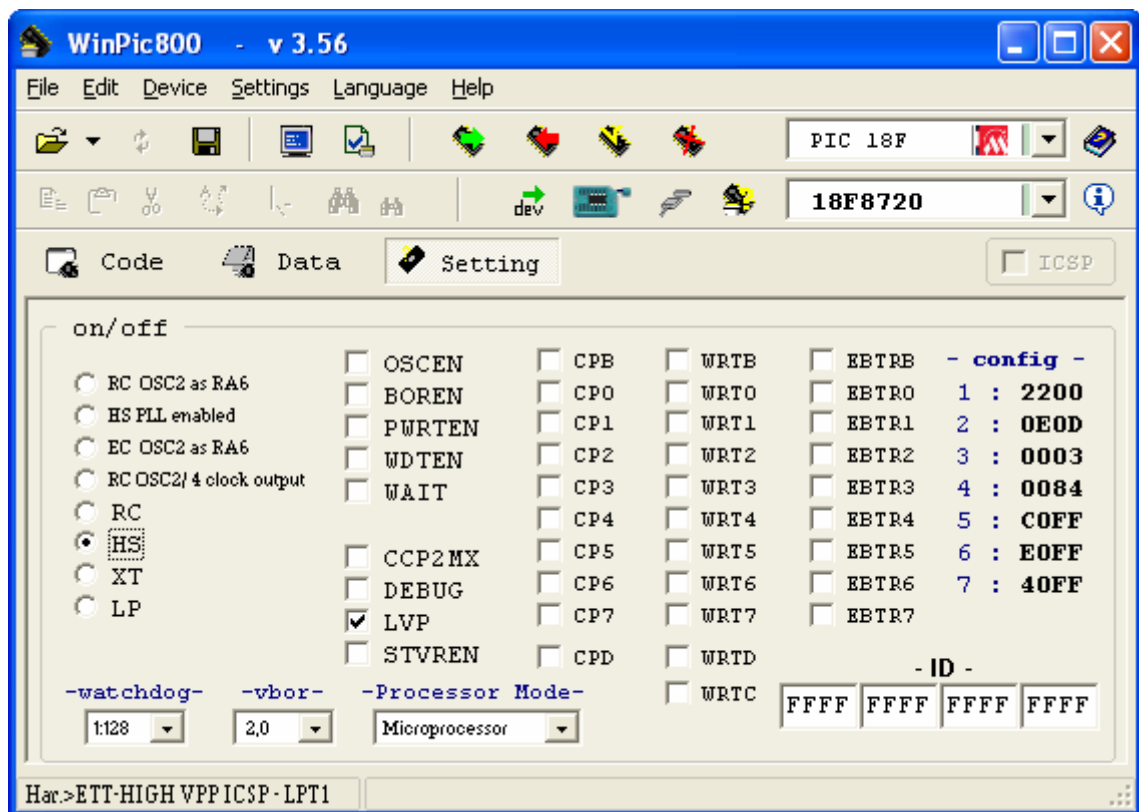


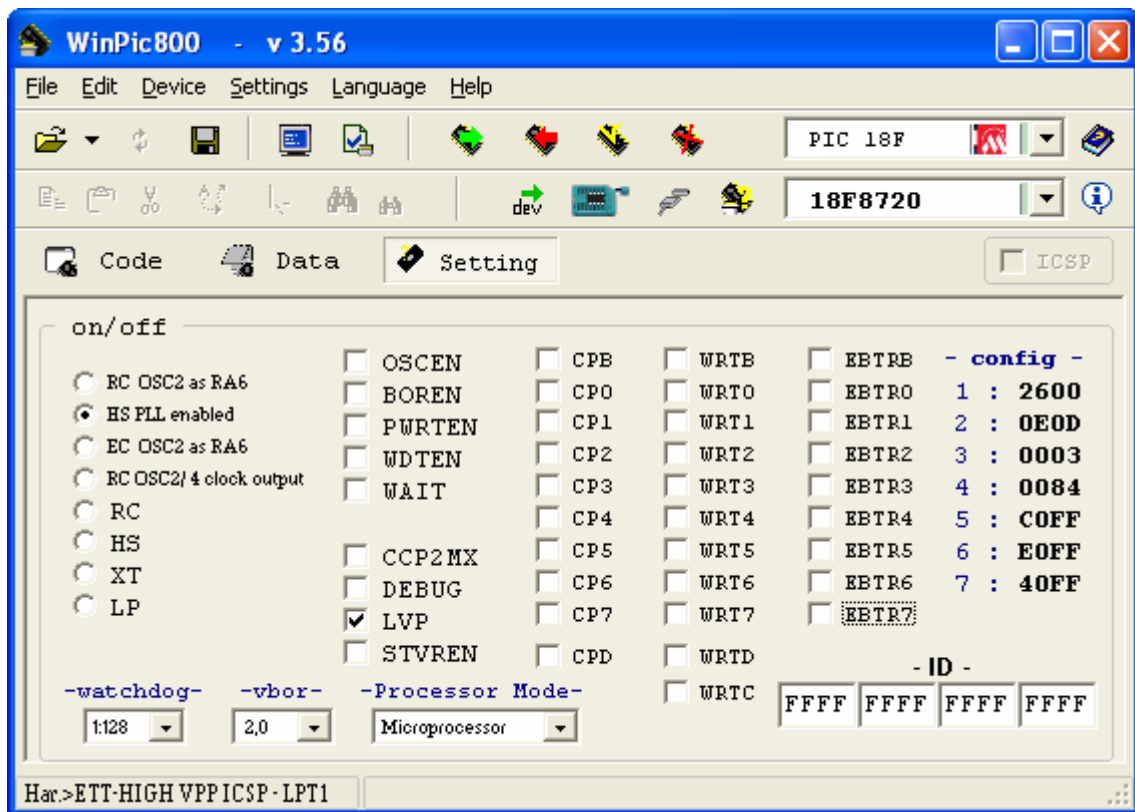
### การกำหนดค่า Configuration ในแบบต่างๆ

เนื่องจากไมโครคอนโทรลเลอร์ PIC 18F8720 มีรายละเอียดในการใช้งาน และฟังก์ชันการทำงานค่อนข้างหลากหลาย ซึ่งขึ้นอยู่กับผู้ใช้งานที่จะต้องกำหนดฟังก์ชัน หรือ รายละเอียดต่างๆ ที่ต้องการ โดยจะขอแนะนำการกำหนดค่าฟังก์ชันการใช้งานที่สอดคล้องกับการใช้งานบอร์ด ET-BASE PIC8720 โดยอ้างอิงกับซอฟต์แวร์โปรแกรมของ WinPic800 ในรูปแบบต่างๆ โดยจะแนะนำ 2 แบบดังต่อไปนี้

- แบบที่ 1 ใช้งานออสซิลเลเตอร์ 10 MHz



- แบบที่ 2 ใช้ข้อสซิลเลเตอร์ 40 MHz



### รายละเอียดต่างๆ ของค่า Configuration

- OSC - การเลือกรูปแบบแหล่งกำเนิดสัญญาณนาฬิกาหลัก

- RC OSC2 as RA6 = วงจรกำเนิดความถี่แบบ Resistor/Capacitor ซึ่งขา OSC2 จะทำหน้าที่เป็น I/O คือ RA6
- HS PLL enabled = Crystal/Resonator แบบความเร็วสูง และ เปิดวงจรคูณความถี่ (FOSCx4) ทำให้สามารถเพิ่มความถี่ขึ้นได้อีกถึง 4 เท่า
- EC OSC2 as RA6 = สัญญาณ Clock จากภายนอกต่อเข้ากับ OSC1 โดยตรง ส่วนขา OSC2 ทำหน้าที่เป็น I/O คือ RA6
- RC OSC2/4 clock output = วงจรกำเนิดความถี่แบบ Resistor/Capacitor ให้สัญญาณ OSC/4 ออกไป ที่ขาสัญญาณ OSC2 (ClkOut)
- RC = วงจรกำเนิดความถี่แบบ Resistor/Capacitor
- HS = Crystal/Resonator แบบความเร็วสูง
- XT = Crystal/Resonator
- LP = Crystal แบบกำลังงานต่ำ

- **OSCEN** = เปิด(Enable) หรือ ปิด(Disable) ฟังก์ชันการทำงานของวงจรสวิตช์สัญญาณนาฬิกา (Oscillator System Clock Switch Enable bit)
- **BOREN** = เปิด(Enable) หรือ ปิด(Disable) การทำงานของวงจรรีเซ็ตโปรแกรมเมื่อแรงดันต่ำกว่าระดับที่กำหนด (-vbor-)
- **PWRTEN** = เปิด(Enable) หรือ ปิด(Disable) การทำงานของ Power-up Timer คือ เปิด-ปิด ฟังก์ชันการหน่วงเวลาการทำงานของ MCU ขณะเริ่มจ่ายแรงดันไป 72 ms ทั้งนี้ก็เพื่อรอให้แรงดันอยู่ในระดับคงที่ก่อนที่จะให้ MCU ประมวลผลคำสั่งแรก
- **WDTEN** = วงจรรีเซ็ตแบบวอตช์ด็อกไทมเมอร์ ( Watchdog Timer Reset Enable) เป็นบิตที่ใช้ปิด-เปิด การทำงานของวงจร รีเซ็ตแบบวอตช์ด็อกไทมเมอร์
- **WAIT** = External Bus Data Wait Enable bit
- **CCP2MX** = เปิด (Enable) – ปิด (Disable) วงจรมัลติเพล็กซ์ที่ทำหน้าที่ เลือกขาสัญญาณ input/output ของโมดูล CCP2 ดังนี้

กรณี Microcontroller mode

☒ **CCP2MX** = RC1

☐ **CCP2MX** = RE7

กรณี Microprocessor, Microprocessor with Boot Block and Extended Microcontroller modes

☒ **CCP2MX** = RC1

☐ **CCP2MX** = RB3

- **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

- **LVP** = เปิด(Enable) หรือ ปิด(Disable) ฟังก์ชันการโปรแกรมแบบ Low Voltage programming ซึ่งสำหรับ ET-BASE PIC40 จะต้องเปิดการทำงานของฟังก์ชันนี้เสมอไม่เช่นนั้นจะทำให้ไม่สามารถทำการดาวน์โหลดโปรแกรมได้
- **STVREN** = เปิด(Enable) หรือ ปิด(Disable) ฟังก์ชันการรีเซ็ตที่เกิดจาก Stack Full/Underflow
- **CPB** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Boot Block (000000-0001FFh)
- **CP0** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 0 (000200-003FFFh)
- **CP1** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 1 (004000-007FFFh)
- **CP2** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 2 (008000-00BFFFh)

- **CP3** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 3 (00C000-00FFFFh)
- **CP4** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 4 (010000-013FFFh)
- **CP5** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 5 (014000-017FFFh)
- **CP6** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 6 (018000-01BFFFh)
- **CP7** = ปกป้องข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 7 (01C000-01FFFFh)
- **CPD** = ปกป้องข้อมูลหน่วยความจำ EEPROM ภายในของ PIC
  
- **WRTB** = ปกป้องการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Boot Block (000000-0001FFFh)
- **WRT0** = ปกป้องการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 0 (000200-003FFFh)
- **WRT1** = ปกป้องการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 1 (004000-007FFFh)
- **WRT2** = ปกป้องการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 2 (008000-00BFFFh)
- **WRT3** = ปกป้องการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 3 (00C000-00FFFFh)
- **WRT4** = ปกป้องการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 4 (010000-013FFFh)
- **WRT5** = ปกป้องการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 5 (014000-017FFFh)
- **WRT6** = ปกป้องการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 6 (018000-01BFFFh)
- **WRT7** = ปกป้องการเขียนข้อมูลหน่วยความจำโปรแกรมในตำแหน่ง Block 7 (01C000-01FFFFh)
- **WRTD** = ปกป้องการเขียนข้อมูลหน่วยความจำ EEPROM ภายในของ PIC
- **WRTC** = ปกป้องการเขียนข้อมูลในพื้นที่ของ Configuration Register (300000-3000FFFh)
- **EBTRB** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Boot Block (000000-0001FFFh)
- **EBTR0** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Block 0 (000200-003FFFh)
- **EBTR1** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Block 1 (004000-007FFFh)
- **EBTR2** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Block 2 (008000-00BFFFh)
- **EBTR3** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Block 3 (00C000-00FFFFh)
- **EBTR4** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Block 4 (010000-013FFFh)
- **EBTR5** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Block 5 (014000-017FFFh)
- **EBTR6** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Block 6 (018000-01BFFFh)
- **EBTR7** = ปกป้องการอ่านหน่วยความจำแบบ Table Read ในตำแหน่ง Block 7 (01C000-01FFFFh)

**- watchdog-**

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

- |                               |                             |
|-------------------------------|-----------------------------|
| <input type="radio"/> 1 : 128 | <input type="radio"/> 1 : 4 |
| <input type="radio"/> 1 : 64  | <input type="radio"/> 1 : 2 |
| <input type="radio"/> 1 : 32  | <input type="radio"/> 1 : 1 |
| <input type="radio"/> 1 : 16  |                             |
| <input type="radio"/> 1 : 8   |                             |

**- vbor -**

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

- |                           |                                     |
|---------------------------|-------------------------------------|
| <input type="radio"/> 2,0 | = ตั้งระดับแรงดัน vbor ไว้ที่ 2.0 V |
| <input type="radio"/> 2,7 | = ตั้งระดับแรงดัน vbor ไว้ที่ 2.7 V |
| <input type="radio"/> 4,2 | = ตั้งระดับแรงดัน vbor ไว้ที่ 4.2 V |
| <input type="radio"/> 4,5 | = ตั้งระดับแรงดัน vbor ไว้ที่ 4.5 V |

**-Processor Mode-**

- ☐ Microcontroller mode
- ☐ Microprocessor mode
- ☐ Microprocessor with Boot Block mode
- ☐ Extended Microcontroller mode

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