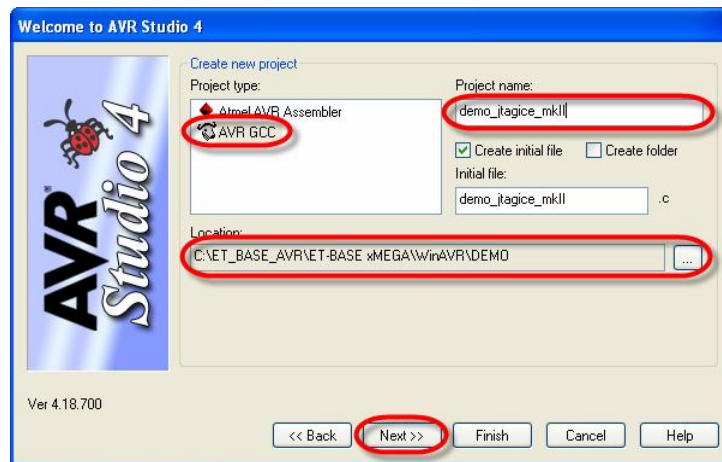
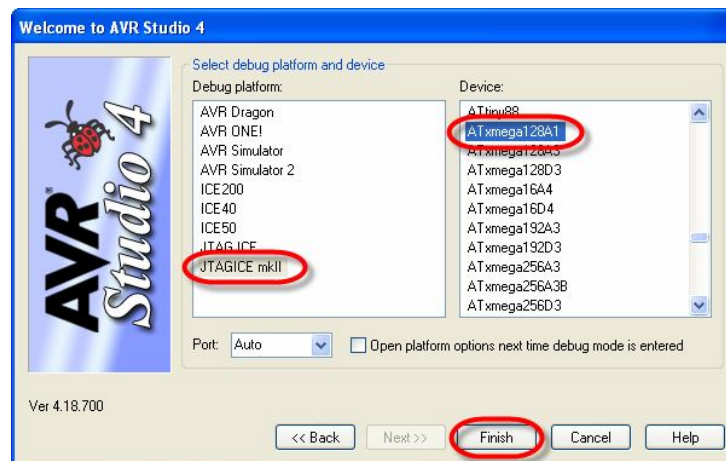


ตัวอย่างการพัฒนาโปรแกรม ด้วย JTAGICE mkII

1. สั่ง Run Program AVR Studio4 แล้วเลือกสร้าง project ใหม่ โดยเลือกที่ project → New project จากนั้นเลือกกำหนดตัวเลือกต่างๆให้กับโปรแกรมโดยให้เลือก Project type เป็น AVR GCC ส่วน Location สำหรับบันทึก project ให้ระบุตำแหน่ง Folder ที่ต้องการใช้บันทึกไฟล์ และ Code ต่างของ project และกำหนดชื่อของ Project name ตามต้องการในที่นี้ให้ลองกำหนดเป็น demo_jtagice_mkII และ ให้เลือก Create initial file ไว้ด้วย ดังนี้



2. เมื่อกำหนดค่าตัวเลือกต่างๆ ให้กับโปรแกรมเรียบร้อยแล้ว ให้เลือกที่ Next แล้วกำหนดค่าใน Debug platform เป็น AVR Dragon และเลือก Device เป็น ATxmega128A1 ซึ่งเมื่อสร้าง project เสร็จโปรแกรมจะสร้างไฟล์ภาษาซีให้ โดยมีชื่อเดียวกับ project ไฟล์ ซึ่งในที่นี้จะเป็นไฟล์ชื่อ demo_jtagice_mkII.c ให้เองโดยอัตโนมัติ เพียงแต่ไฟล์ดังกล่าวจะยังไม่มี code ใดๆ บรรจไว้ให้ เป็นเพียงหน้ากระดาษเปล่าๆ ซึ่งต้องรอให้เราเขียน code เพิ่มเข้าไปเอง ดังรูป



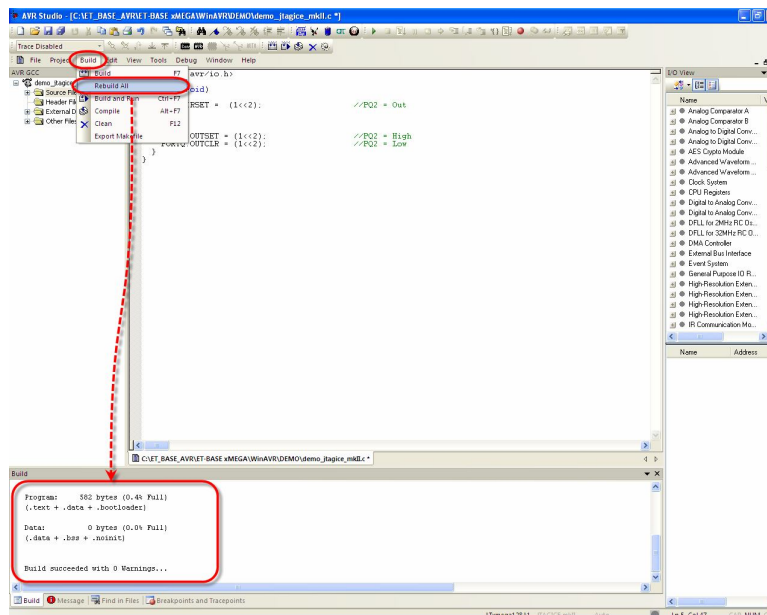
3. ให้พิมพ์คำสั่งของโปรแกรมสำหรับทดสอบการทำงาน ในหน้าต่าง Text Editor ของโปรแกรม โดยในที่นี้จะทดสอบด้วย Code โปรแกรม สำหรับทำหน้าที่ทดสอบการทำงานของบอร์ดในเบื้องต้น โดยทำหน้าที่ ON/OFF LED ซึ่งต่อควบคุมจากขา PQ2 โดยโปรแกรมตัวอย่างจะยังไม่มีส่วนของการหน่วงเวลาเข้ามาเกี่ยวข้อง ซึ่งเมื่อต้องการ Run ด้วยความเร็วจริงต้องเพิ่มคำสั่งเกี่ยวกับการหน่วงเวลาแทรกเข้าไปในภายหลังด้วย ดังตัวอย่าง

```
#include <avr/io.h>

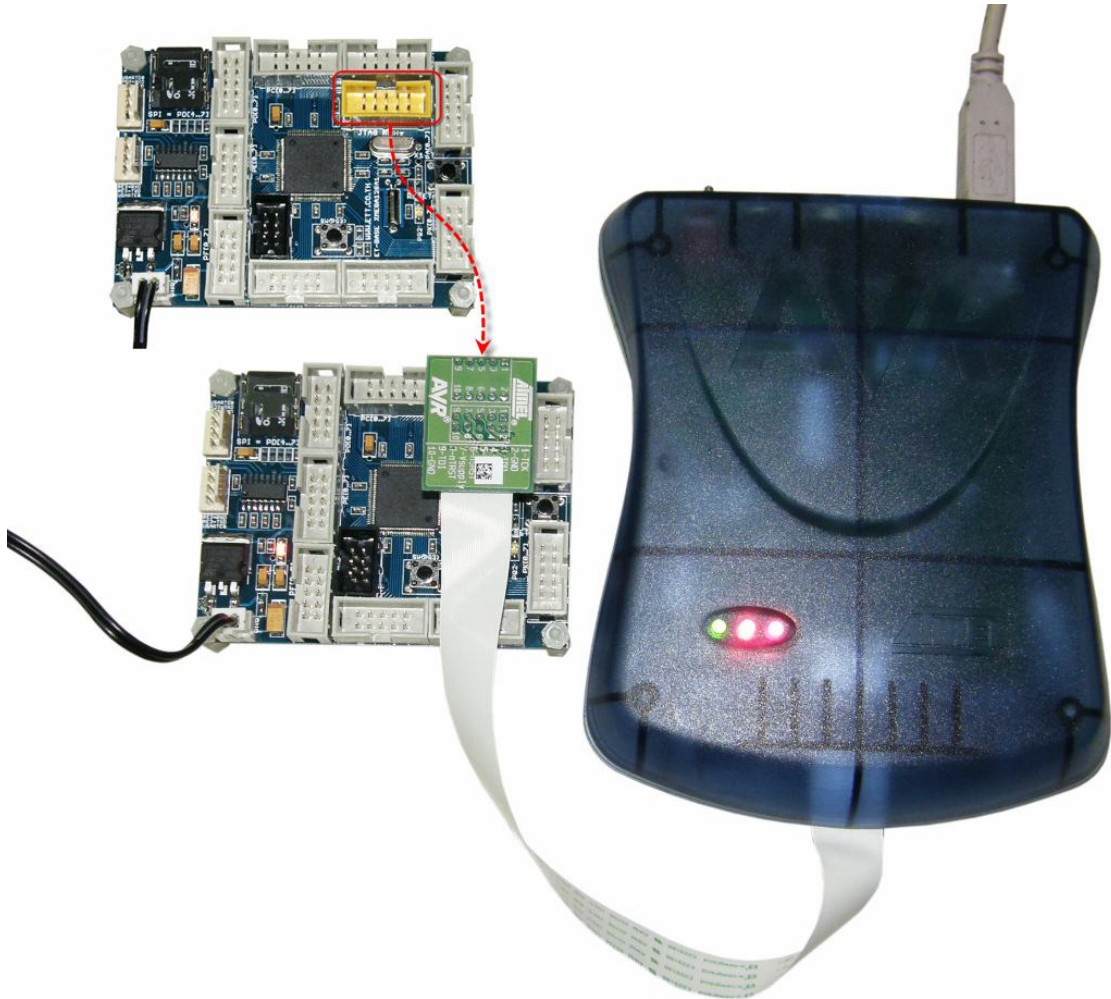
int main(void)
{
    PORTQ.DIRSET = (1<<2);           //PQ2 = Out

    while(1)
    {
        PORTQ.OUTSET = (1<<2);       //PQ2 = High
        PORTQ.OUTCLR = (1<<2);       //PQ2 = Low
    }
}
```

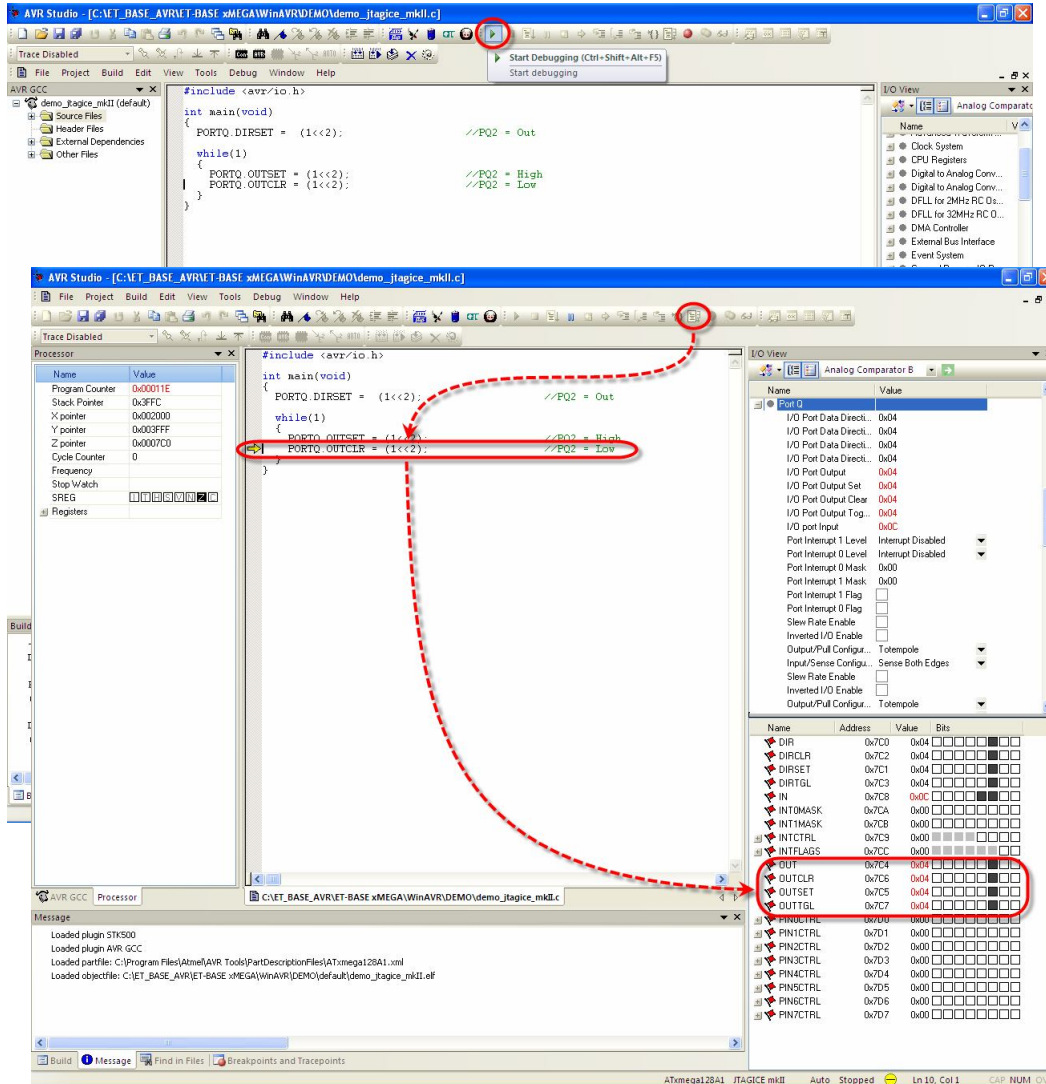
4. หลังจากพิมพ์ Code โปรแกรมเสร็จแล้วให้สั่งแปลโปรแกรม โดยเลือกที่ build → rebuild all ซึ่งถ้าทุกอย่างถูกต้อง ผลการแปลคำสั่งจะได้ผลลัพธ์เป็น “Build succeeded with 0 Warnings...” และจะรายงานผลการแปลพร้อมขนาดหน่วยความจำที่ใช้ไป และจะได้ Output เป็น HEX File ที่มีชื่อเดียวกันกับ project ที่สร้างไว้ โดยจะบรรจุอยู่ใน Directory ย่อยชื่อ default ดังรูป



5. ต่อสาย USB ให้กับ JTAGICE mkII พร้อมกับต่อสายสัญญาณ JTAG จาก JTAGICE mkII เข้ากับหัว JTAG ของบอร์ด ET-BASE xMEGA128A1 ดังรูป



6. สั่ง Start Debug ซึ่งโปรแกรมจะเปลี่ยนเข้าไปทำงานใน Debug Mode ทันที จากนั้นให้ลองทดสอบ สั่งงาน Debug ต่างๆดู เช่น เลือกทดสอบแบบ Auto Step แล้วดูผลการทำงานซึ่งจะเห็นมีลูกศรสีเหลืองชี้หน้าบรรทัดคำสั่ง ที่โปรแกรมกำลังทำงานอยู่ในขณะนั้น พร้อมกับเห็นการเปลี่ยนแปลงค่ารีจิสเตอร์ต่างๆ และที่บอร์ดเองก็จะเป็น LED กระพริบตามการทำงานของโปรแกรมด้วยเช่นเดียวกัน ซึ่งจะเห็นว่าการทำงานต่างๆของโปรแกรมเหมือนจริงทุกประการ ยกเว้นเรื่องความเร็ว



ซึ่งผู้ใช้สามารถทดสอบการทำงานของโปรแกรมผ่านความสามารถในการ Debug ได้มากมายหลายแบบ ไม่ว่าจะเป็นการตั้ง Break เพื่อหยุดการทำงานของโปรแกรมที่สงสัยเพื่อตรวจสอบผลการทำงานต่างๆ ณ. จุดนั้นว่า ถูกต้องตามที่ออกแบบโปรแกรมไว้หรือไม่ ซึ่งเทคนิควิธีต่างๆเกี่ยวกับการใช้งาน Debug ของ JTAGICE mkII สามารถศึกษาเพิ่มเติมได้จากคู่มือการใช้งานของ JTAGICE mkII เอง

นอกจากจะใช้ JTAGICE mkII ในการทำงานแบบ Debug เพื่อทดสอบการทำงานหรือตรวจสอบความผิดพลาดของโปรแกรมแล้ว เรายังสามารถใช้ JTAGICE mkII ทำการโปรแกรม Hex ให้กับ MCU ภายในบอร์ดเพื่อนำไปใช้งานจริงได้ด้วย โดยโปรแกรมที่จะใช้ Run จริง จะต้องแก้ไขเพิ่มเติมให้สมบูรณ์ครบถ้วน เช่นจากโปรแกรมตัวอย่างการทดลองที่ผ่านมาก็จะต้องมีการเพิ่มส่วนของโปรแกรมหน่วงเวลาเข้าไปด้วย ไม่เช่นนั้นเราจะมองไม่เห็นการติด ดับ ของ LED เนื่องจากในขณะที่ทำงานจริงๆ MCU จะทำงานเร็วมากดังนั้นจึงต้องมีการเพิ่มเติมส่วนของโปรแกรมหน่วงเวลาเข้าไปด้วยดังตัวอย่าง

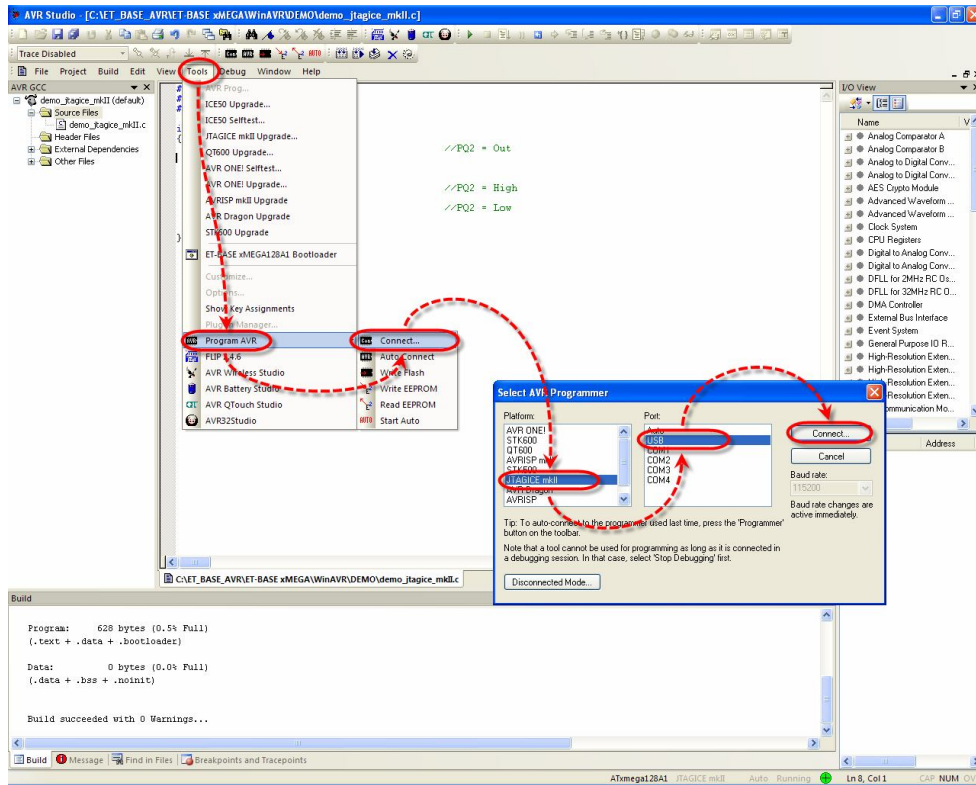
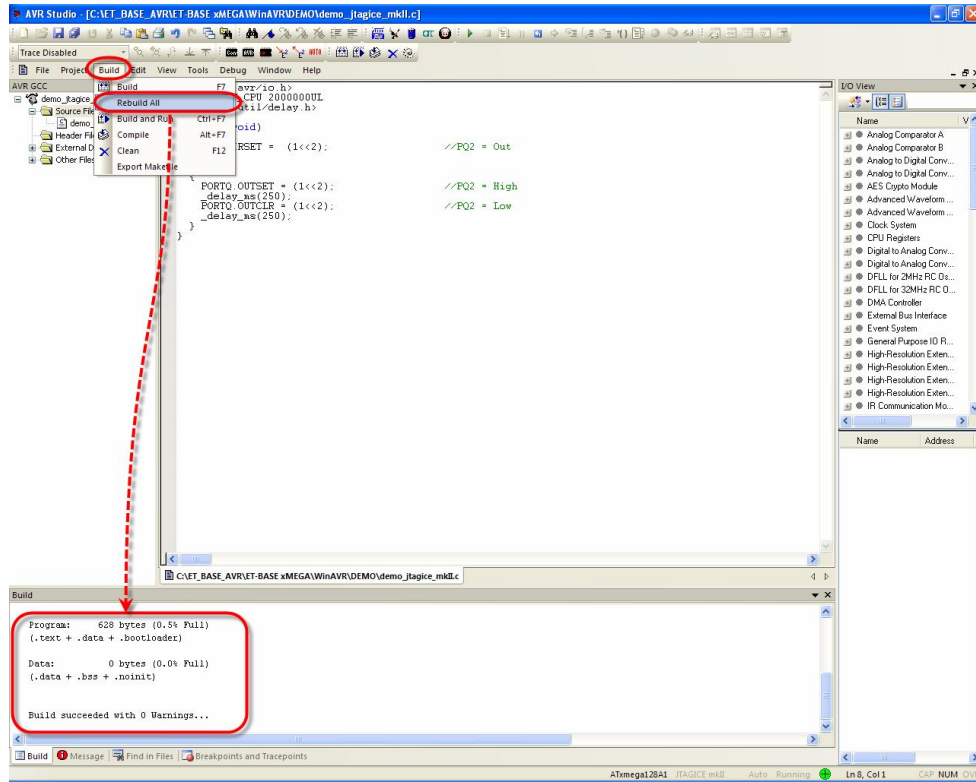
```
#include <avr/io.h>
#define F_CPU 2000000UL
#include <util/delay.h>

int main(void)
{
    PORTQ.DIRSET = (1<<2); //PQ2 = Out

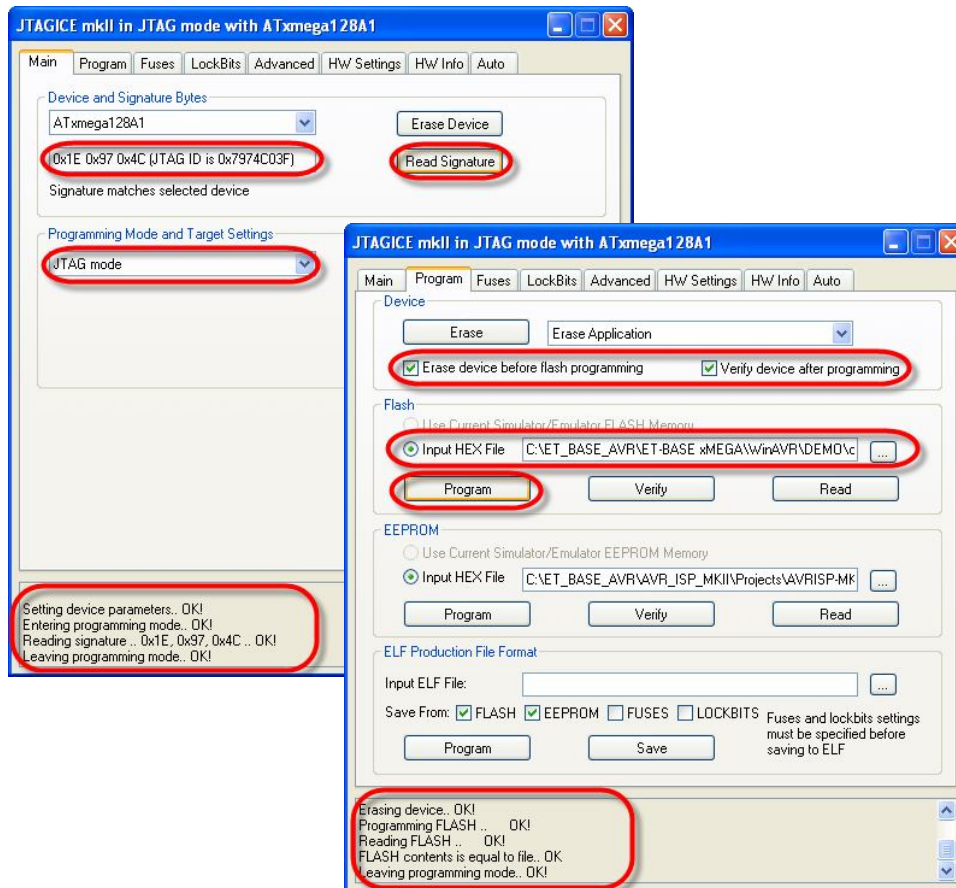
    while(1)
    {
        PORTQ.OUTSET = (1<<2); //PQ2 = High
        _delay_ms(250);
        PORTQ.OUTCLR = (1<<2); //PQ2 = Low
        _delay_ms(250);
    }
}
```

เมื่อทำการแก้ไขโปรแกรมเรียบร้อยแล้วให้ สั่งแปลโปรแกรมซ้ำใหม่อีกครั้ง โดยเลือกที่เมนู คำสั่ง **Build** → **Rebuild ALL**

จากนั้นให้เลือกที่เมนู **Tools** → **Program AVR** → **Connect** และในส่วนช่อง **Select AVR Programmer** ให้เลือก **JTAGICE mkII** → **USB** → **Connect** ซึ่ง JTAGICE mkII สามารถที่จะเชื่อมต่อกับบอร์ด ET-BASE xMEGA128A1 ในโหมด **Programming** ได้ 2 แบบ คือ เชื่อมต่อผ่านทาง ขั้ว **JTAG mode** ดังวิธีการข้างต้นที่ผ่านมา หรือ เชื่อมต่อผ่านทาง **PDI mode** ซึ่งทั้ง 2 แบบสามารถทำงานได้เหมือนกัน แต่ในกรณีที่จะใช้ **PDI mode** จะต้องใช้สายแพร์ **Convert** เพื่อเชื่อมต่อสัญญาณ ด้วย เนื่องจาก JTAGICE mkII ไม่ได้ทำขั้วแบบ **PDI 6 Pin** ไว้ให้โดยตรง โดยใช้สายสีแดง(9) เป็น **PDI Data**, สายสีเขียว(6)เป็น **PDI Clock(RESET)**, สายสีม่วง(4)เป็น **+VDD** และ สายสีขาว(2)เป็น **GND** แต่เพื่อความสะดวกในการใช้งานแนะนำให้ใช้การเชื่อมต่อแบบ **JTAG mode** จะดีกว่า



เมื่อเข้าสู่หน้าต่างของ JTAGICE mkII ใน Tab ของ Main ให้เลือกกำหนดเบอร์ของ MCU เป็น ATxmega128A1 แล้วเลือก Programming Mode and Target Settings เป็น JTAG mode ถ้าเชื่อมต่อแบบ JTAG หรือ เลือกเป็น PDI ถ้าเชื่อมต่อแบบ PDI จากนั้นให้ทดสอบ Read Signature ของ MCU ดูว่าสามารถอ่านได้ถูกต้องหรือไม่ เมื่อเรียบร้อยแล้วให้เปลี่ยนไปที่ Tab ของ Program แล้วเลือก Erase device before flash programming และ Verify device after programming แล้วเลือก กำหนด Hex File ที่ได้จากการแปลคำสั่งในโปรแกรม แล้วเลือก Program ดังรูป



ซึ่งในกรณีที่ใช้โปรแกรม AVR Studio Version 4.18 build 700 ร่วมกับ JTAGICE mkII ในโหมด programming นี้พบว่ายังมีข้อผิดพลาดอยู่ คือ เมื่อสั่ง Program เสร็จเรียบร้อยแล้ว MCU จะยังคงค้างการทำงานอยู่ใน JTAG Mode อยู่ ถึงแม้ว่าจะกดรีเซ็ตแล้วก็ตาม MCU ก็จะไม่ทำงาน ซึ่งต้องแก้ปัญหาโดยการสาย JTAG ออกจากขั้ว JTAG ของบอร์ดแล้วทำการ Power-Up ใหม่ก่อนด้วย 1 ครั้ง MCU จึงจะสามารถเริ่มทำงานตามโปรแกรมที่ได้สั่งโปรแกรมไว้แล้ว