ET-AVR JTAG (RS232) V1.0

ET-AVR JTAG (RS232) V1.0 เป็นบอร์ดที่ออกแบบมาเพื่อใช้ในการดาวน์โหลด Hex File และ ดีบัก ให้กับ MCU ตระกูล AVR ของ Atmel โดยผ่านทาง JTAG Interface ซึ่งสามารถใช้ได้กับ MCU ที่มีโมดูล JTAG Interface เท่านั้น โดยต้องใช้ร่วมกับโปรแกรม AVR Studio 4.XX

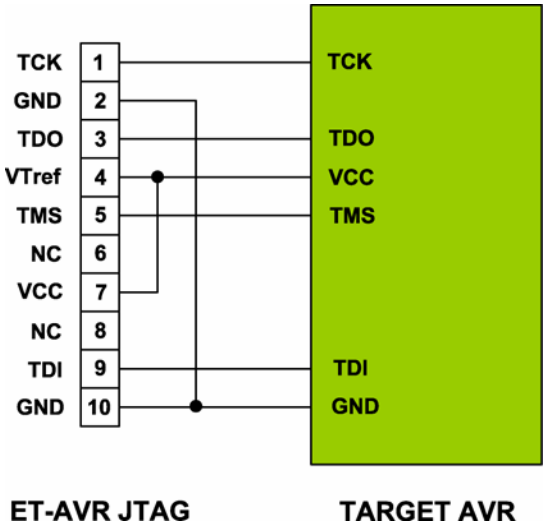
คุณสมบัติของ ET-AVR JTAG (RS232) V1.0

1. มีคุณสมบัติเทียบเท่า AVR JTAG ICE ของ ATMEL
2. สนับสนุนการดีบักแบบเรียลไทม์
3. โปรแกรมและดีบัก MCU ที่มีโมดูล JTAG Interface
4. สามารถอัพเกรด Firmware โดยตรงผ่านโปรแกรม AVR Studio 4 เพื่อให้สามารถใช้กับ MCU เบอร์ใหม่ ๆ ได้ ซึ่ง Firmware จะติดมากับโปรแกรม AVR studio 4
5. สามารถใช้ได้กับระบบไฟเลี้ยงตั้งแต่ 2.7V – 5.5V
6. ใช้แรงดันจากบอร์ด Target
7. การติดต่อสื่อสารผ่านพอร์ตอนุกรม (RS232)
8. มี LED แสดงสถานการณ์ทำงาน Power , Activity

เบอร์ของไมโครคอนโทรลเลอร์ AVR ที่สามารถใช้ได้กับ ET-AVR JTAG (AVR studio 4.12)

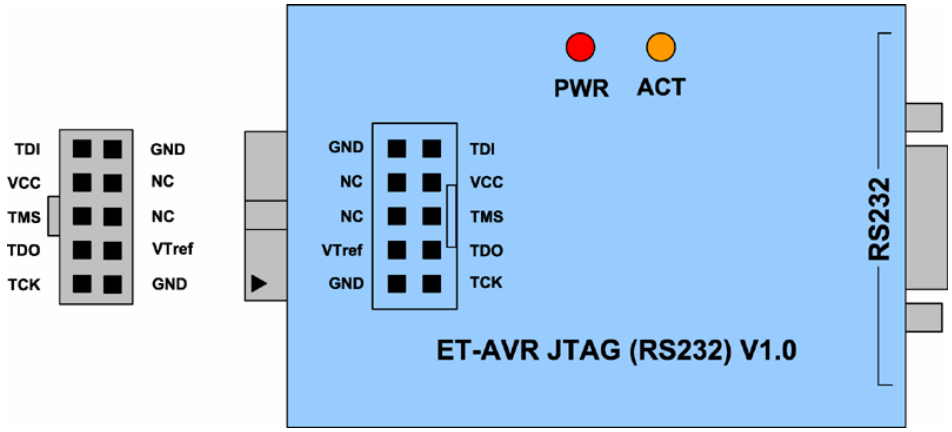
- ATmega16 , ATmega16L
- ATmega162 , ATmega162L
- ATmega169 , ATmega169L , ATmega169V
- ATmega32 , ATmega32L
- ATmega323 , ATmega323L
- ATmega64 , ATmega64L
- ATmega128 , ATmega128L
- AT90CAN128

การเชื่อมต่อ ET-JTAG AVR กับไมโครคอนโทรลเลอร์ AVR

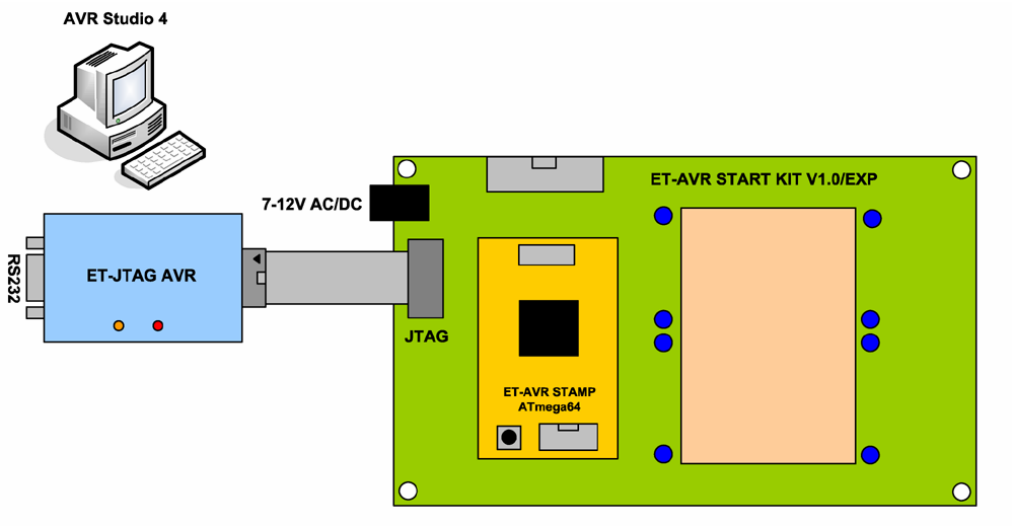


แสดงการเชื่อมต่อ ET-AVR JTAG กับ ไมโครคอนโทรลเลอร์ AVR

การต่อใช้งานจะใช้สายสัญญาณ TCK , TDO , TMS , TDI , VCC , GND เท่านั้นส่วนขาสัญญาณ VTref ไม่จำเป็นต้องต่อก็ได้เนื่องจากวงจรของ ET-JTAG AVR ขานี้จะต่อกับ VCC อยู่แล้ว



แสดงตำแหน่งขาสัญญาณของ ET-AVR JTAG

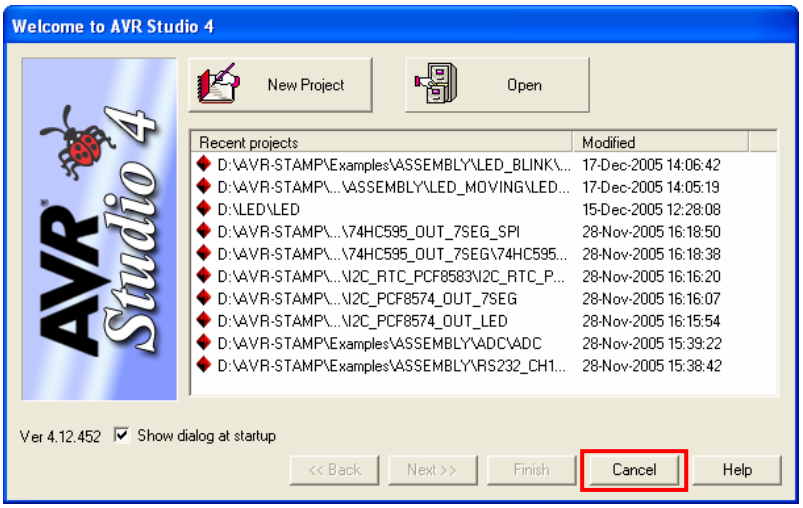


แสดงการเชื่อมต่อ ET-AVR JTAG กับ ET-AVR START KIT V1.0/EXP

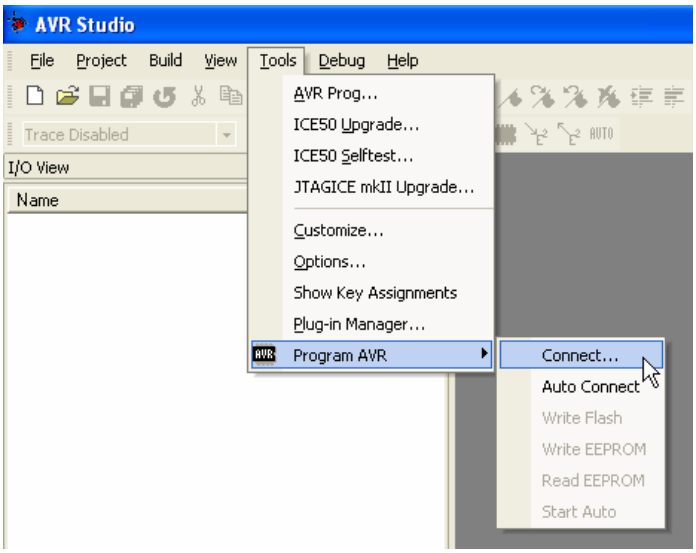
การใช้งาน ET-AVR JTAG ในโหมดโปรแกรม

การในงานในโหมดโปรแกรมนี้สามารถกำหนด Security Bits และ Configuration Bits เช่นเดียวกับโปรแกรม PonyProg2000 ซึ่งวิธีการใช้งานมีดังนี้

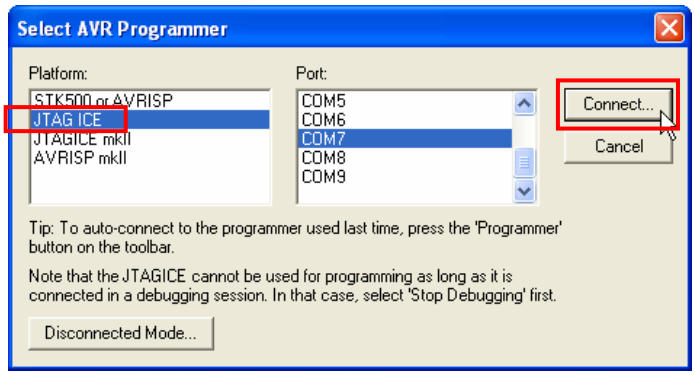
1. ทำการเปิดโปรแกรม AVR Studio จากนั้นจะปรากฏหน้าต่าง Welcome to AVR Studio ให้คลิกที่ Cancel เพื่อปิดหน้าต่างนี้



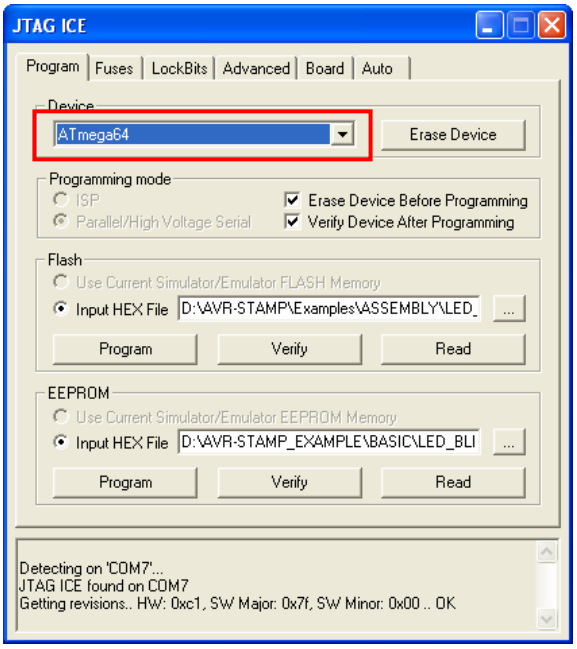
2.ทำการเลือกที่เมนู Tools → Program → AVR Connect… ดังรูป



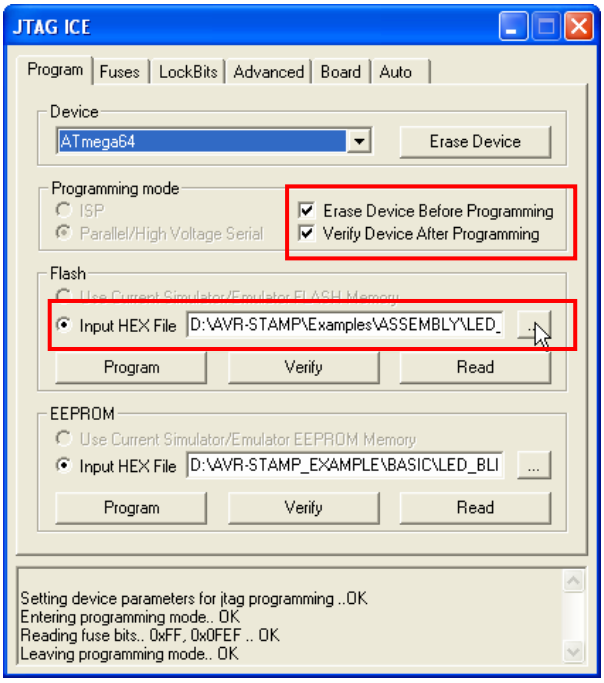
2. จากนั้นจะปรากฏหน้าต่าง Select AVR Programmer ดังรูป ให้ทำการเลือก Platform เป็น JTAG ICE และ Port ที่ต่อ ET-AVR JTAG ต่ออยู่ และคลิกปุ่ม Connect…

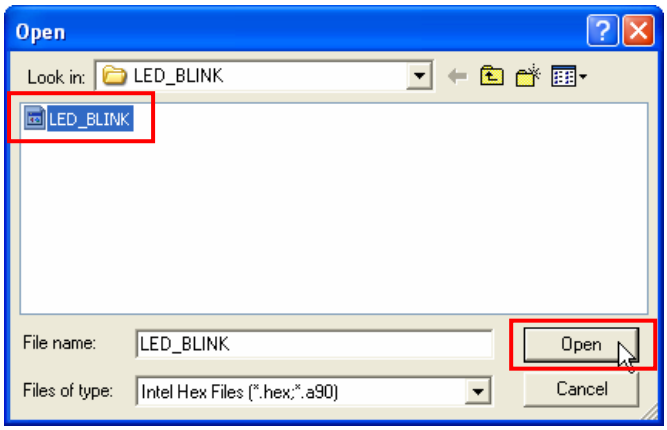


3. ถ้าโปรแกรม AVR Studio สามารถติดต่อกับ ET-JTAG AVR ได้จะปรากฏหน้าต่าง JTAG ICE ดังรูป ทำการเลือกเบอร์ MCU จากช่อง Device ในที่นี้เลือกเป็น ATmega64

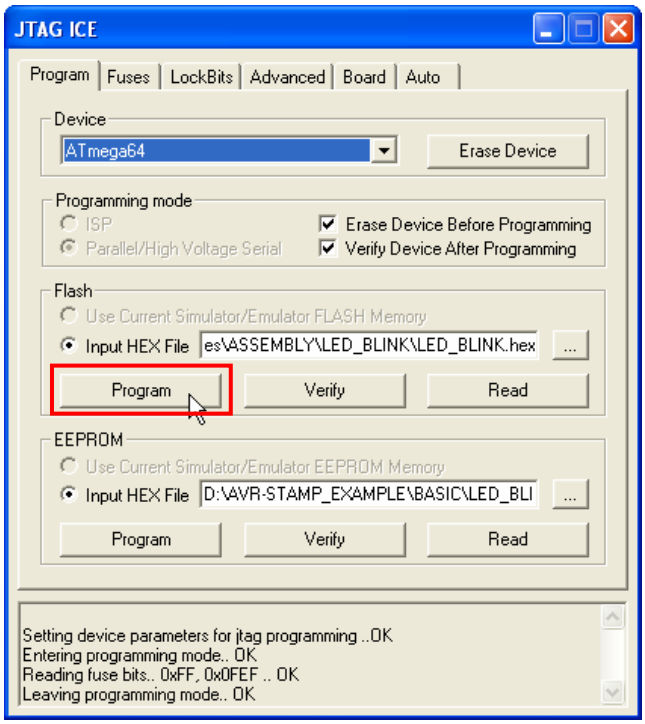


4. สั่งเปิดไฟล์สำหรับที่จะใช้โปรแกรมให้กับ CPU โดยเลือกจากช่อง Input HEX File พร้อมทั้งระบุชื่อและที่อยู่ของ HEX File ที่จะใช้โปรแกรมให้เรียบร้อย ส่วนในช่อง Programming mode ให้เลือก Erase Device Before Programming เพื่อทำการลบข้อมูลก่อนการโปรแกรม และ Verify Device After Programming เพื่อทำการตรวจสอบความถูกต้องของข้อมูลหลังจากโปรแกรม

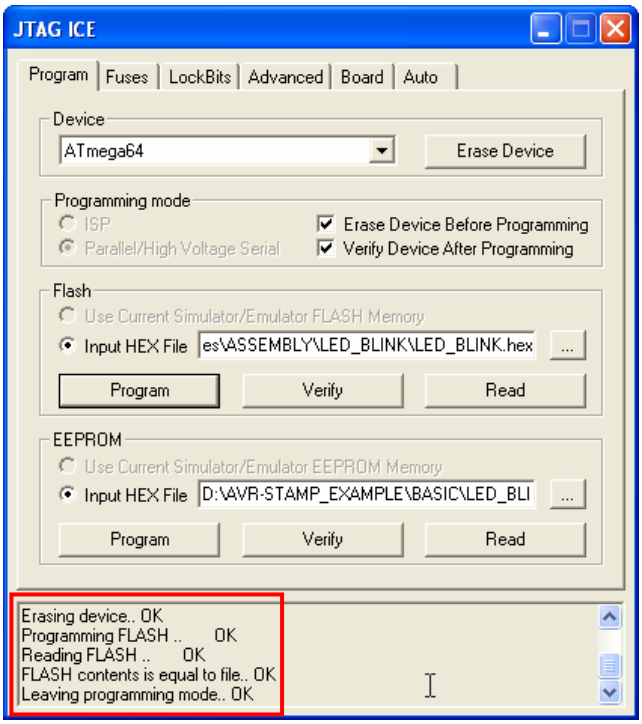




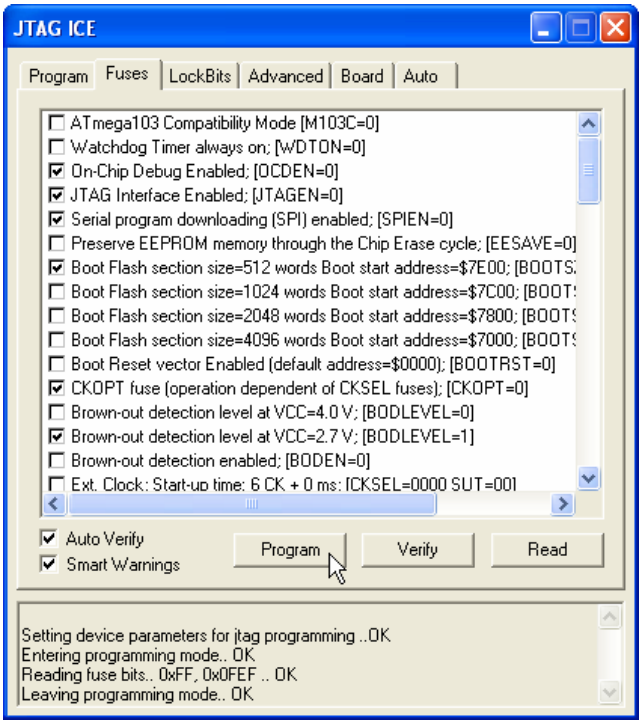
5. ในกรณีที่ไม่ต้องการแก้ไขค่า Fuses และ LockBits ก็สามารถกดปุ่ม Program เพื่อโปรแกรม Hex Files ให้กับ MCU ได้เลย เนื่องจากค่า Fuses และค่า LockBits เมื่อโปรแกรมไปครั้งหนึ่งแล้วค่าจะยังเหมือนเดิม ไม่ถูกลบไปพร้อมกับส่วนของโปรแกรม ถ้าจะแก้ไขค่าก็สามารถโปรแกรมเข้าไปทีหลังโดยที่ไม่กระทบต่อส่วนของโปรแกรม

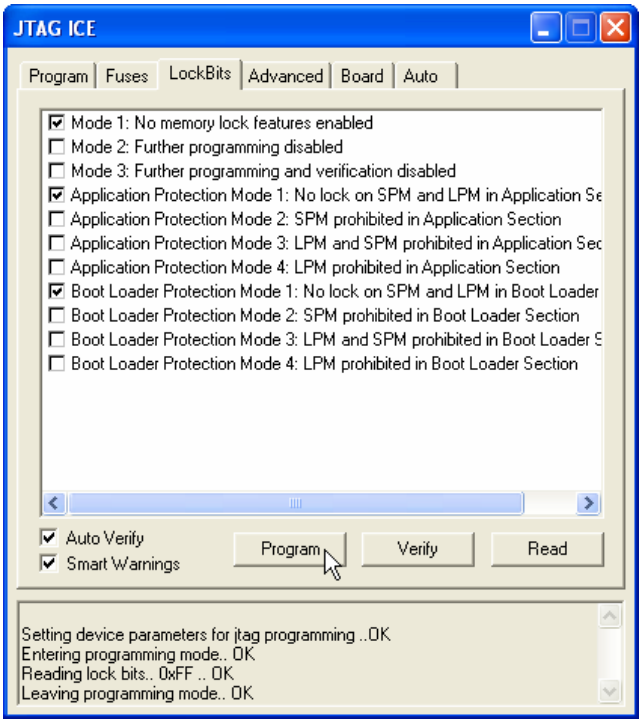


6. เมื่อการโปรแกรม Hex Files ให้กับ MCU ไม่มีข้อผิดพลาดจะได้ข้อความดังรูป



7. ส่วนถ้าต้องการแก้ไขค่า Fuses และค่า LockBits ก็สามารถเลือกไปที่ Fuses และ Lockbits ทำการตั้งค่าและโปรแกรมไปทีหลังได้ตามรูป





หมายเหตุ

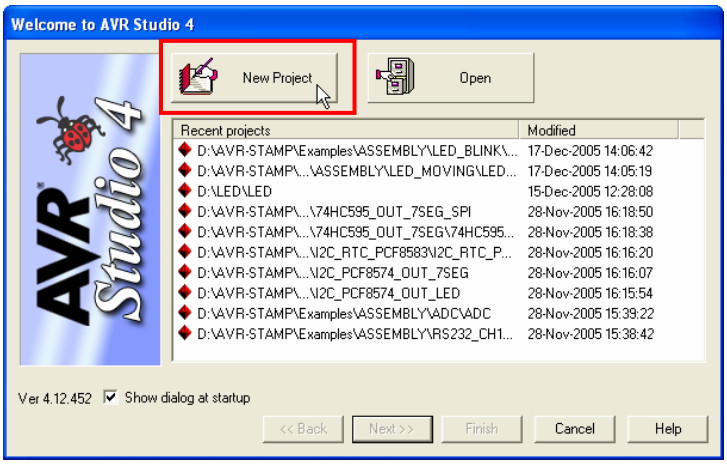
ถ้าในการใช้งานบอร์ดเพื่อทดลองต่างๆ ค่า Lockbits นี้ไม่จำเป็นต้องโปรแกรม ซึ่งค่านี้จะใช้เพื่อป้องกันการอ่านและเขียนข้อมูลทับ MCU

การใช้งาน ET-AVR JTAG ในโหมดดีบัก

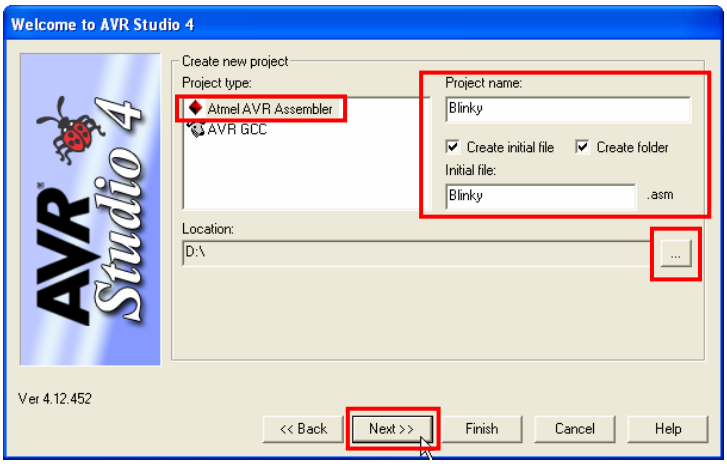
การทำงานในโหมดนี้จะใช้เพื่อดีบัก ดูสถานะการทำงานของ MCU โดยสามารถดีบักทีละ Step หรือแบบอัตโนมัติ โดยระหว่างการดีบักค่าต่าง ๆ ของ MCU จะเปลี่ยนแปลงตามโปรแกรมทำให้สามารถดูค่าผลลัพธ์ต่างๆ ได้ทันทีซึ่งการดีบักนี้สามารถทำได้ทั้งภาษาแอสเซมบลีและภาษาซี เช่น ถ้าเขียนโปรแกรมไฟวิ่งก็จะเห็นไฟวิ่งตามสถานะการดีบักเป็นต้น ซึ่งขั้นตอนการใช้งานในโหมดดีบักจะเป็นดังนี้ โดยจะเป็นตัวอย่างไฟกระพริบ 1 ดวง เพื่อให้ง่ายต่อการสังเกต

ตัวอย่างการดีบักโดยใช้ภาษาแอสเซมบลี

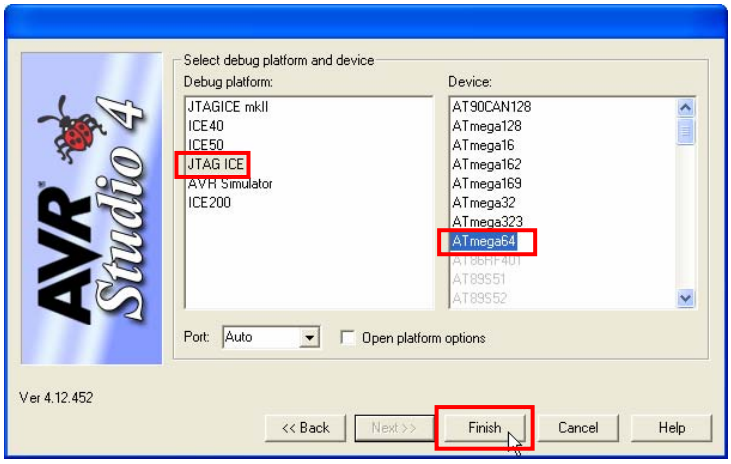
1. ทำการเปิดโปรแกรม AVR Studio จากนั้นจะปรากฏหน้าต่าง Welcome to AVR Studio ให้คลิกที่ New Project เพื่อสร้างโปรเจกต์ใหม่ดังรูป



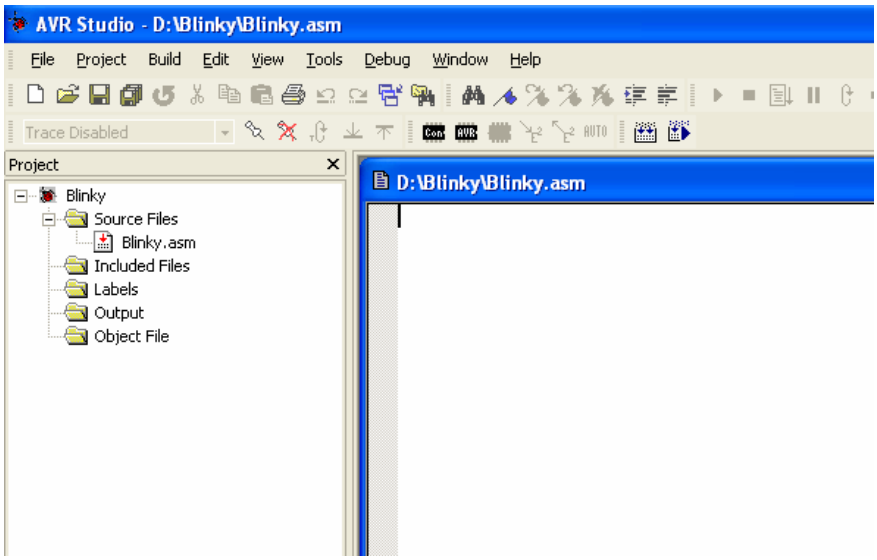
2. เลือก Project type เป็น Atmel AVR Assembler เพื่อเขียนโปรแกรมเป็นภาษาแอสเซมบลี ทำการตั้งชื่อโปรเจกต์ในช่อง Project name เลือกที่ช่อง Create initial file เพื่อสร้างไฟล์แอสเซมบลีพร้อมกับสร้างไฟล์โปรเจคต์ เลือกที่ช่อง Create folder เพื่อสร้างโฟลเดอร์สำหรับเก็บไฟล์โปรเจกต์ จากนั้นทำการเลือกไดเร็กทอรี่ที่จะเก็บไฟล์โปรเจคและคลิกปุ่ม Next ดังรูป



3. เลือก Debug platform เป็น JTAG ICE และ Device เป็น ATmega64 และคลิกปุ่ม Finish ดังรูป



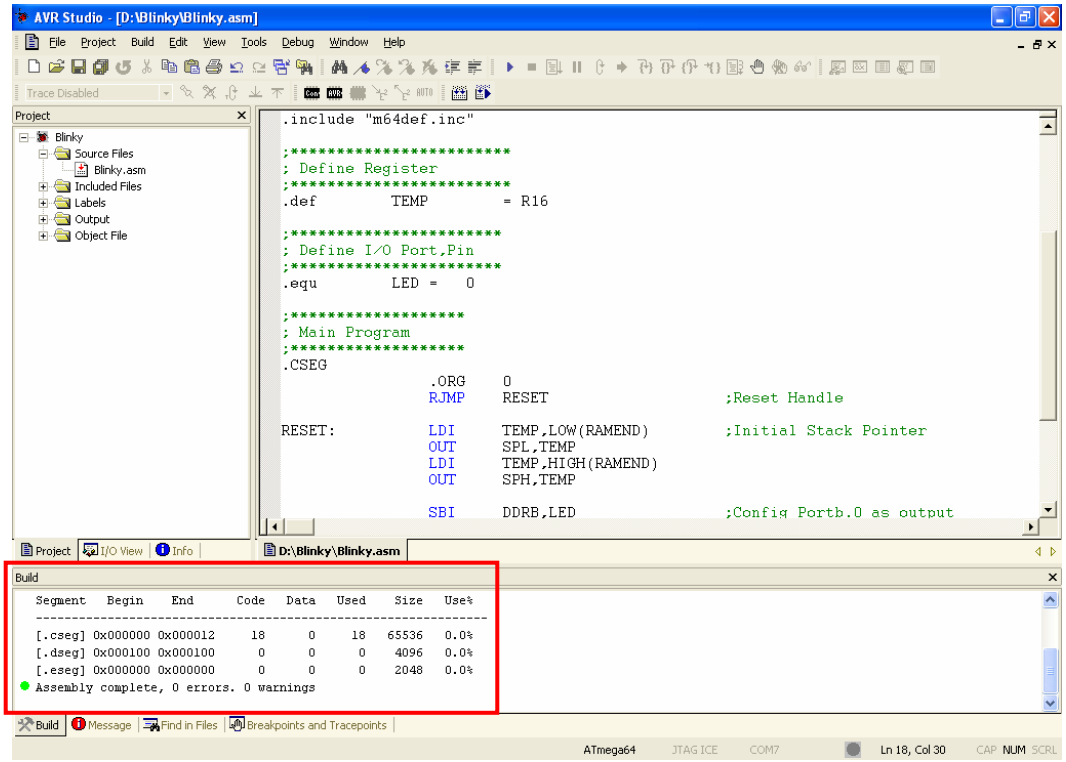
4. จากนั้นจะปรากฏหน้าต่าง Text Editor สำหรับเขียนโปรแกรมดังรูป



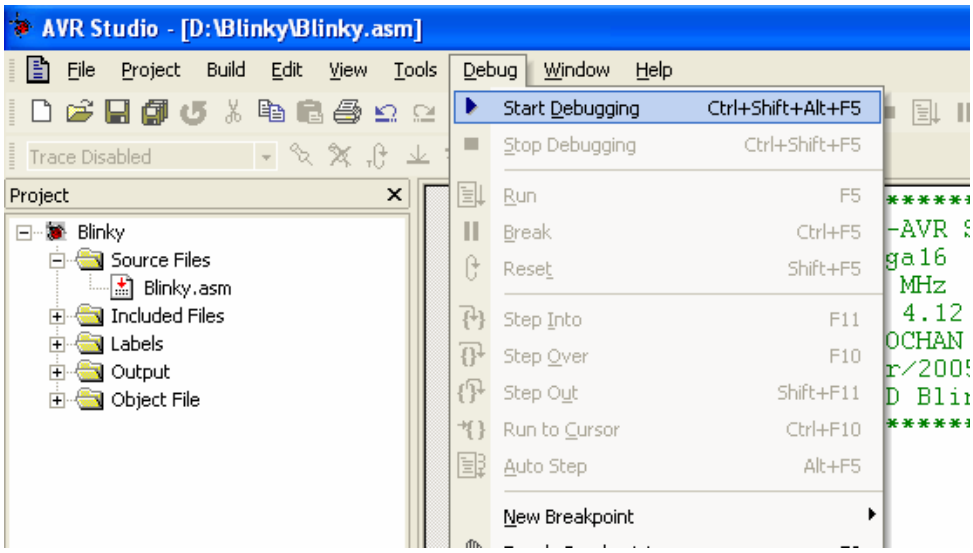
5. ทำการพิมพ์โปรแกรมตัวอย่างภาษาแอสเซมบลี ดังตัวอย่าง จากตัวอย่างไม่ได้ใช้โปรแกรมหน่วงเวลา เพื่อว่าที่จะสังเกตเห็นการดีบักอย่างทันทีไม่มีการหน่วงเวลา

```
;*********************************************************
;* Examples Program For "ET-AVR STAMP ATmega64" Board  *
;* Target MCU  : Atmel ATmega16                        *
;* Frequency   : X-TAL : 16 MHz                        *
;* Compiler    : AVR Studio 4.12 (AVR Assembler 2)     *
;* Create By   : ADISAK CHOOCHAN (WWW.ETT.CO.TH)       *
;* Last Update : 1/September/2005                      *
;* Description : Example LED Blink on Portb.0          *
;*********************************************************

;Connect PB0 to LED1

.include "m64def.inc"

;************************
; Define Register
;************************
.def      TEMP      = R16

;************************
; Define I/O Port,Pin
;************************
.equ      LED  =    0

;*******************
; Main Program
;*******************
.CSEG
          .ORG   0
          RJMP  RESET               ;Reset Handle

RESET:    LDI  TEMP,LOW(RAMEND)     ;Initial Stack Pointer
          OUT  SPL,TEMP
          LDI  TEMP,HIGH(RAMEND)
          OUT  SPH,TEMP

          SBI  DDRB,LED             ;Config Portb.0 as output

MAIN:     SBI  PORTB,LED            ;LED Off
          CBI  PORTB,LED            ;LED On
          RJMP MAIN                 ;Loop
```

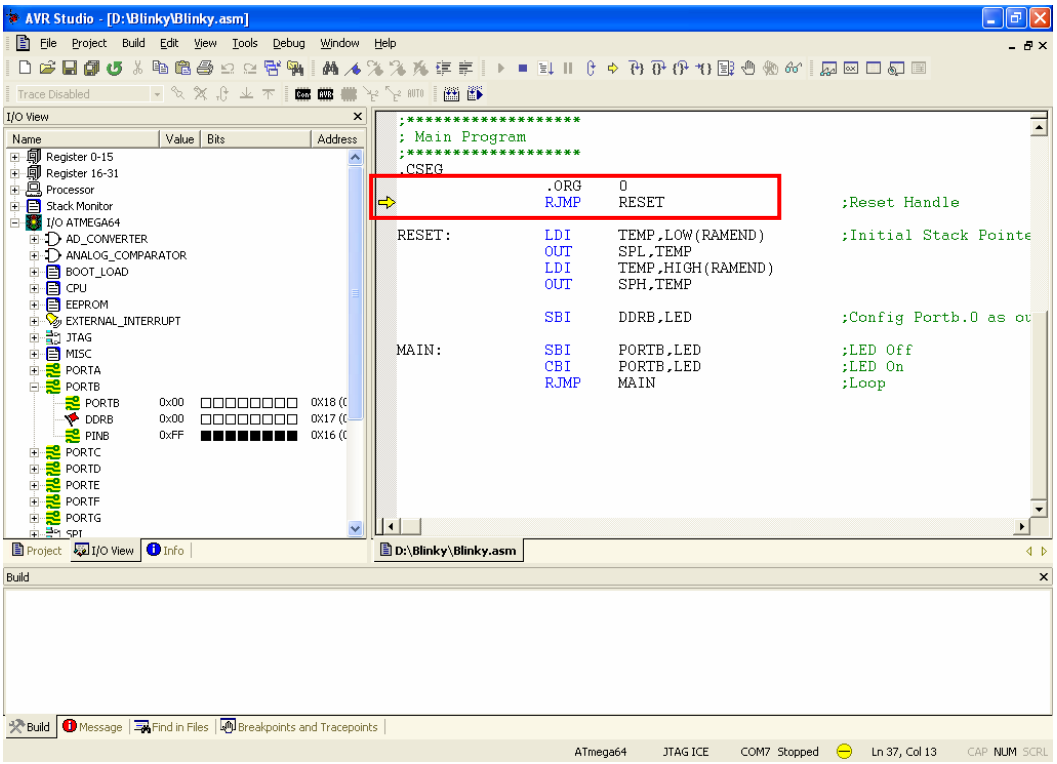
6. ให้ทำการสั่งแปลโปรแกรมที่เราเขียนขึ้น โดยการคลิกเมาส์ที่เมนูคำสั่ง Build → Build ซึ่งหลังจากแปลโปรแกรมแล้วได้ผลถูกต้องและไม่เกิดข้อผิดพลาดใด ๆ จะปรากฏข้อความ 0 errors 0 warnings ดังรูป



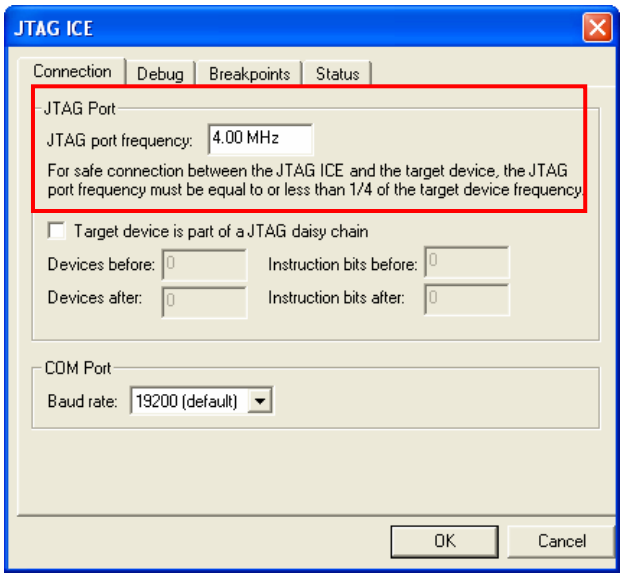
7. คลิกที่เมนูคำสั่ง Debug → Start Debugging ดังรูป



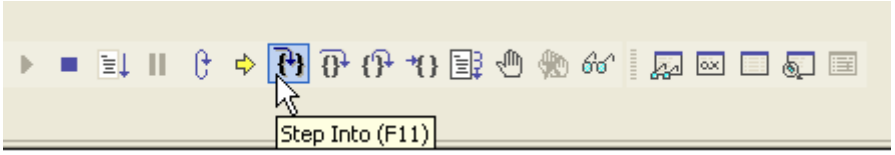
8. จากนั้นโปรแกรมจะทำการโหลดข้อมูลต่างลง MCU และเข้าสู่การดีบัก โดยจะมีเครื่องหมายแสดงจุดเริ่มต้นของโปรแกรมดังรูป โดยทางด้านขวาจะปรากฏหน้าต่าง I/O View แสดงค่ารีจิสเตอร์ต่าง ๆ ของ MCU



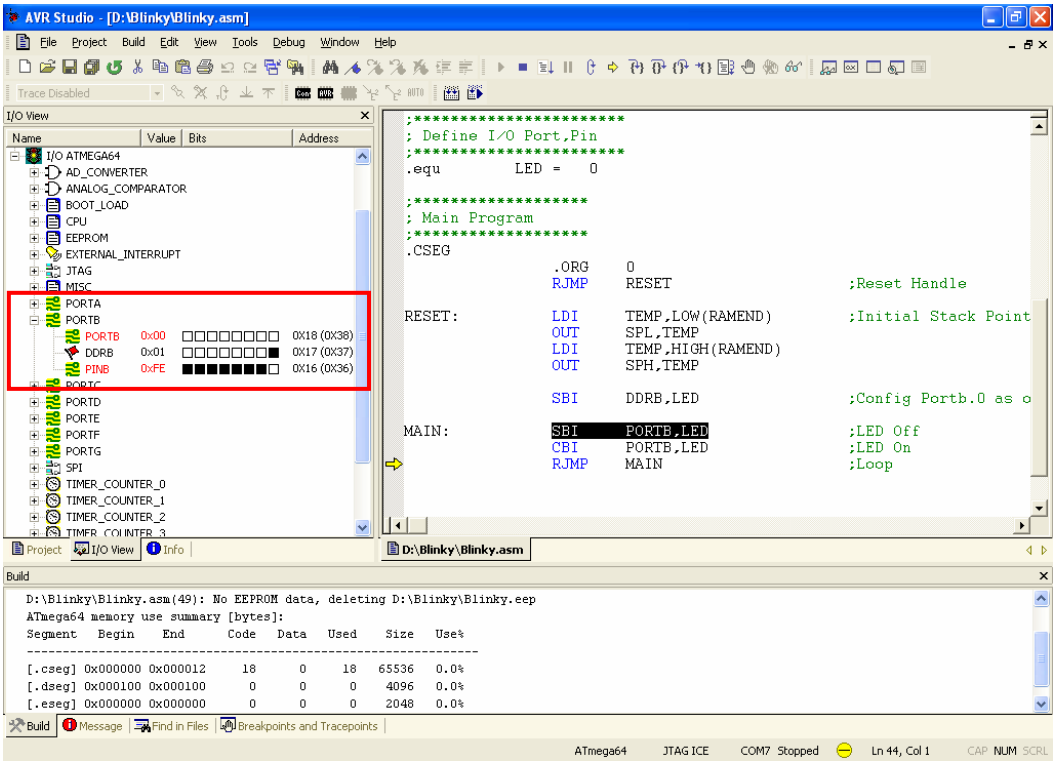
9. เราสามารถที่จะปรับความเร็วในการดีบักได้ โดยการเลือกที่เมนูคำสั่ง Debug → JTAG ICE Options โดยที่สามารถปรับความถี่ของพอร์ต JTAG โดยค่าความถี่นี้ต้องไม่เกิน 1/4 ของความถี่ที่เราใช้งาน



10. ทำการเลือกรูปแบบของการดีบักซึ่งสามารถเลือกได้ทั้งแบบอัตโนมัติ (Auto Step)หรือจะเลือกทีละขั้นตอน โดยเลือกที่แถบเครื่องมือดังรูป ในที่นี้ทดลองเลือกแบบ Step



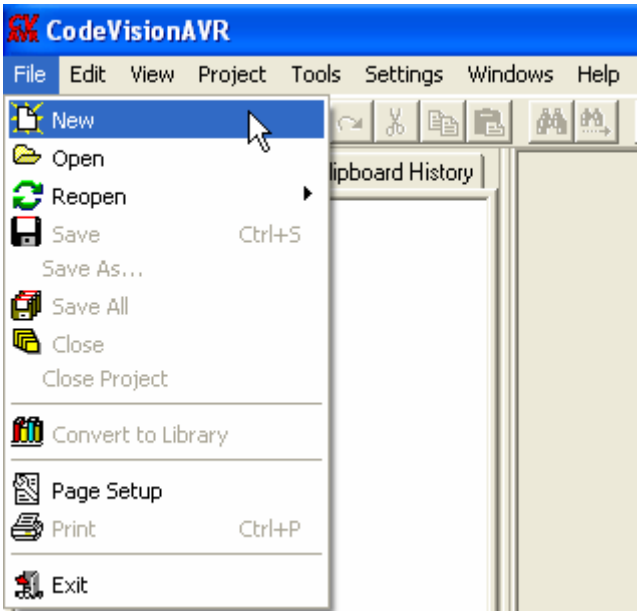
11. จะสังเกตเห็นว่าเมื่อโปรแกรมดีบักผ่านคำสั่ง `SBI PORTB,LED` ซึ่งเป็นคำสั่งให้ PORTB.0 เป็นลอจิก 1 LED ที่อยู่กับ PORTB.0 จะยังไม่สว่างเนื่องจากวงจรของบอร์ดต่อวงจรให้ LED ทำงานที่ลอจิก 0 ทำการกดปุ่ม Step Info อีกครั้งเมื่อผ่านคำสั่ง `CBI PORTB,LED` จะสังเกตว่า LED จะสว่าง ซึ่งค่าต่าง ๆ ในหน้าต่าง I/O View ก็จะเปลี่ยนแปลงตามโปรแกรมด้วย



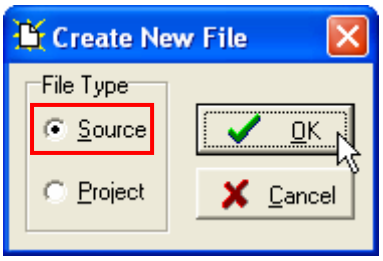
ตัวอย่างการดีบักโดยใช้ภาษาซี

นอกจากภาษาแอสเซมบลีแล้วโปรแกรม AVR Studio ยังสามารถดีบักซอฟต์แวร์ที่เขียนด้วยภาษาซีได้ด้วย ซึ่งตัวอย่างนี้จะเสนอวิธีการดีบักโดยใช้ภาษาซี โดยใช้โปรแกรม CodeVisionAVR C Compiler ร่วมกับโปรแกรม AVRStudio ในการดีบัก

1. เปิดโปรแกรม CodeVisionAVR C Compiler และคลิกเลือกที่เมนูคำสั่ง File → New ดังรูป



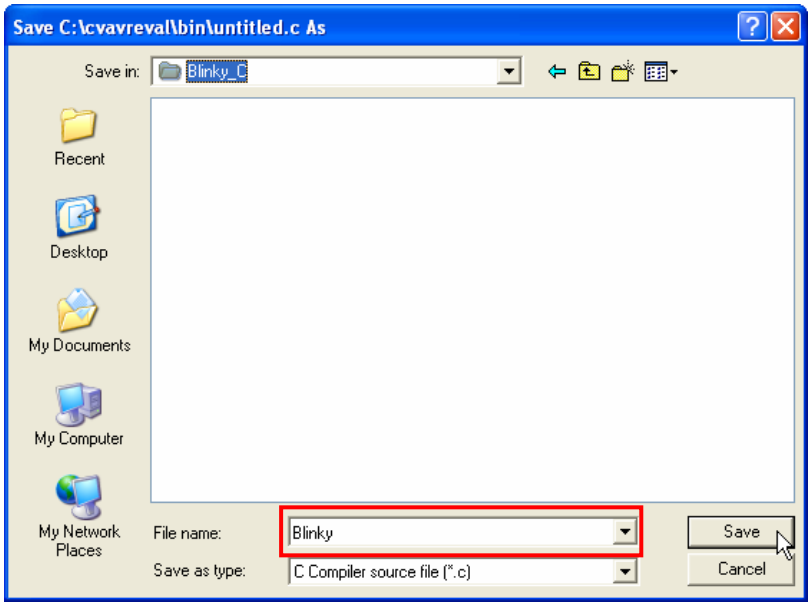
2. เลือก File Type เป็น Source เพื่อสร้างไฟล์ภาษาซีใหม่และคลิกปุ่ม OK ดังรูป



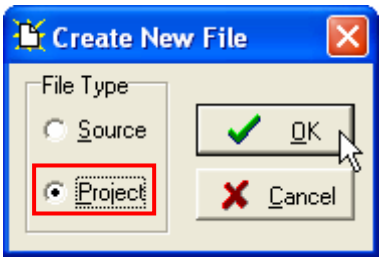
3. จากนั้นจะปรากฏหน้าต่าง Editor ให้ทำการเขียนโปรแกรมดังตัวอย่าง

```
//*************************************************/;
//*Hardware    : ET-AVR STAMP (ATmega64)        */;
//*CPU         : ATMEL-ATmega64                 */;
//* X-TAL       : 16.00 MHz                      */;
//* Filename    : Main.C                         */;
//*Complier    : CodeVisionAVR V1.24.7d         */;
//* Last Update : 9-12-2005(ETT CO.,LTD)         */;
//*            : WWW.ETT.CO.TH                  */;
//*Description : Example LED Blink on Portb.0   */;
//*************************************************/;
//*CodeVisionAVR Complier Option Setting        */;
//* Chip type             : ATmega64             */;
//* Program type          : Application          */;
//*Clock frequency        : 16.000000MHz         */;
//* Memory model          : Small                */;
//*External SRAM size     : 0                    */;
//* Data Stack size       : 1024                 */;
//*************************************************/;

#include <mega64.h>            // ATmega64 MCU
#include <delay.h>             // Delay functions

void main(void)
{
  PORTB=0x00;                  // PB7..0 = 0
  DDRB=0x01;                   // PB0 = Output

 //Loop Blink LED on PB0
  while (1)
  {
    PORTB |= 0x01;             // PB0 = 1 (OFF LED)
    PORTB &= 0XFE;             // PB0 = 0 (ON LED)
  }
}
```

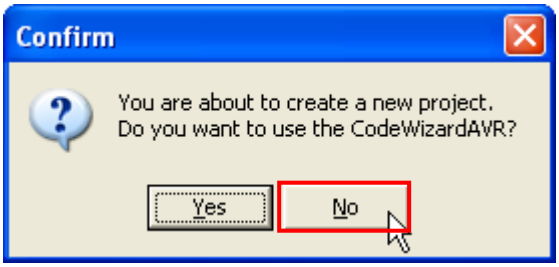
4. ทำการบันทึกโปรแกรมภาษาซีที่เขียนโดยเลือกเมนู File → Save ทำการตั้งชื่อไฟล์และกดปุ่ม Save ดังรูป



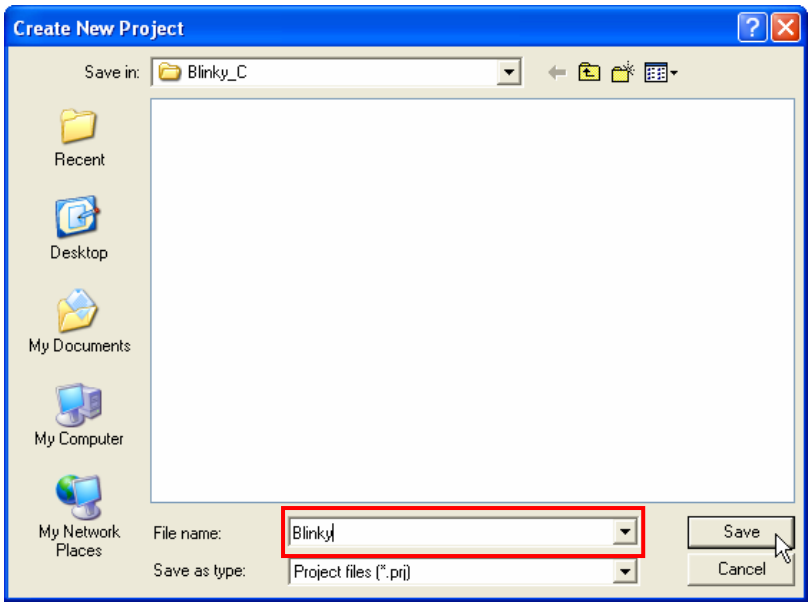
5. เลือกที่เมนู File → New และเลือก File Type เป็น Project เพื่อสร้างโปรเจกต์ใหม่และคลิกปุ่ม OK ดังรูป



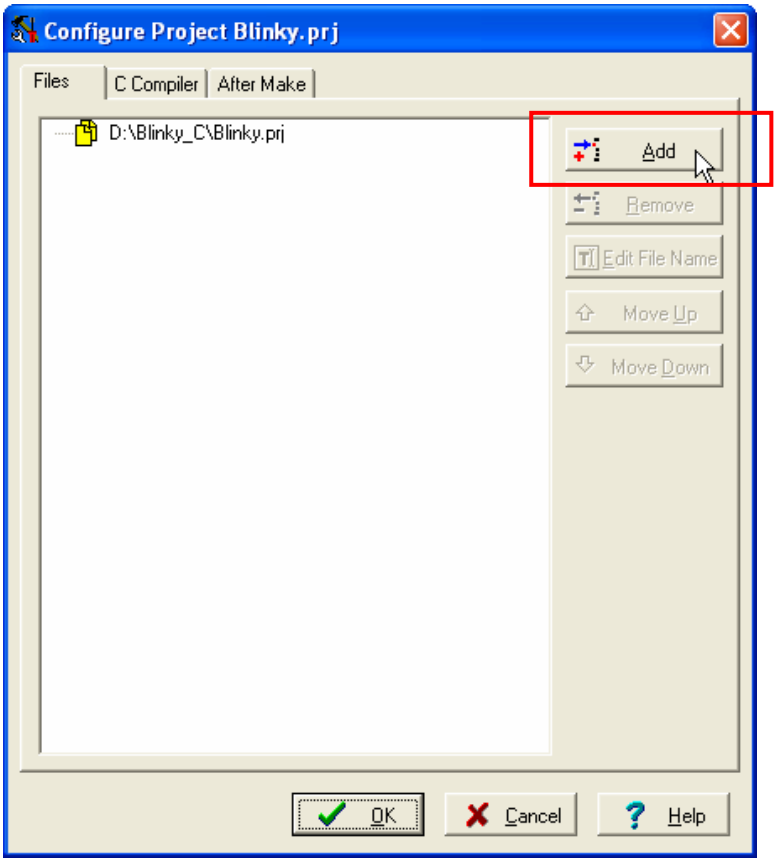
6. คลิกปุ่ม No เพื่อไม่ใช้ตัวช่วยในการสร้างโปรเจกต์ (CodeWizard)

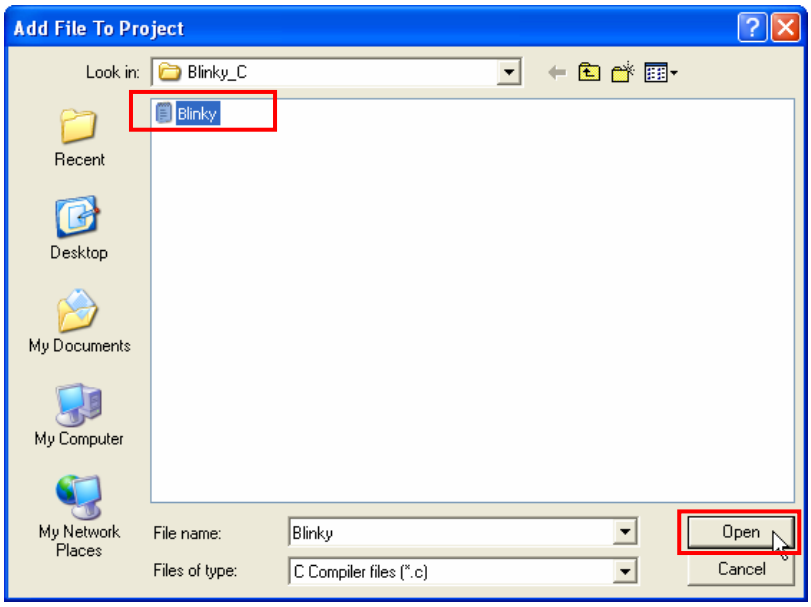


7. ทำการตั้งชื่อโปรเจกต์ตามต้องการและคลิกปุ่ม Save ดังรูป

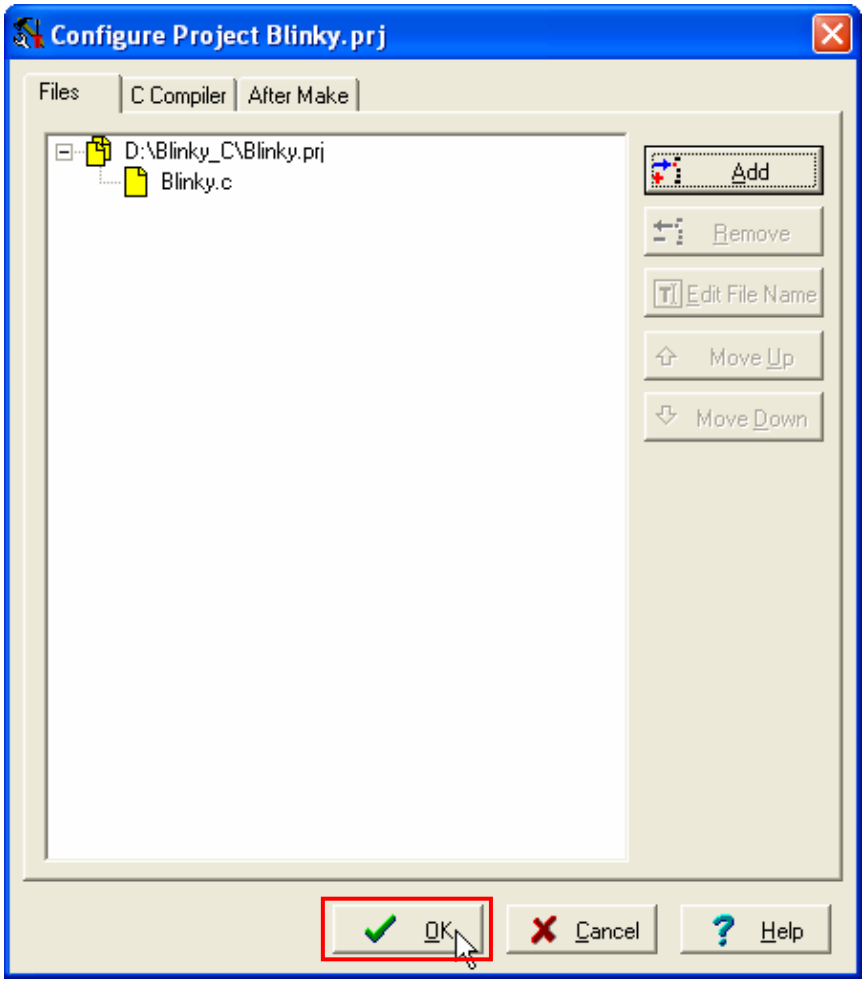


8. ทำการเพิ่มไฟล์ภาษาซีที่เขียนไว้ก่อนหน้านี้เข้ามาในโปรเจกต์โดยการคลิกปุ่ม Add ดังรูป

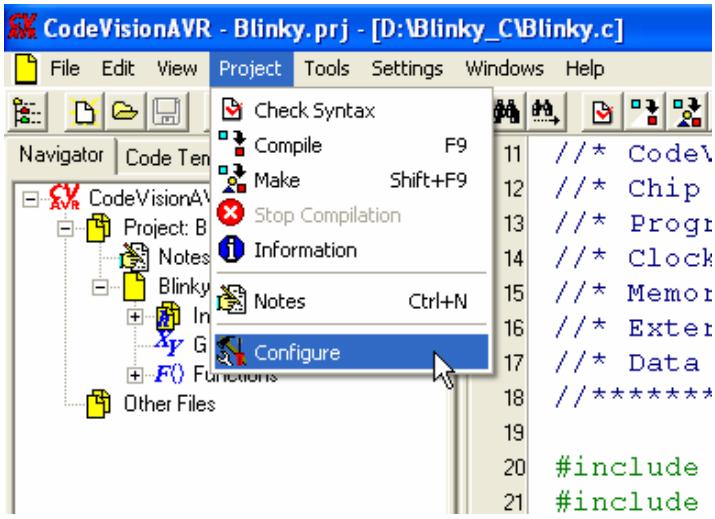


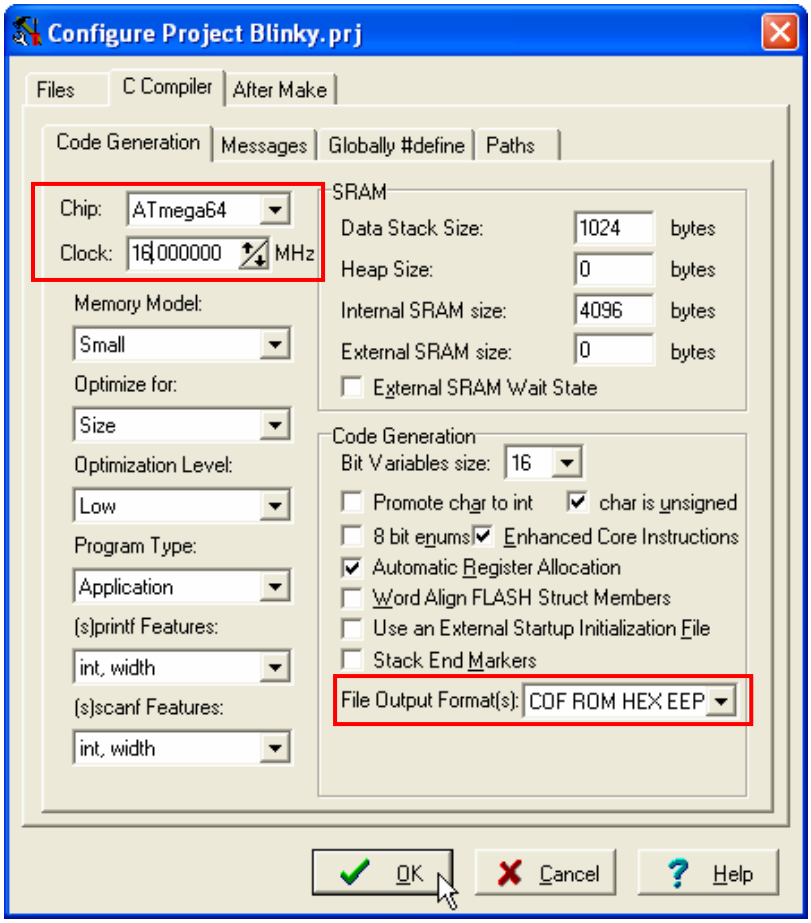


9. เมื่อทุกอย่างเรียบร้อยคลิกปุ่ม OK ดังรูป

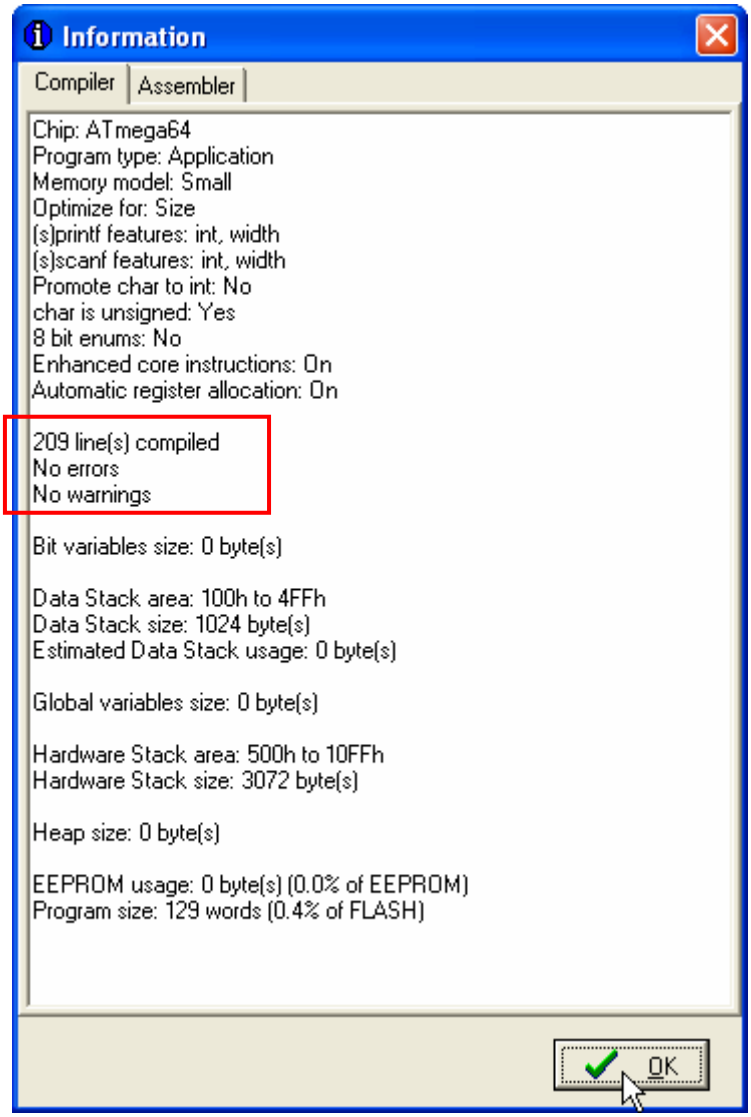


10. ทำการตั้งค่าต่างของโปรเจกต์โดยการคลิกเมาส์ที่เมนูคำสั่ง Project → Configure จากนั้น ทำการกำหนดเบอร์ MCU เป็น ATmega64 ค่าคริสตอลเท่ากับ 16.000000 MHz และ File Output Format(s) เป็น COF ROM HEX EEP

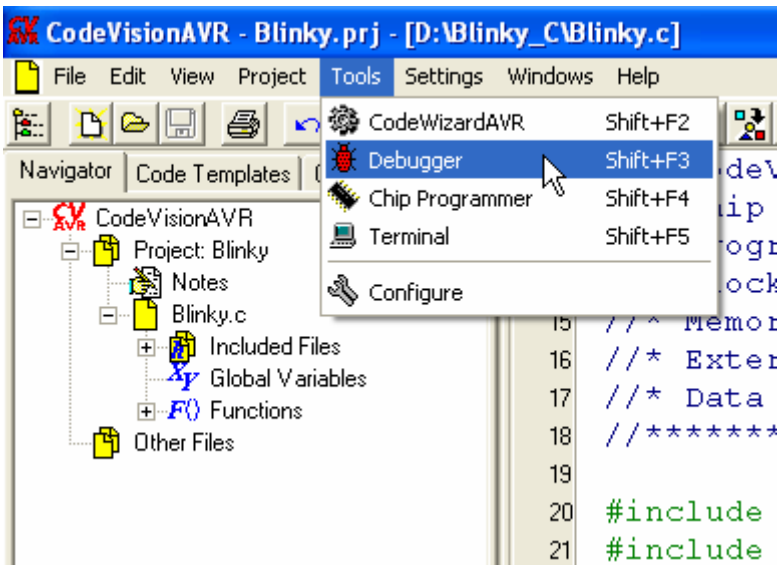


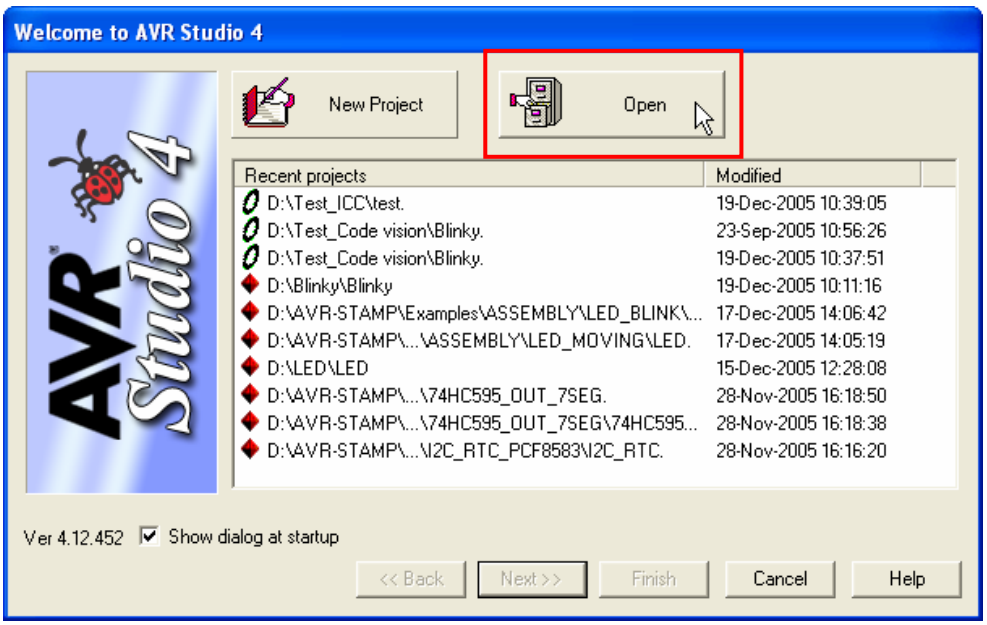


10. ให้ทำการสั่งแปลโปรแกรมที่เราเขียนขึ้น โดยการคลิกเมาส์ที่เมนูคำสั่ง Project → Make ซึ่งหลังจากแปลโปรแกรมแล้วได้ผลถูกต้องและไม่เกิดข้อผิดพลาดใด ๆ จะปรากฏข้อความ No errors, No warnings ดังรูป

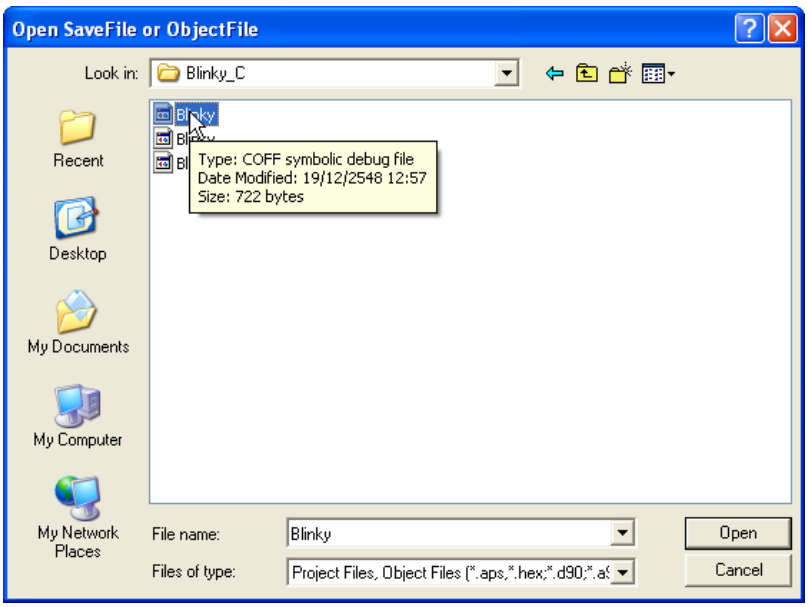


11. คลิกเมาส์ที่เมนูคำสั่ง Tools → Debugger เพื่อเข้าสู่การดีบัก จากนั้นโปรแกรม CodeVisionAVR จะเปิดโปรแกรม AVR Studio ขึ้นมาดังรูป ให้คลิกที่ปุ่ม Open

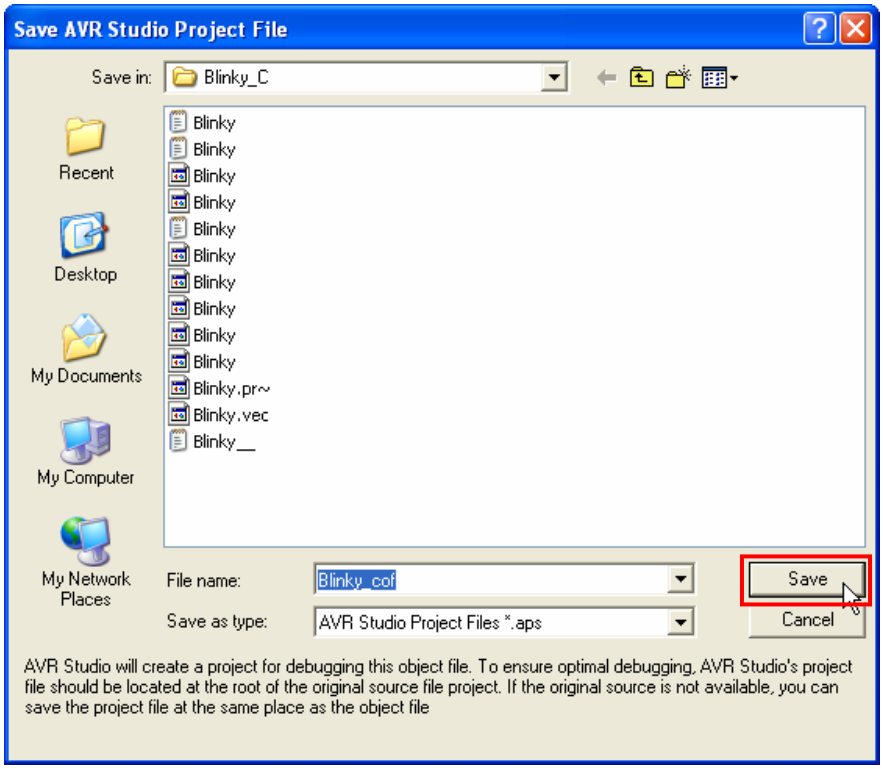




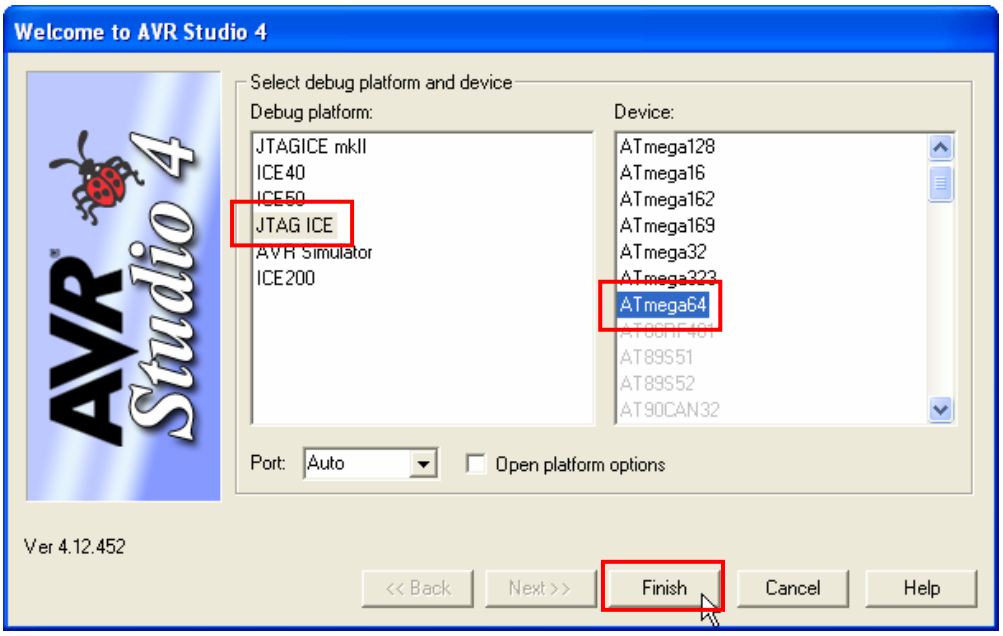
12. เลือกไฟล์ COFF symbolic debug file ที่ได้จากการการสั่งแปลโปรแกรมซึ่งอยู่ในโปรเจกต์ที่ได้สร้างไว้ดังรูป



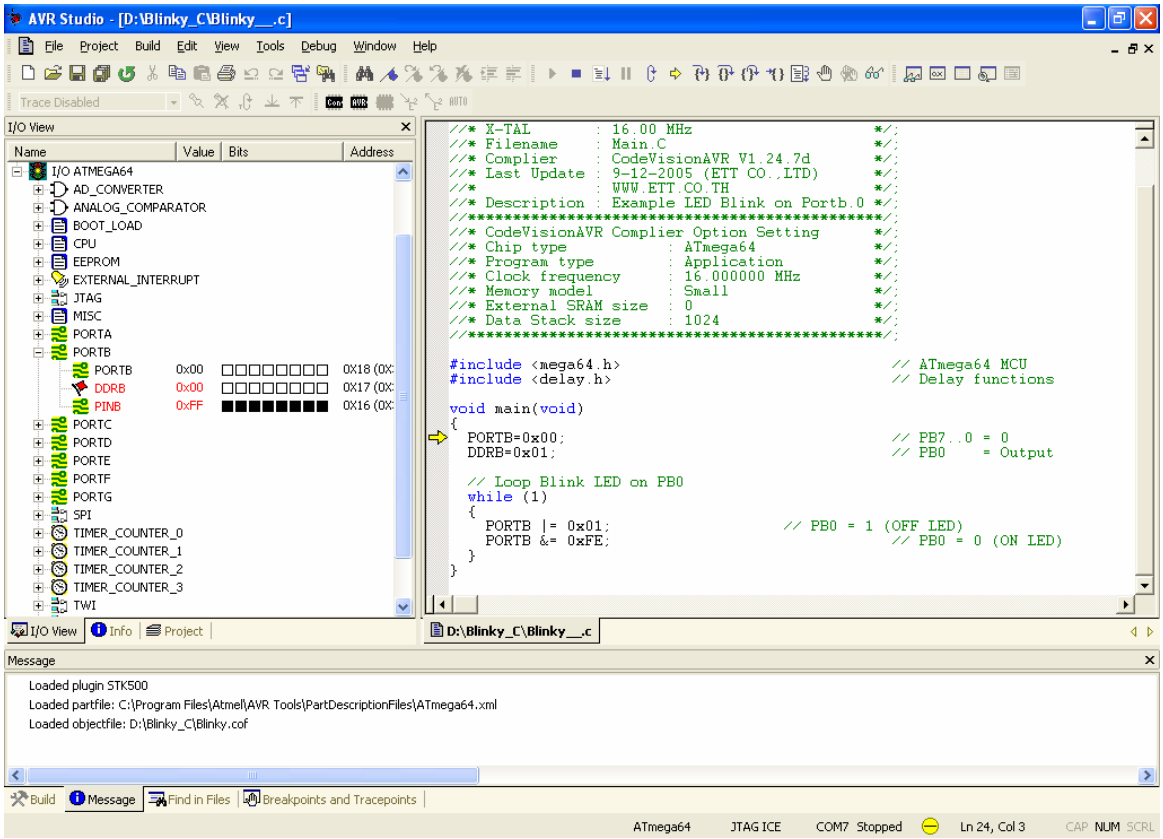
13. จากนั้นโปรแกรมจะให้ทำการบันทึกโปรเจกต์ ให้คลิกปุ่ม Save เพื่อบันทึก



14. เลือก Debug platform เป็น JTAG ICE และ Device เป็น ATmega64 และคลิกปุ่ม Finish ดังรูป



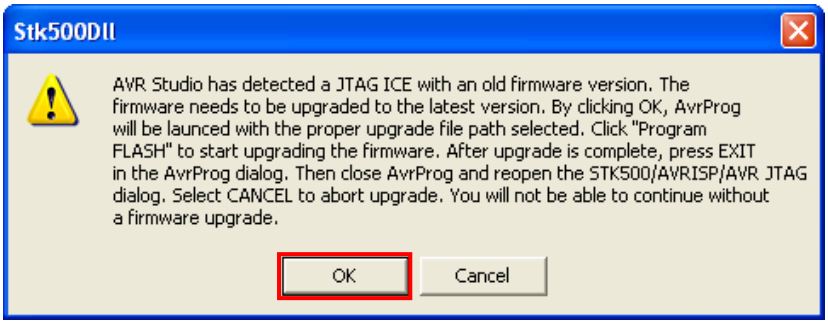
15. เมื่อทุกอย่างเรียบร้อยโปรแกรม AVR Studio จะทำการโหลดโปรแกรมที่เขียนด้วยภาษาซีเข้ามา หลังจากนี้ก็สามารถทำการดีบักค่าต่าง ๆ เหมือนตัวอย่างการดีบักด้วยภาษาแอสเซมบลี



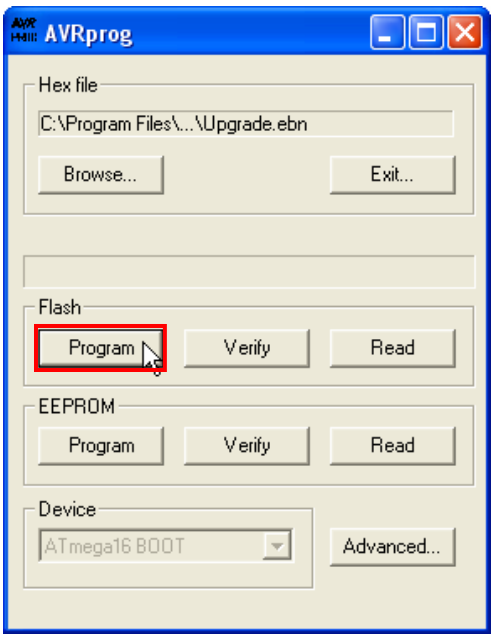
ขั้นตอนการอัพเกรด Firmware ของ ET-JTAG AVR

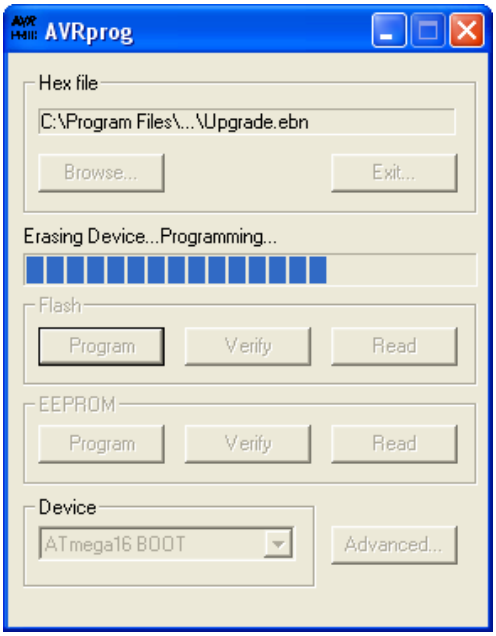
การอัพเกรด Firmware จะทำให้ ET-JTAG AVR สามารถใช้ได้กับ MCU เบอร์ใหม่ ๆ โดย Firmware นี้จะมาพร้อมกับโปรแกรม AVR Studio ซึ่งเมื่อทำการเชื่อมต่อ ET-JTAG AVR เข้าโปรแกรม AVR Studio แล้วถ้ามี Firmware ตัวใหม่จะมีข้อความเตือนให้ทำการอัพเกรด Firmware ซึ่งวิธีการอัพเกรด Firmware มีดังนี้

1. เมื่อทำการติดต่อกับ ET-JTAG AVR และมี Firmware ตัวใหม่จะมีข้อเตือนดังรูป ให้คลิกปุ่ม OK เพื่อทำการอัพเกรด Firmware

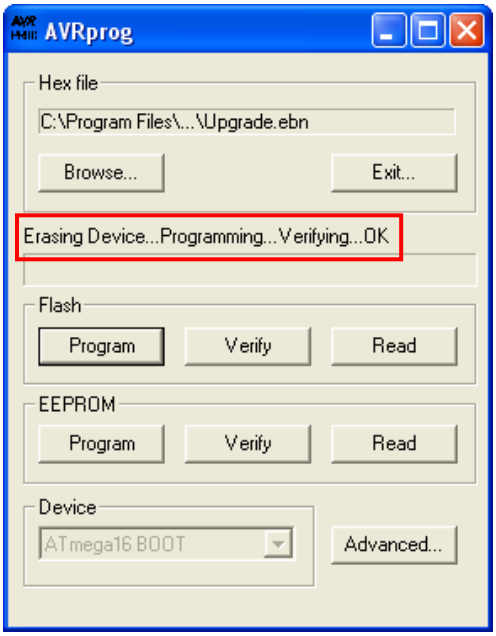


2. จากนั้นโปรแกรมจะทำการเรียกโปรแกรม AVR Prog ขึ้นมาให้ทำการคลิกที่ปุ่ม Browse เพื่อเลือกไฟล์ Upgrade.ebn ซึ่งปกติจะอยู่ที่ไดเร็กทอรี่ C:\Program Files\Atmel\AVR Tools\JTAGICE (โดยปกติโปรแกรมจะเลือกให้อัตโนมัติแล้ว) จากนั้นคลิกที่ปุ่ม Program เพื่อเริ่มทำการอัพเกรดดังรูป



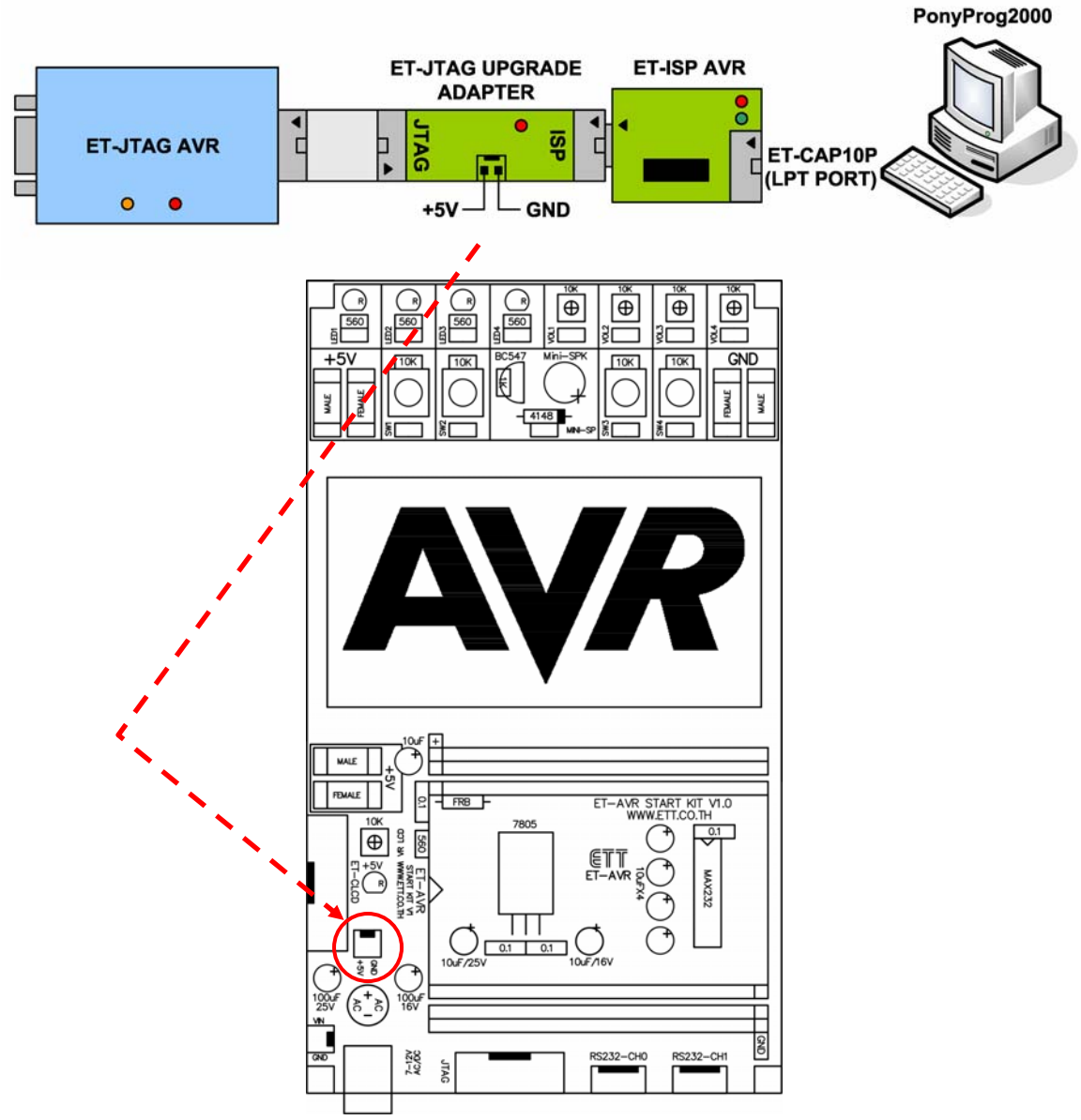


3. ปิดโปรแกรม AVR Prog เมื่อการโปรแกรมเรียบร้อย จากนั้นปลดแหล่งจ่ายไฟที่เลี้ยง ET-JTAG AVR ซึ่งตอนนี้ Firmware ตัวใหม่ก็ได้อัพเกรดเรียบร้อยแล้ว

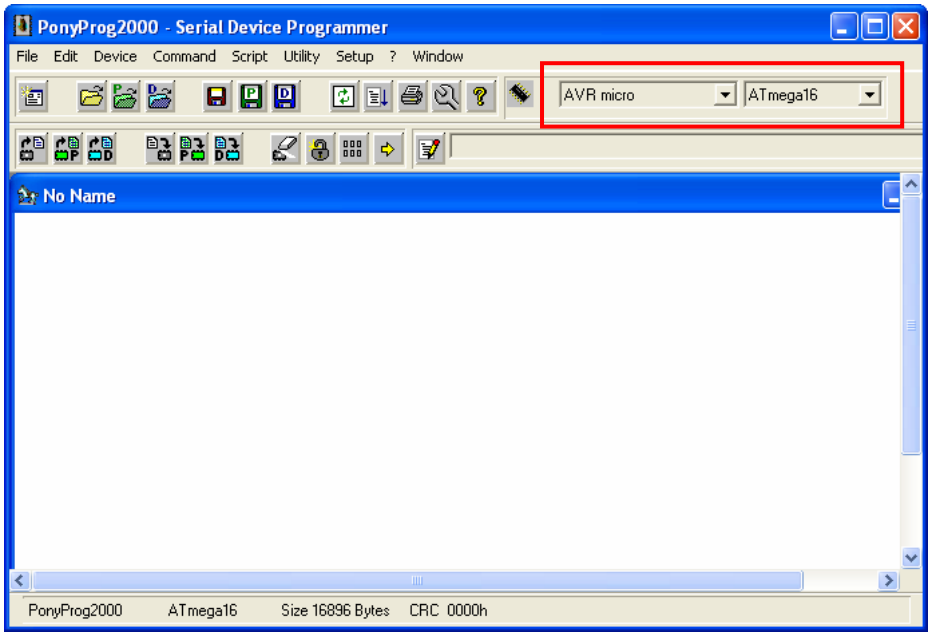


หมายเหตุ ในกรณีที่ไม่สามารถอัพเกรด Firmware แบบอัตโนมัติได้ ซึ่งส่วนมากจะมีปัญหากับชุดแปลงสัญญาณ USB TO SERIAL ก็สามารถอัพเกรดได้อีกวิธีหนึ่ง โดยต้องใช้ร่วมกับ ET-JTAG UPGRADE ADAPTER และ ET-ISP AVR ซึ่งมีวิธีการดังนี้

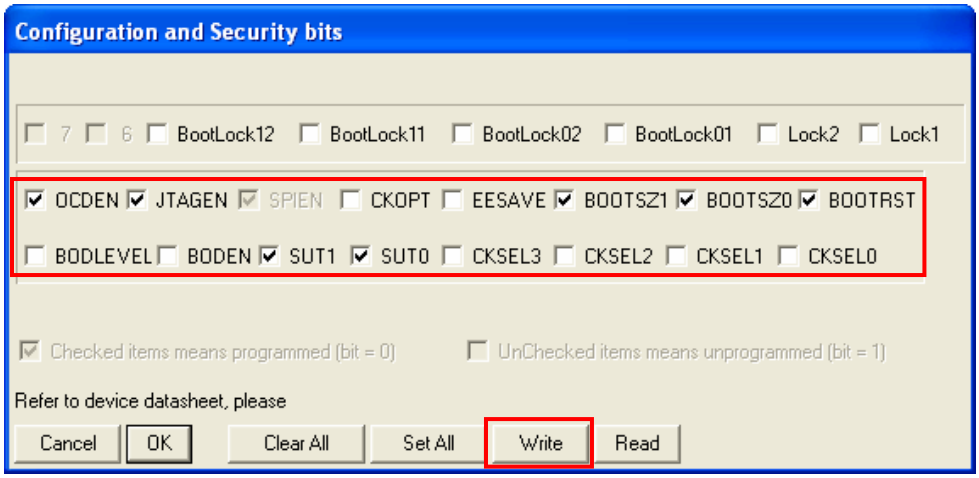
1. ต่อ ET-JTAG AVR เข้ากับ ET-JTAG UPGRADE ADAPTER และ ET-ISP AVR โดยที่ต่อ ET-ISP AVR เข้ากับพอร์ตขนานของคอมพิวเตอร์ โดยใช้ ET-CAP10P ดังรูป ด้วยเหตุที่ ET-JTAG AVR ต้องการไฟเลี้ยงจากภายนอก ดังนั้นจึงต้องรับไฟเลี้ยงผ่านทาง ET-JTAG UPGRADE ADAPTER โดยที่ไฟเลี้ยง 5V นี้สามารถต่อมาจากบอร์ด ET-AVR START KIT V1 เนื่องจาก ได้เตรียมขั้วต่อไว้แล้ว



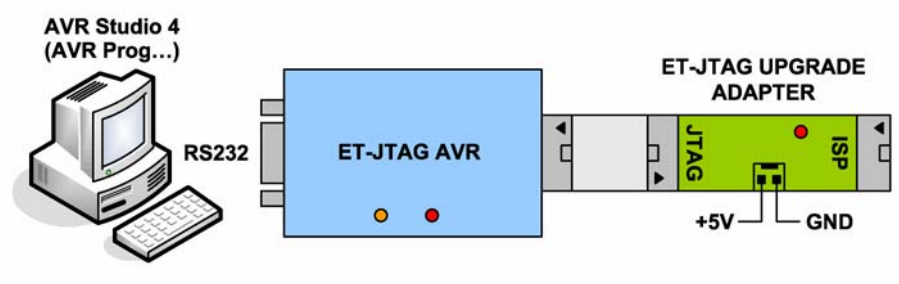
2. เปิดโปรแกรม PonyProg2000 เลือกกำหนดเบอร์ CPU จาก Device → AVR Micro →เป็น ATmega16



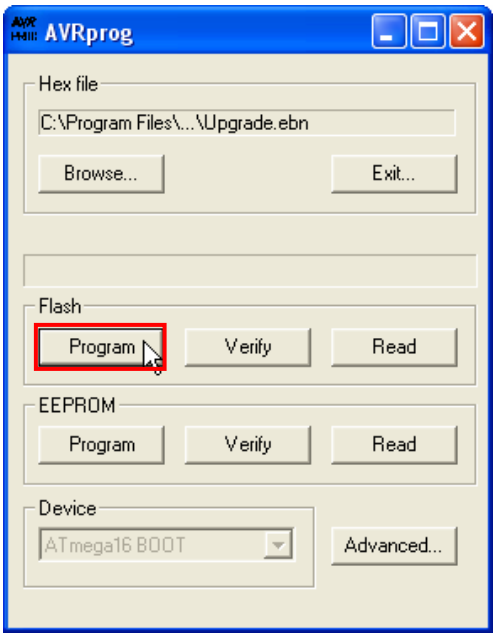
3. เลือกที่เมนู Command → Security and Configuration Bits ทำการเลือก Fuse Bit ดังรูป จากนั้นคลิกปุ่ม Write

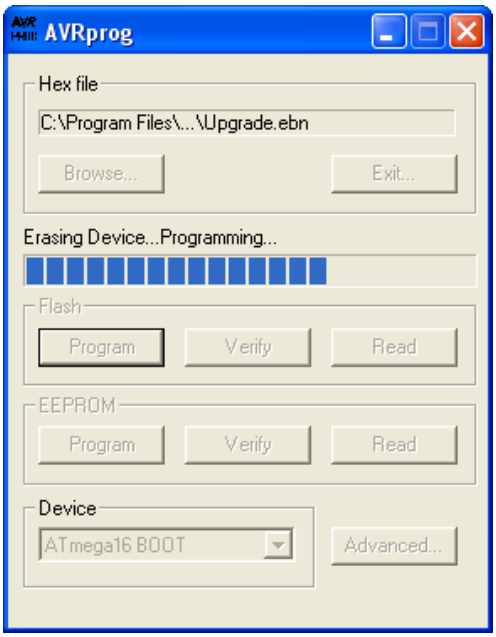


4. ปลดแหล่งจ่ายไฟ 5V ออกจากบอร์ด ET-JTAG UPGRADE ADAPTER และถอด ET-ISP AVR ออกจาก ET-JTAG UPGRADE ADAPTER ทำการเชื่อมต่อ ET-JTAG AVR เข้ากับคอมพิวเตอร์ทางพอร์ต RS232 และก็ต่อไฟเลี้ยง 5V กลับไปตามเดิม ดังรูป

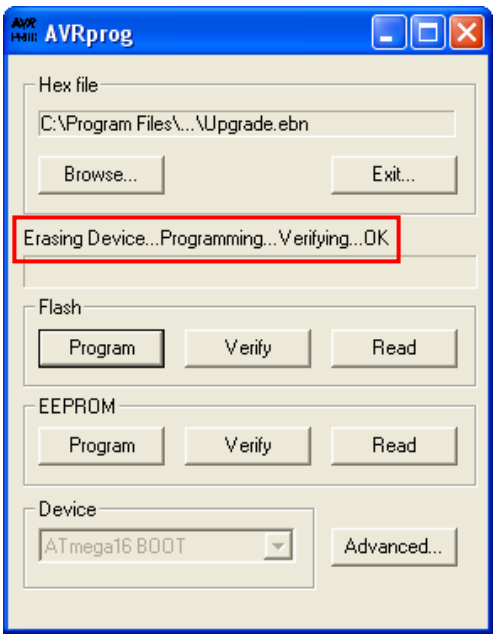


5. เปิดโปรแกรม AVR Studio 4 เลือกที่เมนู Tools → AVR Prog... จากนั้นให้ทำการคลิกที่ปุ่ม Browse เพื่อเลือกไฟล์ Upgrade.ebn ซึ่งปกติจะอยู่ที่ไดเร็กทอรี่ C:\Program Files\Atmel\AVR Tools\JTAGICE (โดยปกติโปรแกรมจะเลือกให้อัตโนมัติแล้ว) จากนั้นคลิกที่ปุ่ม Program เพื่อเริ่มทำการอัพเกรดดังรูป

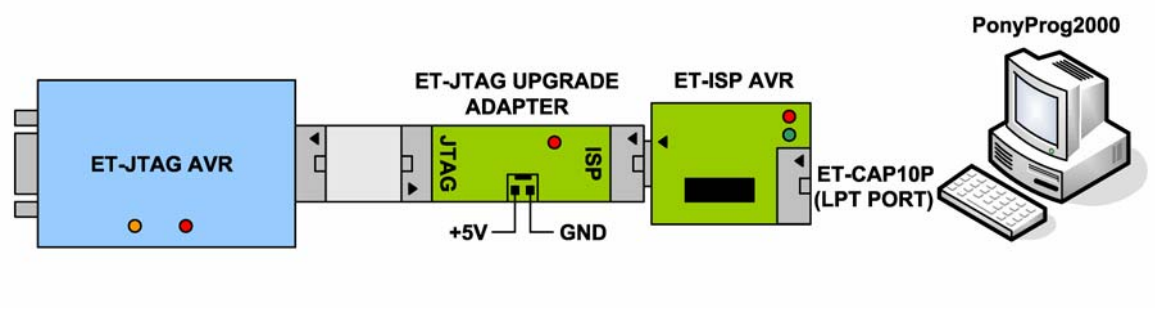




6. ปิดโปรแกรม AVR Prog เมื่อการโปรแกรมเรียบร้อย จากนั้นปลดแหล่งจ่ายไฟ 5V ออกจากบอร์ด ET-JTAG UPGRADE ADAPTER



7. ทำการต่อวงจรเหมือนข้อที่ 1 โดยถอด ET-JTAG AVR ออกจากคอมพิวเตอร์ และต่อไฟเลี้ยง 5V เข้าที่บอร์ด ET-JTAG UPGRADE ADAPTER



8. เปิดโปรแกรม PonyProg2000 และทำการเลือก Fuse Bit ดังรูปจากนั้นคลิกปุ่ม Write ซึ่งตอนนี้ Firmware ตัวใหม่ก็ได้อัพเกรดเรียบร้อยแล้ว