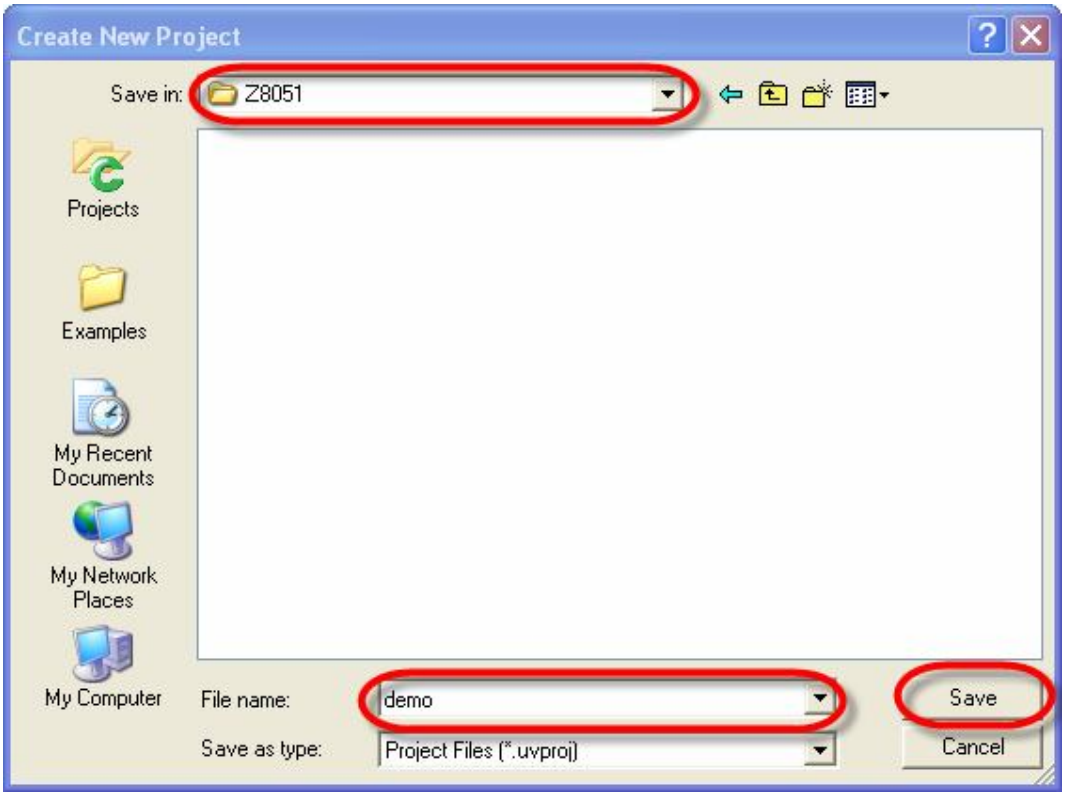
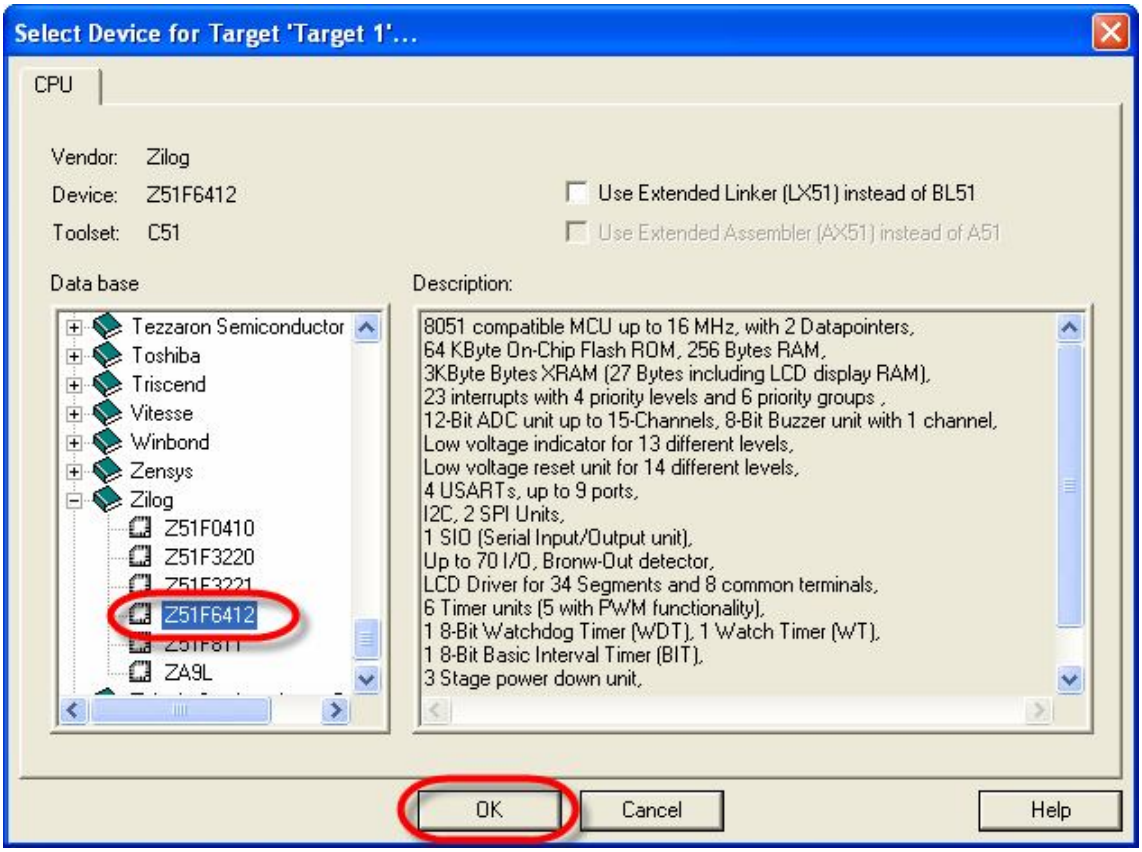
การพัฒนาโปรแกรมของบอร์ด ET-BASE Z51F6412 ด้วย Keil-C51(V9.50a)

ตามปรกติแล้ว Z8051 สามารถพัฒนาโปรแกรมด้วยตัวแปรภาษาที่รองรับการทำงานร่วมกับ MCU ตระกูล MCS51 ทั่วๆไปได้อยู่แล้ว เพราะ Z8051 ใช้ชุดคำสั่งมาตรฐานเดียวกับ MCS51 เพียงแต่ต้องมีการประกาศชื่อแต่ตำแหน่งของ ริจิสเตอร์ในส่วนที่ใช้ควบคุมและเข้าถึงอุปกรณ์ Peripheral I/O ต่างๆเพิ่มเติมเข้าไปในส่วนของ Special Function Register (SFR Register) ซึ่งส่วนนี้ต้องยึดรูปแบบไวยากรณ์คำสั่งตามข้อกำหนดของโปรแกรมแปลภาษาที่ใช้นั้นๆด้วย แต่สำหรับในกรณีของ Keil-C51(V9.50a) ซึ่งเป็นโปรแกรมแปลภาษา C Compiler สำหรับไมโครคอนโทรลเลอร์ตระกูล MCS51 และได้รับการพัฒนาปรับปรุงให้รองรับการใช้งานร่วมกับไมโครคอนโทรลเลอร์ตระกูล Z8051 ของ Zilog Inc. เป็นที่เรียบร้อยแล้ว ผู้ใช้สามารถกำหนดเลือกเบอร์ของ Z8051 สำหรับเขียนโปรแกรมได้ทันที

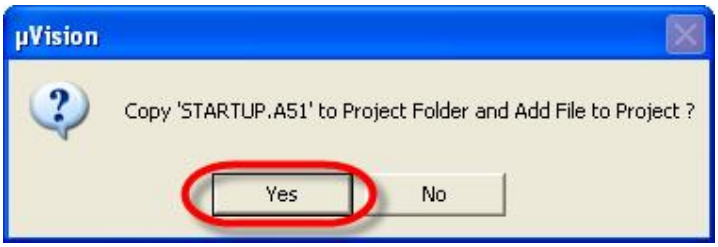
1. เปิดโปรแกรม uVision ซึ่งเป็นโปรแกรม Text Editor ของ keil-C51 ขึ้นมา จากนั้นให้ทำการสั่งสร้าง Project ขึ้นมาใหม่ โดยให้เลือกที่เมนู Project → New uVision Project... แล้วเลือกกำหนดชื่อ Project ตามต้องการ ในที่นี้กำหนดเป็น demo แล้วเลือก Save ดังรูป



เมื่อสั่งบันทึก Project เรียบร้อยแล้วจะมีหน้าต่าง Dialog สำหรับให้เลือกกำหนดเบอร์ MCU ที่จะใช้ ในที่นี้ให้เลือกเบอร์ MCU ของ Zilog เป็นเบอร์ Z51F6412 แล้วเลือก add ไฟล์ STARTUP.A51 ซึ่งเป็นไฟล์สำหรับ Initial หน่วยความจำสำหรับ MCU รวมไว้ใน Project ด้วย ดังตัวอย่าง



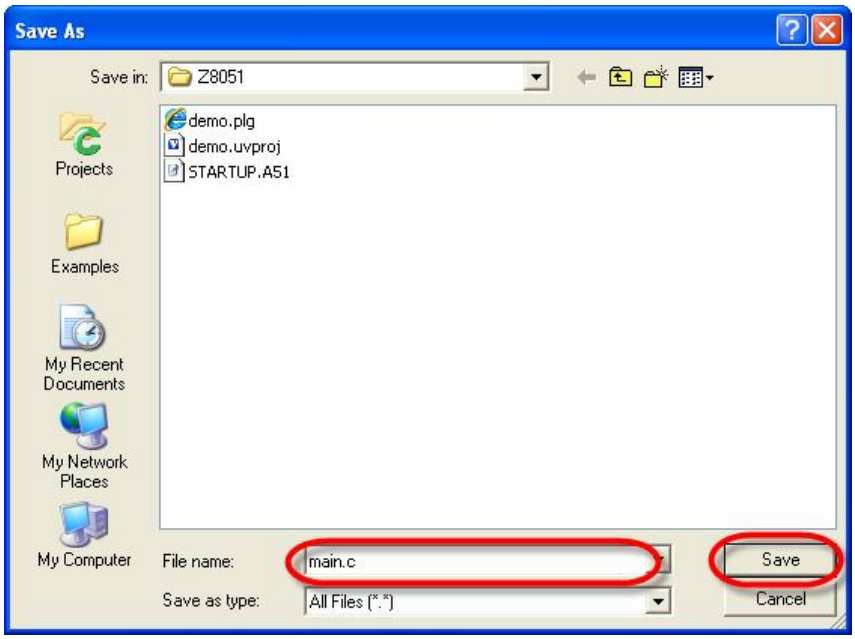
↓



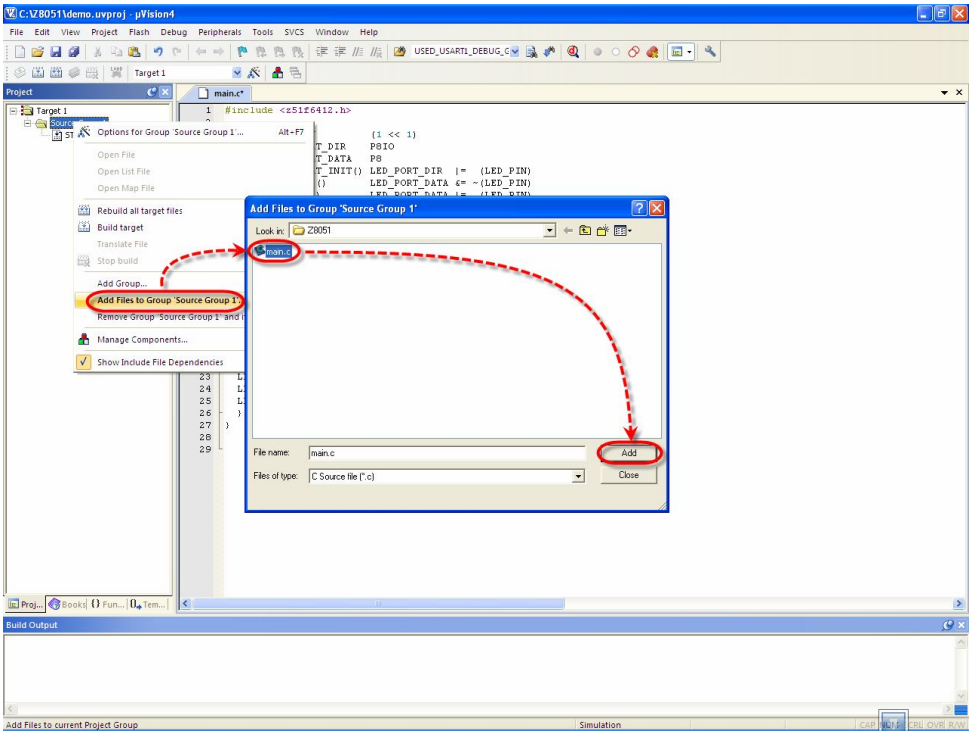
2. เลือกเมนู File → New... แล้วทำการพิมพ์คำสั่งภาษาซีตามตัวอย่าง แล้วสั่งบันทึก โดยเลือกเมนูคำสั่ง File → Save... พร้อมกับกำหนดชื่อไฟล์ที่จะบันทึกเป็น main.c ดังตัวอย่าง

```
#include <z51f6412.h>

#define LED_PIN         (1 << 1)
#define LED_PORT_DIR    P8IO
#define LED_PORT_DATA   P8
#define LED_PORT_INIT() LED_PORT_DIR  |=  (LED_PIN)
#define LED_OFF()       LED_PORT_DATA &= ~(LED_PIN)
#define LED_ON()        LED_PORT_DATA |=  (LED_PIN)

void delay(unsigned int i)
{
  while(i > 0) {i--;}
  return;
}

void main(void)
{
  PLLCR = 0x00;
  SCCR = 0x24;
  LED_PORT_INIT();

  while(1)
  {
    LED_OFF();
    delay(20000);
    LED_ON();
    delay(20000);
  }
}
```

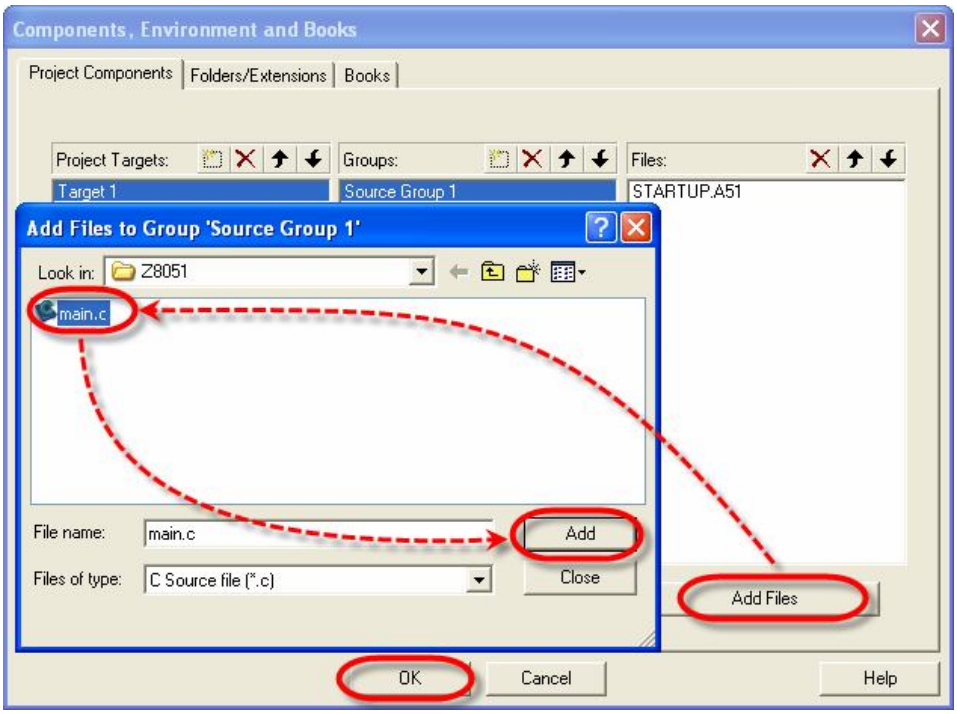
↓



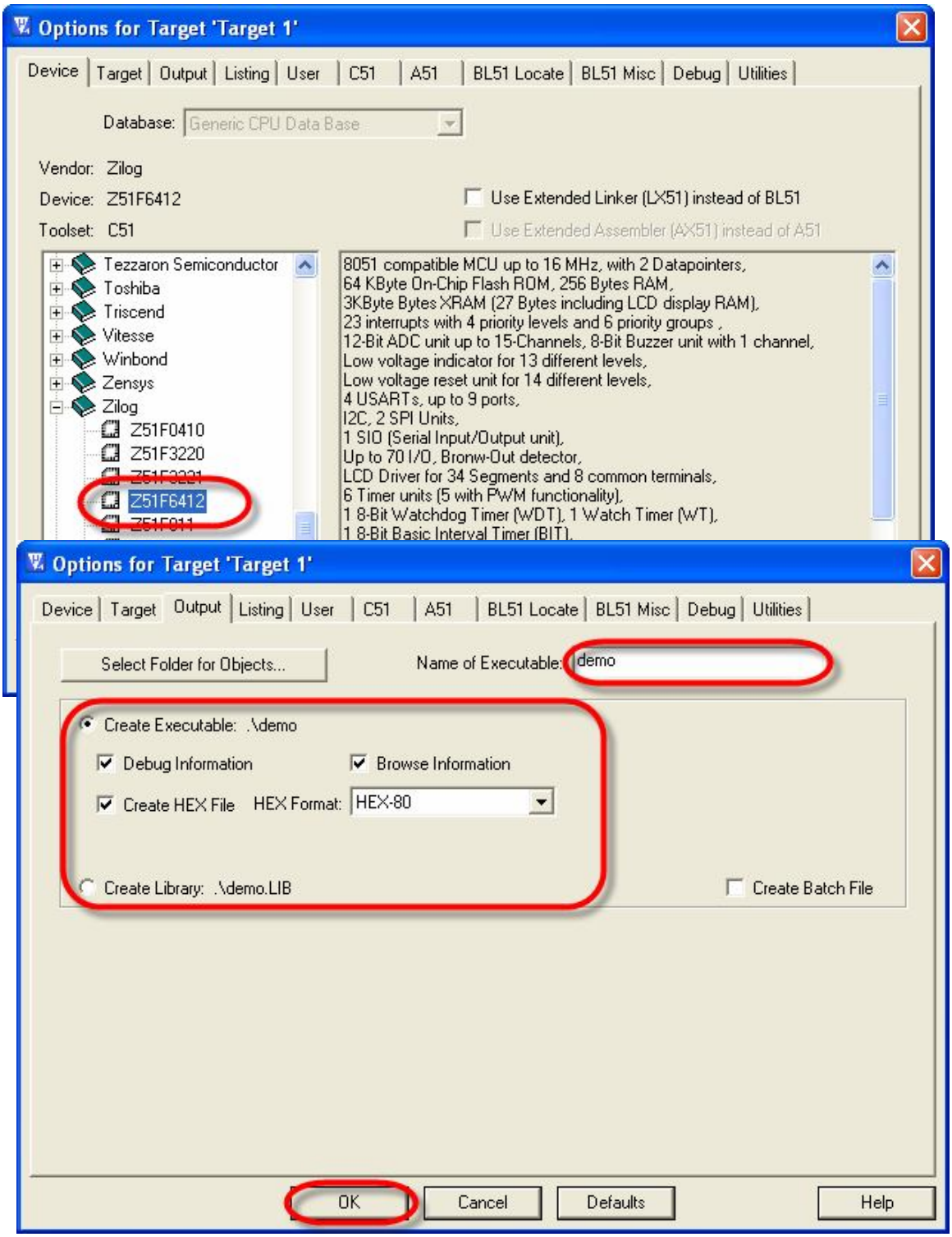
3. คลิกเมาส์ที่ Source Group1 แล้วคลิกขวา เลือก Add File to Group Source Group1 แล้วเลือกไฟล์ Source Code เป็น main.c เลือก Add ดังรูป



หรือเลือกที่เมนู Project → Manage →Components,Environment,Books... →Project Components ดังตัวอย่าง



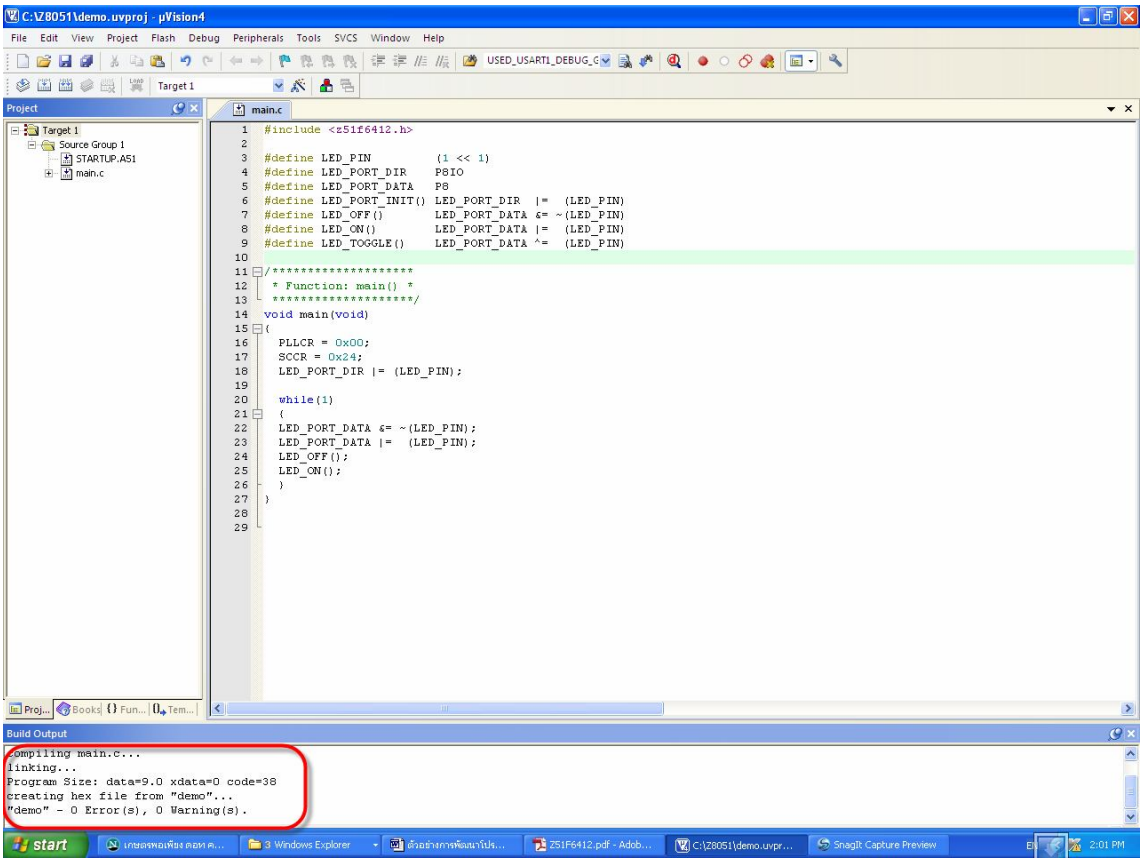
4. เลือกเมนู Project → Options for Target 'Target1'... แล้วคลิกเมาส์ที่ Tab ของ Output พร้อมกับเลือก Enable ที่เช็คบล๊อกของ Debug Information และ Create Hex File โดยเลือกรูปกำหนดรูปแบบของ Hex File เป็น HEX-80 พร้อมกับกำหนดชื่อ Output เป็น demo แล้วเลือก OK ดังตัวอย่าง



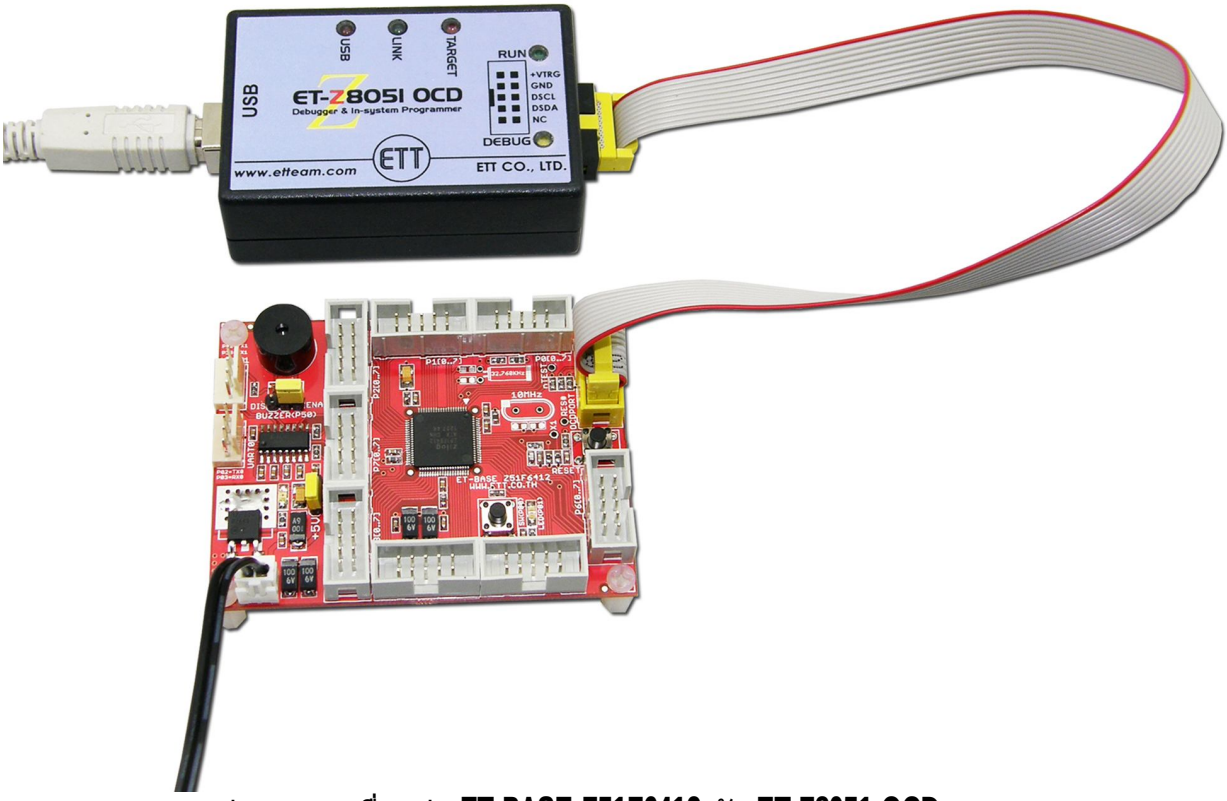
5. สั่งแปลคำสั่งของโปรแกรมที่เขียนขึ้นโดยเลือกที่เมนูProject → Rebuilds all target file ซึ่งถ้าทุกอย่างถูกต้องควรได้ผลการแปลคำสั่งเป็น 0 Error(s) และ 0 Warning(s) และได้ Output File จากการแปลคำสั่งชื่อ demo.hex ดังตัวอย่าง

```
Rebuild target 'Target 1'
assembling STARTUP.A51...
compiling main.c...
linking...
Program Size: data=9.0 xdata=0 code=63
creating hex file from "demo"...
"demo" - 0 Error(s), 0 Warning(s).
```

↓



การเชื่อมต่อ MCU กับ ET-Z8051 OCD ใน Debug Mode

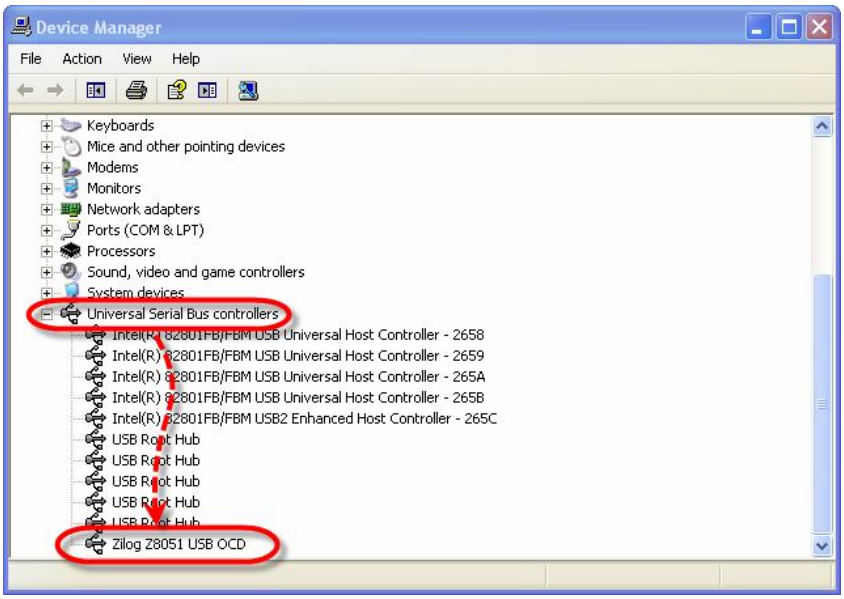


รูปแสดงการเชื่อมต่อ ET-BASE Z51F6412 กับ ET-Z8051 OCD

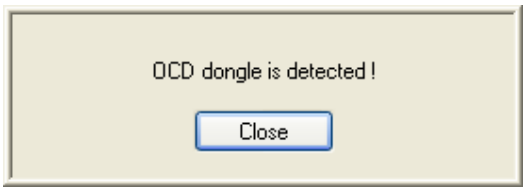
ในโหมดนี้จะเหมาะกับการใช้งานในรูปแบบที่อยู่ในขั้นตอนของการพัฒนาโปรแกรมเพราะสามารถสั่งโปรแกรม MCU พร้อมทั้งควบคุม ทดสอบ หยุดการทำงาน รวมไปถึงการตรวจสอบและปรับแต่งค่ารีจิสเตอร์ ต่างๆในขณะที่ MCU ทำงานอยู่จริงได้ด้วย ทำให้ไม่ต้องคอย ลบ และโปรแกรม MCU ซ้ำบ่อยๆ โดยในโหมดนี้จะใช้งานร่วมกับโปรแกรม Zilog Z8051 OCD ซึ่งปัจจุบัน (สิงหาคม 2555) โปรแกรมจะได้รับการปรับปรุงเป็นรุ่น Zilog Z8051 OCD Version1.147 โดยมีขั้นตอนการใช้งานพอสังเขปดังนี้

1. ทำการติดตั้งโปรแกรม Zilog Z8051 OCD ให้เรียบร้อย โดย Run File ชื่อ “Z8051_1.1.exe”
2. ทำการเชื่อมต่อสาย USB ของ ET-Z8051 OCD เข้ากับคอมพิวเตอร์ PC พร้อมกับทำการติดตั้ง Driver ของอุปกรณ์ให้เรียบร้อย ซึ่งขั้นตอนนี้จะกระทำเพียงครั้งแรกครั้งเดียวเท่านั้น โดยอุปกรณ์ ET-Z8051 OCD ของ อีทีที จะใช้ Driver ชุดเดียวกับ Zilog Z8051 OCD ของ Zilog Inc. ซึ่งตามปรกติ Driver จะถูกติดตั้งเตรียมไว้พร้อมกันกับการติดตั้งโปรแกรม ชื่อ “Z8051_1.1.exe” ในขั้นตอนที่ 1 ซึ่งถ้าติดตั้งโปรแกรมตามค่ามาตรฐาน Driver จะอยู่ที่ “\device drivers\OCD USB\” ภายใต้ Directory ที่ทำการติดตั้งโปรแกรมไว้ โดยปัจจุบันจะมี Driver สำหรับรองรับกับระบบปฏิบัติการ Windows สำหรับเครื่องคอมพิวเตอร์ทั้งแบบ 32บิต และ 64บิต คือ
   a. “C:\Program Files\Zilog\Z8051_1.1\device drivers\OCD USB\x32” สำหรับ 32บิต
   b. “C:\Program Files\Zilog\Z8051_1.1\device drivers\OCD USB\x64” สำหรับ 64บิต

ซึ่งถ้าทุกอย่างถูกต้องเรียบร้อยแล้ว LED USB สีแดง ที่ตัวเครื่อง ET-Z8051 OCD จะติดสว่าง และเมื่อเข้าไปตรวจสอบที่ Device Manager จะต้องพบอุปกรณ์ USB ใน Tab ของ Universal Serial Bus controller ควรพบอุปกรณ์ชื่อ “Zilog Z8051 USB OCD” ดังตัวอย่าง

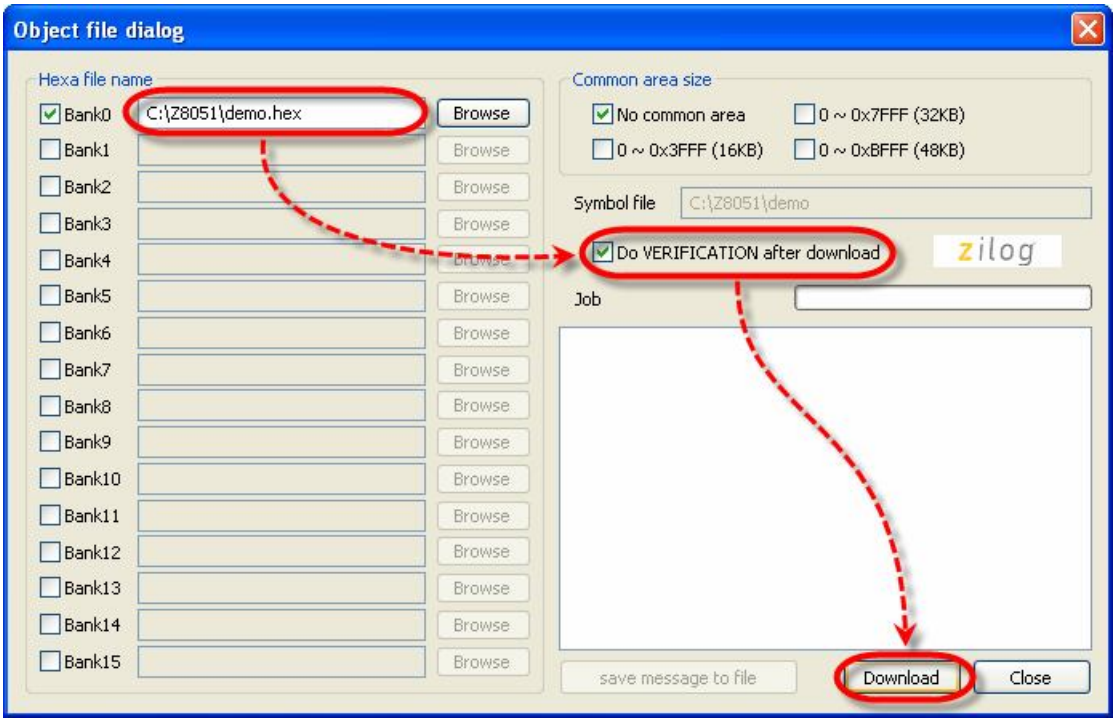


3. สั่ง Run โปรแกรม Zilog Z8051 OCD V1.147 ซึ่งเมื่อโปรแกรมเริ่มทำงานในครั้งแรก จะปรากฏ Dialog ข้อความแจ้งการตรวจพบ อุปกรณ์ Z8051 OCD ดังตัวอย่าง พร้อมกับที่อุปกรณ์ OCD ของ ET-Z8051 OCD จะปรากฏ LED LINK สีเขียว ติดสว่างให้เห็น เพื่อแสดงว่าอุปกรณ์ OCD พร้อมทำงาน

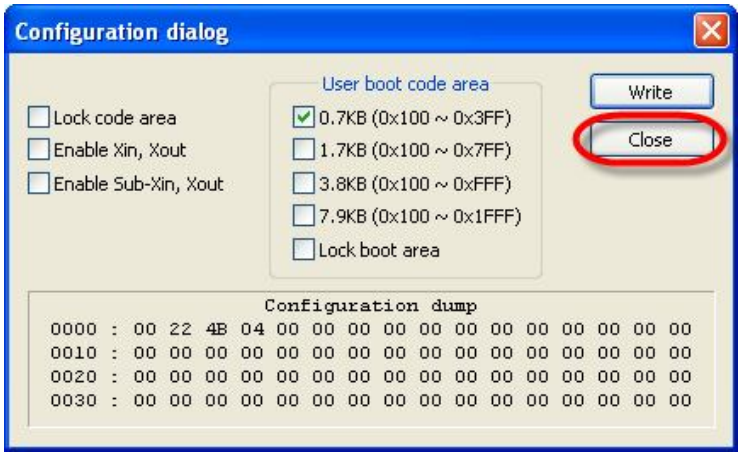


4. ถ้ามีการจ่ายไฟเลี้ยงวงจรให้บอร์ด ET-BASE Z51F6412 ไว้แล้วให้ทำการปลดสายไฟเลี้ยงของบอร์ด ET-BASE Z51F6412 ออกจากบอร์ดให้เรียบร้อยก่อน แล้วจึงทำการต่อสายแพร์ขนาด 10Pin ระหว่างเครื่อง ET-Z8051 OCD เข้ากับขั้วต่อ IDE10Pin สีเหลือง (PORT-OCD) ของบอร์ด ET-BASE Z51F6412 ให้เรียบร้อย จากนั้นจึงทำการจ่ายไฟเข้าบอร์ด ET-BASE Z51F6412 ซึ่งถ้าทุกอย่างถูกต้องหลอด LED สถานะของเครื่อง ET-Z8051 OCD จะติดสว่างหมดทั้ง 3 ดวง คือ USB LINK และ TARGET และที่หน้าจอโปรแกรมจะปรากฏ Dialog ข้อความ 2 ข้อความ ตามลำดับ ให้เห็น และปรากฏเบอร์ "Z516412ARX/ATX" ในช่อง Device name ดังตัวอย่าง

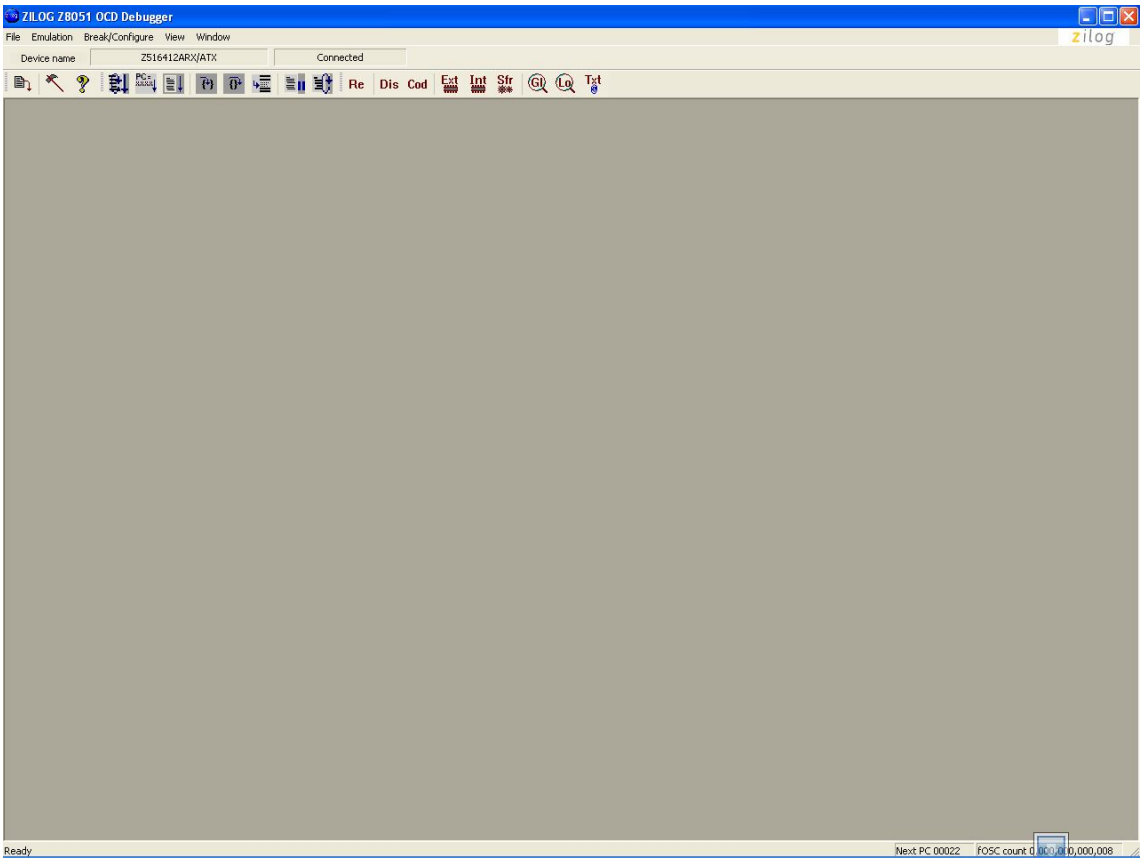
5. มาถึงขั้นตอนนี้ก็แสดงว่าการเชื่อมต่อระหว่าง MCU ในบอร์ด ET-BASE Z51F6412 กับเครื่อง OCD ของ ET-Z8051 OCD สามารถเชื่อมต่อกันได้อย่างถูกต้องเป็นที่เรียบร้อยแล้ว จากนี้ไปผู้ใช้ก็สามารถสั่งงานโปรแกรม Zilog Z8051 OCD ให้ทำงานต่างๆตามที่ต้องการได้แล้ว โดยขั้นตอนต่อไปที่ต้องทำคือทำการสั่งโปรแกรม Hex File ที่ต้องการจะ Download ให้กับ MCU ในบอร์ด โดยให้เลือกที่เมนู File → Load HEX แล้ว กำหนดชื่อไฟล์ที่ต้องการ แล้วสั่ง Download ดังตัวอย่าง



ส่วนของ Configuration ถ้าไม่มีการเปลี่ยนแปลงแก้ไขค่าตัวเลือกใดๆ ให้เลือก Close ได้เลย แต่ถ้าต้องการเปลี่ยนแปลงแก้ไขค่าใดก็ให้ทำการเลือกกำหนดค่าตามต้องการแล้วเลือก Write ในที่นี้ให้เลือก Close เพื่อกำหนดค่า Configuration ตามค่ามาตรฐานเดิมจากโรงงานดังตัวอย่าง



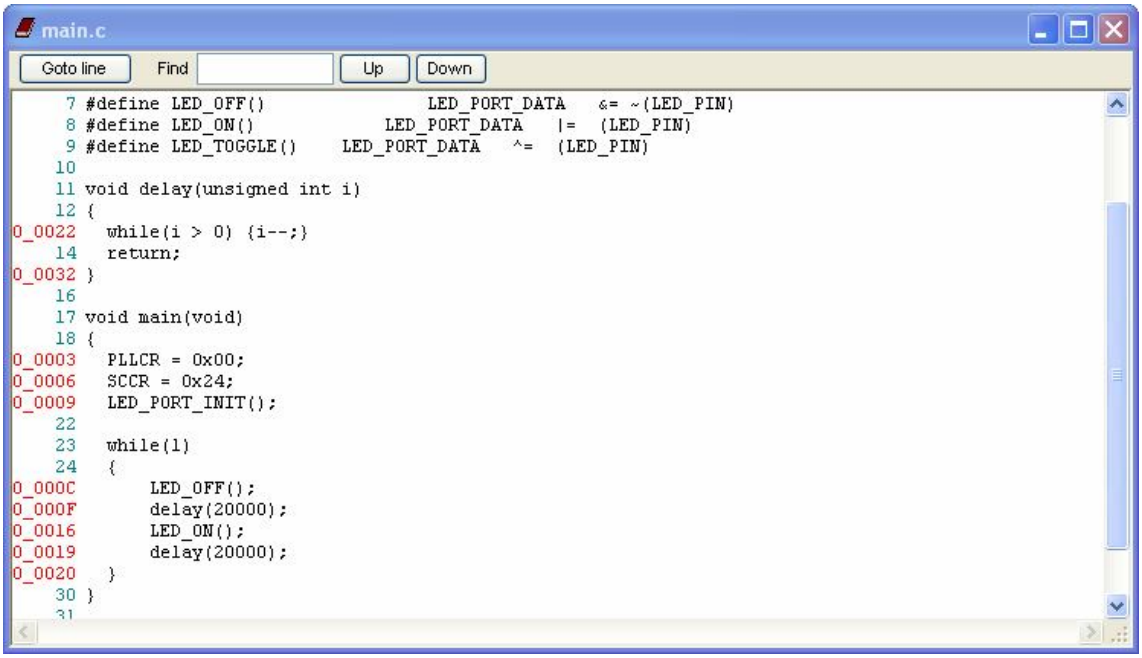
6. หลังจากสั่ง Download Hex File ให้กับ MCU เรียบร้อยแล้ว จากนี้ไปก็สามารถสั่งงาน MCU ให้ทำงานต่างๆตามต้องการได้ทันที ซึ่งอาจเป็นการสั่ง Run(Go) เพื่อดูผลการทำงานจริง หรือกำหนดตำแหน่งสำหรับหยุดการทำงาน (Break) หรือ สั่ง Run ทีละคำสั่ง(Step) เพื่อตรวจสอบผลการทำงานของโปรแกรมว่าถูกต้องตามที่ออกแบบไว้หรือไม่ ซึ่งในครั้งแรกโปรแกรมจะแสดงหน้าต่างโปรแกรมว่างๆ เนื่องจากเรายังไม่ได้กำหนดการแสดงผลของโปรแกรม ว่าต้องการให้โปรแกรมแสดงค่าของอะไรบ้าง ดังตัวอย่าง



โดยเราสามารถเลือกกำหนดให้หน้าจอของโปรแกรมแสดงค่าต่างๆของ Code โปรแกรม หรือ ตัวแปร และรีจิสเตอร์ต่างๆของ MCU ได้ตามต้องการ ซึ่งผู้ใช้สามารถปรับแต่งขนาดหน้าต่างการแสดงผลของส่วนต่างๆได้เองตามความเหมาะสม โดยเลือกที่เมนูคำสั่ง View แล้วเลือกสิ่งที่ต้องการแสดงผล หรือเลือกที่เมนูสัญลักษณ์ในส่วนของ Window open bar ดังตัวอย่าง

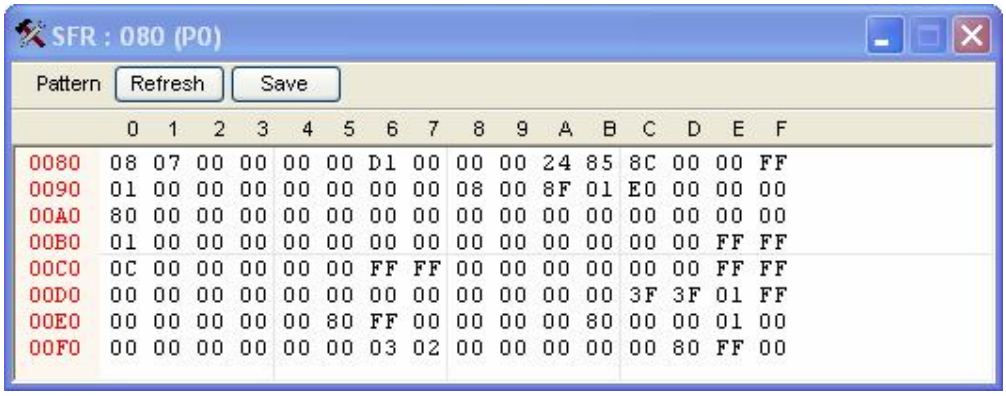
การแสดง Code คำสั่ง

การสั่งแสดง Code คำสั่ง ของโปรแกรม ให้เลือกที่เมนู View → Text file แล้วเลือกไฟล์ Source Code ที่เราต้องการดูค่า ซึ่งในที่นี้ให้เลือก main.c โดยโปรแกรมจะแสดงบรรทัด Code คำสั่งและตำแหน่งแอดเดรสของคำสั่งที่ได้จากการแปลคำสั่งให้เห็น ดังตัวอย่าง



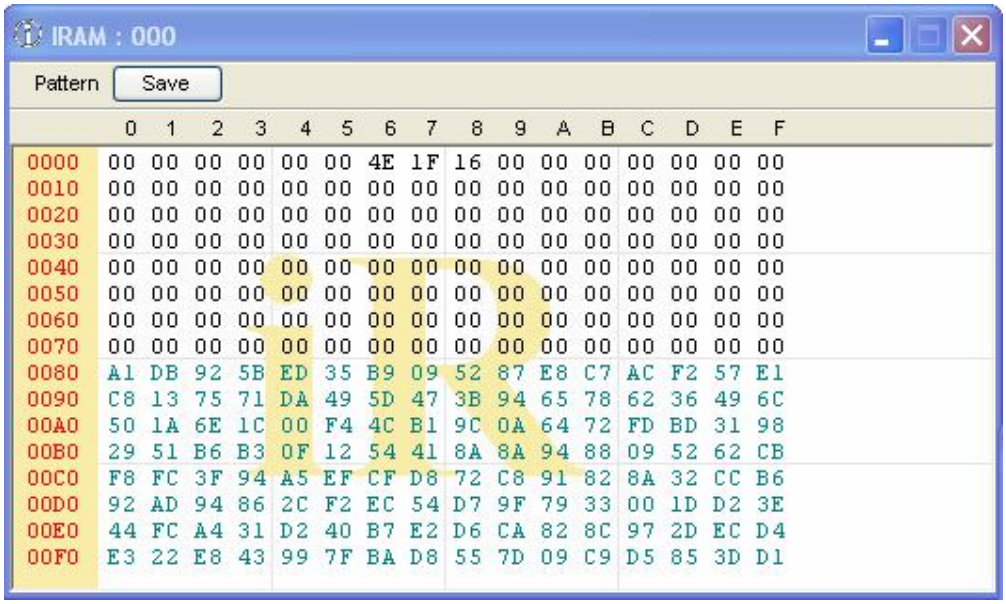
การแสดงค่า รีจิสเตอร์ และ SFR Register

การสั่งแสดงค่าของ Special Function Register ให้เลือกที่เมนู View → SFR dump ซึ่งโปรแกรมจะแสดงหน้าต่างแสดงค่า SFR Register พร้อมกับค่าในรีจิสเตอร์ต่างๆให้เห็นดังตัวอย่าง



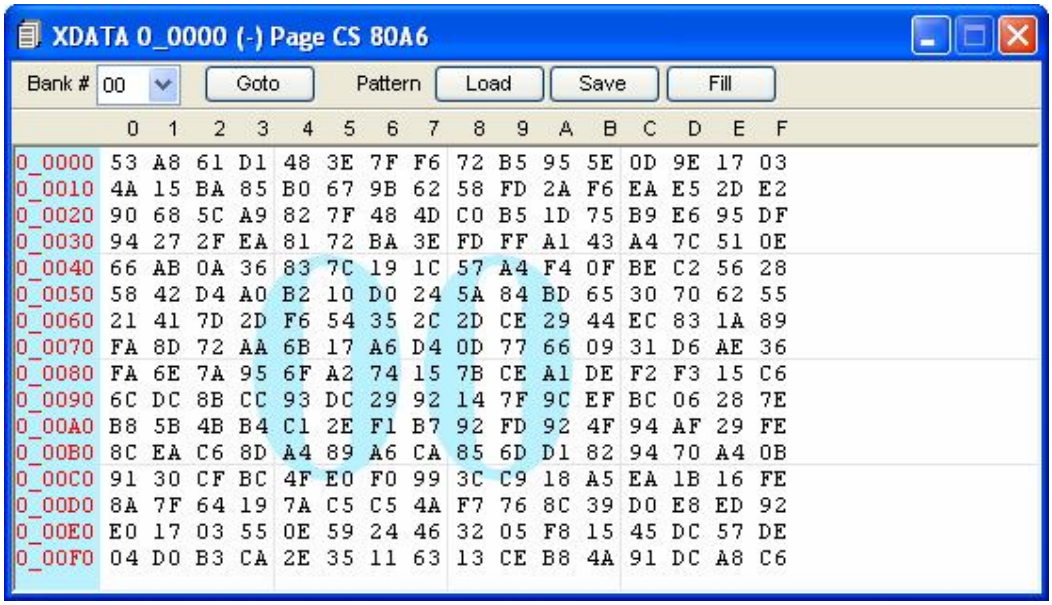
การแสดงค่าของ Internal RAM

การสั่งแสดงค่าของ Internal RAM ให้เลือกที่เมนู View → IRAM dump ซึ่งโปรแกรมจะแสดงหน้าต่างแสดงค่า Internal RAM พร้อมกับค่าในหน่วยความจำให้เห็นดังตัวอย่าง



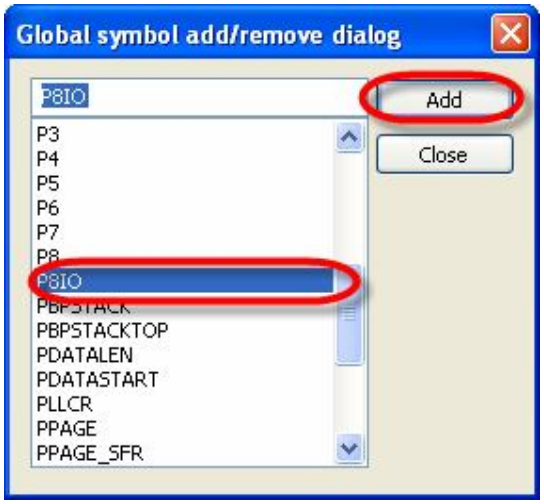
การแสดงค่าของ External RAM(XRAM)

การสั่งแสดงค่าของ Internal RAM ให้เลือกที่เมนู View → XDATA dump ซึ่งโปรแกรมจะแสดงหน้าต่างแสดงค่า XRAM พร้อมกับค่าในหน่วยความจำให้เห็นดังตัวอย่าง



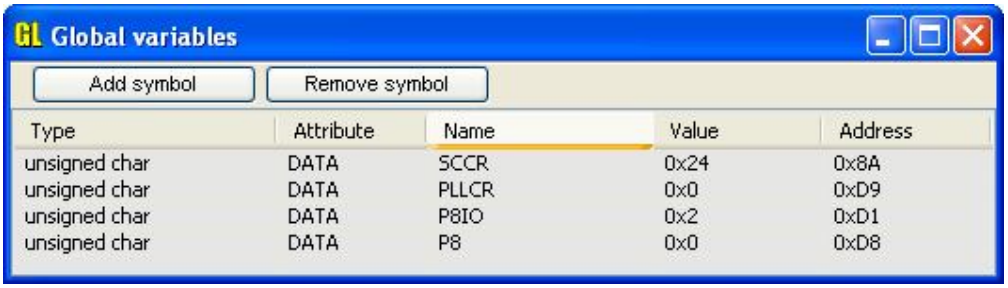
การแสดงค่าตัวแปร Global

การแสดงค่าตัวแปร Global เลือกเมนู View → Watch Global จากนั้นเลือกหัวข้อ Add symbol แล้วคลิกเมาส์เลือกชื่อตัวแปร หรือ รีจิสเตอร์ที่เราต้องการตรวจสอบค่าแล้วเลือก Add หรือถ้าต้องการปิดการแสดงผลของรีจิสเตอร์ตัวใดก็สามารถยกเลิกโดยเลือกที่ Remove symbol ตามต้องการ ดังตัวอย่าง



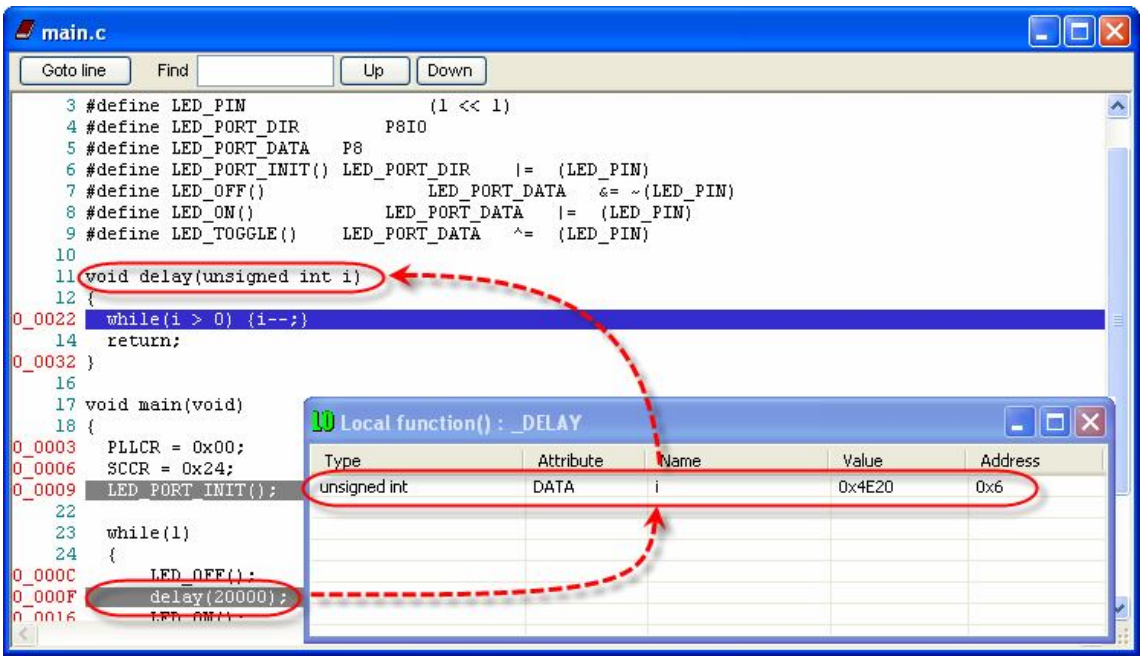
↑↓

↓↑

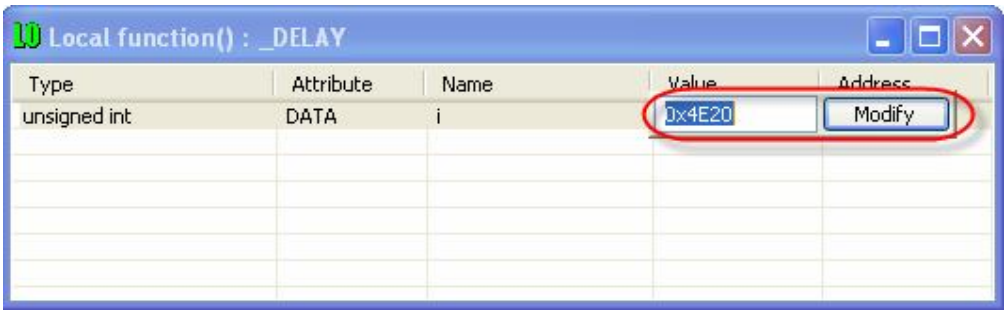


การแสดงค่าตัวแปร Local

การแสดงค่าตัวแปร Local เลือกเมนู View →Watch Local ซึ่งโปรแกรมจะแสดงหน้าต่างสำหรับแสดงค่าตัวแปร Local ที่ใช้ในโปรแกรม โดยแสดงชนิดตัวแปร ชื่อตัวแปร ตำแหน่งหน่วยความจำที่ใช้สร้างตัวแปร และค่าในตัวแปรให้เราทราบ ซึ่งตัวแปรเหล่านี้จะแสดงค่าให้เราทราบเมื่อโปรแกรมทำงานถึงบรรทัดคำสั่งที่มีการผ่านค่าให้ตัวแปรแล้วเท่านั้น เช่น เมื่อมีการเรียกใช้ฟังก์ชั่นหน่วงเวลา Delay จะมีการผ่านค่าตัวแปรไปให้กับตัวแปร i ซึ่งประกาศเป็น unsigned int ไว้ ก็จะปรากฏค่ารายละเอียดของตัวแปรนี้ให้เราเห็น ซึ่งเราสามารถสั่งหยุด และเข้าไปแก้ไขค่าให้กับตัวแปร เพื่อทดสอบการทำงานให้กับตัวแปรได้ เช่น เราอาจทดสอบปรับค่า Value ให้กับ ตัวแปร i เพื่อทดสอบค่าการหน่วงเวลาที่เหมาะสมดูได้ โดยคลิกเมาส์ที่ช่อง Value และแก้ไขค่าตามที่ต้องการแล้วเลือก Modify ดังตัวอย่าง

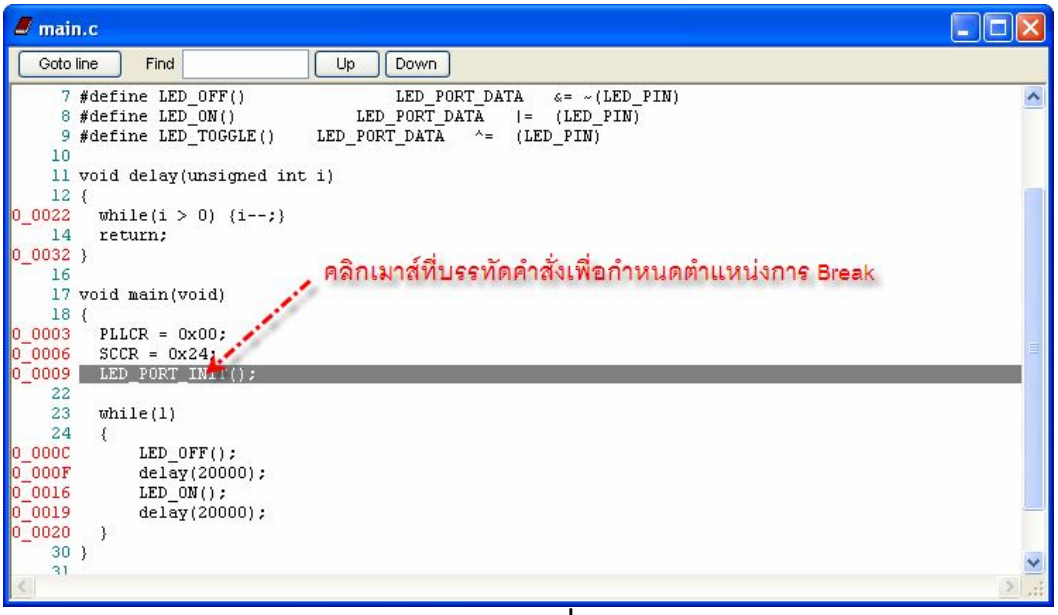


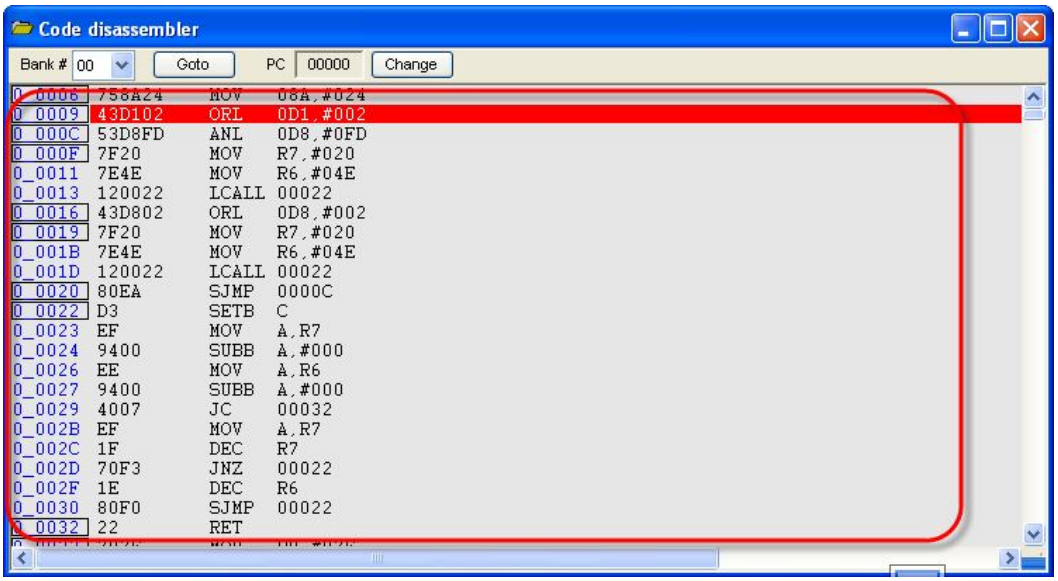
↓



การกำหนดตำแหน่ง Break

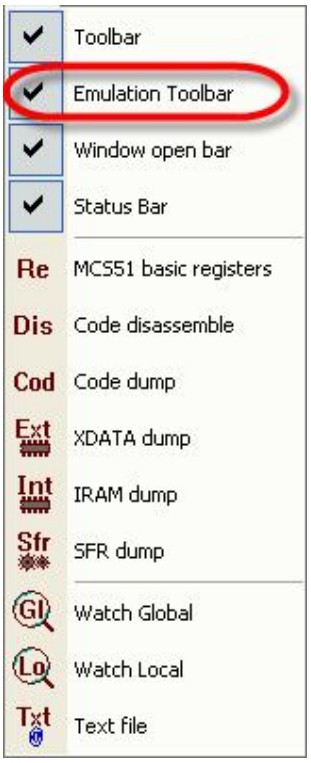
การกำหนดตำแหน่ง Break เพื่อหยุดการทำงานของโปรแกรมในบรรทัดคำสั่งที่เราต้องการเพื่อดูว่าเมื่อ MCU ทำงานถึงบรรทัดคำสั่งที่เรากำหนดแล้ว ค่าของรีจิสเตอร์ และ ตัวแปรต่างๆ มีค่าเป็นอย่างไรถูกต้องหรือไม่ โดยเราสามารถกำหนดตำแหน่งการ Break ได้จากหน้าต่างของ Text File ที่แสดงค่าของ Source Code โดยถ้าต้องการกำหนดการ Break ที่บรรทัดคำสั่งใดก็ให้คลิกเมาส์ที่บรรทัดคำสั่งนั้น ถ้าต้องการยกเลิกก็คลิกเมาส์ซ้ำที่ตำแหน่งเดิม ซึ่งถ้าเราสั่งแสดงค่าของ Code dissembler ไว้ด้วย ที่หน้าต่างของ Code disable ก็จะมีแถบสี แสดงบรรทัดคำสั่งภาษา Assemble ที่ตำแหน่งแอดเดรสเดียวกันกับบรรทัดคำสั่งที่เราสั่ง Break ด้วย ดังตัวอย่าง



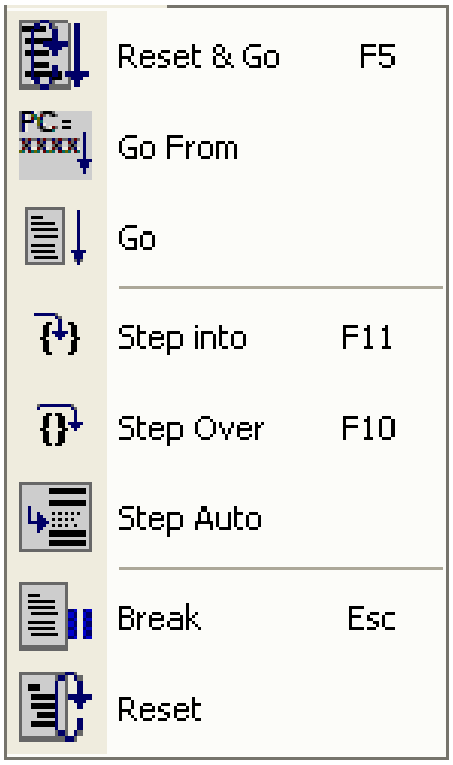


การสั่งงานโปแกรม

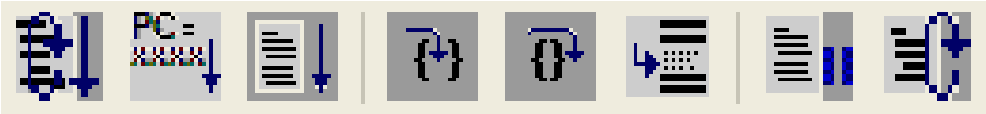
การสั่งงานโปรแกรม Z8051 OCD สามารถเลือกที่เมนู Emulation แล้วเลือกหัวข้อคำสั่งที่ต้องการ หรือสามารถเลือกที่เมนูสัญลักษณ์ ของ Emulation Toolbar โดยตรงก็ได้ โดยจะต้องทำการสั่งแสดงผลของ Emulation Toolbar ในเมนู View ไว้ดังตัวอย่าง



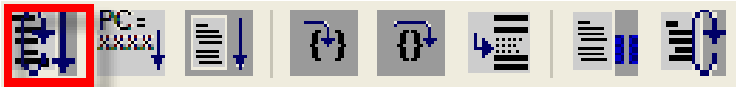
การแสดง **Emulation Toolbar** ในเมนู **View**



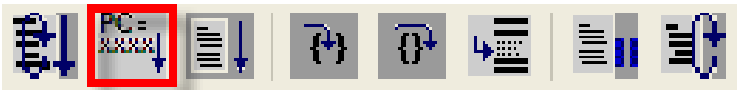
เมนูคำสั่งใน **Emulation**



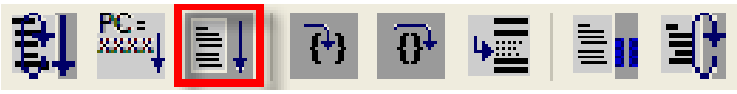
Emulation Toolbar

Reset & Go 

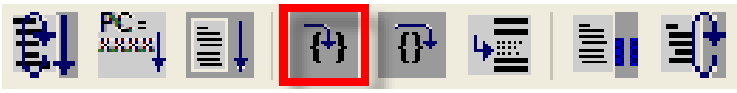
คำสั่งนี้ใช้สำหรับสั่งให้ Reset และ Run ด้วยความเร็วจริง โดยโปรแกรมจะสั่ง Reset การทำงานของ MCU และสั่งให้ MCU เริ่มต้นทำงานจากตำแหน่งแอดเดรส 0x0000

Go Form 

คำสั่งนี้ใช้สั่ง Run ด้วยความเร็วจริง แต่สามารถกำหนดตำแหน่งแอดเดรสเริ่มต้นที่ต้องการให้โปรแกรมทำงานได้

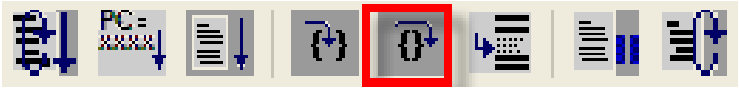
Go 

คำสั่งนี้ใช้สั่ง Run ด้วยความเร็วจริง โดยโปรแกรมจะเริ่มต้นทำงานต่อจากตำแหน่งแอดเดรสที่ PC ชี้อยู่ในขณะนั้น

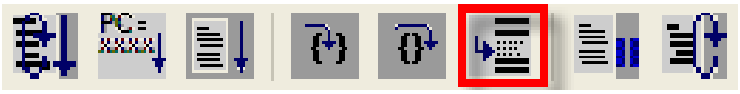
Step into 

ใช้สำหรับสั่ง Run โปรแกรมทีละคำสั่งเพื่อตรวจสอบผลการทำงานของโปรแกรมโดยละเอียด โดยเมื่อ MCU ทำงานตามคำสั่งเสร็จเรียบร้อยแล้วก็จะหยุดการทำงานเพื่อให้เราตรวจสอบผลการทำงาน โดยจะแสดงออกทั้งทาง Hardware และ Software โดยทาง Hardware ก็จะเกิดจากเปลี่ยนแปลงตามความเป็นจริงที่เกิดขึ้น ส่วนทาง Software ก็จะเกิดการเปลี่ยนแปลงทางรีจิสเตอร์ และ หน่วยความจำ

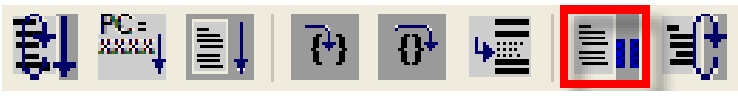
โดยคำสั่งนี้เหมาะสำหรับใช้ตรวจสอบความถูกต้องสมบูรณ์ของการทำงานของโปรแกรมย่อยหรือฟังก์ชั่นย่อยของโปรแกรมที่เขียนขึ้นโดยแยกทดสอบเป็นส่วนๆ

Step Over 

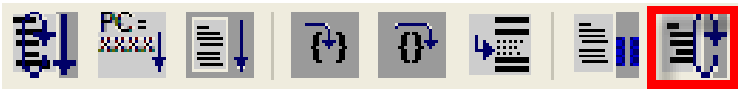
ใช้สำหรับสั่ง Run โปรแกรมทีละคำสั่ง ใช้สำหรับตรวจสอบการทำงานของ Main Program เพื่อทดสอบวงรอบการทำงานของโปรแกรมว่าถูกต้องดีหรือไม่ เหมาะสำหรับ ใช้ตรวจสอบผลการทำงานของโปรแกรม Main หลังจากทำการทดสอบการทำงานของแต่ละฟังก์ชั่นย่อยๆเสร็จสมบูรณ์หมดแล้ว

Step Auto 

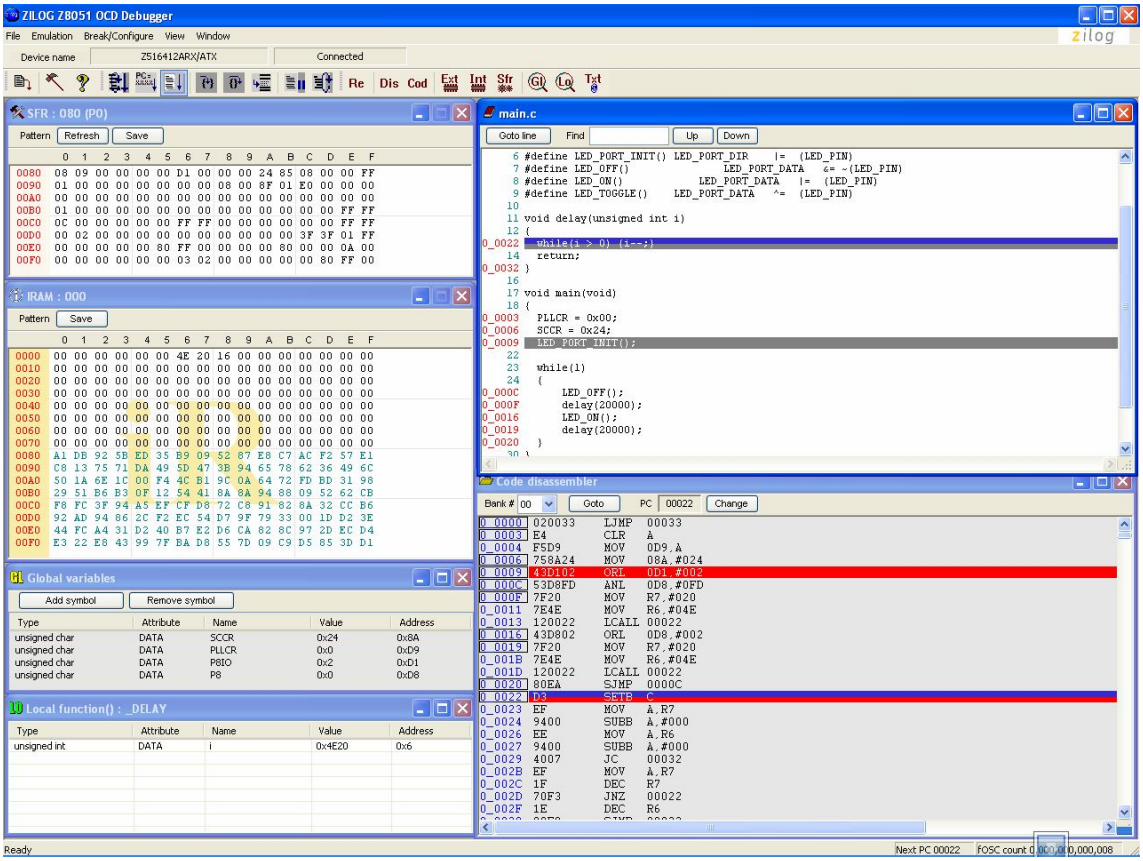
คำสั่งนี้ใช้สั่ง Run โปรแกรมแบบทีละ Step แบบต่อเนื่องกันไป โดยจะมีการแสดงค่าของรีจิสเตอร์ ตัวแปร และหน่วยความจำต่างๆ ที่เปลี่ยนแปลงไปจากการทำงานของคำสั่งให้เห็นตลอดเวลาด้วย ซึ่งการทำงานของโปรแกรมจะทำงานเป็น Step ดังนั้นความเร็วในการทำงานจะเกิดการสะดุด เป็นจังหวะการทำงานของโปรแกรมจึงช้ากว่าการ Run ตามปรกติ

Break 

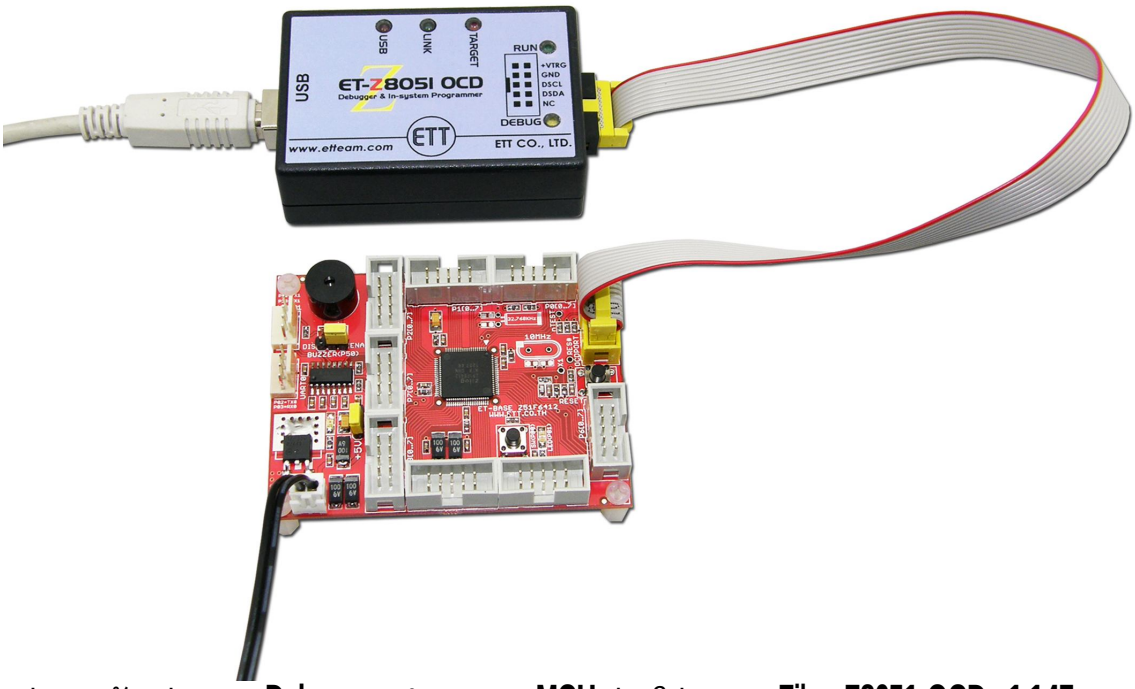
คำสั่งนี้ใช้สำหรับสั่งหยุดการทำงานของโปรแกรมที่กำลัง Run อยู่ในขณะนั้น ใช้สำหรับสั่งหยุดการทำงานของโปรแกรมเพื่อตรวจสอบค่า เปลี่ยนแปลง แก้ไขค่า ของรีจิสเตอร์ ตัวแปร และหน่วยความจำต่างๆ

Reset 

คำสั่งนี้ใช้สำหรับสั่งรีเซ็ตการทำงานของ MCU และกำหนดค่า PC ให้กลับไปเริ่มต้นเตรียมพร้อมที่ตำแหน่งแอดเดรส 0x0000



↑↓



รูปแสดงตัวอย่าง การ **Debug** การทำงานของ **MCU** ผ่านโปรแกรม **Zilog Z8051 OCD v1.147**