择	说明
J2	BRD+5V	通过电源(而非USB)为开发板供电
J14	+5V	器件电源电压
J24	开路	使用+5V(而非3.3V)
J7	闭合	使能端口D <rd0:3>上的LED</rd0:3>
J21	闭合	使能端口B <rb0:3>上的LED</rb0:3>
J36	OSC1至RA7	OSC1 CLKIN(8 MHz外部振荡器)
J37	OSC2至RA6	OSC2 CLKOUT(8 MHz外部振荡器)
J51	PGD至RB7	ICSPDAT
J52	PGC至RB6	ISCPCLK

#### 表1-2: 未使用的跳线选择

跳线	选择	说明
JP2	闭合	不使用LCD
J22、J23、J53和J54	开路	不使用LCD
J15和J16	开路	不使用Digilent Pmod™连接器
J43、J44、J45、J46和J47	开路	不使用mikroBUS
J41、J42、J48、J49和J50	开路	不使用mikroBUS
J4和J31	VCAP	不使用 RA5 和 RA4

# B.5 获取Microchip调试工具

可在如下开发工具网页中找到仿真器和调试器: http://www.microchip.com/development-tools

#### 请注意以下有关 Microchip 器件代码保护功能的要点:

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip确信:在正常使用的情况下, Microchip系列产品是当今市场上同类产品中最安全的产品之一。
- 目前,仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知,所有这些行为都不是以 Microchip 数据手册中规定的 操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是"牢不可破"的。

代码保护功能处于持续发展中。 Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视 为违反了 《数字器件千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下,能访问您的 软件或其他受版权保护的成果,您有权依据该法案提起诉讼,从而制止这种行为。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分,因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便 利,它们可能由更新之信息所替代。确保应用符合技术规范, 是您自身应负的责任。Microchip 对这些信息不作任何明示或 暗示、书面或口头、法定或其他形式的声明或担保,包括但不 限于针对其使用情况、质量、性能、适销性或特定用途的适用 性的声明或担保。Microchip 对因这些信息及使用这些信息而 引起的后果不承担任何责任。如果将 Microchip 器件用于生命 维持和/或生命安全应用,一切风险由买方自负。买方同意在 由此引发任何一切伤害、索赔、诉讼或费用时,会维护和保障 Microchip 免于承担法律责任,并加以赔偿。除非另外声明,在 Microchip 知识产权保护下,不得暗中或以其他方式转让任何 许可证。

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印 度的设计中心均通过了 ISO/TS-16849:2009 认证。Microchip 的 PIC<sup>®</sup> MCU 与 dsPIC<sup>®</sup> DSC、KEELo<sup>®</sup> 跳码器件、单行 EEPROM、单片机外 设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外, Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

# QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV — ISO/TS 16949—

#### 商标

Microchip 的名称和徽标组合、Microchip 徽标、AnyRate、 AVR、AVR 徽标、AVR Freaks、BeaconThings、BitCloud、 CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、 Heldo、JukeBlox、KEELOQ、KEELOQ 徽标、Kleer、 LANCheck、LINK MD、maXStylus、maXTouch、MediaLB、 megaAVR、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、 picoPower、PICSTART、PIC32 徽标、Prochip Designer、 QTouch、RightTouch、SAM-BA、SpyNIC、SST、SST 徽标、 SuperFlash、tinyAVR、UNI/O 及 XMEGA 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

ClockWorks、The Embedded Control Solutions Company、 EtherSynch、Hyper Speed Control、HyperLight Load、 IntelliMOS、mTouch、Precision Edge 和 Quiet-Wire 均为 Microchip Technology Inc. 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BodyCom、chipKIT、chipKIT 徽标、 CodeGuard、CryptoAuthentication、CryptoCompanion、 CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、Inter-Chip Connectivity、JitterBlocker、 KleerNet、KleerNet 徽标、Mindi、MiWi、motorBench、MPASM、 MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、 NetDetach、Omniscient Code Generation、PICDEM. PICDEM.net、PICkit、PICtail、PureSilicon、QMatrix、RightTouch 徽标、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、 SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、 WiperLock、Wireless DNA和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Inc. 在美国的服务标记。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地 区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2016-2018, Microchip Technology Inc. 版权所有。

ISBN: 978-1-5224-2528-1



# 全球销售及服务网点

#### 美洲

公司总部 Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 1-480-792-7200 Fax: 1-480-792-7277 技术支持: http://www.microchip.com/ support 网址: www.microchip.com

亚特兰大 Atlanta Duluth, GA Tel: 1-678-957-9614 Fax: 1-678-957-1455

**奥斯汀 Austin, TX** Tel: 1-512-257-3370

波士顿 Boston Westborough, MA Tel: 1-774-760-0087 Fax: 1-774-760-0088

芝加哥 Chicago Itasca, IL Tel: 1-630-285-0071 Fax: 1-630-285-0075

达拉斯 Dallas Addison, TX Tel: 1-972-818-7423 Fax: 1-972-818-2924

底特律 Detroit Novi, MI Tel: 1-248-848-4000

**休斯敦 Houston, TX** Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis Noblesville, IN Tel: 1-317-773-8323 Fax: 1-317-773-5453 Tel: 1-317-536-2380

洛杉矶 Los Angeles Mission Viejo, CA Tel: 1-949-462-9523 Fax: 1-949-462-9608 Tel: 1-951-273-7800

**罗利 Raleigh, NC** Tel: 1-919-844-7510

**纽约 New York, NY** Tel: 1-631-435-6000

圣何塞 San Jose, CA Tel: 1-408-735-9110 Tel: 1-408-436-4270

加拿大多伦多 Toronto Tel: 1-905-695-1980 Fax: 1-905-695-2078

DS50002400B CN 第30页

亚太地区 中国 - 北京 Tel: 86-10-8569-7000 中国 - 成都

Tel: 86-28-8665-5511 中国 - 重庆 Tel: 86-23-8980-9588 中国 - 东莞 Tel: 86-769, 8702-9880

Tel: 86-769-8702-9880 中国 - 广州 Tel: 86-20-8755-8029

中国 - 杭州 Tel: 86-571-8792-8115

中国 - 南京 Tel: 86-25-8473-2460

中国 - 青岛 Tel: 86-532-8502-7355 中国 - 上海

Tel: 86-21-3326-8000 中国 - 沈阳

Tel: 86-24-2334-2829 中国 - 深圳 Tel: 86-755-8864-2200

中国 - 苏州 Tel: 86-186-6233-1526

中国 - 武汉 Tel: 86-27-5980-5300

中国 - 西安 Tel: 86-29-8833-7252 中国 - 厦门 Tel: 86-592-238-8138 中国 - 香港特别行政区 Tel: 852-2943-5100

**中国 - 珠海** Tel: 86-756-321-0040

**台湾地区 - 高雄** Tel: 886-7-213-7830

台湾地区 - 台北 Tel: 886-2-2508-8600

**台湾地区 - 新竹** Tel: 886-3-577-8366 亚太地区 澳大利亚 Australia - Sydney Tel: 61-2-9868-6733

印度 India - Bangalore Tel: 91-80-3090-4444

印度 India - New Delhi Tel: 91-11-4160-8631

印度 India - Pune Tel: 91-20-4121-0141 日本 Japan - Osaka

Tel: 81-6-6152-7160 日本 Japan - Tokyo Tel: 81-3-6880-3770

**韩国 Korea - Daegu** Tel: 82-53-744-4301

**韩国 Korea - Seoul** Tel: 82-2-554-7200

> 马来西亚 Malaysia - Kuala Lumpur Tel: 60-3-7651-7906

马来西亚 Malaysia - Penang Tel: 60-4-227-8870

菲律宾 Philippines - Manila Tel: 63-2-634-9065

**新加坡 Singapore** Tel: 65-6334-8870

**泰国 Thailand - Bangkok** Tel: 66-2-694-1351

越南 Vietnam - Ho Chi Minh Tel: 84-28-5448-2100

荷兰 Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340

**挪威 Norway - Trondheim** Tel: 47-7289-7561

波兰 Poland - Warsaw Tel: 48-22-3325737

罗马尼亚 Romania - Bucharest Tel: 40-21-407-87-50

西班牙 Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91

**瑞典 Sweden - Gothenberg** Tel: 46-31-704-60-40

**瑞典 Sweden - Stockholm** Tel: 46-8-5090-4654

英国 UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820

### 欧洲

奥地利 Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393

丹麦 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829

芬兰 Finland - Espoo Tel: 358-9-4520-820

法国 France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

德国 Germany - Garching Tel: 49-8931-9700 德国 Germany - Haan

Tel: 49-2129-3766400

**德国 Germany - Heilbronn** Tel: 49-7131-67-3636

德国 Germany - Karlsruhe Tel: 49-721-625370

**德国 Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44

德国 Germany - Rosenheim Tel: 49-8031-354-560

以色列 Israel - Ra'anana Tel: 972-9-744-7705

意大利 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781

意大利 Italy - Padova Tel: 39-049-7625286

10/25/17