ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น 188 460 ไมโครคอนโทรลเลอร์

Experiment MCL-01: PIC MCU External Interrupt

วัตถุประสงค์

- ้1. เพื่อฝึกการเขียนโปรแกรมภาษาแอสเซมบลีและการบันทึกโปรแกรมลงในไมโครคอนโทรลเลอร์ PIC 16F628A
- 2. เพื่อศึกษาการทำงานของระบบ external interrupt ของ 16F628A

อุปกรณ์และเครื่องมือ

- เครื่องคอมพิวเตอร์พร้อมโปรแกรม MPLAB สำหรับการเขียนโปรแกรมภาษาแอสเซมบลีและ WinPIC800 สำหรับการ บันทึกโปรแกรมลงในไมโครคอนโทรลเลอร์
- เครื่องบันทึกโปรแกรมลงในไมโครคอนโทรลเลอร์ ETT EMPIC V2.0 พร้อม power supply และสายเชื่อมต่อกับเครื่อง คอมพิวเตอร์
- 3. PIC MCU 16F628A
- 4. Crystal Oscillator 4.0 MHz.
- 5. Ceramic capacitor 30 pF x 2
- 6. Digital Experiment Box

การใช้งาน WinPic800 ร่วมกับบอร์ด ET-EMPIC V2.0

โปรแกรม WinPic800 เป็น Freeware ใช้สำหรับ download hex file ให้กับไมโครคอนโทรลเลอร์ตระกูล PIC และ dsPIC ขั้นตอนของการใช้งานโปรแกรม WinPic800 มีดังนี้

- 1. ติดตั้งโปรแกรม WinPic800 ในเครื่องคอมพิวเตอร์ให้เรียบร้อยพร้อมใช้งาน
- เขียนโปรแกรมและแปลคำสั่งของโปรแกรมที่เขียนให้เป็น hex File โดยในการทดลองนี้จะใช้โปรแกรม MPLAB ของ Microchip
- สิดตั้งชุด ET-CAP10PIN V2.0 เข้ากับพอร์ตขนาน (LPT Port หรือ Printer Port) ของเครื่องคอมพิวเตอร์เพื่อใช้ใน การสั่ง download hex file ด้วยโปรแกรม WinPic800
- 5. ทำการจ่ายไฟให้กับบอร์ด ET-EMPIC V2.0
- 6. Run WinPic800 เพื่อเริ่มต้นขั้นตอนของการ download hex file ซึ่งโปรแกรมจะปรากฏบนหน้าจอดังภาพ

🔷 WinPic800	- v 3.56					<u> </u>
File Edit Devi	ce Settings	Language H	elp			
🚔 🔻 🕸		2.	🕨 🧇 🔮	🍹 🐐	PIC 16F	🚳 💽 🧇
E: C %	248 🔖	M 😐	dev 📰	r 🕫 👙	16F628A	· (
🔽 Code	Code 🗳 Data 🛷 Setting					
0x0000: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?. 🔺
0x0008: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?.
0x0010: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?.
0x0018: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?.
0x0020: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?.
0x0028: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?.
0x0030: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?.
0x0038: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?.
0x0040: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?.
0x0048: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?.
0x0050: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?.
0x0058: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?.
0x0060: 3F	FF 3FFF	3FFF 3FFF	3FFF 3FF	F 3FFF 3FFF	?.?.?.?.	?.?.?.?. 🔻
Har.>ETT-HIGH VPP ICSP - LPT1						

- 7. เลือกเบอร์ของไมโครคอนโทรลเลอร์ที่ต้องการจะใช้งานให้ถูกต้อง
- 8. ตรวจสอบการตั้งค่าของโปรแกรม WinPic800 ว่าได้ถูกกำหนดให้ใช้งานกับ ET-EMPIC V2.0 โดยไปที่เมนู Settings --> Hardware จะปรากฏหน้าจอดังภาพ

WinPic800 [Hardware Settings]					
Selection of hardware					
3DIP8					
ART2003	Manufacturer : ETT CO., LTD				
ETT-HIGH VPP ICSP					
ETT-LOW VPP ICSP	* High Voltage Programming (+13V VPP)				
GTP-04					
GTP-07	- Support Development Board -				
GTP-244					
GTP-Lite	: CP-PIC V3 / V3 Plus				
GTP-USB [Plus]	: CP-PIC V3 EXP / V3 EXP Plus				
INEX-Parallel Port	: CP-PIC V4 / V4 Plus				
JDM Programmer 🚽	: ET-dsPIC 30F2010				
	: ET-EMPIC V1/V2 Emulator & Programmer				
- Addr - \$0378	Information : www.ett.co.th www.etteam.com				
	- Free Schematics -				
	Cancel Apply edits				

โดยจะต้องตั้งค่าให้เป็น ETT-HIGH VPP ICSP ซึ่งจะเป็นการโปรแกรมตัวไมโครคอนโทรลเลอร์โดยใช้แรงดันสูง จาก นั้นเลือกที่ Apply edits

- การโปรแกรมตัวไมโครคอนโทรลเลอร์ ให้เลือกเปิด hex file ที่ได้จาก assembler จากนั้น ตรวจสอบว่าซอฟต์แวร์ สามารถตรวจพบไมโครคอนโทรลเลอร์ได้หรือไม่โดยใช้คำสั่ง "Device Detect PIC" ซึ่งโปรแกรมจะทำการอ่านรหัส ID Code ของไมโครคอนโทรลเลอร์พร้อมกับแสดงเบอร์ที่ตรวจพบให้ทราบ ถ้าการตรวจสอบเบอร์เกิดความผิด พลาด โปรแกรมแสดงเบอร์ไม่ถูกต้อง ให้ลองตรวจสอบสายสัญญาณ และการวางไมโครคอนโทรลเลอร์ลงใน ZIF Socket
- 10. ตรวจสอบค่า Configuration ที่จะโปรแกรมลงในไมโครคอนโทรลเลอร์ ว่ากำหนดไว้ถูกต้องหรือไม่ โดยไปที่ Setting Tab จะปรากฏ Configuration Option ต่างๆ ให้เลือก สำหรับการทดลองนี้ จะกำหนดดังนี้

🔷 WinPic800 🕘 v 3.56					
File Edit Device Settings	Language Help				
🛎 🔹 😫 📃	🖸 🤹 🧇 🐝 🐐 PIC 16F	🚳 🔽 🧇			
e ca y 🌾 🖓	🗛 🚽 🚓 📰 🖉 🏂 🚺 16F628A				
Code 🖓 Data 🛛 Setting					
D FFFF 0x20002003	on/off - 0sc - CP - C LP Off (XT) C 0000h-07FFh C HS C ExtClk C INTRC I/O C INTRC CLKOUT WDTE BOREN C RC IO M PWRTE LVP C RC CLKOUT M NCLRE CPD 0x2007: 3F89				
Har.>ETT-HIGH VPP ICSP - LPT1					

โดยตัวเลือกต่างๆ มีความหมายดังนี้

- OSC ชนิดของ Oscillator ที่จะใช้ ได้แก่ LP : Low-power crystal XT : Crystal / resonator HS : High-speed crystal / resonator ExtClk : External clock source INTRC I/O : Internal RC oscillator, I/O function on RA6/OSC2/CLKOUT pin INTRC CLKOUT : Internal RC oscillator, clock signal output on RA6/OSC2/CLKOUT pin RC I/O : External RC oscillator, I/O function on RA6/OSC2/CLKOUT pin RC CLKOUT : External RC oscillator, clock signal output on RA6/OSC2/CLKOUT pin RC CLKOUT : External RC oscillator, clock signal output on RA6/OSC2/CLKOUT pin CP - Code Protection (Program Memory) WDTE - Watchdog Timer Enable PWRTE - Power-up Timer Enable
- MCLRE Master Clear pin Enable
- LVP Low-voltage programming

BOREN - Brown-out Reset Enable **CPD -** Code Protection (Data memory)

 สั่งให้โปรแกรม hex file ลงในไมโครคอนโทรลเลอร์โดยใช้คำสั่ง "Program All" หรือคลิกเมาส์ที่ Button คำสั่ง หลังจากขั้นตอนนี้เสร็จสิ้นลงถ้าไม่เกิดความผิดพลาดใดๆ ก็จะปรากฏหน้าจอดังต่อไปนี้

WinPic800						
Program - 1	6F628A			Vdd 🥝		
Programming Programming Programming Programming	Programming Code: 20 Programming ID: Programming Data: 1 Programming Config. :					
100%						
Cancel	Progress		Accept			
Clock - 1 us	Pgm - 6 ms		CPU - 2933 Mhz			
Close this window when finished						

ขั้นตอนการทดลอง

1. ต่อวงจรดังต่อไปนี้บนกล่องทดลอง Digital box



2. ป้อนและแปลโปรแกรมภาษาแอสเซมบลีต่อไปนี้ใน MPLAB

	list p	p=16f628			
W	equ	0	;	or	use #include <p16f628.inc></p16f628.inc>
f	equ	1	;	to	avoid this declaration
porta	equ	0x05			
portb	equ	0x06			
intcon	equ	0x0b			
cmcon	equ	0x1f			
	org	0x000			
	goto	start			

	org	0x004	
	goto	int serv	
start	movlw option	b'01111111'	;select Ext. Int. Rising Edge
	bcf	intcon,1	;clear INT Flag
	bsf	intcon,7	;enable GIE
	bsf	intcon.4	enable Ext. Int.
	clrf	porta	initialize PORTA
	movlw	0x07	turn off analog comparator
	movwf	cmcon	; cain off analog comparator
	movlw	b!111111111	, assign PORTA as input
	trie	porta	,assign ionin as input
	moulu	bi00000011	accian DOPTR A ac input
	tuoviw	D 0000001	,assign FORIB.0 as input
	LIIS aluf	portb	
1 1	CITI	porto	
test	DTISS	porta,U	; CNECK PORTA.U
	goto	cirb_i	; if $PORTA.U = U$, goto clrb_1
	bsi	portb,1	; if $PORTA.0 = 1$, $PORTB.1 = 1$
	goto	test	;check PORTA.0 again
clrb_1	bcf	portb,1	; PORTB.1 = 0
	goto	test	;check PORTA.0 again
int_serv	bsf	portb,2	;if Ext. Int. occurs
loop	goto	loop	;trap in ISR, reset required
	end		

- 3. ทำการ download hex file ที่ได้จากขั้นตอนที่ 1 ลงในไมโครคอนโทรลเลอร์
- 4. ทดสอบการทำงาน โดยสังเกตุจาก LED แสดงสถานะของ interrupt
- 5. ดัดแปลงวงจรใหม่ดังรูป



- 6. ให้ทำการแก้ไขโปรแกรมจากขั้นตอนที่ 2 เพื่อให้แสดงจำนวนครั้งของการเกิด interrupt บน LED ขนาด 4 bit
- 7. ดัดแปลงวงจรใหม่ดังรูป



 ให้ทำการแก้ไขโปรแกรมจากขั้นตอนที่ 2 เพื่อเปลี่ยนให้ไมโครคอนโทรลเลอร์รับสัญญาณ interrupt แบบ RB port change (RB4-RB7) แล้วแสดงผลด้วย LED 1-bit ว่าเกิดการ interrupt ขึ้น

คำถามท้ายการทดลอง

จากการทดลองในขั้นตอนที่ 8 ถ้าต้องการจะทราบว่าสัญญาณ interrupt เกิดขึ้นที่ port RB7, RB6, RB5 หรือ RB4 จะ ต้องแก้ไขวงจรและโปรแกรมอย่างไร <u>ให้แสดงวงจรและโปรแกรม</u> ที่สามารถทำให้ไมโครคอนโทรลเลอร์รับรู้ได้ว่าเกิด interrupt ขึ้นที่ขาใด

<u>หมายเหตุ:</u> ในการเขียนโปรแกรมสำหรับคำถามท้ายการทดลอง สามารถใช้ Simulator ใน MPLAB เพื่อทดสอบการทำงานได้ ก่อนที่จะโปรแกรมลงในไมโครคอนโทรลเลอร์