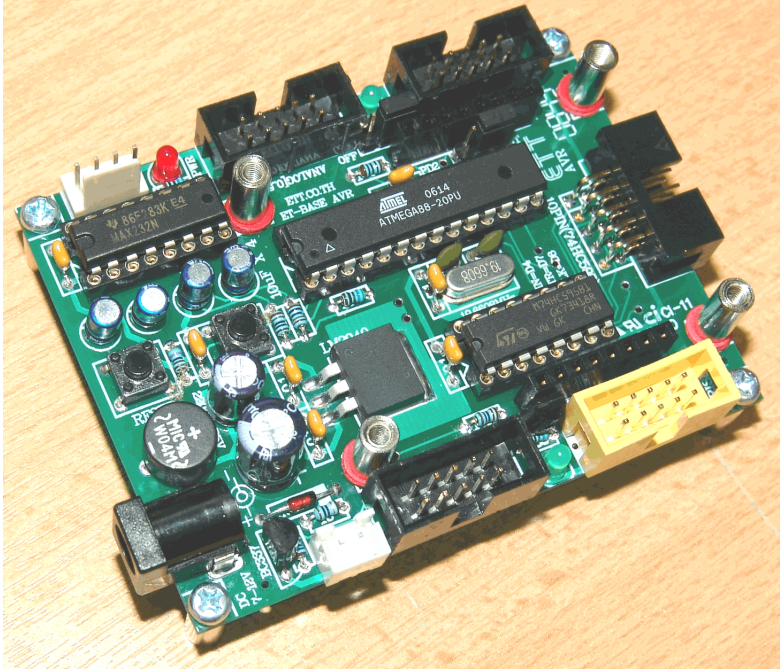
ET-BASE AVR EASY88



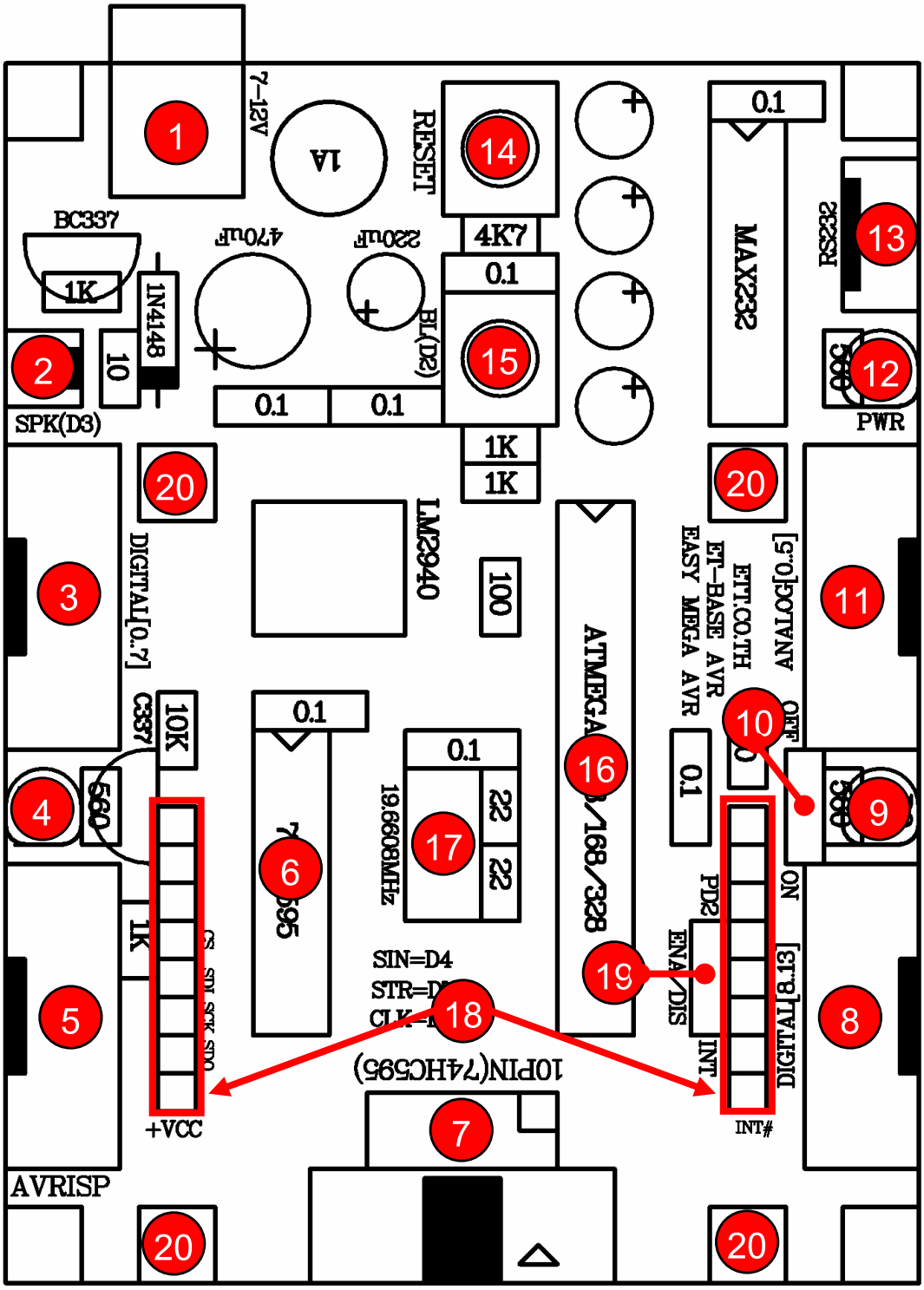
ET-BASE AVR EASY88 เป็นบอร์ดไมโครคอนโทรเลอร์ในตระกูล AVR โดยบอร์ดเลือกใช้ไมโครคอนโทรเลอร์ ตระกูล AVR เบอร์ ATMEGA88 ของ ATMEL เป็น MCU ประจำบอร์ด โดย MCU รุ่นนี้จะบรรจุอยู่ภายในตัวถังแบบ 28 Pin DIP โดย MCU ตัวนี้จะมีจุดเด่น คือ เป็นไมโครคอนโทรลเลอร์ขนาดเล็กแต่เพียบพร้อมไปด้วยทรัพยากรพื้นฐานต่างๆอย่างครบถ้วน เหมาะแก่การใช้ในดารศึกษาเรียนรู้สำหรับผู้เริ่มต้น และยังสามารถนำไปประยุกต์ใช้งานต่างๆได้โดยง่าย ซึ่ง MCU สามารถทำงานได้ด้วยความถี่สูงสุด 20MHz ที่ 1 Clock / Machine Cycle นอกจากนี้แล้วยังมีความเพียบพร้อมด้วยอุปกรณ์พื้นฐานต่างๆที่จำเป็นต่อการใช้งาน ไม่ว่าจะเป็นหน่วยความจำสำหรับเก็บข้อมูลแบบ EEPROM ขนาด 512 Byte และหน่วยความจำใช้งานแบบ SRAM อีก 1 K Byte ส่วนในด้านของอุปกรณ์ Peripheral นั้นก็นับว่าครบถ้วนเหมาะแก่การนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมและประมวลผลต่างๆได้เป็นอย่างดี โดยจะมีทั้งระบบฮาร์ดแวร์ของ SPI, UART, I2C, Watchdog, Timer/Counter, PWM และ ADC ฯลฯ

โดยการออกแบบโครงสร้างของบอร์ดนั้น จะเน้นเรื่องขนาดของบอร์ดให้มีขนาดเล็กเพื่อให้ง่ายต่อการนำไปประยุกต์ใช้งาน และสะดวกต่อการพัฒนาโปรแกรม โดยลักษณะของบอร์ดจะเน้นความคุ้มค่าและมีความอ่อนตัว ทั้งด้านของการศึกษาทดลอง และ การนำไปประยุกต์ใช้งานจริงๆ โดยในด้านของการศึกษาทดลองนั้น สามารถเลือกซื้อบอร์ดทดลอง Input / Output ขนาดเล็กของ ET-MINI I/O แบบต่างๆ ที่ทาง อีทีที ออกแบบและผลิตขึ้นมาสนับสนุนเพื่อเป็นทางเลือกให้ผู้ใช้ได้เลือกชุดอุปกรณ์ที่ตนเองสนใจศึกษาทดลอง เพื่อนำมาติดตั้งใช้งานร่วมกับบอร์ดของ ET-BASE AVR EASY88 ได้อย่างง่ายดาย สำหรับกรณีที่จะนำบอร์ดไปเขียนโปรแกรมเพื่อสร้างเป็นชิ้นงานจริงๆนั้น ก็สามารถนำไปดัดแปลง หรือ เชื่อมต่อกับอุปกรณ์ต่างๆได้โดยง่ายตามความเหมาะสม ซึ่งเรียกได้ว่า บอร์ดเดียวใช้ได้ทั้งเรียนรู้และใช้งาน

คุณสมบัติของบอร์ด

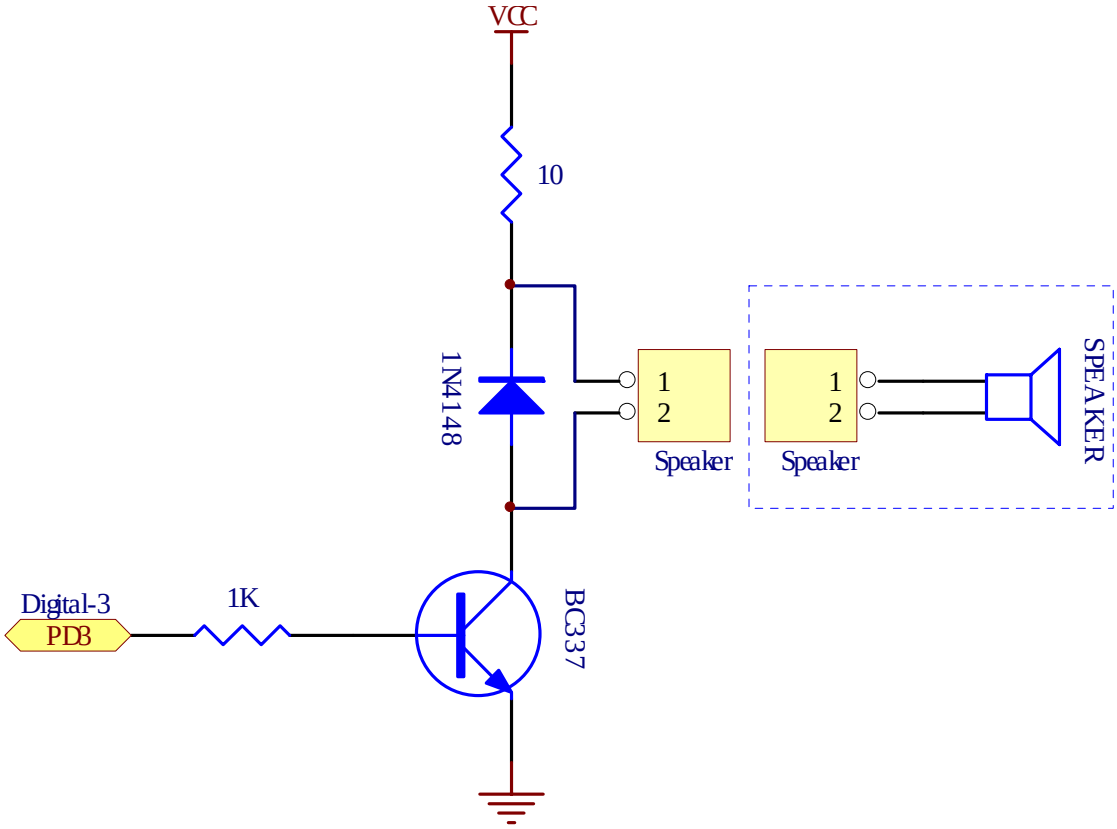
- เลือกใช้ MCU ตระกูล AVR เบอร์ ATMEGA88 ของ ATMEL เป็น MCU ประจำบอร์ด โดยเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาแบบ Crystal Oscillator ค่า 19.6608 MHz เพื่อให้สามารถใช้งานพอร์ตสื่อสารอนุกรม RS232 ได้อย่างลงตัว
- สามารถเปลี่ยนไปติดตั้งใช้งาน MCU ตระกูล AVR ขนาด 28 PIN เบอร์อื่นๆในอนุกรมเดียวกันได้ เช่น ATMEGA8,ATMEGA48,ATMEGA168 และ ATMEGA328 เป็นต้น โดย MCU เหล่านี้จะมีตำแหน่งขาที่เข้ากันได้ สามารถติดตั้งใช้งานในบอร์ดได้ทันที โดยไม่ต้องดัดแปลงแก้ไขวงจร
- มีหน่วยความจำ 8 KByte Flash / 512 Byte EEPROM / 1024 Byte SRAM
- มีพอร์ต I/O ขนาด 20 บิต จำนวน 3 พอร์ต (PB(6บิต),PC(6บิต),PD(8Bit))
    - มีวงจรสื่อสารอนุกรม UART จำนวน 1 พอร์ต
    - มีวงจรสื่อสาร SPI จำนวน 1 พอร์ต
    - มีวงจรสื่อสาร I2C จำนวน 1 พอร์ต
    - มีวงจร Timer/Counter ขนาด 16 บิต 1 ชุด และ Timer/Counter ขนาด 8 บิต 2 ชุด
    - มีวงจร ADC ขนาด 10บิต จำนวน 6 ช่อง
- MCU ประจำบอร์ดได้รับการติดตั้ง Bootloader สำหรับใช้ Upload Code ให้บอร์ดผ่านทางพอร์ตสื่อสารอนุกรม RS232 ได้ทันที โดยไม่ต้องใช้เครื่องโปรแกรมภายนอก สามารถใช้การพัฒนาโปรแกรมได้ทั้ง ภาษาแอสแซมบลี้ ภาษาซี ภาษาเบสิก และ ภาษา C++ ของ Arduino ได้ทันที
- มีขั้วต่อสัญญาณ I/O แบบ TTL แบบ Header 2x5 จำนวน 3 ชุด (PB,PC และ PD)
- มีสวิตซ์ RESET พร้อมวงจร External Reset แบบ RC Reset ภายในบอร์ด
- มีสวิตซ์ BL(PD2) สำหรับใช้รีเซ็ตบอร์ดเข้าทำงานใน Bootloader และใช้ทดสอบ Input
- มีขั้วต่อ RS232 สำหรับใช้งาน และ Upload Code ด้วย Bootloader ผ่าน RS232
- มีขั้วต่อ AVRISP แบบ 10PIN IDE มาตรฐาน AVRISP สำหรับใช้โปรแกรม Code ให้บอร์ด ในกรณีที่ไม่ต้องการใช้การ Upload Code ผ่านทาง RS232 ของ Bootloader
- มีวงจรขยาย Output ด้วย 74HC595 พร้อมขั้วต่อสัญญาณ Output แบบ IDE 10 Pin จำนวน 1 ชุด
- มี LED แสดงสถานะแหล่งจ่าย Power และ 2 LED สำหรับใช้ทดสอบการทำงานของบอร์ด
- ใช้กับแหล่งจ่ายไฟ 7 ถึง 10VAC/DC พร้อมวงจร Bridge Rectifier และ Regulate 1A(Low Drop)
- มีฐานสำหรับใช้ติดตั้งบอร์ดทดลอง I/O ขนาดมาตรฐาน ET-MINI I/O SET เพื่อใช้งานร่วมกับบอร์ดทดลองต่างๆ เช่น DC Motor, Stepping Motor, I2C RTC,I2C I/O,I2C EEPROM, SPI ฯลฯ
- ขนาด PCB Size เล็กเพียง 8 x 6 Cm.

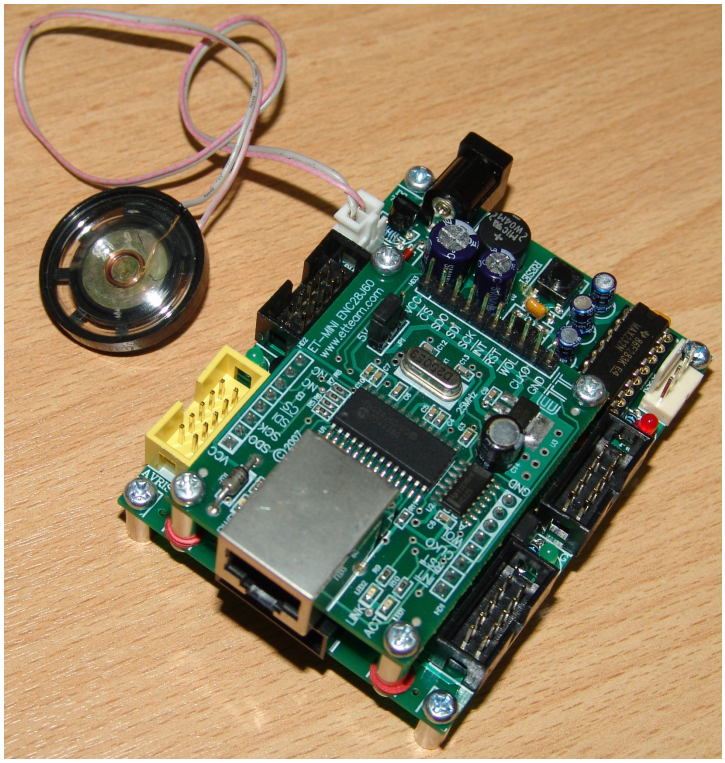
โครงสร้างบอร์ด ET-BASE AVR EASY88



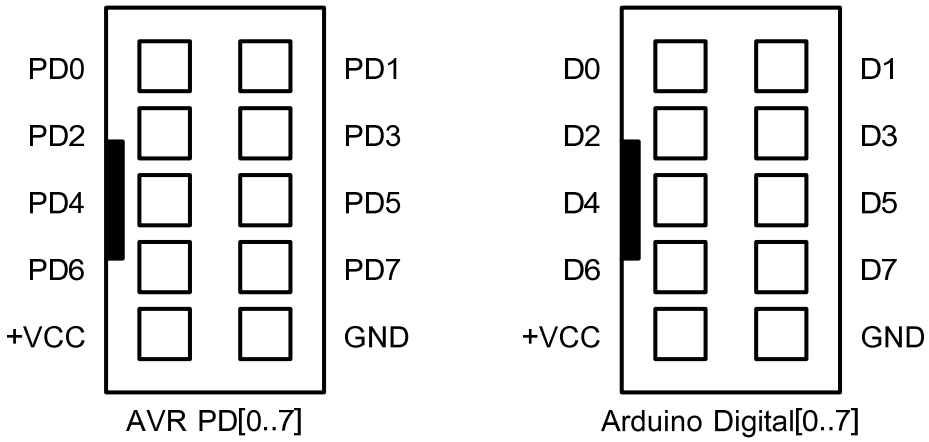
รูปแสดง โครงสร้างของบอร์ด ET-BASE AVR EASY88

- **หมายเลข** 1 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด ใช้กับแหล่งจ่าย 7-10VAC/DC
- **หมายเลข** 2 เป็นขั้วต่อสำหรับใช้ต่อกับ ลำโพง ซึ่งถูก Drive ผ่านทรานซิสเตอร์ BC337 ดังวงจร

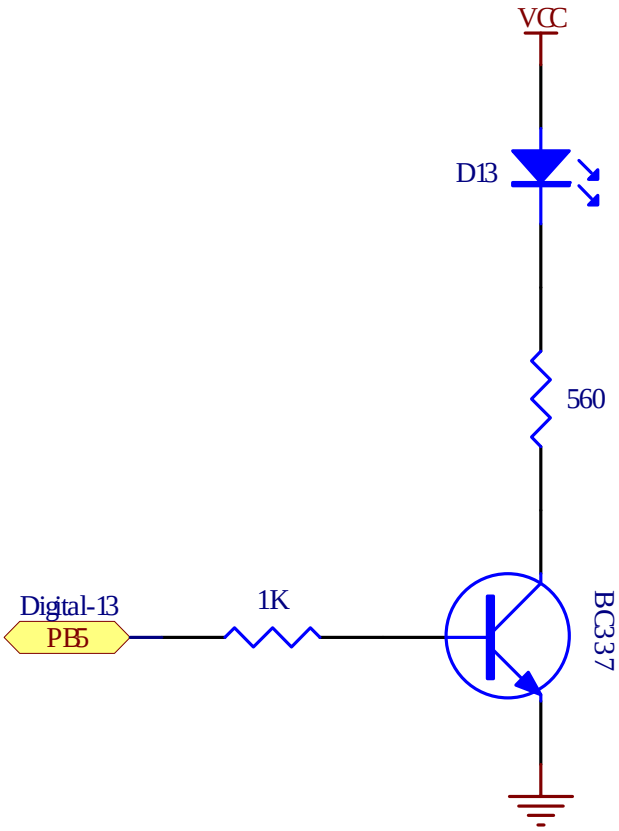




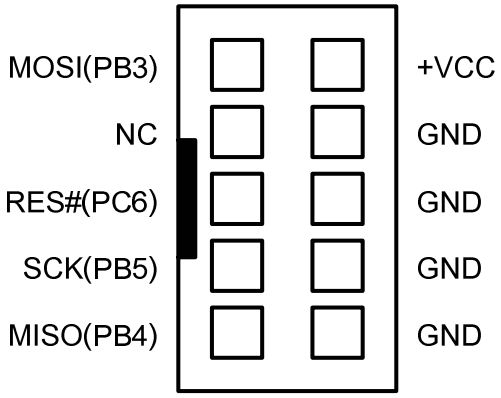
- **หมายเลข** 3 เป็นขั้วต่อสัญญาณจาก PD[0..7] ซึ่งในกรณีใช้การพัฒนาโปรแกรมด้วย Arduino จะเป็นขาสัญญาณของ Digital[0..7]



- **หมายเลข** 4 เป็น LED ใช้แสดงสถานะของขาสัญญาณ PB[5] หรือ Digital[13] ของ Arduino ซึ่ง LED นี้จะถูก Drive ผ่านทรานซิสเตอร์ BC337 ดังวงจร

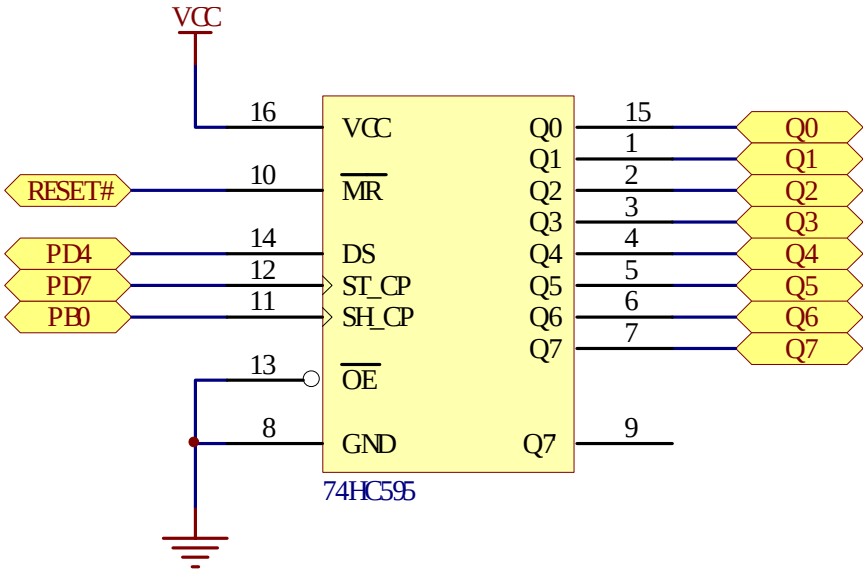


- **หมายเลข** 5 เป็นขั้วต่อ AVRISP ใช้สำหรับ Download Code ให้กับ MCU ในกรณีที่ใช้การพัฒนาโปรแกรมของบอร์ดเป็นแบบ MCU ของ AVR ตามปรกติโดยไม่ผ่านระบบ Bootloader โดยขั้วต่อ AVRISP นี้จะสามารถใช้งานได้กับเครื่องโปรแกรมทุกรุ่นที่รองรับการใช้งานกับ ATMEGA88 และใช้ขั้วต่อ ตรงตามมาตรฐาน AVRISP ดังรูป



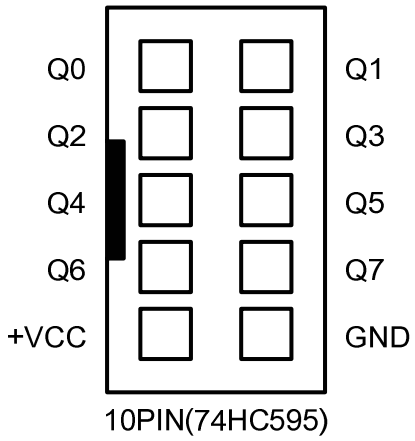
AVRISP

- **หมายเลข** 6 เป็นไอซีเบอร์ 74HC595 ซึ่งใช้ขยาย Output Port ขนาด 8 บิต โดยมีการต่อวงจรร่วมกับ MCU ที่ใช้ในบอร์ด ดังวงจร

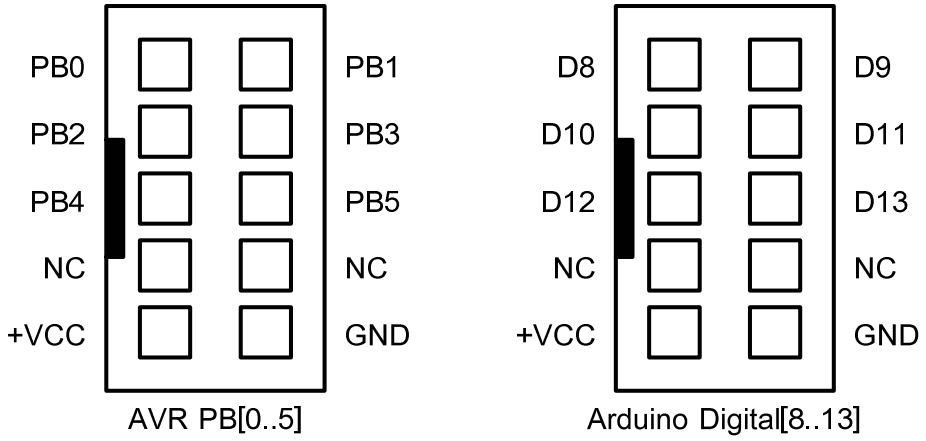


โดย Output ของ 74HC595 นี้สามารถนำไปประยุกต์ใช้งานเพื่อทำหน้าที่เป็น Output ทั่วๆไป หรือใช้สำหรับเชื่อมต่อกับ Character LCD ในแบบ 4 Bit Mode ก็ได้เช่นเดียวกัน

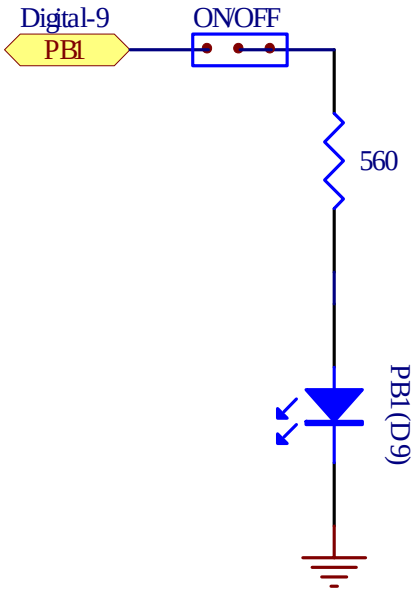
- **หมายเลข** 7 เป็นขั้วต่อสัญญาณ Output จาก 74HC595 ซึ่งมีขนาด 8บิต ก็คือ Q[0..7] โดยมีการจัดเรียงขาสัญญาณดังนี้



- **หมายเลข** 8 เป็นขั้วต่อสัญญาณจาก PB[0..5] ซึ่งในกรณีใช้การพัฒนาโปรแกรมด้วย Arduino จะเป็นขาสัญญาณของ Digital[8..13]

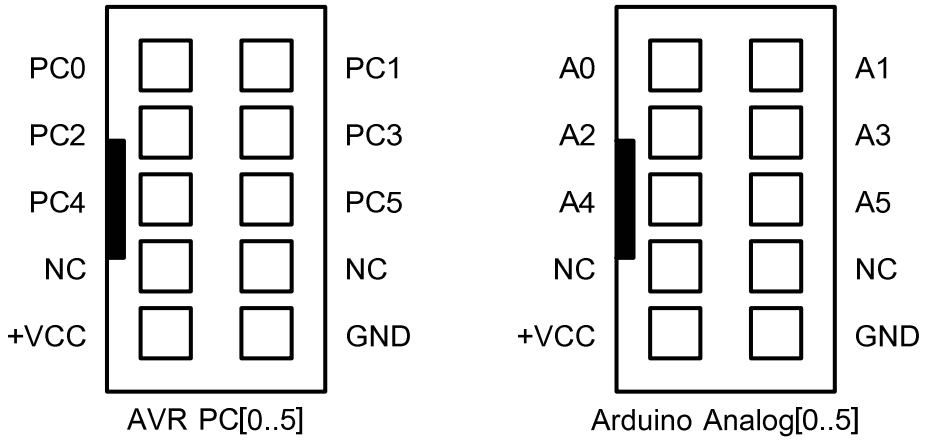


- **หมายเลข** 9 เป็น LED ใช้แสดงสถานะของขาสัญญาณ PB[1] หรือ Digital[9] ของ Arduino ซึ่ง LED นี้จะถูกต่อวงจรแบบ Source Current จากขาสัญญาณของ MCU โดยมี Jumper เป็นตัวตัดต่อสัญญาณระหว่าง PB[1] กับ LED ซึ่ง LED นี้สามารถแสดงผลได้ 2 แบบ คือ ใช้ทดสอบการแสดงผลแบบ ON/OFF เมื่อกำหนดขาสัญญาณ PB[1] เป็นแบบ Digital Output และ ใช้ทดสอบการแสดงผลแบบ Dimmer เมื่อกำหนดขาสัญญาณ PB[1] เป็นแบบ Output PWM ดังวงจร

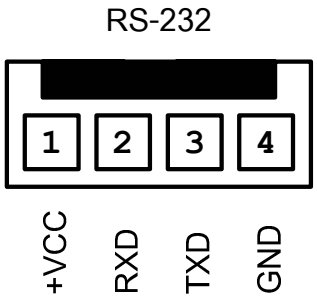


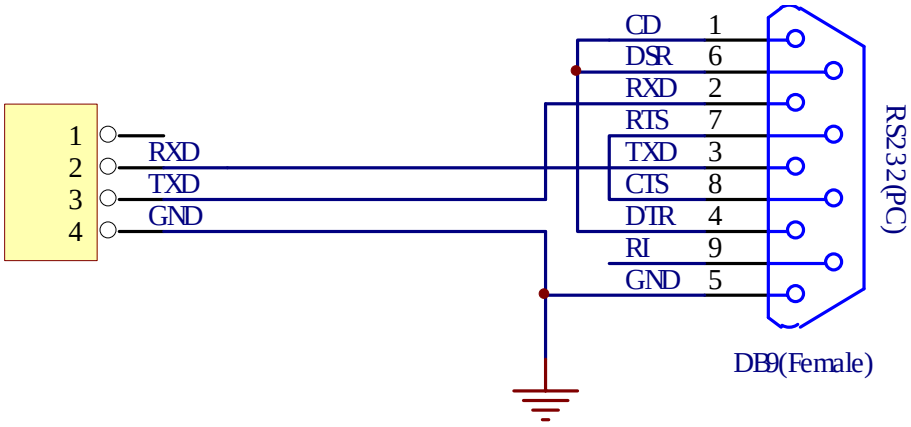
- **หมายเลข** 10 เป็น Jumper สำหรับใช้ในการตัดต่อสัญญาณ PB[1] กับ LED โดยเมื่อเลือกไว้ด้าน ON จะเป็นการต่อสัญญาณ PB[1] เข้ากับ LED แต่เมื่อเลือก OFF จะเป็นการตัดการเชื่อมต่อของ PB[1] ออกจากวงจรแสดงผลของ LED

- **หมายเลข** 11 เป็นขั้วต่อสัญญาณจาก PC[0..5] ซึ่งในกรณีใช้การพัฒนาโปรแกรมด้วย Arduino จะเป็นขาสัญญาณของ Analog[0..5]



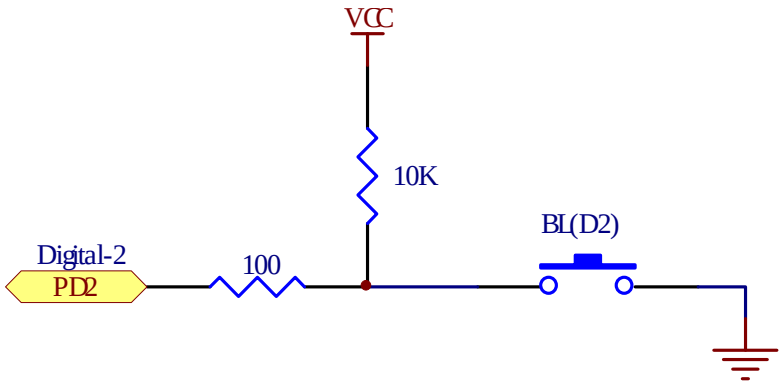
- **หมายเลข** 12 เป็น LED Power ใช้แสดงสถานะของแหล่งจ่ายไฟ +5VDC
- **หมายเลข** 13 คือ ขั้วต่อ RS232 สำหรับใช้งานทั่วไป และ Upload Code ให้กับ MCU ผ่านระบบ Bootloader โดยมีการจัดเรียงสัญญาณดังนี้





รูปแสดง วงจรการต่อสายของ RS232 สำหรับใช้งานกับบอร์ด

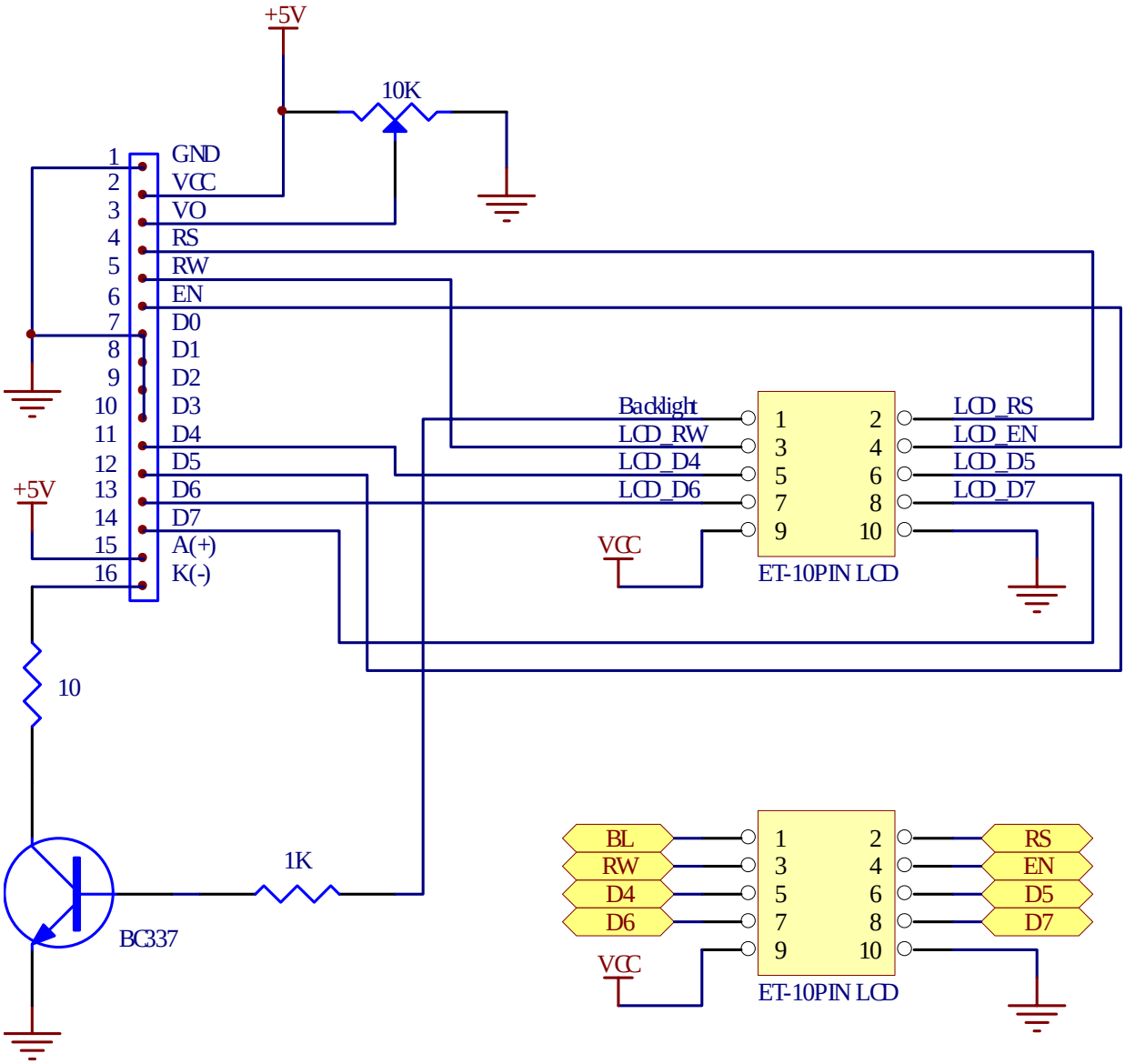
- **หมายเลข** 14 คือ สวิตช์ RESET ใช้สำหรับรีเซ็ตการทำงานของ MCU
- **หมายเลข** 15 คือ สวิตช์ BL(Bootloader) โดยต่อผ่านขาสัญญาณ PD[2] ใช้สำหรับสร้างสัญญาณโลจิก LOW ให้กับขาสัญญาณ PD[2] เพื่อทดสอบการรับค่า Input รวมทั้งการสร้างสัญญาณ Trigger Interrupt ของ INT0 รวมทั้งการใช้สั่งให้ MCU เข้าทำงานใน Bootloader โดยใช้งานร่วมกับสวิตช์ RESET โดยสวิตช์ BL มีการต่อวงจรดังรูป



- **หมายเลข** 16 เป็น MCU ประจำบอร์ด ซึ่งสามารถใช้ได้กับ AVR ขนาด 28ขา ได้หลายเบอร์ เช่น ATMEGA8,ATMEGA88,ATMEGA168 และ ATMEGA328
- **หมายเลข** 17 เป็น Crystal Oscillator ค่าความถี่ 19.6608 MHz
- **หมายเลข** 18 เป็น Header สำหรับรองรับการเชื่อมต่อสัญญาณกับบอร์ด ET-MINI ENC28J60 ของ บริษัท อีทีที จำกัด สำหรับใช้พัฒนาโปรแกรมใช้งานกับระบบ Ethernet LAN
- **หมายเลข** 19 เป็น Jumper สำหรับใช้ตัดต่อขาสัญญาณของ PD2(INT0) ที่เชื่อมต่อระหว่าง PD2(INT0) ของบอร์ด ET-BASE AVR EASY88 กับ INT ของบอร์ด ET-MINI ENC28J60 ซึ่งถ้าเลือกไว้ด้าน ENA หมายถึง Enable ซึ่งจะเป็นการเชื่อมต่อขา INT จาก ENC28J60 เข้ากับขา PD2 หรือ INT0 ของ ATMEGA88 แต่เมื่อเลือกไว้ทางด้าน DIS จะหมายถึง Disable ซึ่งเป็นการตัดการเชื่อมต่อขา INT ของ ENC28J60 ออกจากขา PD2(INT0) ของ ATMEGA88 ซึ่งตามปรกติควรเลือกไว้ที่ด้าน DIS เสมอ
- **หมายเลข** 20 เป็นตำแหน่งฐานรองสำหรับยึดบอร์ดทดลองขนาดเล็กของ บริษัท อีทีที จำกัด ที่มีขนาดมาตรฐาน ในขนาด MINI I/O Size ซึ่งผู้ใช้สามารถนำบอร์ดชุด ET-MINI I/O ต่างๆมาต่อทดลองร่วมกับบอร์ด ET-BASE AVR EASY88 ได้ทันที

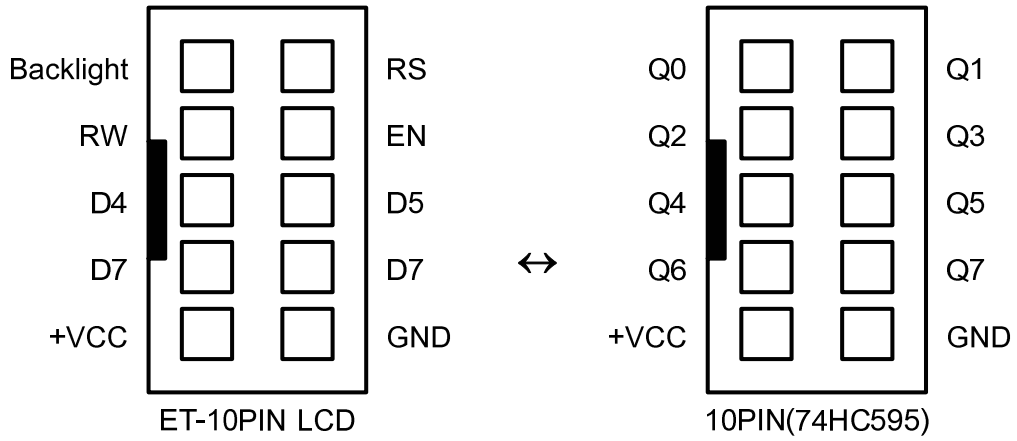
การต่อใช้งาน LCD ร่วมกับ 10PIN 74HC595

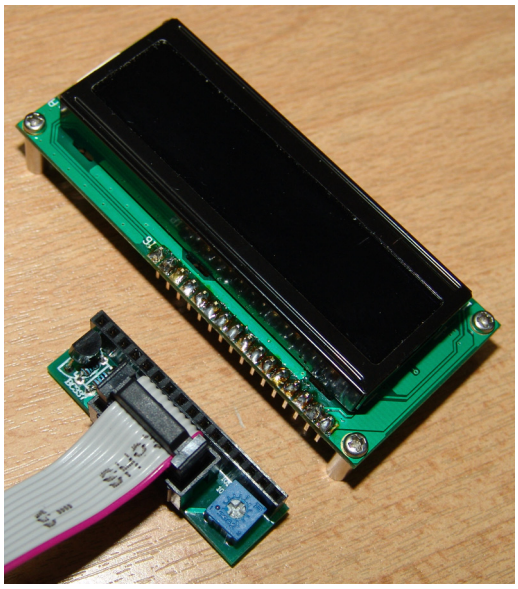
เนื่องจาก MCU เบอร์ ATMEGA88 มีขาสัญญาณอย่างจำกัด เมื่อต้องการนำมาต่อใช้งานกับ LCD จะทำให้ต้องสูญเสียขาสัญญาณไปหลายเส้น ซึ่งอาจส่งผลให้จำนวนสัญญาณไม่เพียงพอที่จะนำไปทำการเชื่อมต่อกับอุปกรณ์อื่นๆ การขยาย Output ด้วย 74HC595 จะช่วยลดการสูญเสียขาสัญญาณใช้งานไปได้เป็นอย่างมาก เนื่องจากเสียขาสัญญาณในการควบคุม 74HC595 เพียง 3 เส้น แต่ได้จำนวน Output เพิ่มขึ้นมาถึง 8 เส้น โดยสัญญาณ Output ที่ได้จาก 74HC595 นี้สามารถนำไปใช้งานเป็น Output Port สำหรับควบคุมการทำงานของอุปกรณ์ต่างๆได้ตามต้องการ รวมทั้งการควบคุมการแสดงผลของ LCD ด้วย ซึ่งในกรณีที่จะนำ Output จาก 74HC595 ไปใช้ควบคุมการแสดงผลของ LCD นั้น เพื่อความสะดวกในการใช้งาน ขอแนะนำให้จัดหาชุด ET-10PIN CLCD (ET-CONV 10 TO LCD) มาเป็นตัวกลางในการเชื่อมต่อระหว่าง ขั้ว IDE 10PIN ของ Output จาก 74HC595 หรือ 10PIN(74HC595) ดังตัวอย่าง



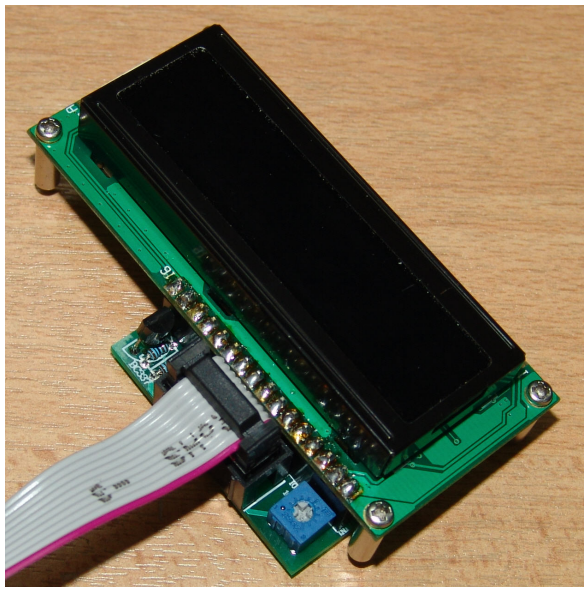
รูปแสดง วงจรของชุด ET-10PIN CLCD (ET-CONV 10 TO LCD)

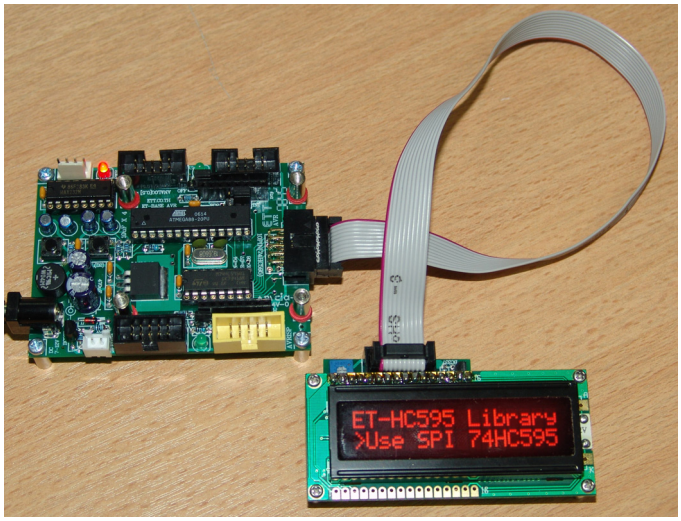
โดยเมื่อนำ LCD มาติดตั้งเข้ากับชุด ET-10PIN CLCD เรียบร้อยแล้วจะทำให้สามารถเชื่อมต่อสายสัญญาณระหว่างขั้ว ET-10PIN CLCD กับ 10PIN(74HC595) ได้โดยง่าย โดยใช้การเชื่อมต่อผ่านทางสายแพร์ขนาด 10PIN ได้ทันทีดังตัวอย่าง





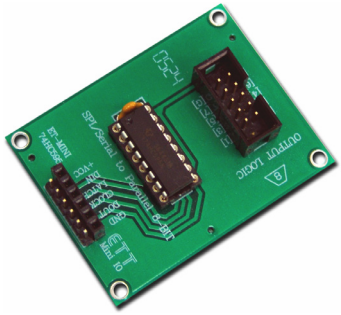
→

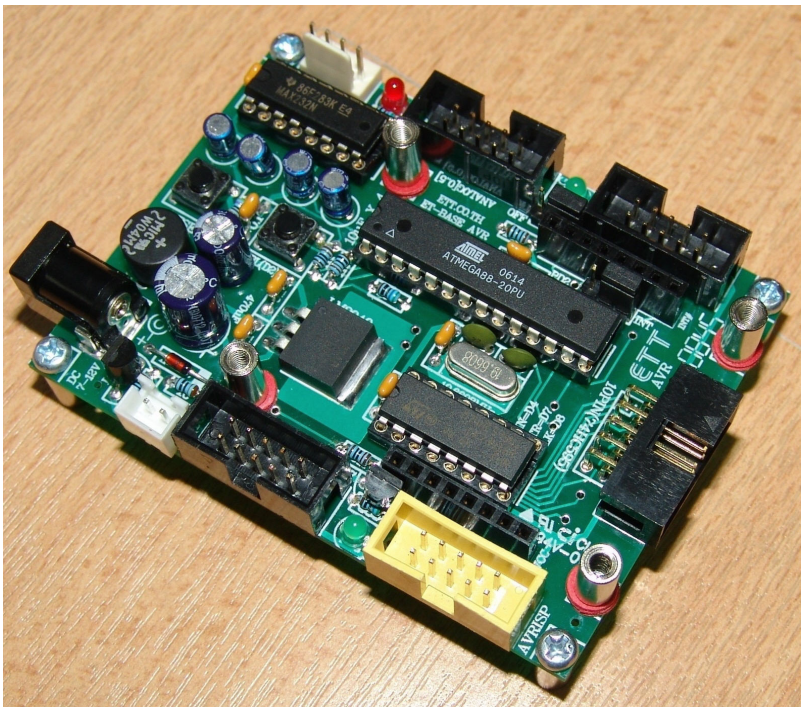


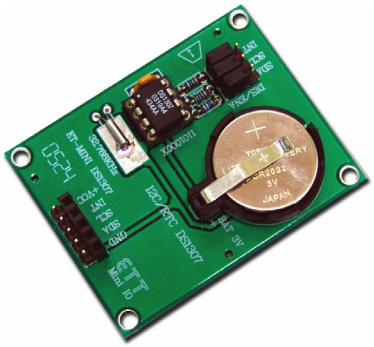
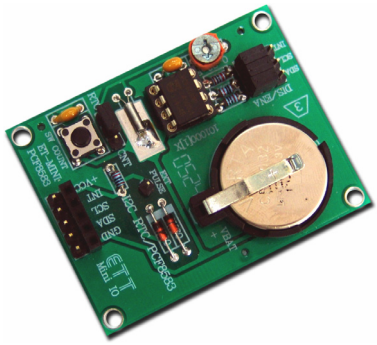
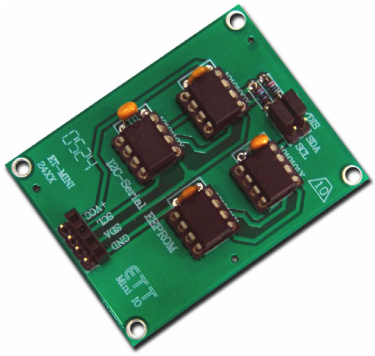


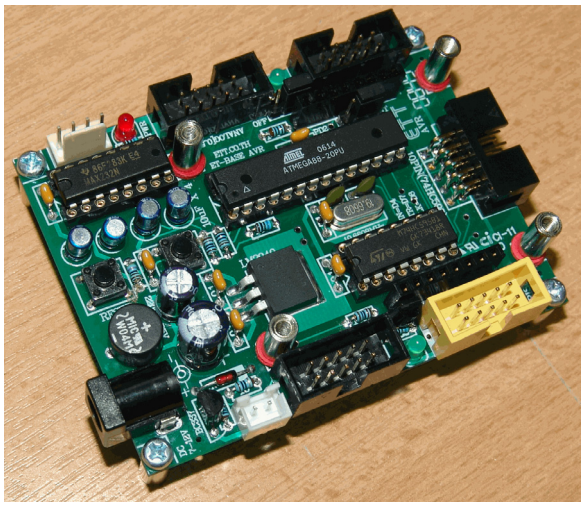
การติดตั้งบอร์ด ET-BASE AVR EASY88 ใช้งานกับ ET-MINI I/O

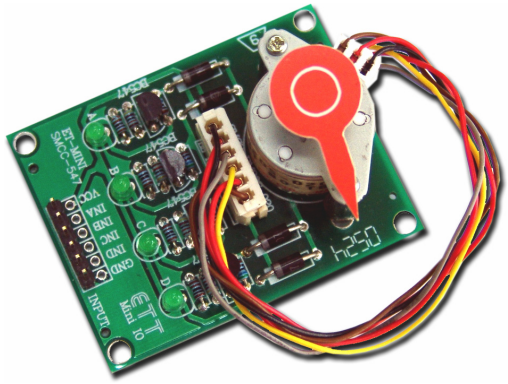
ในการติดตั้งชุดทดลอง ET-MINI I/O SET เข้ากับบอร์ด ET-BASE AVR EASY88 นั้นสามารถวางบอร์ดของ ET-MINI I/O ซ้อนทับลงบนบอร์ด ET-BASE AVR EASY88 ได้ทันที โดยเฉพาะอย่างยิ่งถ้าเป็นบอร์ดรุ่น ET-MINI ENC28J60 จะมีการออกแบบขั้วสัญญาณไว้รองรับโดยเฉพาะเลย สามารถทำการติดตั้งบอร์ดใช้งานร่วมกันโดยไม่ต้องต่อสายสัญญาณใดๆ แต่ถ้าเป็นบอร์ด ET-MINI I/O รุ่นอื่นๆ ต้องทำการต่อสายสัญญาณระหว่างบอร์ด ET-MINI I/O เข้ากับ ET-BASE AVR EASY88 ด้วย ดังตัวอย่าง

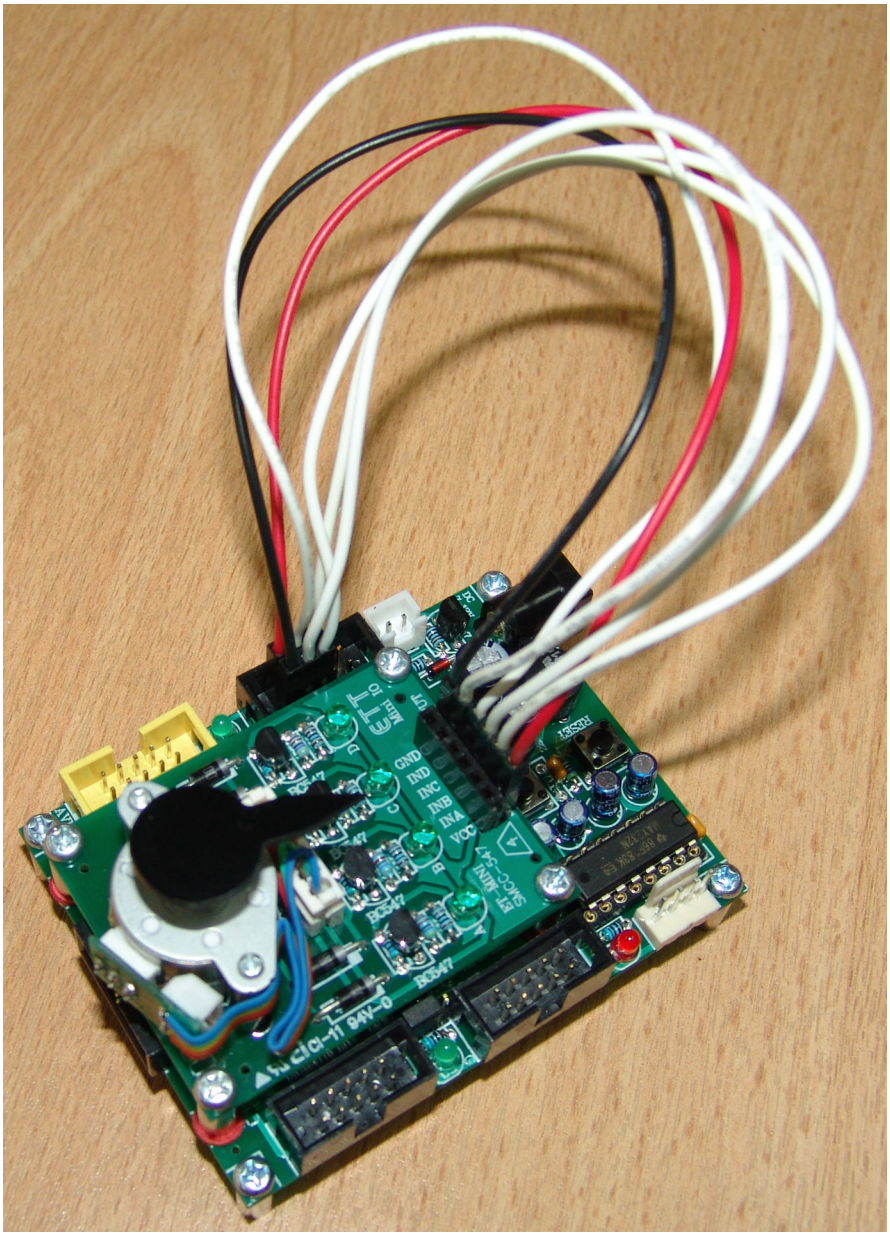




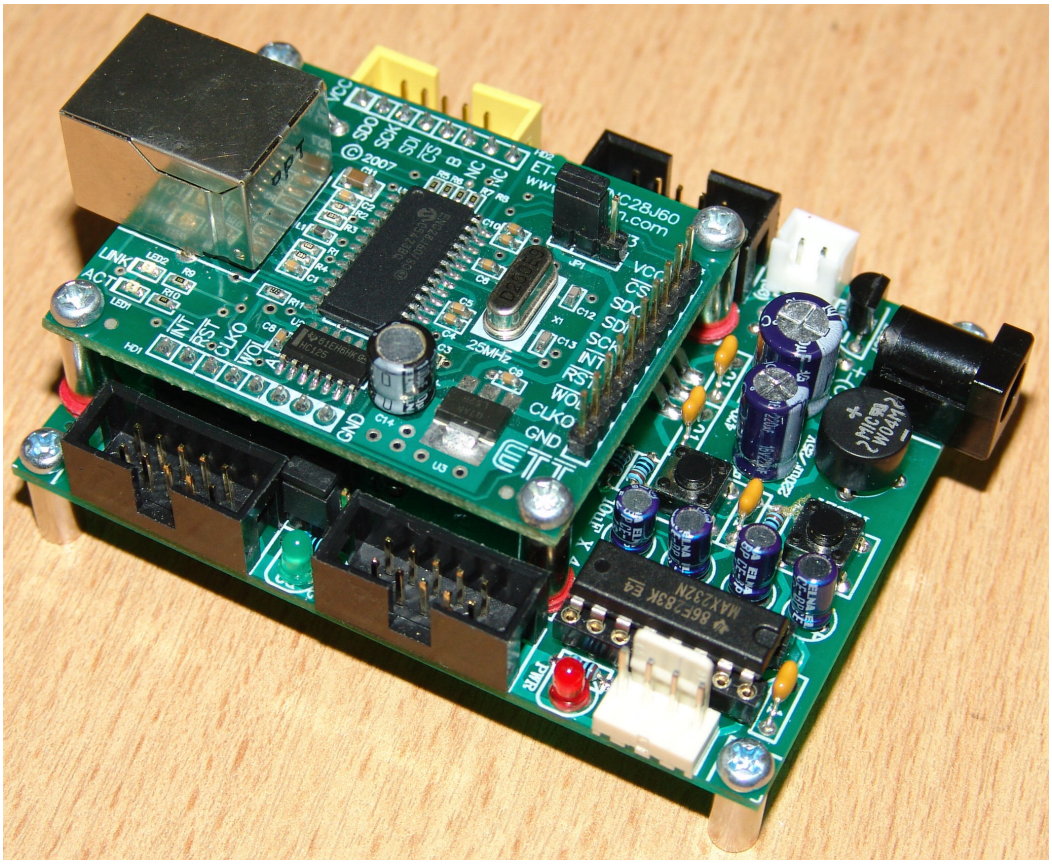






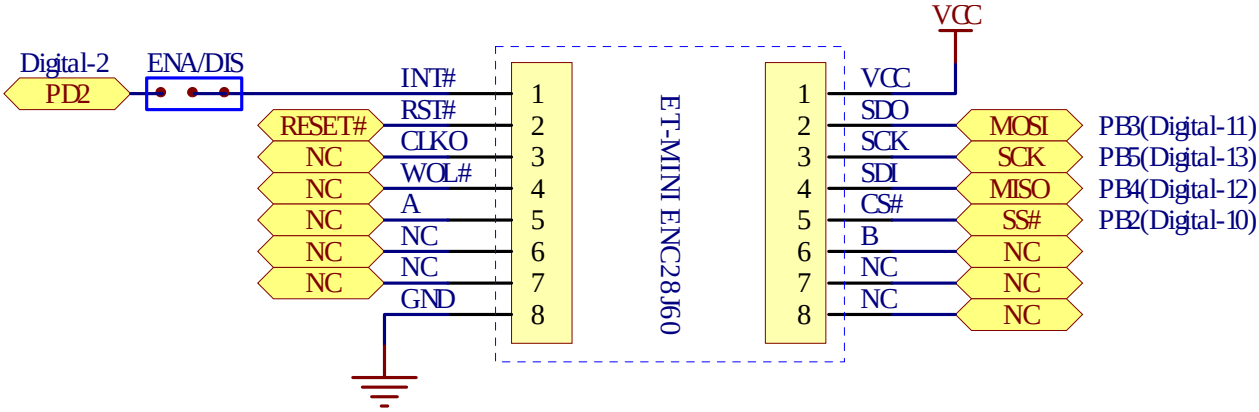
รูปแสดง ตัวอย่างการติดตั้ง ET-MINI SMCC-547 กับ ET-BASE AVR EASY88

การติด ET-BASE AVR EASY88 ใช้งานกับ ET-MINI ENC28J60

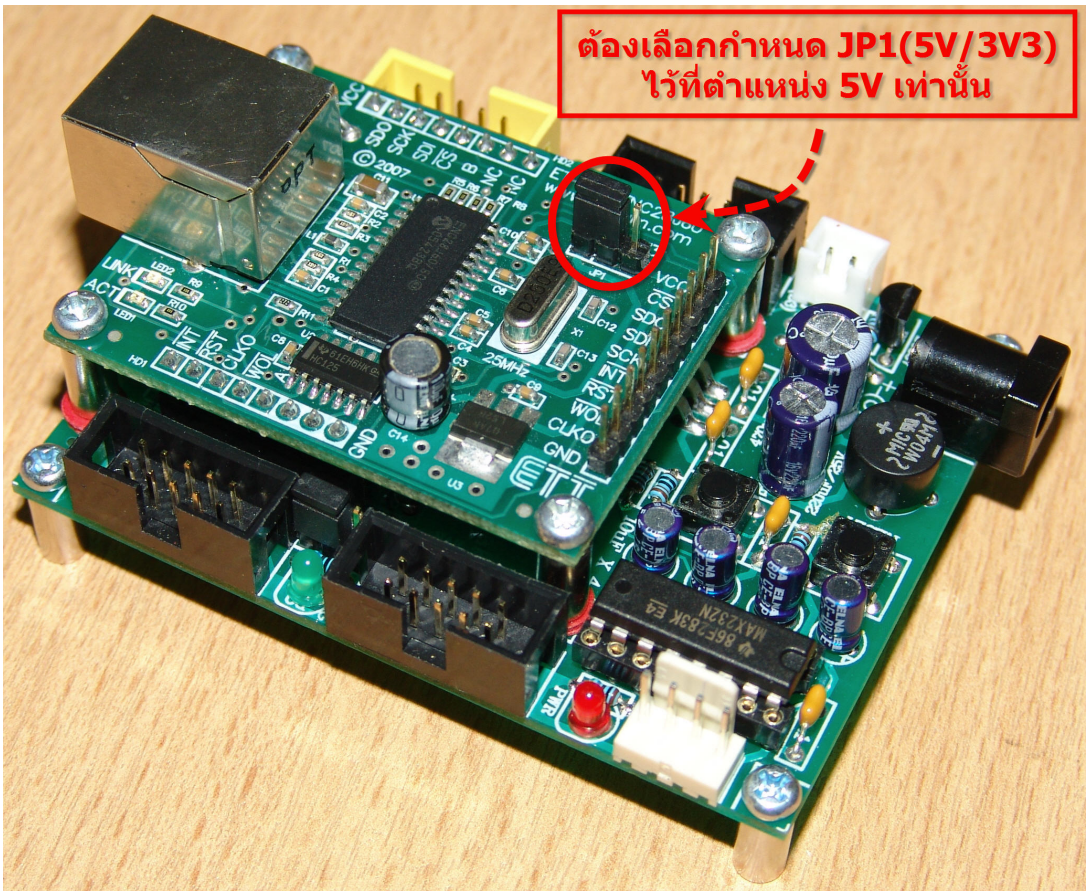


รูปแสดง การติดตั้งบอร์ด ET-MINI ENC28J60 กับบอร์ด ET-BASE AVR EASY88

โดยบอร์ด ET-MINI ENC28J60 เมื่อทำการติดตั้งใช้งานร่วมกับบอร์ด ET-BASE VAR EASY88 แล้ว สัญญาณการเชื่อมต่อจะถูกกำหนดไว้คงที่ตามการจัดสรร I/O ที่ทางอีทีที ได้ออกแบบรองรับไว้แล้ว ไม่สามารถเปลี่ยนแปลงได้ โดยการเชื่อมต่อจะถูกกำหนดดังวงจร



หมายเหตุ ต้องกำหนด Jumper JP1 สำหรับเลือกขนาดแรงดันของแหล่งจ่ายไฟ (5V / 3V3) ที่อยู่บนบอร์ด ET-MINI ENC28J60 ไว้ทางด้าน 5V ด้วย



รูปแสดง การติดตั้งบอร์ด ET-BASE AVR EASY88 กับ ET-MINI ENC28J60

บอร์ด ET-MINI ENC28J60	ทิศทางสัญญาณ	บอร์ด ET-BASE AVR EASY88 (SPI Bus)
$\overline{CS}$	←	PB2(GPIO Out) เป็น Output ออกจาก MCU
SDO	←	PB3(SPI:MOSI) Output ออกจาก MCU
SCK	←	PB5(SPI:SCK) Output ออกจาก MCU
SDI	→	PB4(SPI:MISO) Input เข้าไปยัง MCU
$\overline{INT}$	→	PD2(INT0) Input เข้าไปยัง Interrupt ของ MCU
$\overline{RST}$	←	Output ออกจาก MCU
$\overline{WOL}$	→	-
CLKO	→	-
LEDA	→	-
LEDB	→	-

ตารางแสดง สัญญาณที่ต้องใช้เชื่อมต่อกับบอร์ด ET-MINI ENC28J60

การพัฒนาโปรแกรมของ ET-BASE AVR EASY88 ด้วย Arduino

ตามปรกติแล้วบอร์ด ET-BASE AVR EASY88 จะทำการ ติดตั้งโปรแกรม Bootloader ไว้ให้เรียบร้อยแล้ว โดยใช้ Bootloader ชื่อ “BOOT_EASY88_MANUAL_19_6608MHZ.HEX” ซึ่งเป็น Bootloader ที่ทางทีมงาน อีทีที ได้นำต้นฉบับจาก Arduino มาปรับปรุงเงื่อนไขการทำงานใหม่ เพื่อให้การทำงานสอดคล้องกับระบบฮาร์ดแวร์ของบอร์ด ET-BASE AVR EASY88 ได้ดียิ่งขึ้น โดยโปรแกรม Bootloader นี้จะใช้สำหรับติดต่อสื่อสารเพื่อสั่ง Upload Code จากคอมพิวเตอร์ PC ให้กับ MCU ในบอร์ดทำงาน โดยไม่ต้องใช้เครื่องโปรแกรมภายนอกให้ยุ่งยาก ซึ่ง คุณสมบัติของ Bootloader ที่ทาง อีทีที ปรับปรุงเพิ่มเติมขึ้น มีคุณสมบัติการทำงานเป็นดังนี้

- สื่อสารกับโปรแกรมภายนอกด้วย Protocol แบบ STK500 (STK500V1)
- ใช้ความเร็ว Baudrate 19200 โดยใช้ความถี่ XTAL 19.6608 MHz
- โปรแกรม Bootloader มีขนาด 2KByte ทำงานที่ตำแหน่ง 0x1800-0x1FFF
- ใช้ LED ที่ต่อกับขา Digital-13(PB5) เป็นตัวแสดงสถานะในขณะที่ Bootloader ทำงาน
- โปรแกรมใน Bootloader จะทำงานโดยอัตโนมัติทุกครั้งหลังการรีเซ็ต โดย MCU จะเริ่มต้นทำงานใน Bootloader นี้ก่อนเสมอ จากนั้นจะตรวจสอบสถานะทางโลจิกของขา PD2 โดยถ้าพบว่าขา PD2 เป็น HIGH ก็จะออกจากการทำงานใน Bootloader เพื่อไปเริ่มต้นทำงานตามคำสั่งที่เป็นของผู้ใช้โดยอัตโนมัติ แต่ถ้าพบว่าขา PD2 เป็น LOW จะเข้าทำงานใน Bootloader Programming Mode โดยในขณะที่ Bootloader Programming Mode ทำงาน ตอนเริ่มต้นจะเห็น LED ที่ต่อไว้กับขาสัญญาณ Digital-13(PB5) กระพริบ 3 ครั้ง แล้วติดค้าง เพื่อรอการติดต่อสื่อสารจากโปรแกรมสำหรับสั่งให้ทำการ Upload Code ให้กับ MCU จนกว่าจะถูกสั่งรีเซ็ต

สำหรับบอร์ด ET-BASE AVR EASY88 นั้น จะใช้สวิตซ์ BL(PD2) และสวิตซ์ RESET เป็นตัวเลือกการทำงานของ Bootloader ซึ่งถ้าต้องการให้ MCU ทำงานจาก Code ที่ผู้ใช้เขียนขึ้นก็ให้ใช้การกดสวิตซ์ RESET เพียงอย่างเดียว แต่ถ้าต้องการให้ MCU เข้าสู่การทำงานใน Bootloader Programming Mode ก็ให้กดสวิตซ์ BL(PD2) ค้างไว้แล้วกดสวิตซ์ RESET โดยต้องกดสวิตซ์ BL(PD2) ค้างไว้ก่อนจนกว่าจะปล่อยสวิตซ์ RESET แล้วจึงค่อยปล่อยสวิตซ์ BL(PD2) เป็นลำดับสุดท้าย โดยจะสังเกตเห็น LED ที่ต่อไว้ที่ขา PB5 กระพริบ 3 ครั้งแล้วติดค้างรอไว้ เพื่อบอกให้ทราบว่า MCU เข้าทำงานใน Bootloader แล้ว

แต่อย่างไรก็ตามในการที่จะสามารถทำการติดตั้งโปรแกรม Bootloader Code ให้กับ MCU ได้นั้น ผู้ใช้จำเป็นต้องมีเครื่องมือสำหรับทำหน้าที่ Program Code ให้กับ MCU อยู่ด้วย โดยใช้เครื่องโปรแกรมที่มีขั้วต่อตามมาตรฐานของ “AVRISP” ของ ATMEL แบบ IDE 10 PIN ได้ทันที โดยวิธีการให้ศึกษาเพิ่มเติมจากหัวข้อ “**การโปรแกรม Bootloader ให้กับบอร์ด ET-BASE AVR EASY88**” ในตอนท้ายของคู่มือนี้

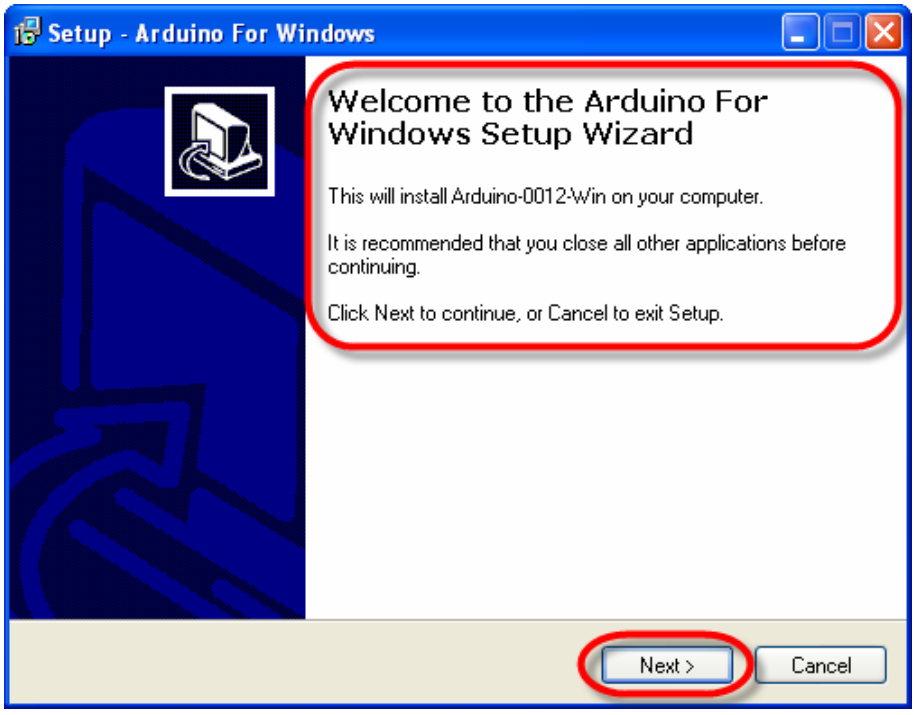
การติดตั้งโปรแกรม Arduino

สำหรับโปรแกรม Arduino นั้น ได้รับการพัฒนาขึ้นมาให้สามารถใช้งานกับระบบปฏิบัติการแบบต่างๆ ได้หลาย Platform ซึ่งปัจจุบัน (เดือน ธันวาคม พศ.2551) โปรแกรมของ Arduino ได้รับการปรับปรุงเป็นรุ่น เวอร์ชั่น “Arduino-0012” แล้ว โดยมีโปรแกรมให้เลือกใช้งาน 4 Platform ทั้ง Windows,Mac OSx และ Linux โดยผู้อ่านสามารถเข้าไป ตรวจสอบ หรือ Download โปรแกรมรุ่นใหม่ๆของ Arduino มาใช้งานได้ฟรีโดยไม่เสียค่าใช้จ่ายใดๆจาก “http://arduino.cc/” หรือ “http://arduino.cc/en/Main/Software” ซึ่งเป็นเวปไซต์ที่ได้รวบรวมรายละเอียดและข่าวคราวความเคลื่อนไหวต่างๆ เกี่ยวกับ Arduino มากมาย ซึ่งข้อมูลต่างๆจะได้รับการปรับปรุงอย่างต่อเนื่องเป็นประจำ

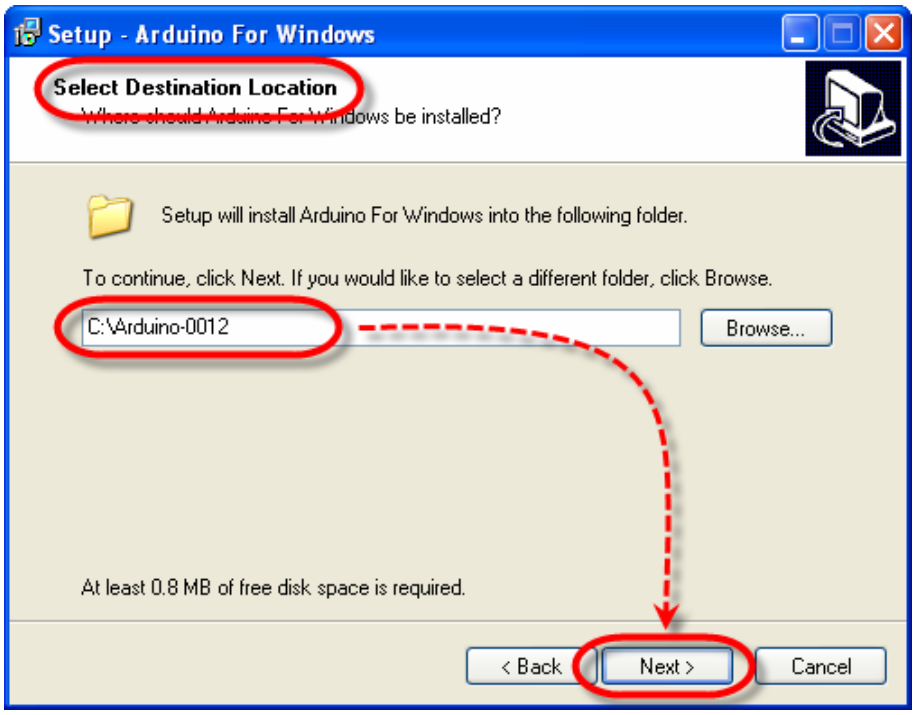
สำหรับกรณี ที่ผู้อ่านใช้งานกับบอร์ด Arduino ของ บริษัท อีทีที นั้น โปรแกรมต่างๆจะถูกรวบรวมไว้ในแผ่น CD ROM เป็นที่เรียบร้อยแล้ว โดยโปรแกรมดังกล่าวจะเป็น รุ่นที่ได้รับการปรับแต่งรายละเอียดให้สามารถใช้งานร่วมกันกับบอร์ดรุ่นต่างๆที่ทางบริษัท อีทีที ผลิตขึ้นมาใหม่ได้ด้วย นอกจากนี้แล้วทางบริษัท อีทีที ยังได้ทำการเพิ่มเติม Library ส่วนที่ทาง อีทีที พัฒนาปรับปรุงขึ้นมาใหม่รวมไว้ในชุดโปรแกรมดังกล่าวด้วยแล้ว พร้อมกับทำการเพิ่ม Install Shield เข้าไปด้วยเพื่อให้ผู้ใช้สามารถทำการติดตั้งใช้งานโปรแกรมได้โดยง่ายเช่นเดียวกันกับการ Install โปรแกรมทั่วๆไป

สำหรับขั้นตอนการติดตั้งโปรแกรมนั้น สามารถทำได้ตามขั้นตอนของ Wizard ในการติดตั้งของโปรแกรมได้ทันที โดยมีลำดับขั้นตอนดังนี้

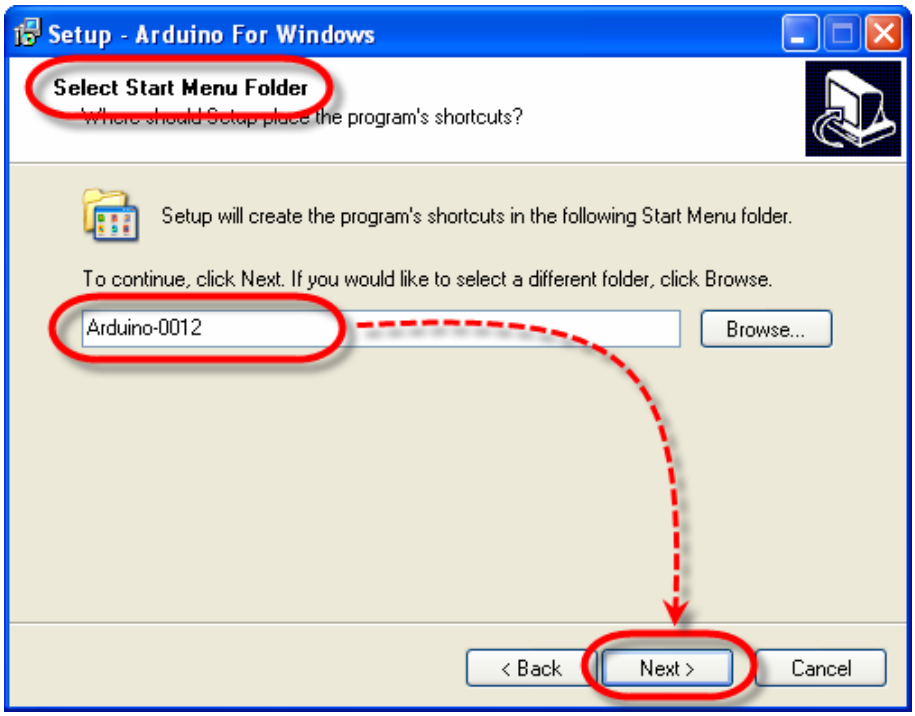
1. สั่ง Run ไฟล์ “ET-ARDUINO-0012-WIN.EXE” ซึ่งจะได้ผลดังรูป



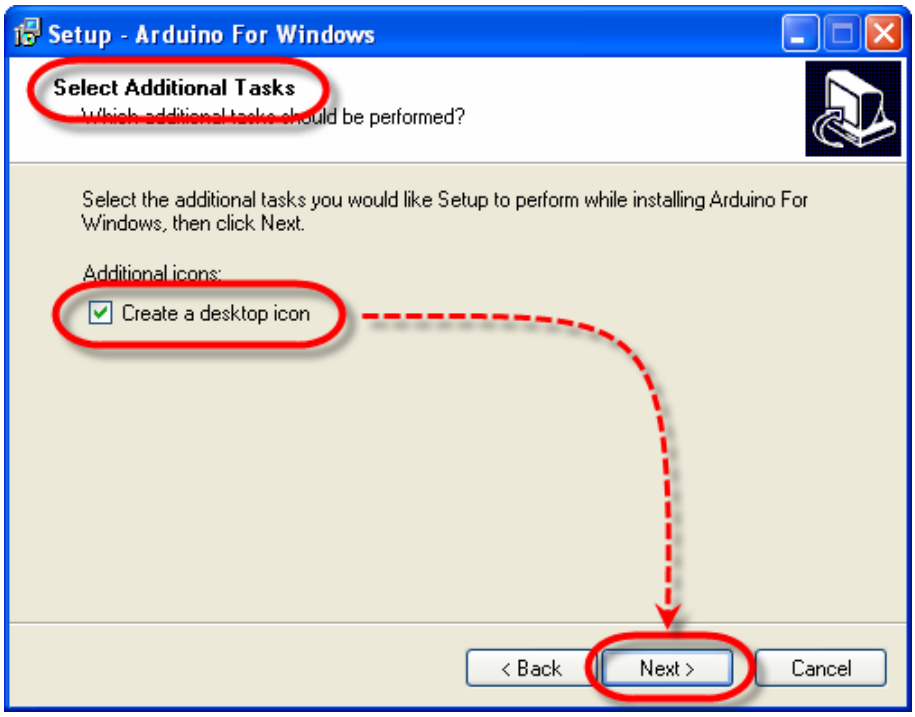
2. ในขั้นตอนนี้โปรแกรมจะให้กำหนด ตำแหน่งโฟลเดอร์ที่จะใช้สำหรับติดตั้งโปรแกรม ซึ่งให้เลือกกำหนดตามค่า Default ของโปรแกรมการติดตั้ง คือ C:\Arduino-0012 แล้วเลือก Next ดังรูป



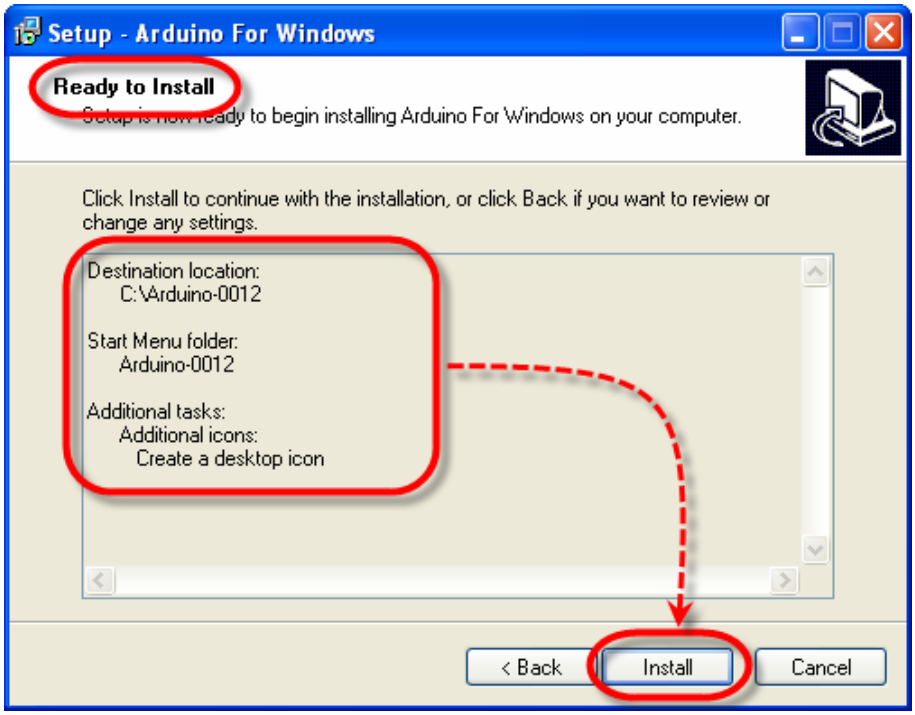
3. ในขั้นตอนนี้โปรแกรมจะให้กำหนด ชื่อโฟลเดอร์ที่จะใช้สำหรับเรียกใช้โปรแกรมผ่านทางเมนูคำสั่งของ Windows ซึ่งในขั้นตอนนี้แนะนำให้เลือกกำหนดตามค่า Default ของโปรแกรมการติดตั้ง คือ C:\Arduino-0012 แล้วเลือก Next ดังรูป



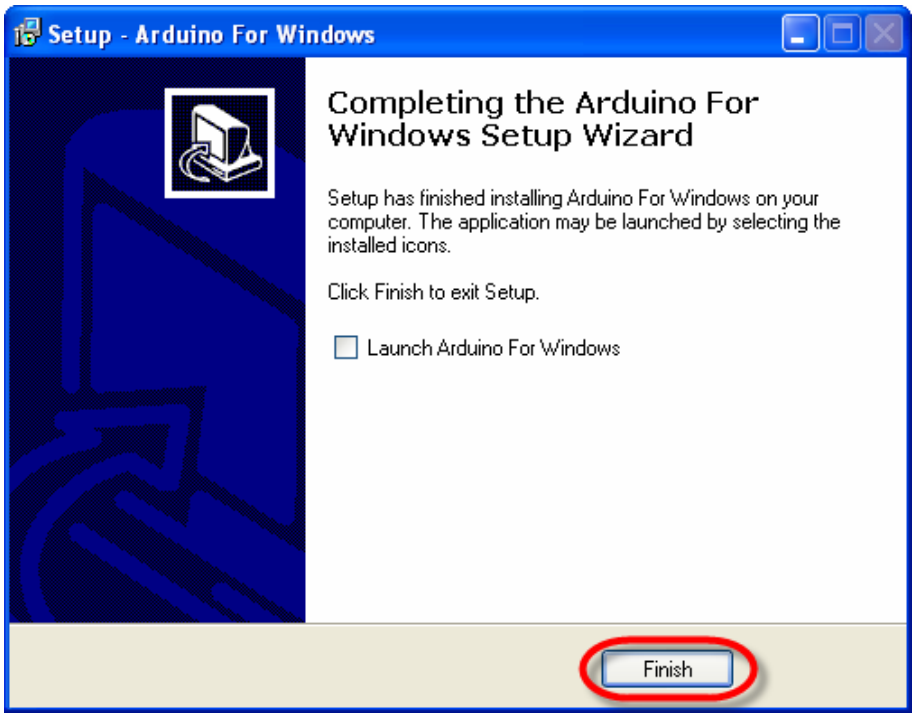
4. ในขั้นตอนนี้ให้เลือก Create a desktop icon ด้วย เพื่อให้โปรแกรมสร้าง Icon สำหรับเรียกใช้งานโปรแกรมที่หน้า Desktop ให้ด้วย แล้วเลือก Next ดังรูป



5. เมื่อถึงขั้นตอนนี้ โปรแกรมก็พร้อมทำการติดตั้งแล้ว โดยโปรแกรมจะแสดงค่าตัวเลือกต่างๆ ที่ถูกกำหนดไว้ในขั้นตอนก่อนหน้านี้ให้ทราบ เมื่อทุกอย่างถูกต้องให้เลือก Install ซึ่งโปรแกรมก็จะเริ่มทำการติดตั้งทันที



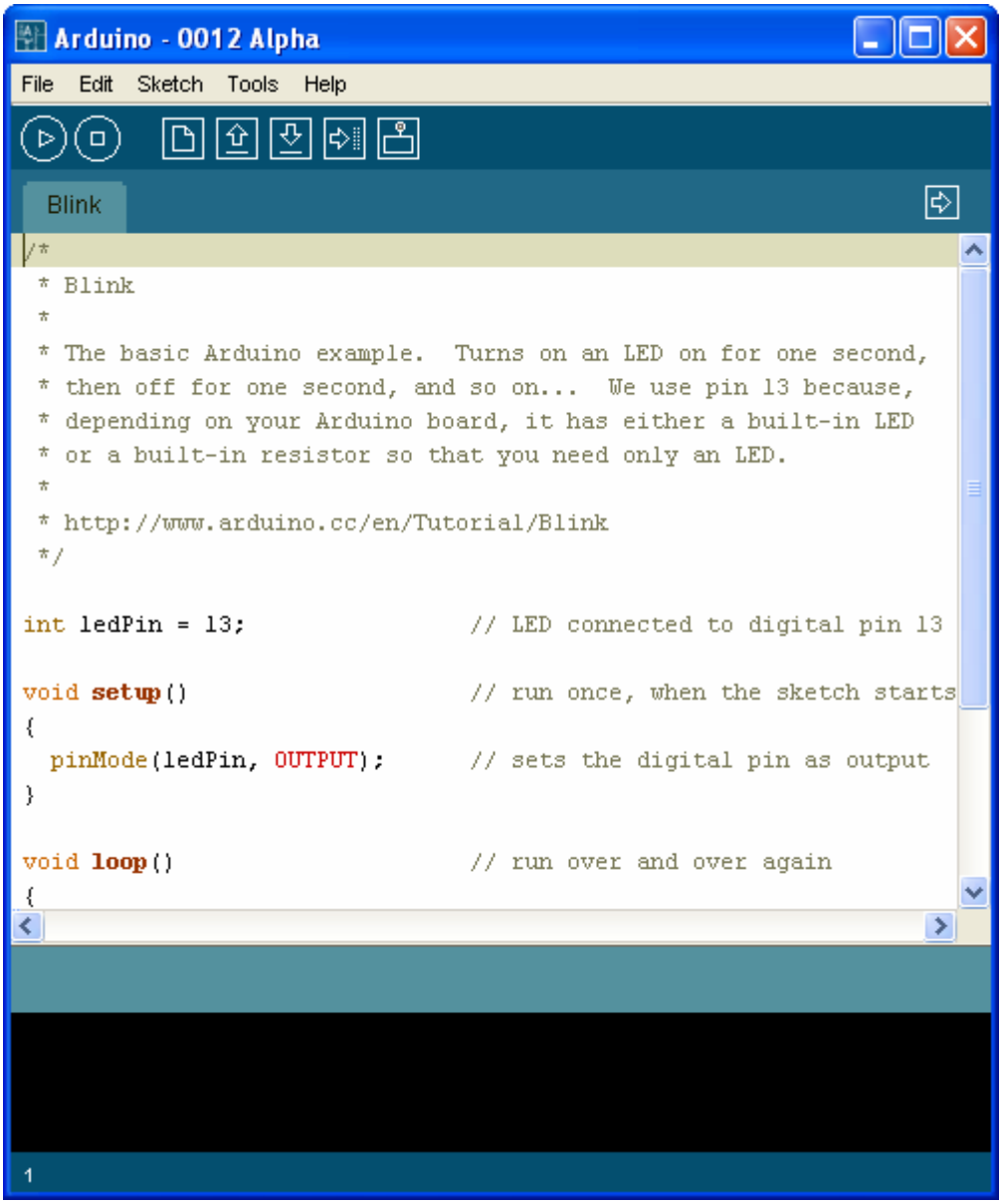
6. ให้รอจนกระทั่งขั้นตอนการติดตั้งเรียบร้อยแล้วเลือก Finish ดังรูป



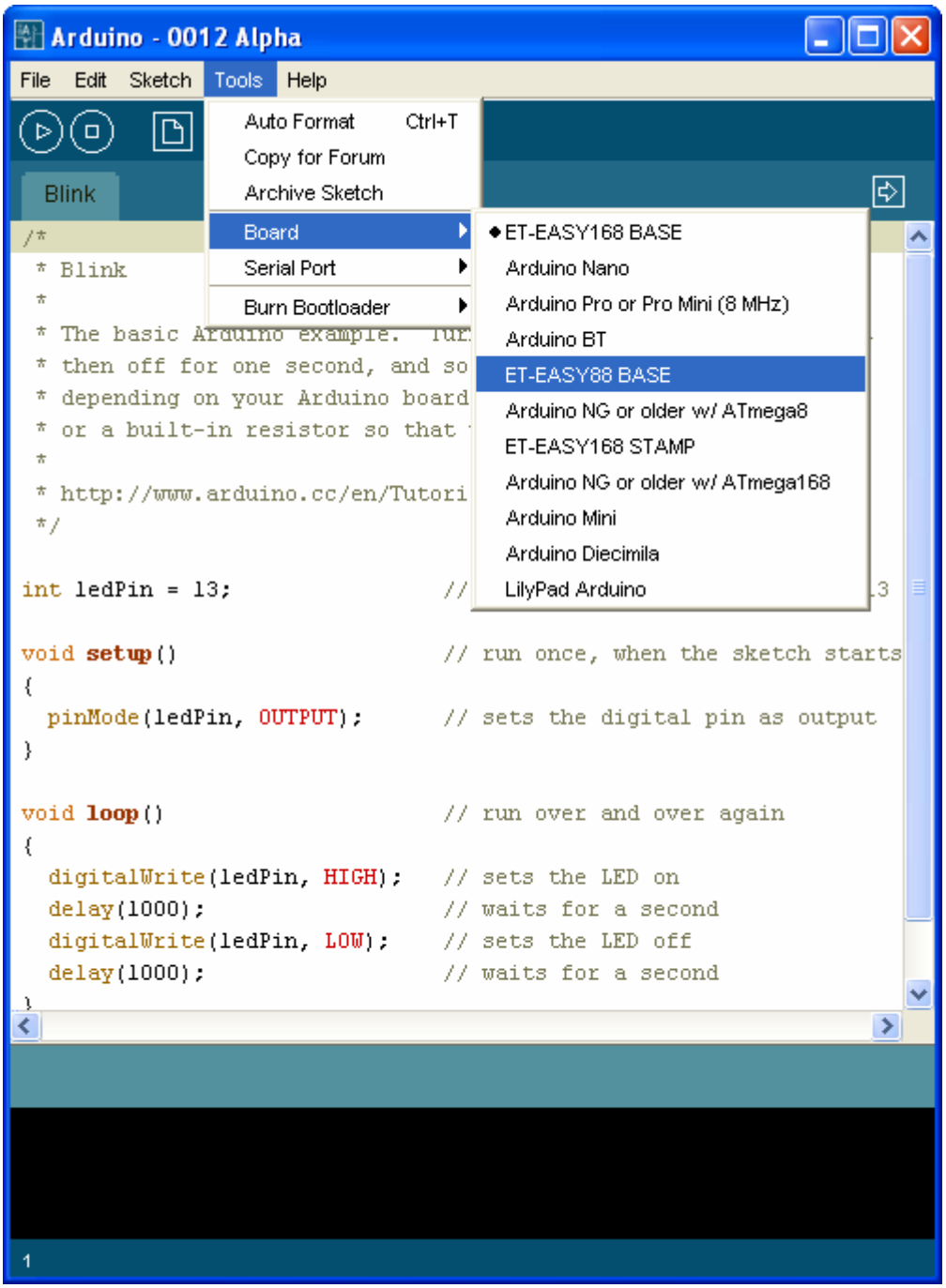
ทดสอบเขียนโปรแกรมใช้งานด้วย Arduino

หลังจากที่เราได้ทำการติดตั้งโปรแกรม Arduino เป็นที่เรียบร้อยแล้ว ก็เป็นอันเสร็จสิ้นขั้นตอนของการเตรียมการแล้ว ลำดับขั้นตอนต่อจากนี้เป็นต้นไป ก็เป็นเรื่องของการใช้งาน การเขียนโปรแกรม และการศึกษาเรียนรู้ต่างๆตามความต้องการแล้ว แต่ก่อนอื่นเราจะต้องทำการติดตั้งโปรแกรมของ Arduino เพื่อใช้เป็นโปรแกรมสำหรับศึกษาเรียนรู้ ซึ่งมีลำดับขั้นตอนดังต่อไปนี้

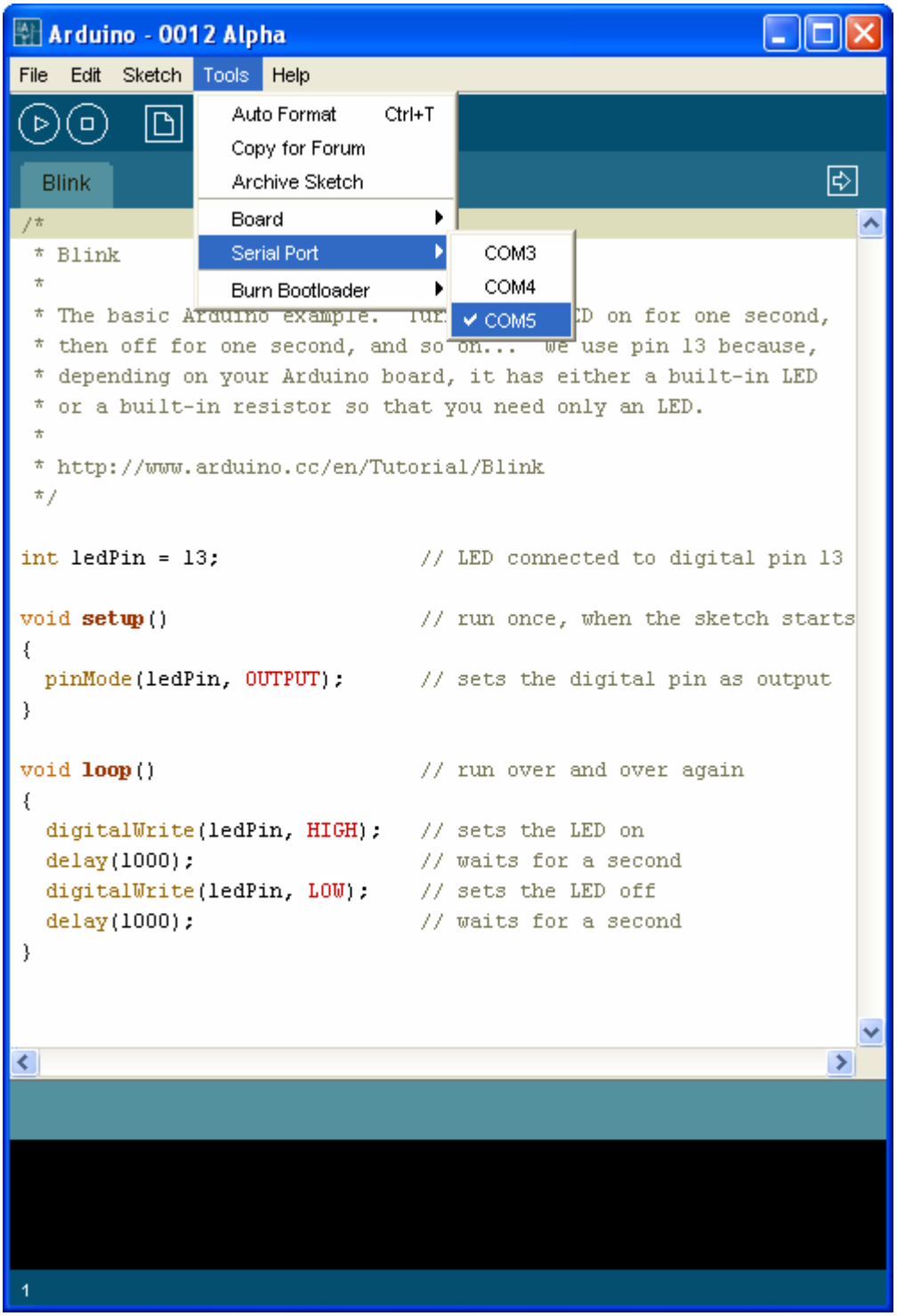
1. ทำการสั่ง Run โปแกรม “arduino.exe” จะได้ผลดังรูป



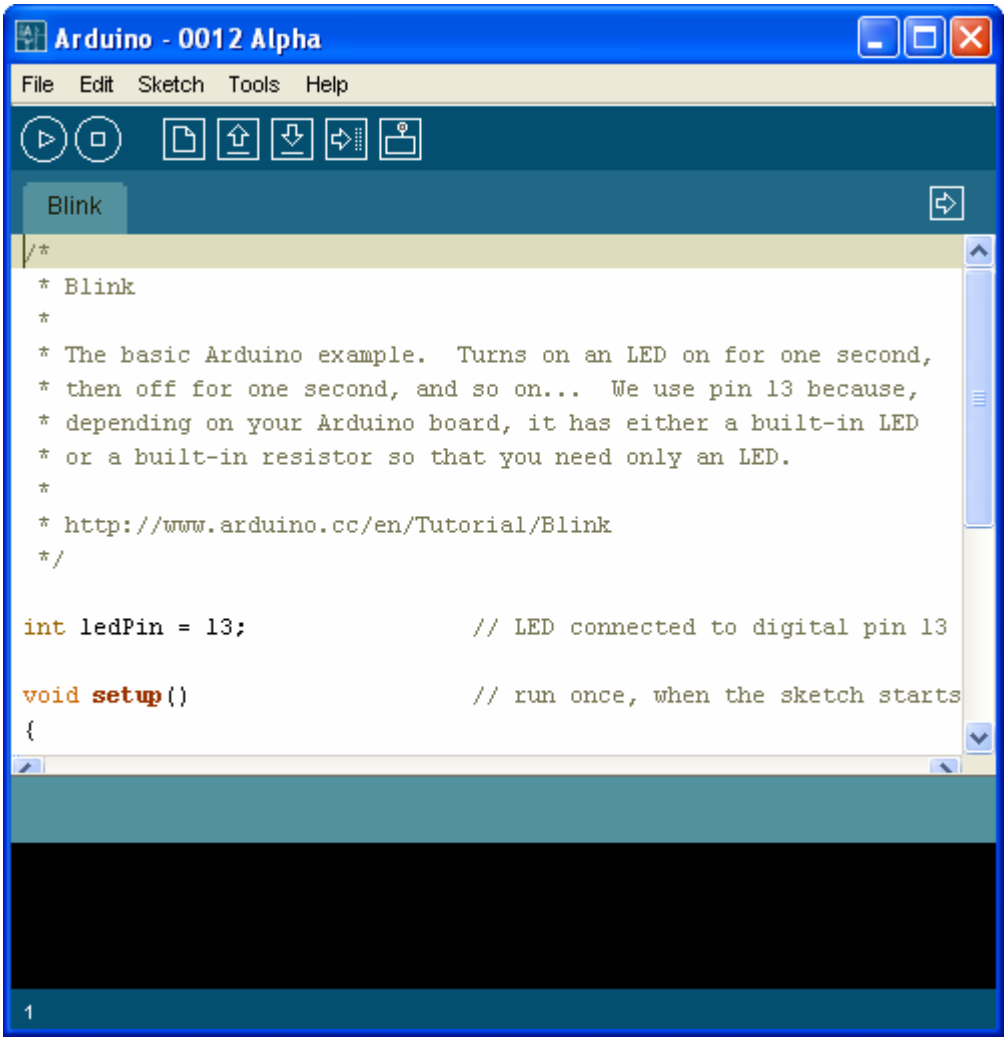
2. ในครั้งแรกของการเรียกใช้งานโปรแกรม ให้ทำการกำหนดระบบฮาร์ดแวร์ที่จะใช้งานกับโปรแกรมของ Arduino ให้เรียบร้อยเสียก่อน เนื่องจากในปัจจุบันนี้ มีการออกแบบวงจรและสร้างฮาร์ดแวร์บอร์ดแบบต่างๆสำหรับนำมาใช้งานร่วมกับโปรแกรมพัฒนาของ Arduino ไว้มากมายหลายรุ่น โดยในกรณีของบอร์ด ET-BASE AVR EASY88 ให้ทำการเลือกกำหนดชื่อบอร์ดเป็น “EASY88 BASE” โดยคลิกเมาส์ที่ “Tools → Board → “ET-EASY88 BASE” ดังรูป



3. เลือกกำหนดหมายเลขพอร์ต สำหรับติดต่อสื่อสารกับบอร์ด ให้ตรงกับหมายเลข Comport ที่ต่อใช้งานไว้จริงในเครื่องคอมพิวเตอร์ PC เช่น ถ้าหมายเลข Comport ของเครื่องคอมพิวเตอร์ PC เป็น COM5 ให้คลิกเมาส์ที่ Tools → Serial Port → COM5 ดังรูป

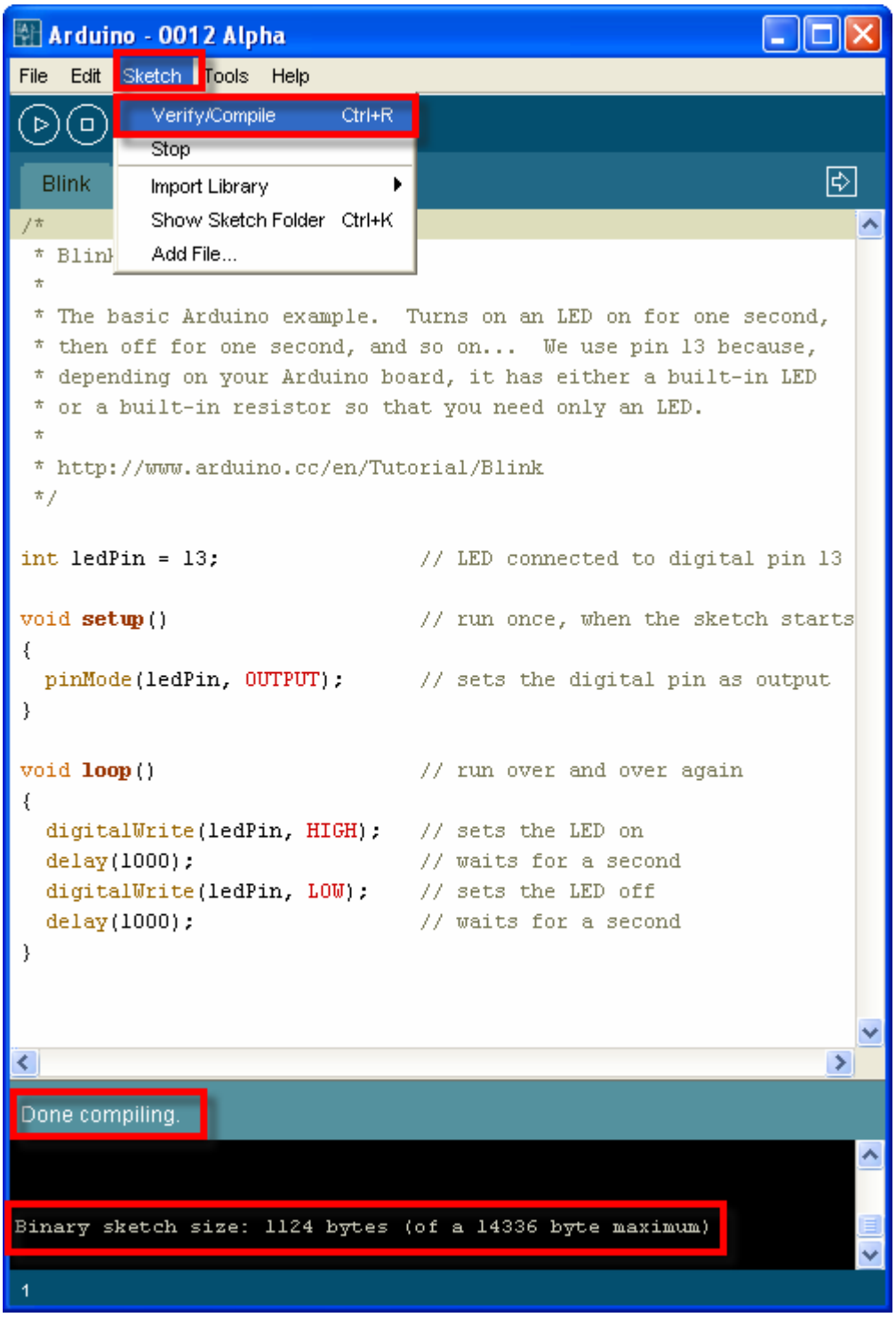


4. ทดสอบเขียนโปรแกรม โดยคลิกเมาส์ที่ File → New แล้วพิมพ์โปรแกรมทดสอบ หรืออาจใช้การสั่งเปิดไฟล์ตัวอย่างที่สร้างไว้แล้วขึ้นมาแทนก็ได้ โดยในที่นี้ขอแนะนำให้ทดสอบด้วยโปรแกรมไฟกระพริบ โดยให้เลือก “File → sketchbook → Examples → Digital → Blink” ซึ่งจะได้ดังรูป

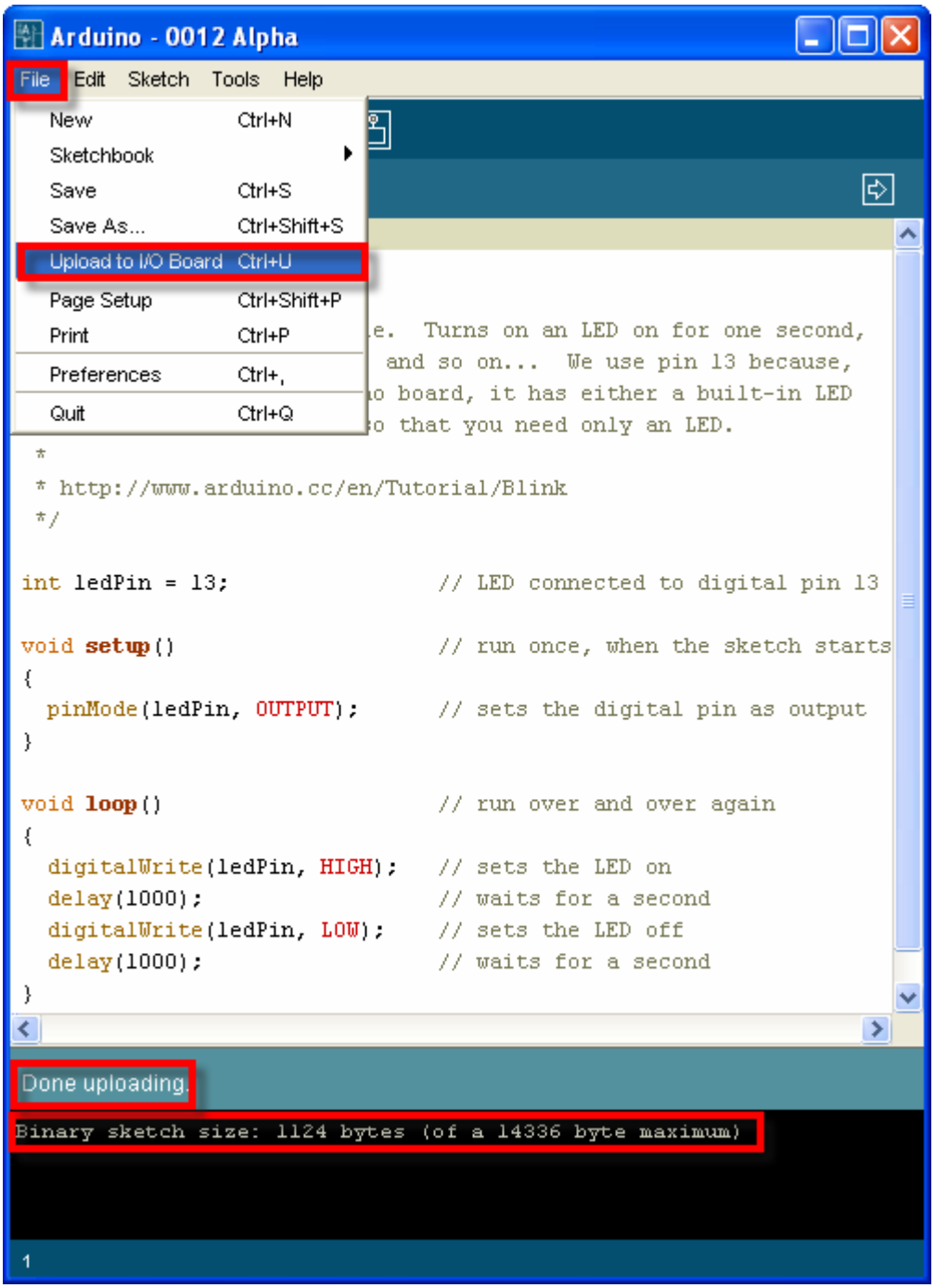



```
int ledPin = 13;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

5. สั่งแปลโปรแกรมโดยคลิกเมาส์ที่ “Sketch → Verify/Compile” ดังตัวอย่าง



6. สั่ง Download Code ให้กับบอร์ด โดยคลิกเมาส์เลือกที่ “File → Upload to I/O Board” แล้วรอสักครู่จนโปรแกรมทำงานเสร็จ ซึ่งควรได้ผลดังรูป

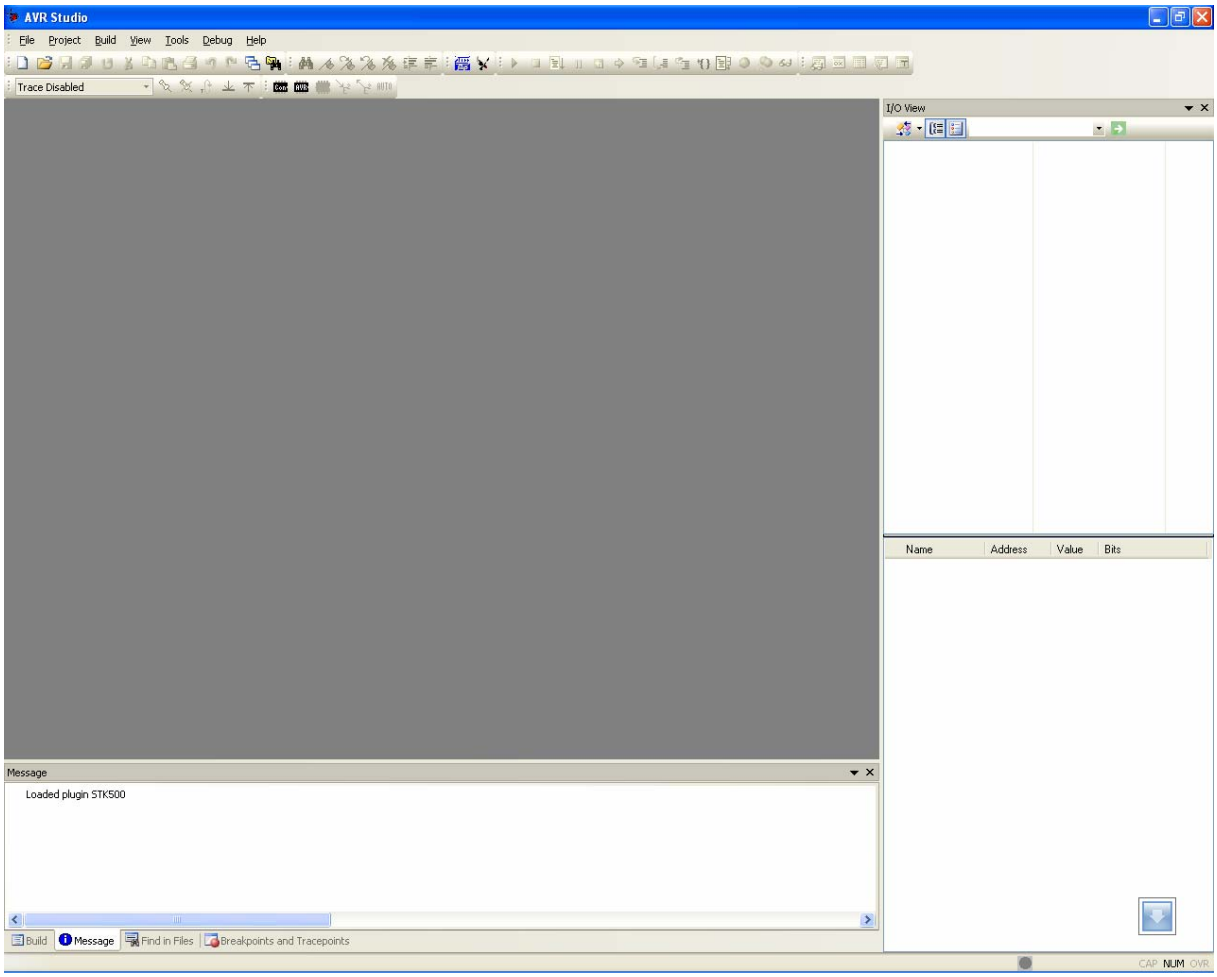


7. หลังจากที่ทำการ Upload Code ให้กับบอร์ดเป็นที่เรียบร้อยแล้ว บอร์ดก็จะเริ่มต้นทำงานตามคำสั่งที่เขียนไว้ในโปรแกรมทันที โดยจะสังเกตเห็น LED กระพริบ ติด และ ดับ สลับกันไปมา ด้วยความเร็วประมาณ 1 วินาที ตลอดเวลา

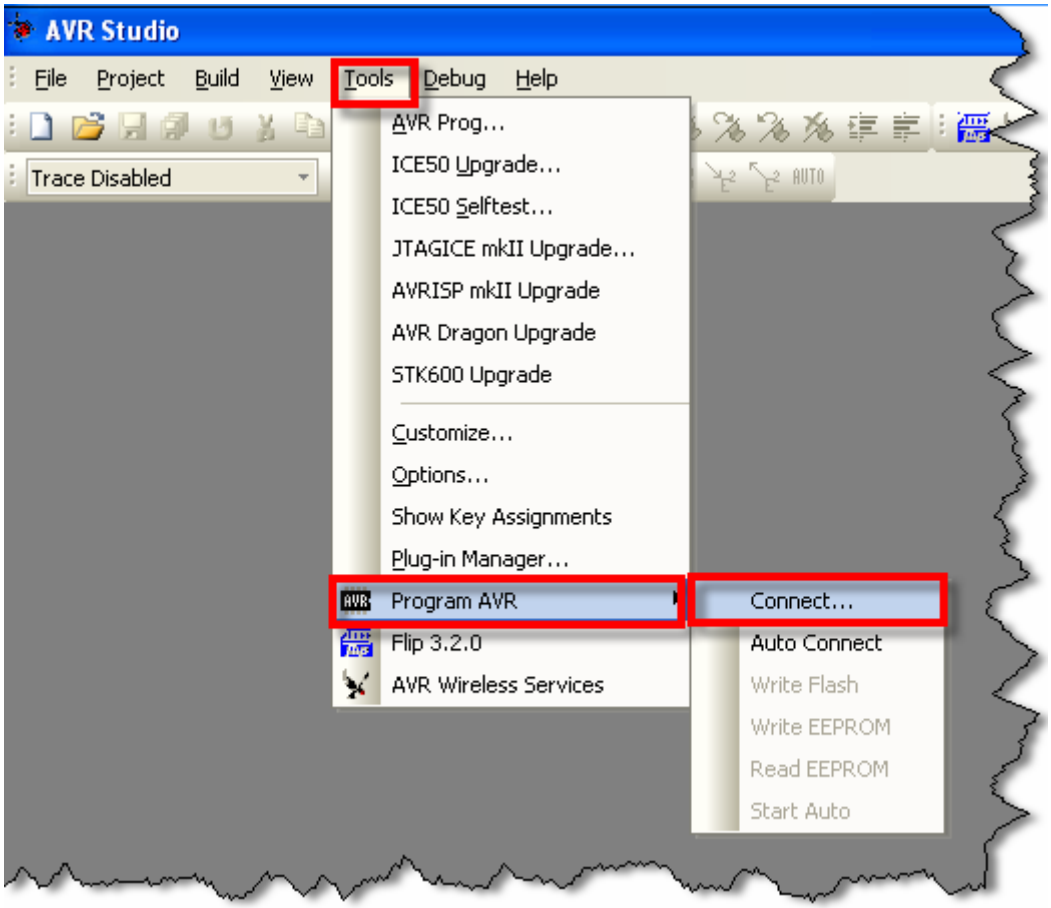
การโปรแกรม Bootloader ให้กับบอร์ด ET-BASE AVR EASY88

ตามปรกติแล้วบอร์ด ET-BASE AVR EASY88 ได้ทำการโปรแกรม Bootloader ไว้ให้เป็นที่เรียบร้อยแล้ว ผู้ใช้สามารถใช้งานได้ทันที แต่อย่างไรก็ตามในกรณีที่ต้องการเปลี่ยนแปลง Bootloader หรือเกิดความผิดพลาดในการใช้งาน จนทำให้ Bootloader เสียหายไป ผู้ใช้ก็ยังสามารถทำการ โปรแกรม Bootloader ให้กับบอร์ดได้ใหม่ โดยบอร์ด ET-BASE AVR EASY88 ได้ออกแบบ ให้มีขั้วต่อ AVRISP สำหรับใช้เป็นช่องทางในการโปรแกรม Code ให้กับ MCU ได้โดยตรง ด้วยเครื่องโปรแกรมทุกรุ่นที่มีขั้วต่อตรงตามมาตรฐาน AVRISP ของ ATMEL ได้ทันที โดยในที่นี้จะขอแสดงตัวอย่าง วิธีการ โปรแกรม Bootloader โดยใช้เครื่องโปรแกรม ของ อีทีที รุ่น “ET-AVR ISP USB V1.0” โดยใช้โปรแกรม “AVR Studio 4” ของ ATMEL เป็นตัวจัดการ ซึ่งมีลำดับขั้นตอนดังต่อไปนี้

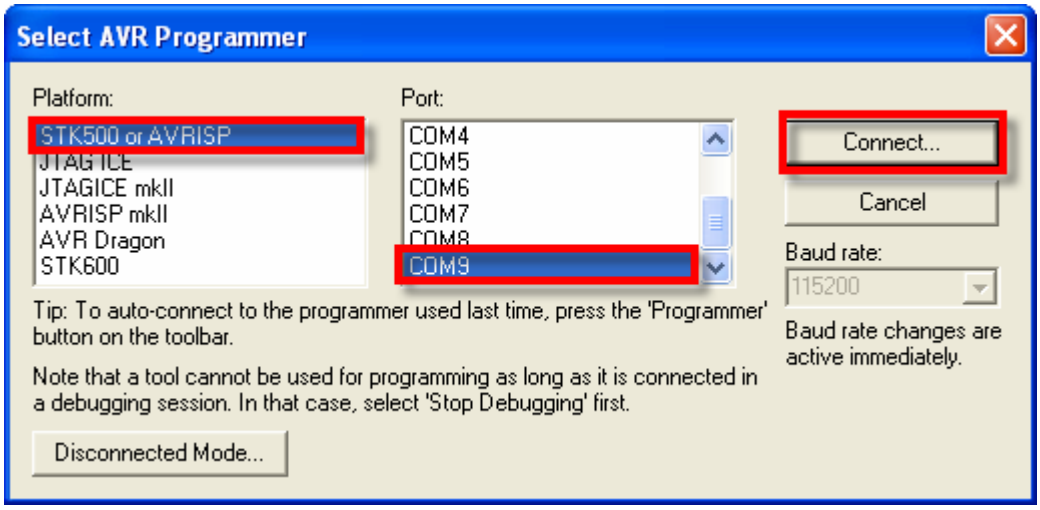
1. ต่อสาย RS232 จากคอมพิวเตอร์ ให้กับบอร์ด ET-BASE AVR EASY88 และจ่ายไฟให้บอร์ด
2. ต่อสาย USB ให้กับเครื่องโปรแกรม ET-AVR ISP USB V1 พร้อมทั้งต่อสายแพร์ 10 Pin ระหว่างขั้วต่อของ AVRISP ของทั้ง 2 บอร์ดเข้าด้วยกัน
3. สั่ง Run โปรแกรม AVR Studio 4 ดังรูป



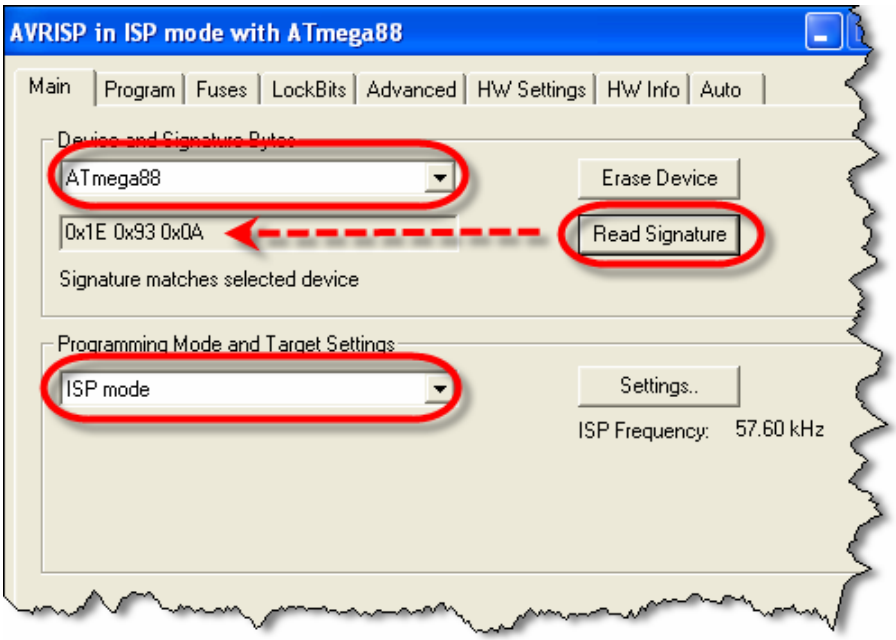
4. เลือกคลิกเมาส์ที่ Tools → Program AVR → Connect.. → STK500 or AVRISP จากนั้นก็ให้เลือกหมายเลขของ Comport ที่เป็นของเครื่องโปรแกรม ET-AVR ISP USB V1 ตามที่ลง Driver ไว้ พร้อมกับเลือก Connect (จากตัวอย่างเป็น Com9) ดังรูป



↓

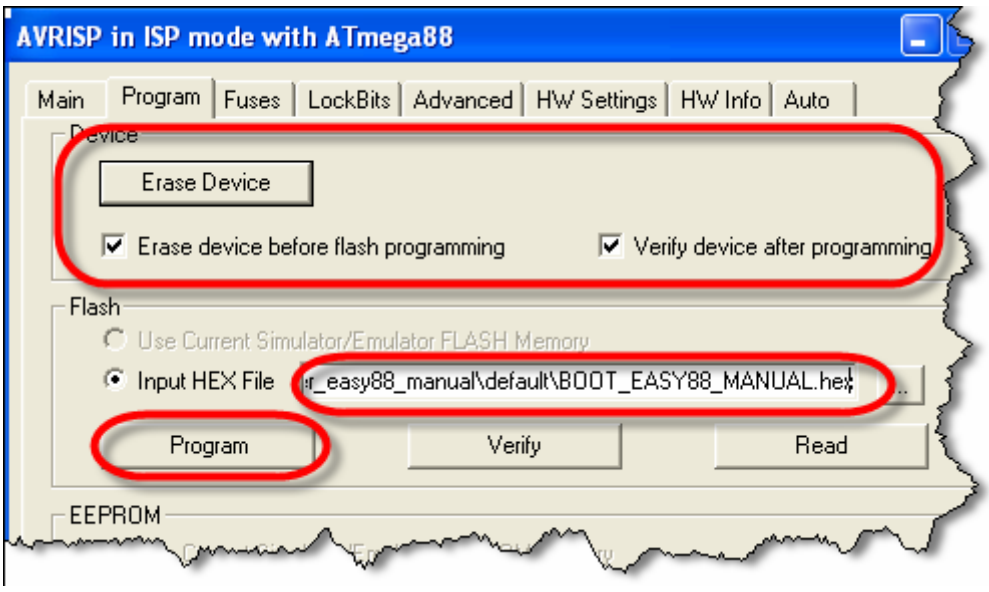


5. หลังจากที่ทำการ Connect เรียบร้อยแล้วให้ทดสอบการเชื่อมต่อโดยเลือกที่ tab ของ Main พร้อมกับเลือกเบอร์ของ MCU เป็น ATmega88 และเลือก Programming Mode and target Settings เป็น ISP mode แล้วทดสอบเลือก Read Signature ดู ซึ่งถ้าทุกอย่างถูกต้องควรได้ผลดังรูป



6. ให้เลือกไปที่ tab ของ Program พร้อมทั้งเลือก ตัวเลือกต่างๆดังนี้

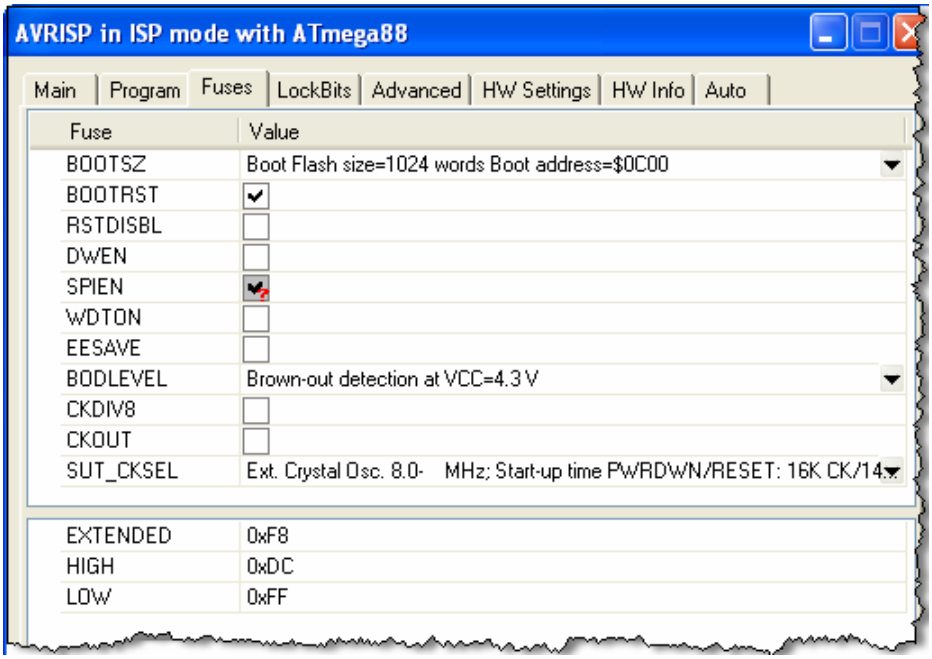
- Device ให้เลือก Erase device before flash programming และ Verify device after programming
- Flash ให้เลือก Input HEX File เป็น BOOT_EASY88_MANUAL_19_6608MHZ.HEX จากนั้นให้เลือกที่ Program เพื่อสั่ง Program Bootloader ให้กับ MCU ซึ่งจะได้ผลดังรูป



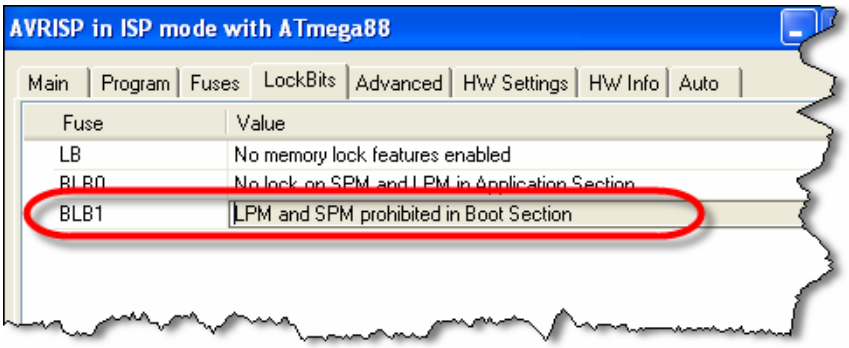
7. หลังจากที่ทำการโปรแกรม Code ให้กับ MCU เรียบร้อยแล้ว ให้เลือกมาที่ Tab ของ Fuses เพื่อทำการสั่งโปรแกรม Fuse Bit ให้กับ MCU โดยให้เลือกกำหนดค่าตัวเลือกดังนี้

- BOOTSZ ให้เลือกเป็น Boot Flash size = 1024 word start address = $0C00
- BOOTRST ให้เลือก Enable
- SPIEN ให้เลือก Enable
- SUT_CKSEL ให้เลือกเป็น Ext.Crystal Osc 8.0MHz;Start-up time PWRDN/RESET : 16K CK/14 ซึ่งเป็นตัวเลือกด้านล่างสุด

เมื่อเลือกตัวเลือกต่างๆครบแล้วจึงสั่ง Program ซึ่งควรได้ผลดังรูป



8. หลังจากทำการสั่ง program Fuse Bit เรียบร้อยแล้ว ให้เลือก Tab มาที่ LockBits แล้วเลือกการ Protect เฉพาะ Bootloader โดยเลือก BLB1 เป็น LPM and SPM prohibited in Boot Section ซึ่งจะได้ค่า Lock Bit เป็น 0xCF แล้วสั่ง program เป็นอันเสร็จสิ้นกระบวนการ Program Bootloader



การพัฒนาโปรแกรมของ ET-BASE AVR EASY88 แบบ AVR Microcontroller

ในกรณีที่ผู้ใช้ต้องการที่จะพัฒนาโปรแกรมให้กับบอร์ด ET-BASE AVR EASY88 แบบปรกติ ในรูปแบบของไมโครคอนโทรลเลอร์ AVR ก็สามารถทำได้ โดยสามารถเลือกภาษาในการเขียนโปรแกรม ได้เอง ซึ่งในกรณีนี้ผู้ใช้สามารถเลือกใช้ภาษาใดๆก็ได้ที่รองรับการใช้งานกับ MCU ตระกูล AVR เบอร์ ATmega88 โดย ผู้ใช้งานสามารถบริหารจัดการระบบทรัพยากรต่างๆที่อยู่ภายในตัว ATmega88 ได้เองทั้งหมด โดยสามารถทำได้ 2 แนวทางคือ

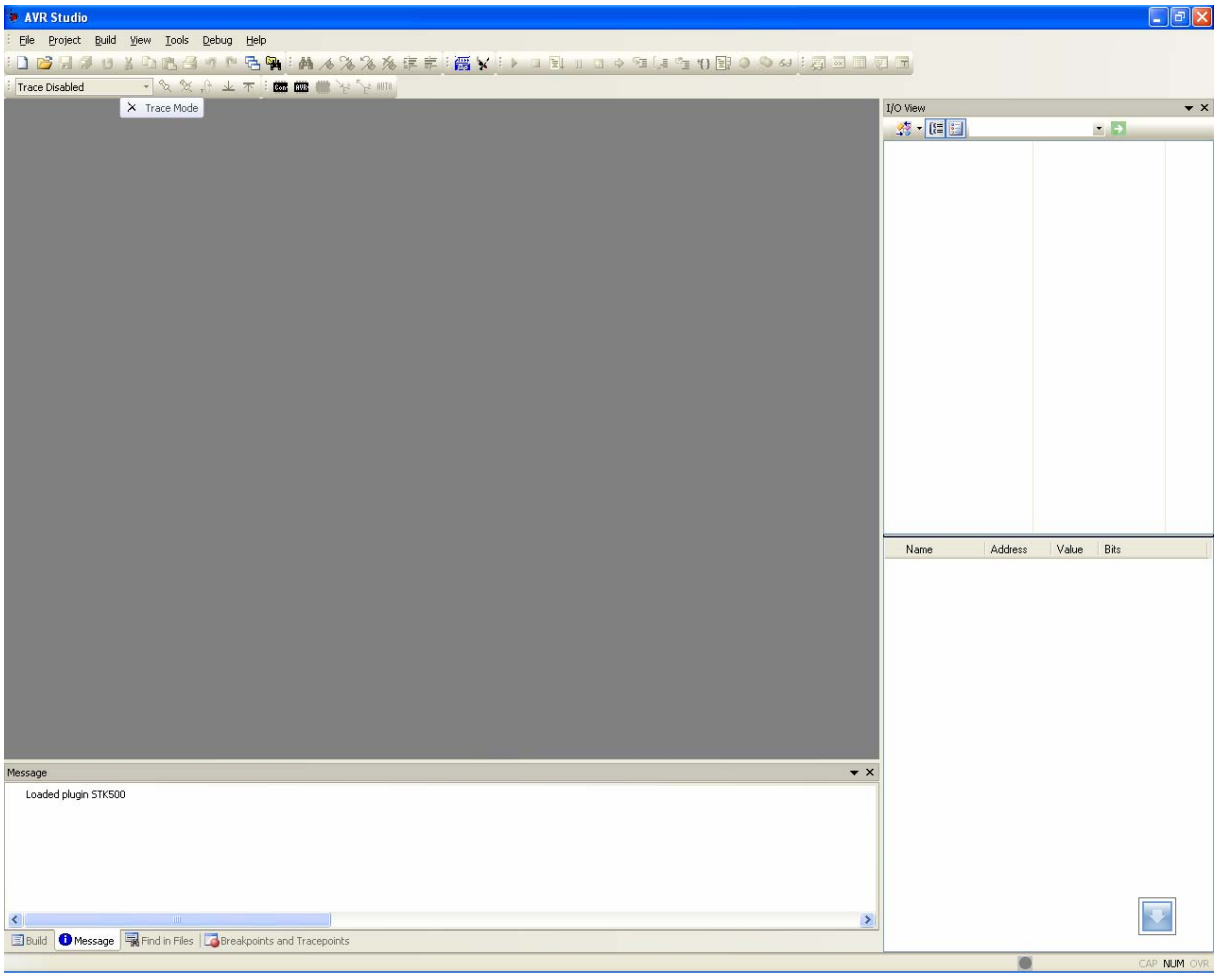
- **การพัฒนาโปรแกรมโดยใช้เครื่องโปรแกรมภายนอก** ในกรณีนี้จะมีข้อดีคือ ไม่สูญเสียทรัพยากรใดๆเลย ผู้ใช้สามารถใช้งานและกำหนดคุณสมบัติการทำงานของทรัพยากรต่างๆที่มีอยู่ใน MCU ได้เอง ตามต้องการ แต่มีข้อเสียคือ ต้องมีเครื่องโปรแกรมภายนอก สำหรับทำหน้าที่โปรแกรม Code ให้กับ MCU โดยสามารถใช้ได้กับเครื่องโปรแกรมทุกรุ่นที่ รองรับการใช้งานกับ MCU เบอร์ ATmega88 และมีขั้วต่อตามมาตรฐาน AVRISP ของ ATMEL

- **การพัฒนาโปรแกรมโดยใช้งานร่วมกับ Bootloader** ในกรณีนี้จะมีข้อดี คือ เมื่อทำการเขียนโปรแกรมเสร็จแล้วสามารถสั่ง Program Code ให้กับ MCU ผ่านทางโปรแกรม Bootloader ที่ติดตั้งไว้แล้วได้ทันที โดยไม่ต้องใช้เครื่องโปรแกรมภายนอก แต่มีข้อจำกัด คือ ต้องเสียพื้นที่หน่วยความจำ Flash สำหรับเก็บ Code โปรแกรมไป 2KByte (0x1800 -0x1FFF) เพื่อใช้ติดตั้งโปรแกรม Bootloader ซึ่งตามปรกติแล้วบอร์ด ET-BASE AVR EASY88 จะทำการ ติดตั้งโปรแกรม Bootloader ไปให้เรียบร้อยแล้ว โดยในกรณีนี้ผู้ใช้จะสามารถเขียนโปรแกรมใช้งานได้ทั้งหมดจำนวน 6 Kbyte จากที่มีอยู่ทั้งหมด 8Kbyte โดยต้องเขียนโปรแกรมให้มีตำแหน่งการทำงานของ Code อยู่ระหว่างตำแหน่งแอดเดรส 0x0000 ถึง 0x17FF (0x0C00 - 0x0FFF K Word) ส่วน SRAM,EEPROM และทรัพยากรอื่นๆที่มีอยู่ใน MCU ผู้ใช้ยังสามารถใช้งานได้ครบตามจำนวนที่มีอยู่ในตัว MCU ทุกประการ

ตัวอย่างการพัฒนาโปรแกรมด้วย WinAVR โดยใช้งานร่วมกับ Bootloader

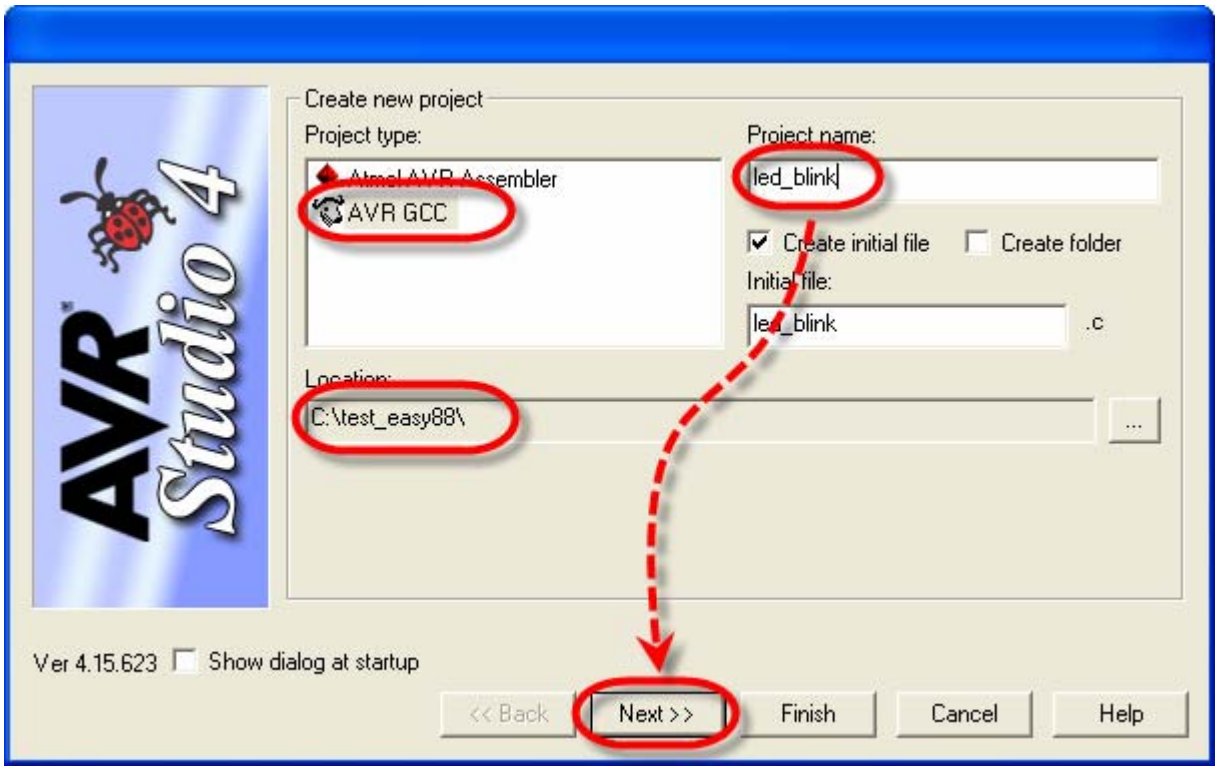
ตัวอย่างที่จะแสดงต่อไปนี้จะแสดงให้เห็นวิธีการพัฒนาโปรแกรมให้กับ ATmega88 ด้วยภาษาซี โดยใช้โปรแกรม AVR Studio4 ของ ATMEL ร่วมกับ Compiler ภาษาซีของ WinAVR

1. สั่ง Run Program AVR Studio4

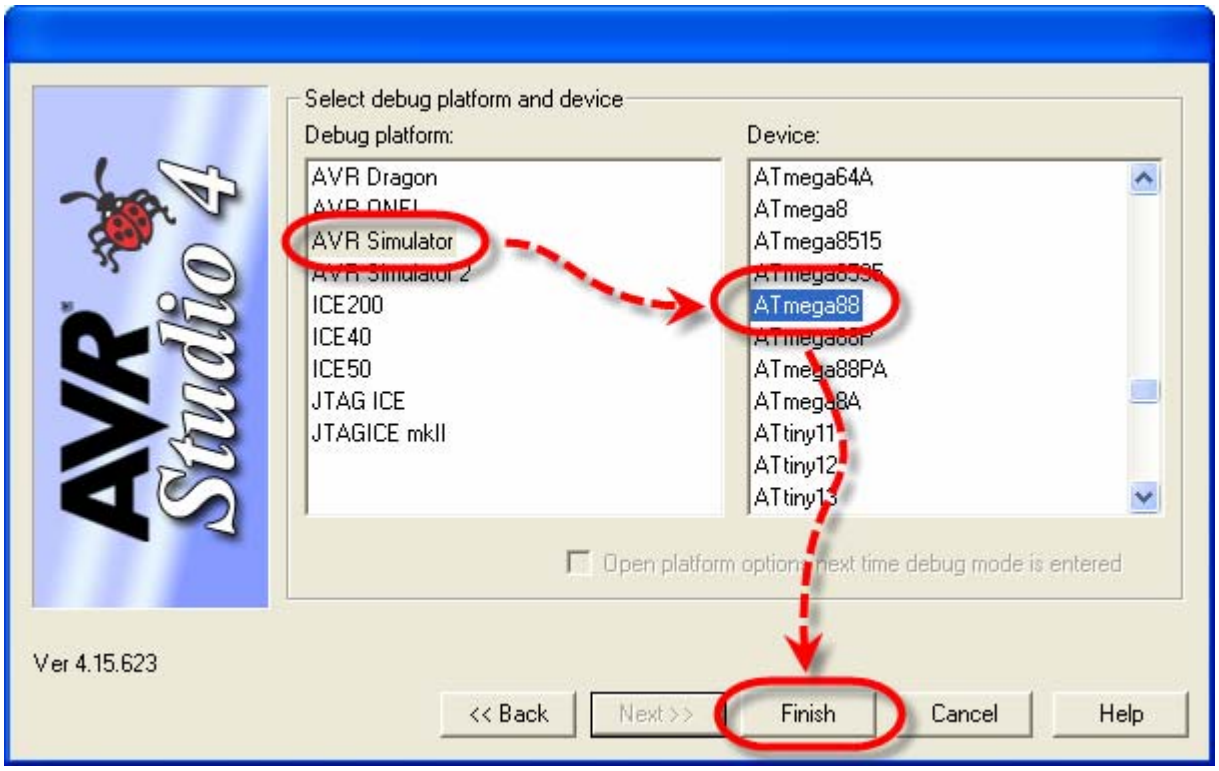


2. สั่งสร้าง project ใหม่ โดยเลือกที่ project → New project จากนั้นเลือกกำหนดตัวเลือกต่างๆ ให้กับโปรแกรมดังนี้
   - Project type เลือกกำหนดเป็น AVR GCC
   - Location สำหรับบันทึก project ให้ระบุตำแหน่ง Folder ที่ต้องการใช้บันทึกไฟล์ และ Code ต่างของ project ในตัวอย่างกำหนดไว้ที่ “C:\test_easy88\”
   - Poject name ให้กำหนดชื่อ project ตามต้องการในตัวอย่างกำหนดเป็น “led_blink” และให้เลือก Create initial file ไว้ด้วย ซึ่งเมื่อเราทำการกำหนดชื่อ project name เสร็จแล้ว โปรแกรมจะสร้างไฟล์ ที่มีชื่อเดียวกันกับ project name ให้เองโดยอัตโนมัติ

3. เมื่อกำหนดค่าตัวเลือกต่างๆ ให้กับโปรแกรมเรียบร้อยแล้ว ให้เลือกที่ Next แล้วกำหนดค่าใน Debug platform เป็น AVR Simulator และเลือก Device เป็น ATmega88 ดังรูป

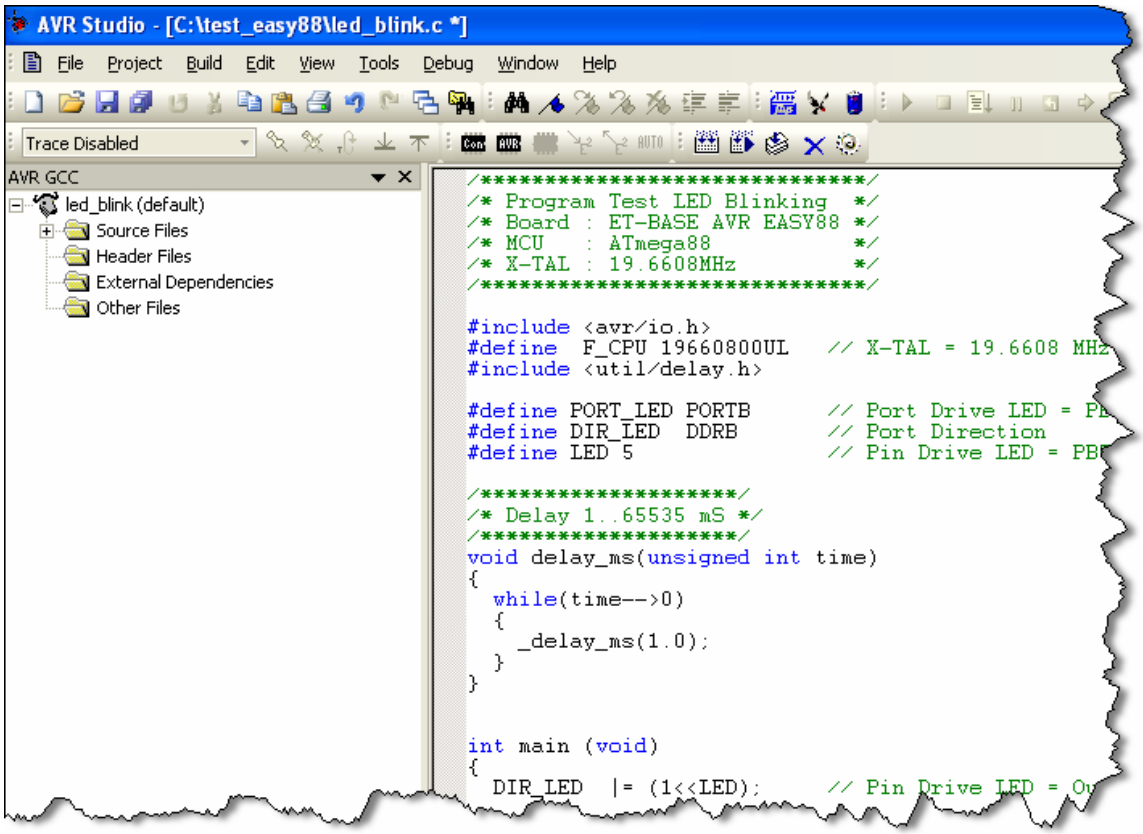


↓

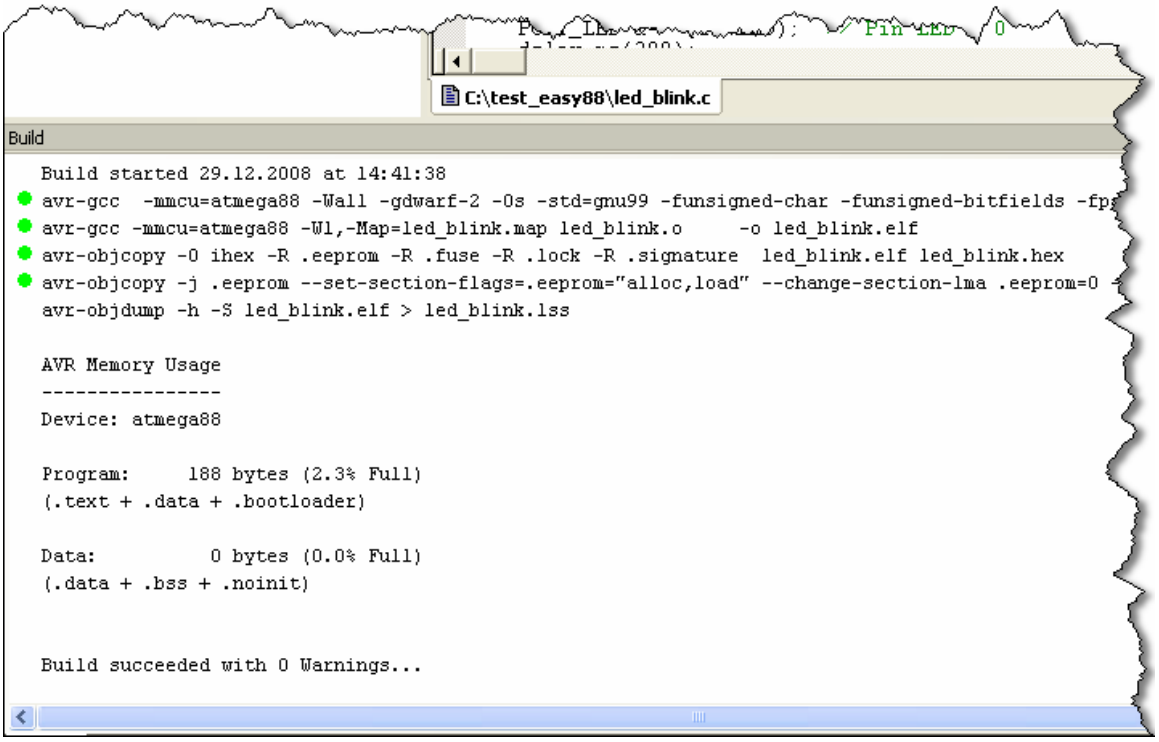


4. ให้พิมพ์คำสั่งของโปรแกรมสำหรับทดสอบการทำงาน ในหน้าต่าง Text Editor ของโปรแกรม โดยในที่นี้จะทดสอบด้วย Codeโปรแกรม สำหรับทำหน้าที่เป็นไฟกระพริบที่ PB5 ซึ่งเป็น LED ที่ติดตั้งไว้บนบอร์ด ET-BASE AVR EASY88 อยู่แล้ว ดังตัวอย่าง

```
/******************************/
/* Program Test LED Blinking  */
/* Board : ET-BASE AVR EASY88 */
/* MCU   : ATmega88           */
/* X-TAL : 19.6608MHz         */
/******************************/

#include <avr/io.h>
#define  F_CPU 19660800UL     // X-TAL = 19.6608 MHz
#include <util/delay.h>

#define PORT_LED PORTB        // Port Drive LED = PB
#define DIR_LED  DDRB         // Port Direction
#define LED 5                 // Pin Drive LED = PB5

/********************/
/* Delay 1..65535 mS */
/********************/
void delay_ms(unsigned int time)
{
  while(time-->0)
  {
    _delay_ms(1.0);
  }
}


int main (void)
{
  DIR_LED  |= (1<<LED);       // Pin Drive LED = Out

  while(1)
  {
    PORT_LED &= ~(1<<LED);    // Pin LED = 0
     delay_ms(200);

    PORT_LED |=  (1<<LED);    // Pin LED = 1
     delay_ms(200);
  }
}
```



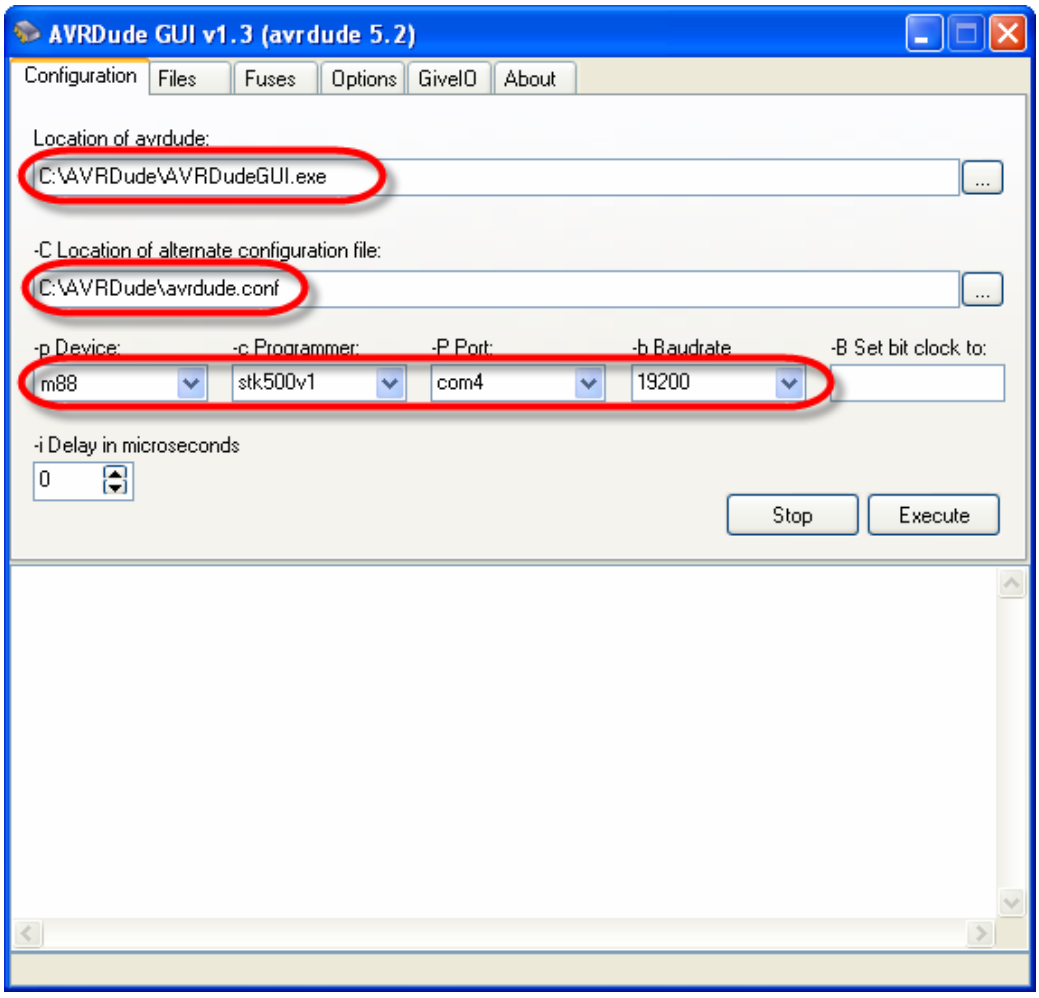
5. หลังจากพิมพ์ Code โปรแกรมเสร็จแล้วให้สั่งแปลโปรแกรม โดยเลือกที่ build →rebuild all ซึ่งผลจากการแปลคำสั่งจะได้เป็น HEX File ที่มีชื่อเดียวกันกับ project ที่สร้างไว้ ดังรูป



การสั่งโปรแกรม Hex File ให้กับบอร์ด ET-BASE AVR EASY88 ผ่าน Bootloader

ในการพัฒนาโปรแกรมของ บอร์ด ET-BASE AVR EASY88 ในรูปแบบของ AVR Microcontroller นั้น หลังจากที่ทำการเขียนโปรแกรม และสั่งแปลคำสั่งจนได้ HEX File เรียบร้อยแล้วในการสั่ง Program HEX Code ที่ได้จากการแปลคำสั่งของโปรแกรมนี้ ผ่านทางโปรแกรม Bootloader ของบอร์ดนั้น ขอแนะนำให้ใช้โปรแกรม AVRDude โดยใช้งานผ่าน AVRDudeGUI ซึ่งมีขั้นตอนดังนี้

1. ทำการสร้างโฟลเดอร์ชื่อ AVRDUDE ไว้ใน Drive C (“C:\AVRDUDE\”) จากนั้นให้ทำการ Copy โปรแกรมของ avrdude และ avrdudeGUI ไว้ใน “C:\AVRDUDE\” โดยจะมีทั้งหมด 3 ไฟล์ด้วยกัน คือ avrdude.exe, avrdude.conf และ avrdudegui.exe
2. สั่ง Run Program AVRDude.EXE แล้วเลือกที่ Tab ของ Configuration ให้เลือกกำหนดค่าต่างๆ ดังต่อไปนี้

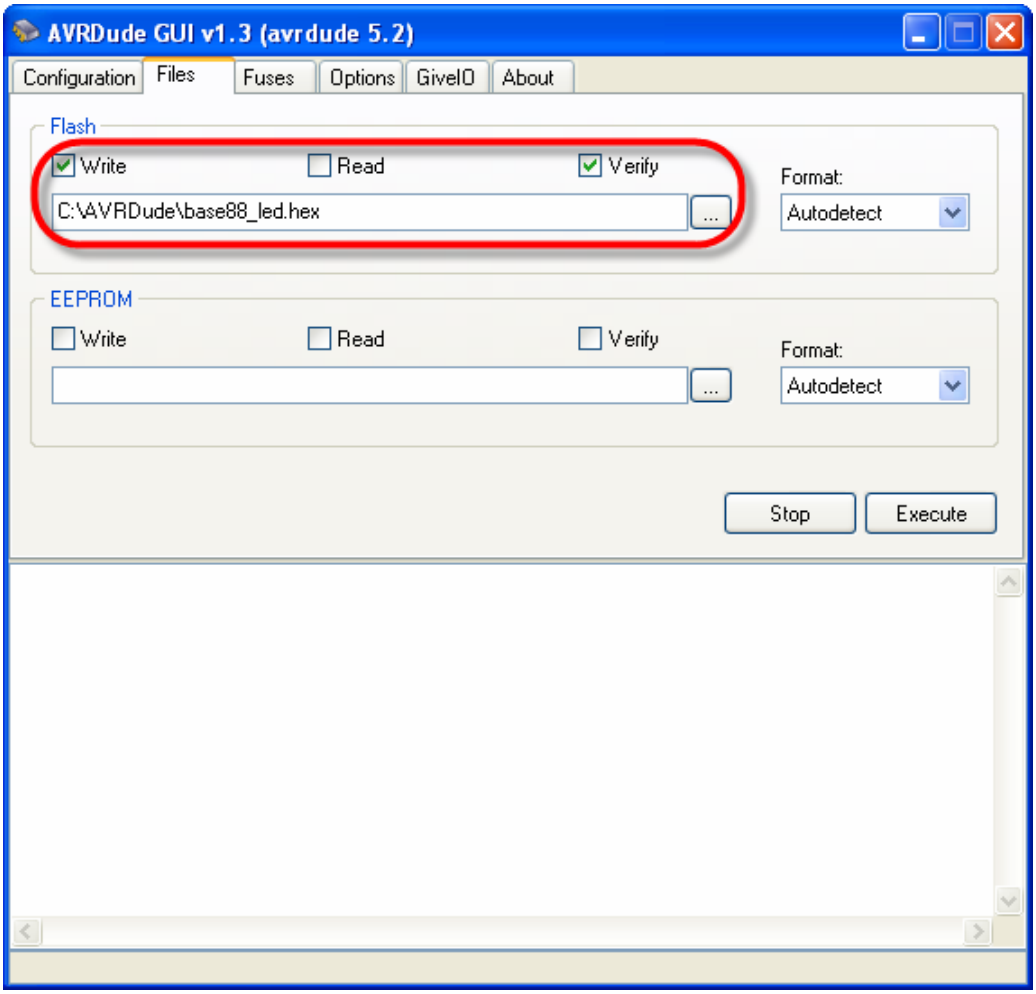


หมายเหตุ ถ้าหมายเลข Comport มากกว่า com8 ให้พิมพ์หมายเลข Comport ลงไปในช่องรับค่าของ –P Port ได้เอง เช่น com9 แต่ถ้าหมายเลข Comport มีค่าเป็น 2 หลักให้ เพิ่มเครื่องหมาย “\\.\” นำหน้าชื่อหมายเลข Comport ลงไปด้วย เช่น ถ้าใช้กับ Com13 ก็ให้กำหนดเป็น “\\.\com13” เป็นต้น

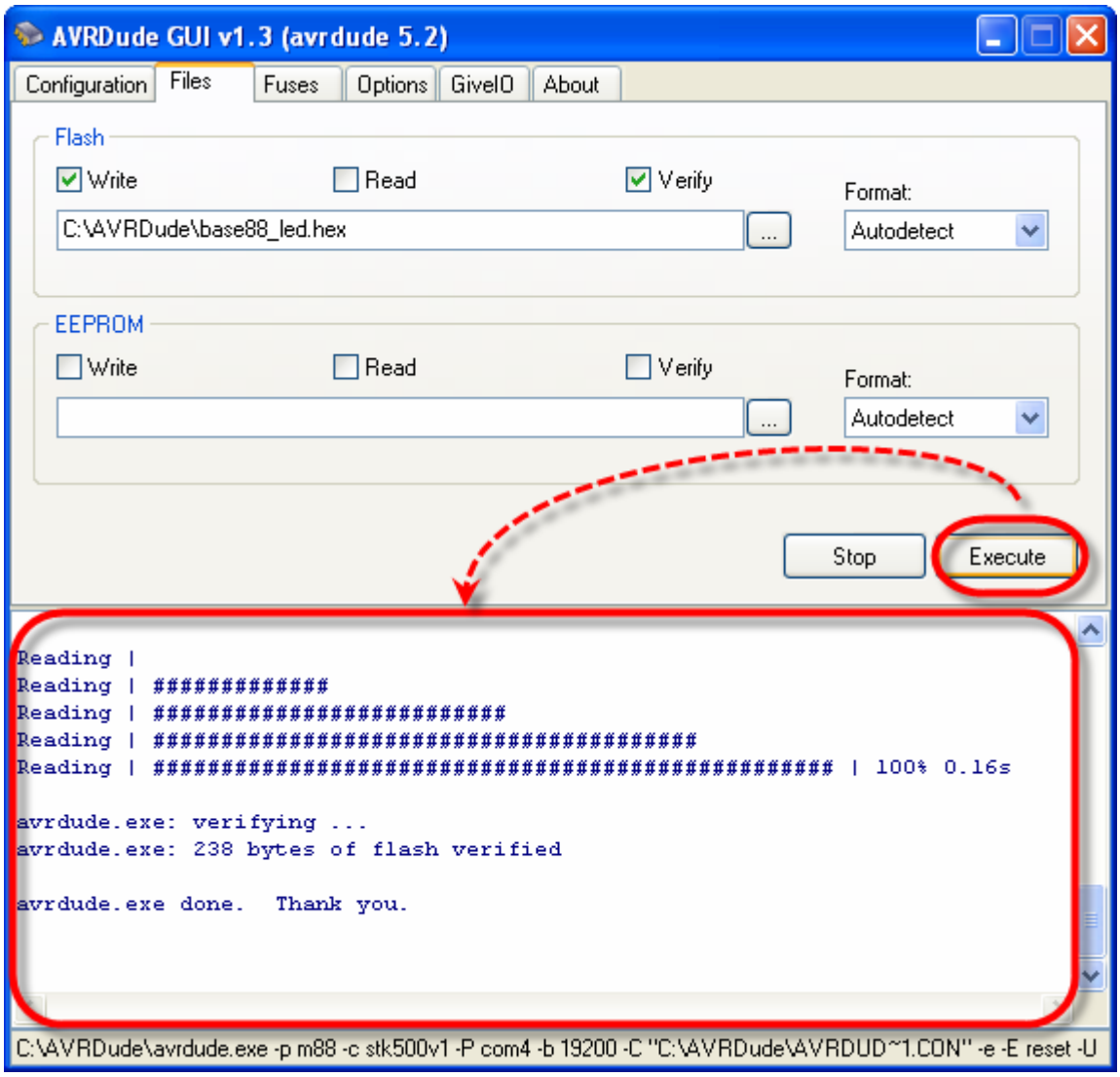
ให้ทำการกำหนดค่า Configuration ให้กับโปรแกรม AVRDudeGUI ดังนี้

- Location of avrdude ให้เลือกไปที่ชื่อและที่อยู่ของไฟล์ avrdude.exe ที่ได้ติดตั้งไว้ตามขั้นตอนที่ 1 ซึ่งก็คือ “C:\AVRDude\avrdude.exe”
- -C Location of alternate configuration file ให้เลือกไปที่ชื่อและที่อยู่ของไฟล์ avrdude.conf ที่ได้ติดตั้งไว้ตามขั้นตอนที่ 1 ซึ่งก็คือ “C:\AVRDude\avrdude.conf”
- -p Device ให้เลือกเป็น m88
- -c Programmer ให้เลือกเป็น stk500
- -p Port ให้เลือกเป็นหมายเลข Comport ตามที่ต่อไว้จริงกับคอมพิวเตอร์ PC
- -b Baudrate ให้เลือกกำหนดเป็น 19200

3. เปลี่ยน Tab ไปที่ File แล้วเลือกกำหนดส่วนของ Flash โดยให้เลือก Write และ Verify พร้อมทำกำหนดชื่อ HEX File ไฟล์ ที่ต้องการจะโปรแกรม ซึ่งในที่นี้จะใช้ HEX File ที่ได้จากตัวอย่างที่เขียนด้วยภาษาซีของ WinAVR ซึ่งจะอยู่ใน “C:\AVRDude\base88_led.hex” ดังรูป



4. เมื่อกำหนดค่าต่างๆให้กับโปรแกรมเรียบร้อยแล้ว ให้เลือกที่ Execute ที่อยู่ใน Tab ของ Files ซึ่งโปรแกรม avrdude จะเริ่มต้นทำการ โปรแกรม HEX File ให้กับบอร์ดทันที ให้รอจนเสร็จดังรูป



5. หลังจากสั่งโปแกรมเสร็จให้กดสวิตช์ RESETบอร์ดจะเริ่มต้นทำงานทันที โดยจะเห็น LED กระพริบติด และ ดับ สลับกันไปมาไม่รู้จบ