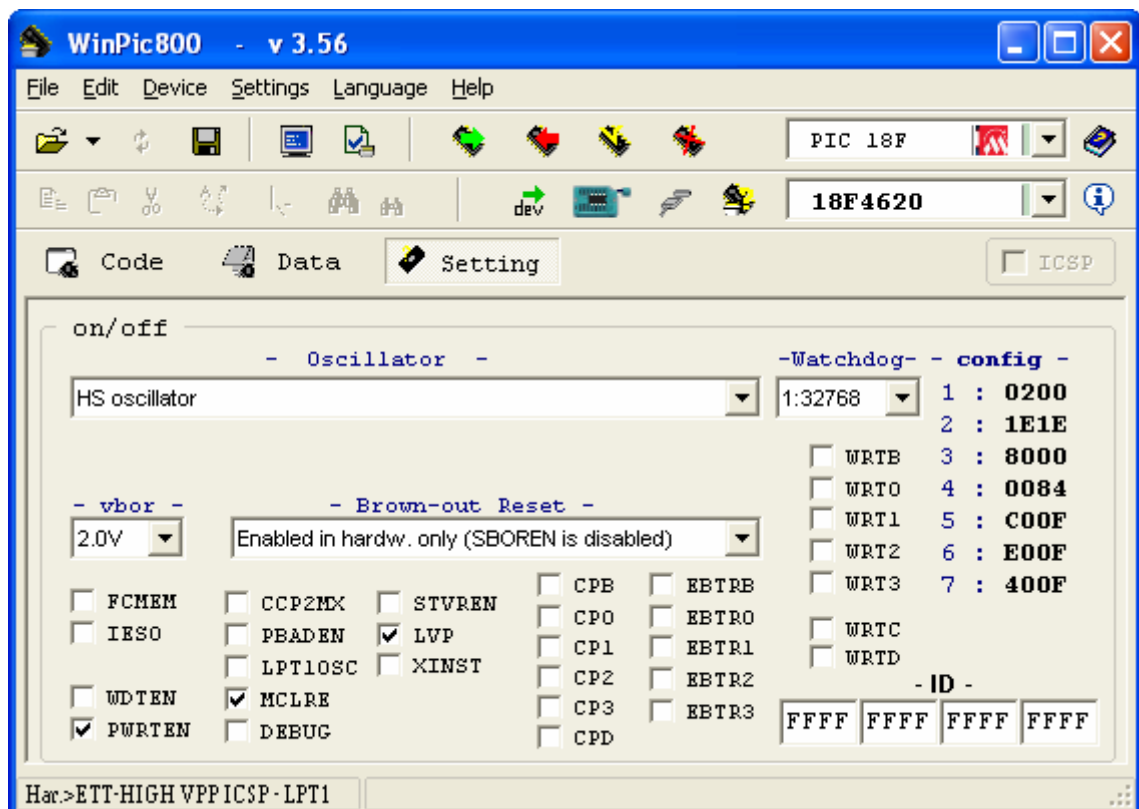


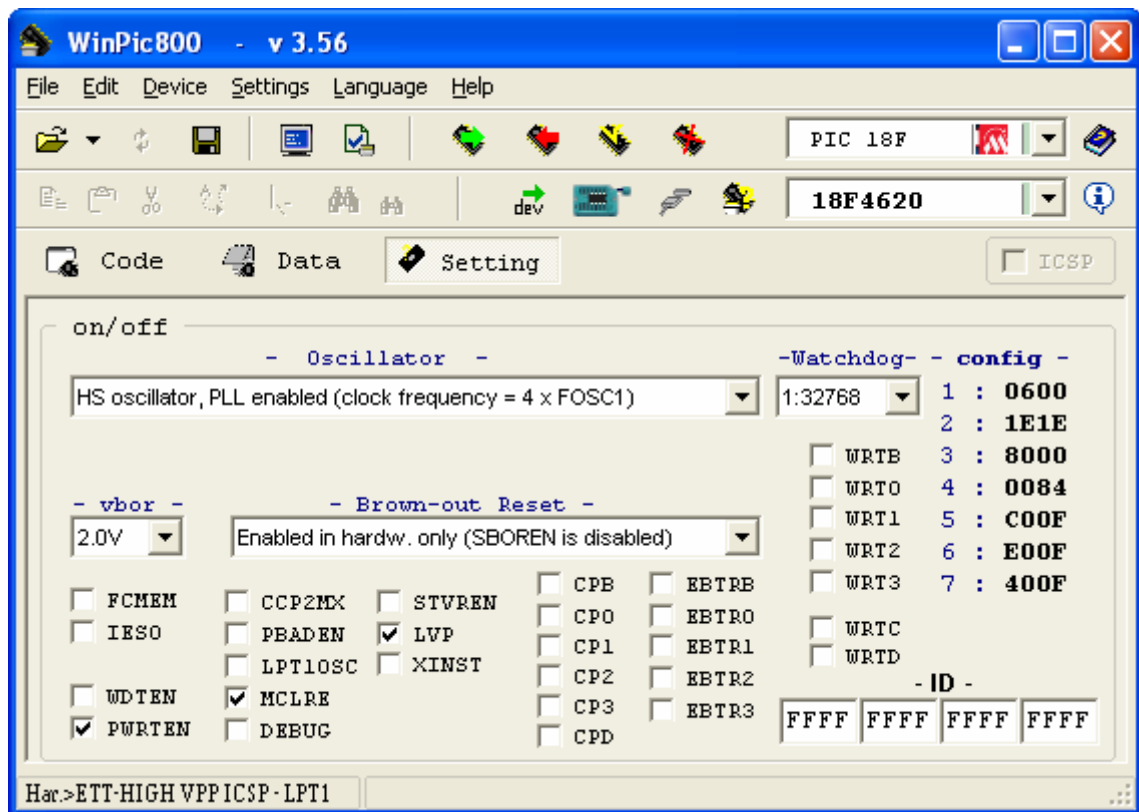
การกำหนดค่า Configuration ในแบบต่างๆ

เนื่องจากไมโครคอนโทรลเลอร์ PIC 18F4620 มีรายละเอียดในการใช้งาน และฟังก์ชันการทำงานค่อนข้างหลากหลาย ซึ่งขึ้นอยู่กับผู้ใช้งานที่จะต้องกำหนดฟังก์ชัน หรือ รายละเอียดต่างๆ ที่ต้องการ โดยจะขอแนะนำการกำหนดค่าฟังก์ชันการใช้งานที่สอดคล้องกับการใช้งานบอร์ด ET-BASE PIC40 โดยอ้างอิงกับซอฟต์แวร์โปรแกรมของ WinPic800 ในรูปแบบต่างๆ โดยจะแนะนำ 2 แบบดังต่อไปนี้

- แบบที่ 1 ใช้งานออสซิลเลเตอร์ 10 MHz



- แบบที่ 2 ใช้ฮอสซิลเลเตอร์ 40 MHz



รายละเอียดต่างๆ ของค่า Configuration

- **Oscillator** - การเลือกรูปแบบแหล่งกำเนิดสัญญาณนาฬิกาหลัก

- **LP** = Crystal แบบกำลังงานต่ำ
- **XT** = Crystal/Resonator
- **HS** = Crystal/Resonator แบบความเร็วสูง
- **External RC oscillator, CLK0 function on RA6** = วงจรกำเนิดความถี่แบบ Resistor/Capacitor ให้สัญญาณ OSC/4 ออกไปที่ขาสัญญาณ RA6
- **EC oscillator, CLK0 function on RA6** = วงจรกำเนิดความถี่โดยใช้ คาปาซิเตอร์ภายนอก และ ให้ขาสัญญาณ RA6 ทำหน้าที่เป็นขาสัญญาณ Clock Out (FOSC/4)
- **EC oscillator, port function on RA6** = วงจรกำเนิดความถี่โดยใช้ คาปาซิเตอร์ภายนอก และ ให้ขาสัญญาณ RA6 ทำหน้าที่เป็น I/O ใช้งานทั่วไป
- **HS oscillator ,PLL enabled (clock frequency = 4 x FOSC1)** = วงจรกำเนิดความถี่ที่ใช้คริสตอลแบบความเร็วสูง และ เปิดการทำงานของวงจรคูณความถี่ (PLL) โดยจะคูณความถี่สัญญาณนาฬิกาขึ้นไปได้อีกถึง 4 เท่า

- **Internal RC oscillator , port function on RA6, port function on RA7** = วงจรกำเนิดสัญญาณนาฬิกาแบบ RC ภายใน โดยที่
 - ขาสัญญาณ RA6 ทำหน้าที่เป็นขาสัญญาณ I/O
 - ขาสัญญาณ RA7 ทำหน้าที่เป็นขาสัญญาณ I/O
- **Internal RC oscillator , CLKO function on RA6, port function on RA7** = วงจรกำเนิดสัญญาณนาฬิกาแบบ RC ภายใน โดยที่
 - ขาสัญญาณ RA6 ทำหน้าที่เป็นขาสัญญาณ CLKO (Clock Output)
 - ขาสัญญาณ RA7 ทำหน้าที่เป็นขาสัญญาณ I/O

ทำหน้าที่เป็น I/O คือ RA6

- vbor -

คือ วงจรรีเซ็ต MCU เมื่อแรงดันต่ำกว่าที่กำหนด (Brown-Out Reset Voltage) โดยสามารถกำหนดแรงดันได้หลายระดับ คือ 2.0V, 2.7V, 4.2V และ 4.5V โดยการทำงานของวงจรรีเซ็ตแบบนี้จะเกิดขึ้นได้ก็ต่อเมื่อมีการ Enable บิต BOREN ไว้แล้วเท่านั้น

- 2.0V = ตั้งระดับแรงดัน vbor ไว้ที่ 2.0 V
- 2.7V = ตั้งระดับแรงดัน vbor ไว้ที่ 2.7 V
- 4.2V = ตั้งระดับแรงดัน vbor ไว้ที่ 4.2 V
- 4.5V = ตั้งระดับแรงดัน vbor ไว้ที่ 4.5 V

- **FCMEM** = เปิด(Enable) หรือ ปิด(Disable) ฟังก์ชัน Fail Safe Clock Monitor
- **IESO** = เปิด(Enable) หรือ ปิด(Disable) ฟังก์ชันการทำงานของวงจรสวิตช์สัญญาณนาฬิกา (Oscillator System Clock Switch Enable bit)
- **WDTEN** = เปิด(Enable) หรือ ปิด(Disable) ฟังก์ชันการรีเซ็ตแบบวอตช์ด็อกไทเมอร์ (Watchdog Timer Reset Enable)
- **PWRTEN** = เปิด(Enable) หรือ ปิด(Disable) การทำงานของ Power-up Timer คือ เปิด-ปิด ฟังก์ชันการหน่วงเวลาการทำงานของ MCU ขณะเริ่มจ่ายแรงดันไป 72 ms ทั้งนี้ก็เพื่อรอให้แรงดันอยู่ในระดับคงที่ก่อนที่จะให้ MCU ประมวลผลคำสั่งแรก

- Brown-out Reset -

เลือกรูปแบบของ Brown-out Reset ดังต่อไปนี้

- **Disabled in hardware and software** = ปิด (Disabled) การทำงานของ Brown-out Reset และ ไม่ อนุญาตให้มีการควบคุมการ Enable โดยซอฟต์แวร์จากการควบคุมบิต SBOREN
- **Enabled and control. By soft.(SBOREN is enabled)** = เปิด(Enable) การทำงานของ Brown-out Reset และ อนุญาตให้มีการควบคุมการ Enable โดยซอฟต์แวร์จากการควบคุมบิต SBOREN

- **Enabled in hardw. only and disabled in Sleep mode** = เปิด(Enable) การทำงานของ Brown-out Reset แต่ให้ปิดการทำงาน (Disable) เมื่อเข้าสู่โหมด Sleep และ ไม่อนุญาตให้มีการควบคุมโดยซอฟต์แวร์จากการควบคุมบิต SBOREN
- **Enabled in hardw. only (SBOREN is disabled)** = เปิด(Enable) การทำงานของ Brown-out Reset และ ไม่อนุญาตให้มีการควบคุมโดยซอฟต์แวร์จากการควบคุมบิต SBOREN
- **CCP2MX** = เปิด (Enable) – ปิด (Disable) วงจรมัลติเพล็กซ์ที่ทำหน้าที่ เลือกขาสัญญาณ input/output ของโมดูล CCP2 ว่าจะเป็น RC1 หรือ RB3
 - ☒ **CCP2MX** = RC1
 - ☐ **CCP2MX** = RB3
- **PBADEN** = เลือกสถานะของ PORTB<4:0> ในสภาวะรีเซ็ต ว่าจะให้เป็นขาสัญญาณ Analog input หรือขาสัญญาณ Digital I/O
 - ☒ **PBADEN** = Analog input
 - ☐ **PBADEN** = Digital I/O
- **LPT1OSC** = เลือกระดับการใช้พลังงานของ Timer1 Oscillator
 - ☒ **LPT1OSC** = Low-power operation
 - ☐ **LPT1OSC** = Higher power operation
- **MCLR** = เปิด (Enable) – ปิด (Disable) ฟังก์ชันการรีเซ็ตของ MCU จากขาสัญญาณ MCLR
 - ☒ **MCLRE** = ใช้งานขาสัญญาณ MCLR/RE3 เป็นขาสัญญาณรีเซ็ตโปรแกรม MCLR
 - ☐ **MCLRE** = ใช้งานขาสัญญาณ MCLR/RE3 เป็นขาสัญญาณ I/O คือ RE3
- **DEBUG** = เปิด(Enable) หรือ ปิด(Disable) การทำงานของ Background Debugger
 - ☒ **DEBUG** = Enable ฟังก์ชัน Background Debugger ขาสัญญาณ RB6 และ RB7 ถูกใช้เป็นขาสัญญาณสำหรับการทำ In-circuit debugger
 - ☐ **DEBUG** = Disable ฟังก์ชัน Background Debugger ขาสัญญาณ RB6 และ RB7 ถูกใช้เป็นขาสัญญาณ I/O ทั่วไป
- **STVREN** = เปิด(Enable) หรือ ปิด(Disable) ฟังก์ชันการรีเซ็ตที่เกิดจาก Stack Full/Underflow
 - ☒ **STVREN** = Enable
 - ☐ **STVREN** = Disable

- **LVP** = เปิด(Enable) หรือ ปิด(Disable) ฟังก์ชันการโปรแกรมแบบ Low Voltage programming ซึ่งสำหรับ ET-BASE PIC40 จะต้องเปิดการทำงานของฟังก์ชันนี้เสมอไมเช่นนั้นจะทำให้ไม่สามารถทำการดาวน์โหลดโปรแกรมได้

☒ LVP = Enable

☐ LVP = Disable

- **XINST** = Enable หรือ Disable คำสั่งเพิ่มเติมของ PIC ในตระกูล 18F ซึ่งมีคำสั่งต่างดังตารางต่อไปนี้

EXTENSIONS TO THE PIC18 INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected
			MSb		LSb		
ADDFSR f, k	Add literal to FSR	1	1110	1000	ffkk	kkkk	None
ADDULNK k	Add literal to FSR2 and return	2	1110	1000	11kk	kkkk	None
CALLW	Call subroutine using WREG	2	0000	0000	0001	0100	None
MOVSF z_s, f_d	Move z_s (source) to 1st word f_d (destination)2nd word	2	1110	1011	0zzz	zzzz	None
MOVSS z_s, z_d	Move z_s (source) to 1st word z_d (destination)2nd word	2	1110	1011	1zzz	zzzz	None
PUSHL k	Store literal at FSR2, decrement FSR2	1	1110	1010	kkkk	kkkk	None
SUBFSR f, k	Subtract literal from FSR	1	1110	1001	ffkk	kkkk	None
SUBULNK k	Subtract literal from FSR2 and return	2	1110	1001	11kk	kkkk	None

☒ XINST = Enable ใช้งานคำสั่งเพิ่มเติม

☐ XINST = Disable ไม่ใช้งานคำสั่งเพิ่มเติม

- **CPB** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Boot Block (000000-0001FFh)
- **CP0** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 0 (000200-001FFFh)
- **CP1** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 1 (002000-003FFFh)
- **CP2** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 2 (004000-005FFFh)
- **CP3** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 3 (006000-007FFFh)
- **CPD** = ปกป้องข้อมูลหน่วยความจำ EEPROM ภายในของ PIC
- **EBTRB** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Boot Block (000000-0001FFh)
- **EBTR0** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Block 0 (000200-001FFFh)
- **EBTR1** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Block 1 (002000-003FFFh)
- **EBTR2** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Block 2 (004000-005FFFh)
- **EBTR3** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Block 3 (006000-007FFFh)

- watchdog-

เป็นฟังก์ชันการรีเซตชนิดหนึ่งของ MCU โดยอาศัยหลักการของไทมเมอร์ซึ่งจะนับสัญญาณนาฬิกาไปเรื่อยๆ (เป็นสัญญาณนาฬิกาของวงจร RC ภายใน MCU) หากไม่มีการเคลียร์ค่าการนับอยู่เสมอๆ และปล่อยให้มันนับไปจนกระทั่งถึงค่าที่กำหนด มันก็จะสร้างสัญญาณไปรีเซต MCU ซึ่งสามารถกำหนดค่าระดับของวอตช์ดอกไทมเมอร์ได้จากบิตต่างๆ ต่อไปนี้

- | | |
|---------------------------------|-------------------------------|
| <input type="radio"/> 1 : 32768 | <input type="radio"/> 1 : 128 |
| <input type="radio"/> 1 : 16384 | <input type="radio"/> 1 : 64 |
| <input type="radio"/> 1 : 8192 | <input type="radio"/> 1 : 32 |
| <input type="radio"/> 1 : 4096 | <input type="radio"/> 1 : 16 |
| <input type="radio"/> 1 : 2048 | <input type="radio"/> 1 : 8 |
| <input type="radio"/> 1 : 1024 | <input type="radio"/> 1 : 4 |
| <input type="radio"/> 1 : 512 | <input type="radio"/> 1 : 2 |
| <input type="radio"/> 1 : 256 | <input type="radio"/> 1 : 1 |

- **WRTB** = ป้องกันการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Boot Block (000000-0001FFh)
- **WRT0** = ป้องกันการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 0 (000200-001FFFh)
- **WRT1** = ป้องกันการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 1 (002000-003FFFh)
- **WRT2** = ป้องกันการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 2 (004000-005FFFh)
- **WRT3** = ป้องกันการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 3 (006000-007FFFh)
- **WRTC** = ป้องกันการเขียนข้อมูลในพื้นที่ของ Configuration Register (300000-3000FFFh)
- **WRTD** = ป้องกันการเขียนข้อมูลหน่วยความจำ EEPROM ภายในของ PIC

***** หมายเหตุ** การทำเครื่องหมายถูก หรือ การคลิกเลือกในช่องต่างๆ ของ Configuration ก็คือการเลือก Enable ฟังก์ชันการทำงานนั้นๆ โดยจะมีผลให้บิตของค่า Configuration นั้นเป็น 1 และ ในทางกลับกันหากไม่ได้ทำเครื่องหมายถูก หรือ การคลิกเลือกในช่องต่างๆ ก็จะเป็นการ Disable การทำงานนั้นๆ