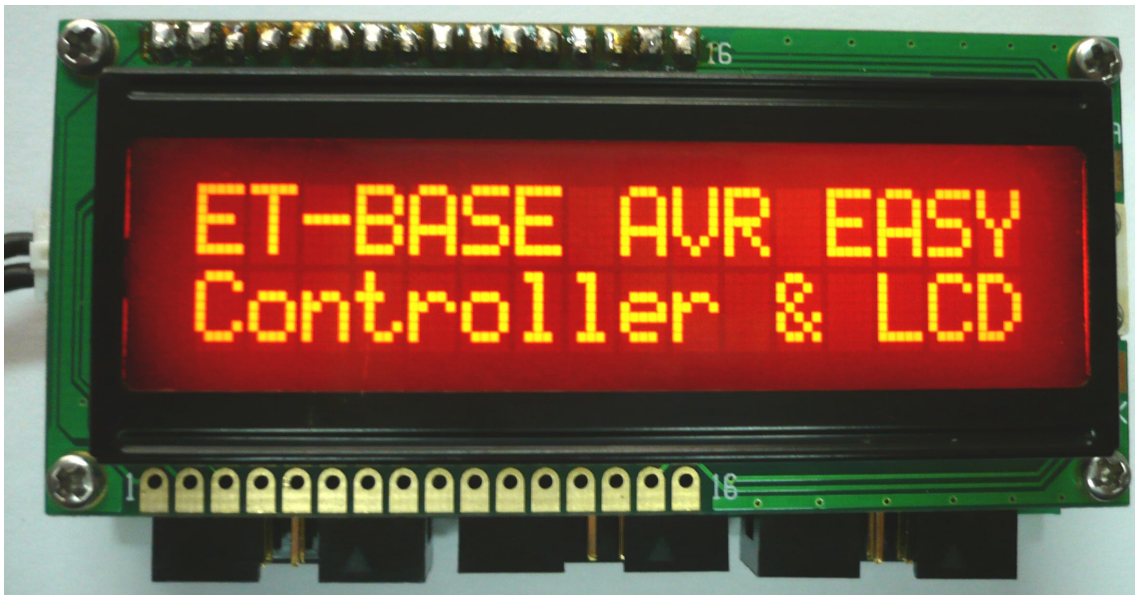


ET-EASY AVR LCD



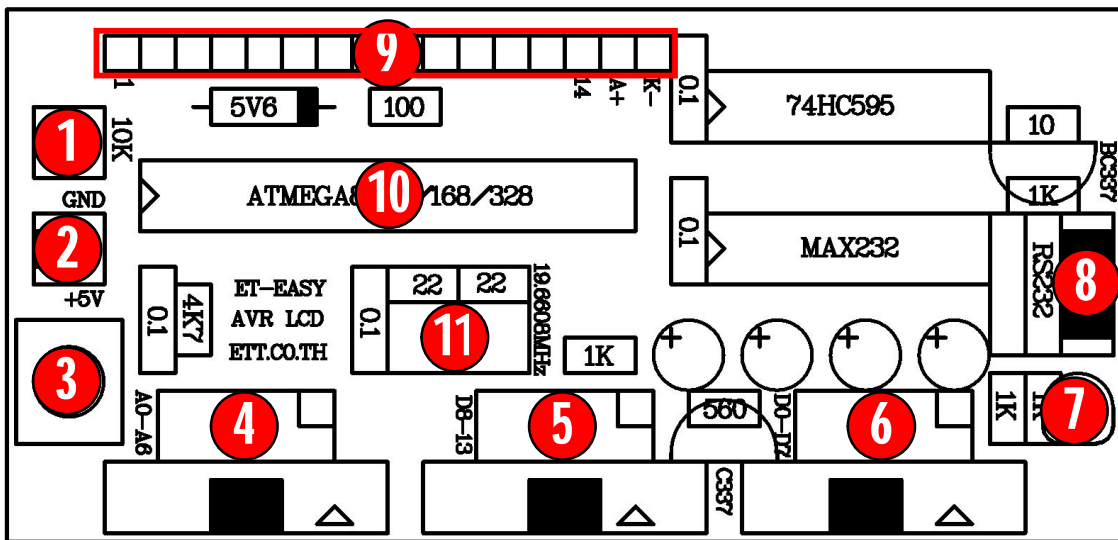
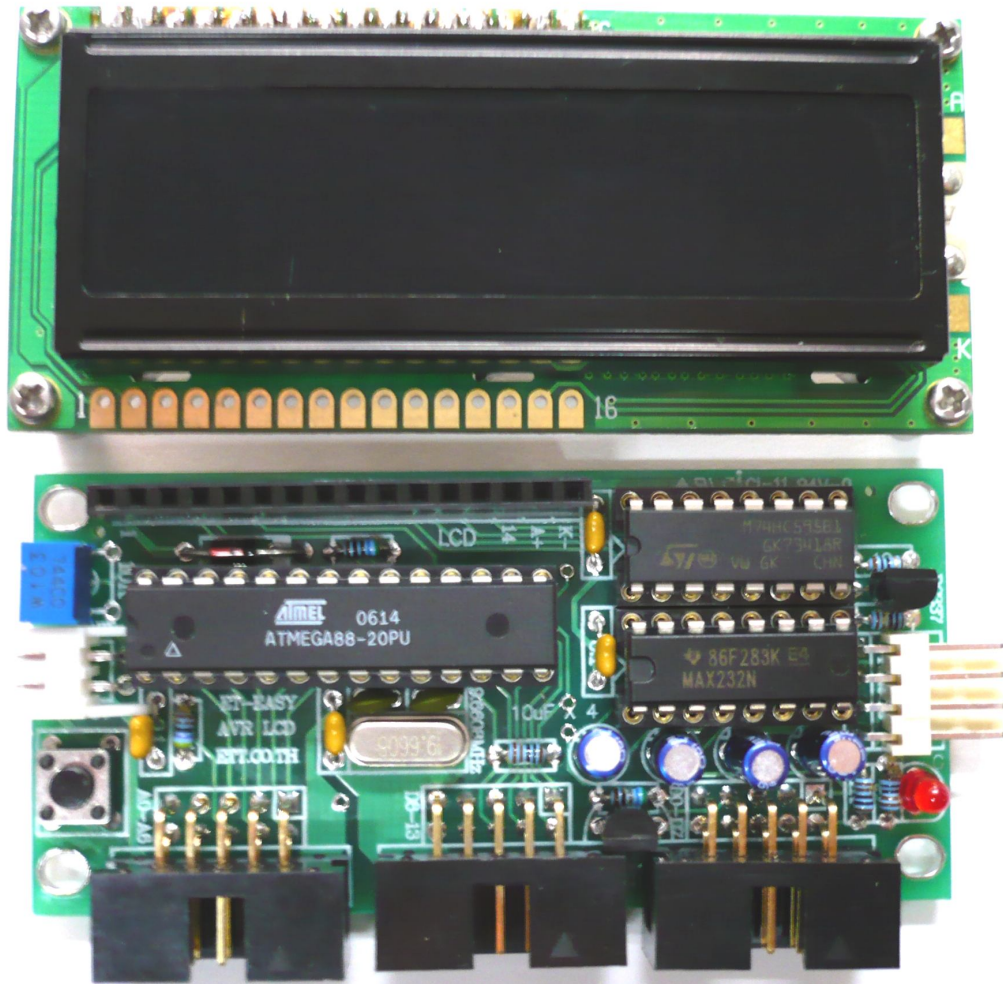
ET-EASY AVR LCD เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล AVR โดยรวมจอแสดงผล LCD แบบ Character ขนาด 16 ตัวอักษร 2 บรรทัดและวงจรถบคุม ติดตั้งพร้อมใช้งานภายในบอร์ด โดยบอร์ดมีขนาดเท่ากับ ขนาดของจอแสดงผล LCD พร้อมขั้วต่อ I/O ขนาด 10 Pin จำนวน 3 ชุด โดยบอร์ดเลือกใช้ไมโครคอนโทรลเลอร์ ตระกูล AVR เบอร์ ATMEGA88 ของ ATMEL เป็น MCU ประจำบอร์ด โดย MCU รุ่นนี้จะบรรจุอยู่ในตัวถังแบบ 28 Pin DIP โดย MCU ตัวนี้จะมีจุดเด่น คือ เป็นไมโครคอนโทรลเลอร์ขนาดเล็กแต่เพียงพร้อมไปด้วยทรัพยากรพื้นฐานต่างๆอย่างครบถ้วน เหมาะแก่การนำไปประยุกต์ใช้งานต่างๆได้โดยง่าย ซึ่ง MCU ภายในบอร์ดทำงานได้ด้วยความเร็ว 19.6608 MHz นอกจากนี้แล้วยังมีความเพียงพร้อมด้วยอุปกรณ์พื้นฐานต่างๆที่จำเป็นต่อการใช้งานไม่ว่าจะเป็นหน่วยความจำสำหรับเก็บข้อมูลแบบ E²PROM ขนาด 512 Byte และหน่วยความจำใช้งานแบบ SRAM อีก 1 K Byte ในด้านของอุปกรณ์ Peripheral นั้นก็นับว่าครบถ้วน เหมาะแก่การนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมและประมวลผลต่างๆได้เป็นอย่างดี โดยจะมีทั้งระบบฮาร์ดแวร์ของ SPI, UART, I2C, Watchdog, Timer/Counter, PWM และ ADC ฯลฯ

โดยการออกแบบโครงสร้างของบอร์ดนั้น จะเน้นเรื่องขนาดของบอร์ดให้มีขนาดเล็กเพื่อให้ง่ายต่อการนำไปประยุกต์ใช้งานและสะดวกต่อการพัฒนาโปรแกรม โดยลักษณะของบอร์ดมีความเหมาะสมต่อการนำไปประยุกต์ใช้งานที่มีความจำเป็นต้องใช้จอแสดงผลแบบ LCD ในการทำงานด้วย และยังสามารถทำการรับส่งข้อมูลกับอุปกรณ์ภายนอกผ่านทางพอร์ตสื่อสารอนุกรม RS232 ได้ทันที สำหรับส่วนของ I/O นั้นก็จะใช้การเชื่อมต่อผ่านทางขั้วต่อแบบ IDE Connector ขนาด 10PIN สามารถเชื่อมต่อสัญญาณผ่านทางสายแพร์ ออกไปใช้งาน หรือ เชื่อมต่อกับบอร์ด I/O ภายนอกต่างๆได้โดยง่าย

คุณสมบัติของบอร์ด

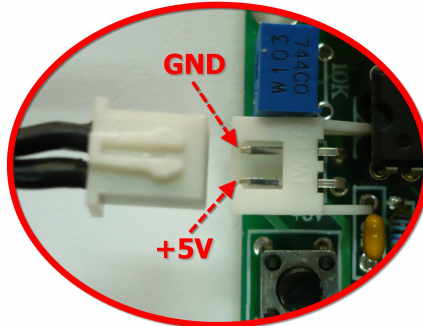
- เลือกใช้ MCU ตระกูล AVR เบอร์ ATMEGA88 ของ ATMEL เป็น MCU ประจำบอร์ด โดยเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาแบบ Crystal Oscillator ค่า 19.6608 MHz เพื่อให้สามารถใช้งานพอร์ตสื่อสารอนุกรม RS232 ได้อย่างลงตัว
- สามารถเปลี่ยนไปติดตั้งใช้งาน MCU ตระกูล AVR ขนาด 28 PIN เบอร์อื่นๆในอนุกรมเดียวกันได้ เช่น ATMEGA8,ATMEGA48,ATMEGA168 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))
 - มีวงจรถือสื่อสารอนุกรม UART จำนวน 1 พอร์ต
 - มีวงจรถือสื่อสาร SPI จำนวน 1 พอร์ต
 - มีวงจรถือสื่อสาร 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
- มีขั้วต่อสัญญาณ I/O แบบ TTL ด้วย Header 2x5 จำนวน 3 ชุด (PB,PC และ PD)
- มีสวิตช์ RESET พร้อมวงจร External Reset แบบ RC Reset ภายในบอร์ด
- บอร์ดได้รับการติดตั้งโปรแกรม Boot Loader ให้กับ MCU ด้วยแล้ว ผู้ใช้สามารถทำการพัฒนาโปรแกรมของบอร์ดด้วย Arduino หรือ ส่งโปรแกรม Hex File ให้กับบอร์ดผ่านทางโปรแกรม Boot Loader ได้ โดยการต่อสวิตช์ Boot Loader (Push Button Switch) เพิ่มเพียง 1 ตัวเท่านั้น
- มีขั้วต่อ RS232 สำหรับใช้งาน แบบ CPA-4Pin มาตรฐาน อีทีที สำหรับใช้งานในการรับส่งข้อมูลทั่วไป หรือ ใช้สำหรับโปรแกรมข้อมูลให้กับบอร์ดผ่าน Boot Loader
- มีวงจรเชื่อมต่อกับ LCD ผ่านทางไอซี 74HC595 โดยใช้ขาสัญญาณ PD4, PD7 และ PB0 ในการควบคุมสัญญาณ LCD พร้อมวงจรปรับความสว่างหน้าจอ และวงจรควบคุมไฟส่องสว่าง Black Light ให้ด้วย ทำให้สามารถประหยัด I/O ในการเชื่อมต่อกับ LCD และเหลือ I/O ไว้ใช้งานอื่นๆได้มากขึ้น
- ใช้กับแหล่งจ่ายไฟตรงขนาด +5VDC โดยต่อผ่านขั้ว Connector CPA ขนาด 2 Pin
- ขนาด PCB Size เล็กเพียง 8 cm x 3.7cm.

โครงสร้างบอร์ด ET-EASY AVR LCD

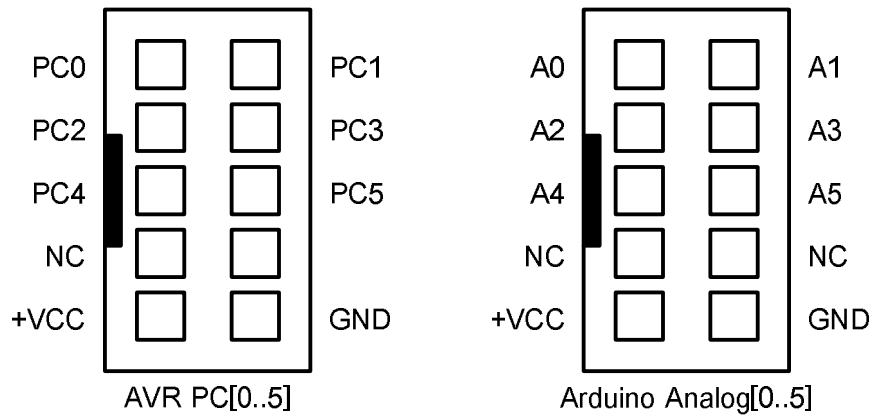


รูปแสดง โครงสร้างของบอร์ด ET-EASY AVR LCD

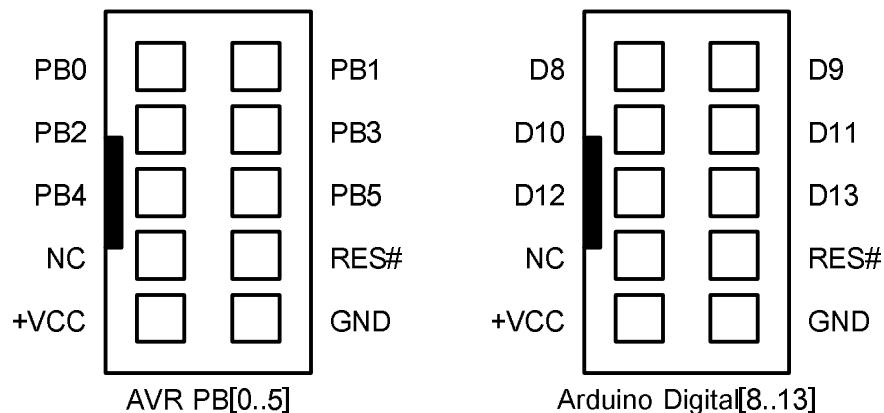
- หมายเลข 1 คือ ตัวต้านทานสำหรับปรับความสว่างของหน้าจอ LCD
- หมายเลข 2 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด ใช้กับแหล่งจ่ายไฟตรงขนาด +5VDC



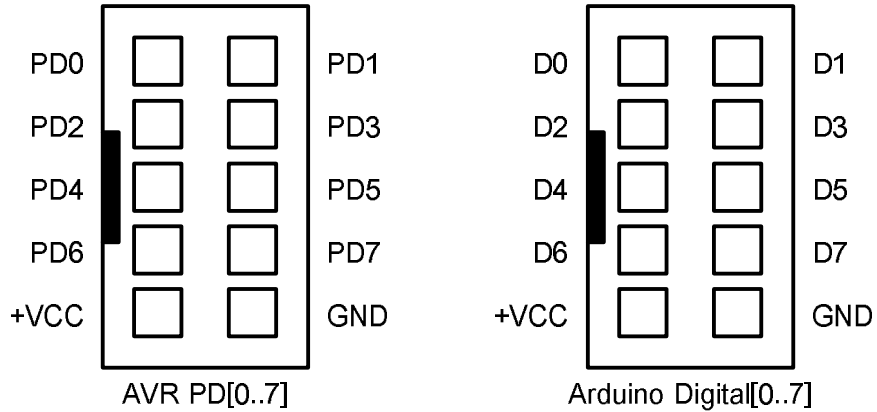
- หมายเลข 3 สวิตช์ RESET ใช้สำหรับรีเซ็ตการทำงานของ MCU
- หมายเลข 4 เป็นขั้วต่อสัญญาณจาก PC[0..5] ซึ่งในกรณีนี้ใช้การพัฒนาโปรแกรมด้วย Arduino จะเป็นขาสัญญาณของ Analog[0..5]



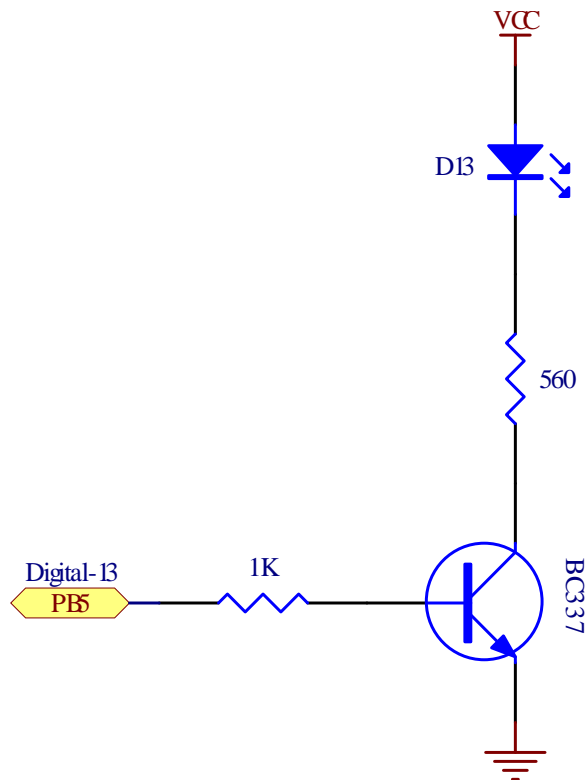
- หมายเลข 5 เป็นขั้วต่อสัญญาณจาก PB[0..5] ซึ่งในกรณีนี้ใช้การพัฒนาโปรแกรมด้วย Arduino จะเป็นขาสัญญาณของ Digital[8..13]



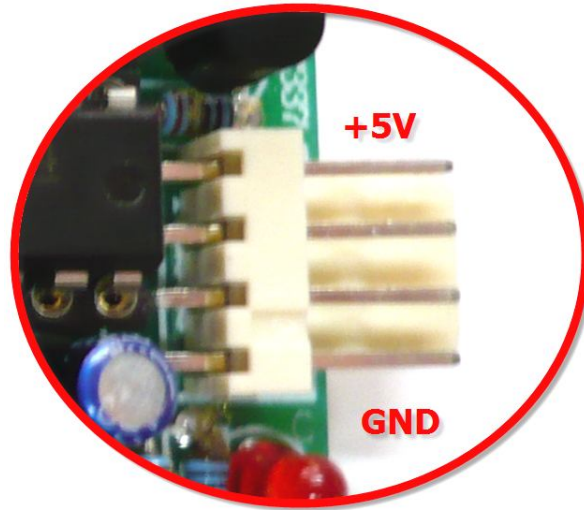
- หมายเลข 6 เป็นขั้วต่อสัญญาณจาก PD[0..7] ซึ่งในกรณีนี้ใช้การพัฒนาโปรแกรมด้วย Arduino จะเป็นขาสัญญาณของ Digital[0..7]



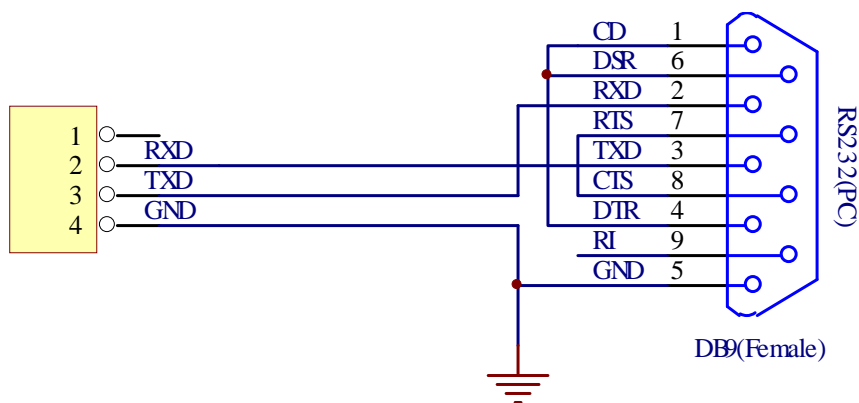
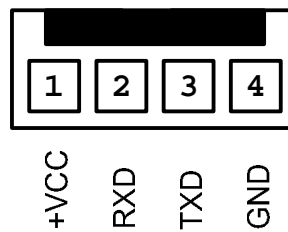
- หมายเลข 7 เป็น LED ใช้แสดงสถานะของขาสัญญาณ PB[5] หรือ Digital[13] ของ Arduino ซึ่ง LED นี้จะถูก Drive ผ่านทรานซิสเตอร์ BC337 ดังวงจร



- หมายเลข **8** คือ ขั้วต่อ RS232 สำหรับใช้งานทั่วไป และ Upload Code ให้กับ MCU ผ่านระบบ Bootloader โดยมีการจัดเรียงสัญญาณดังนี้



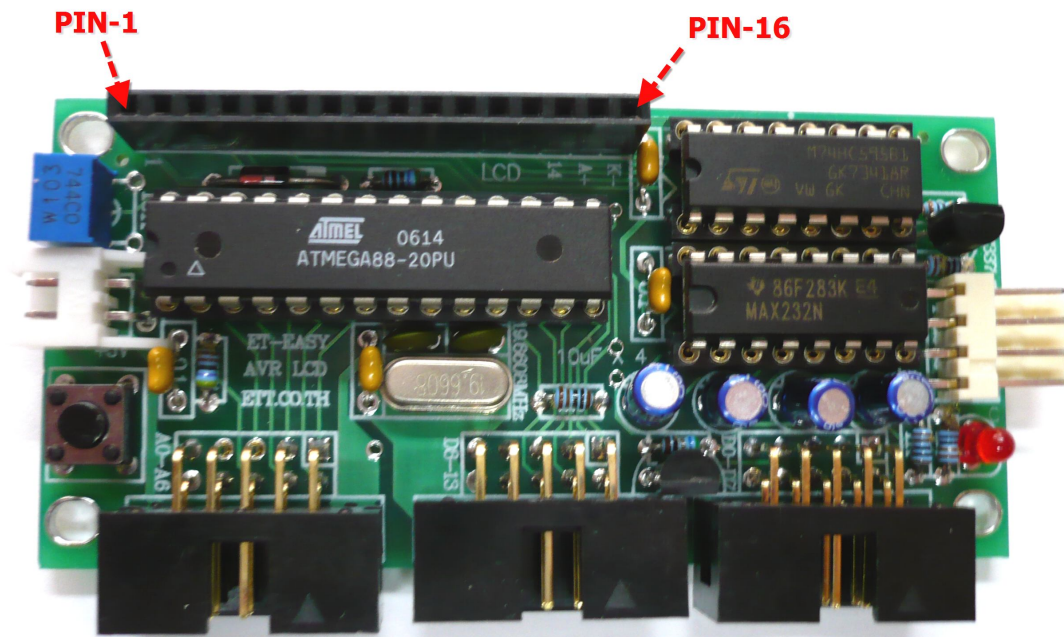
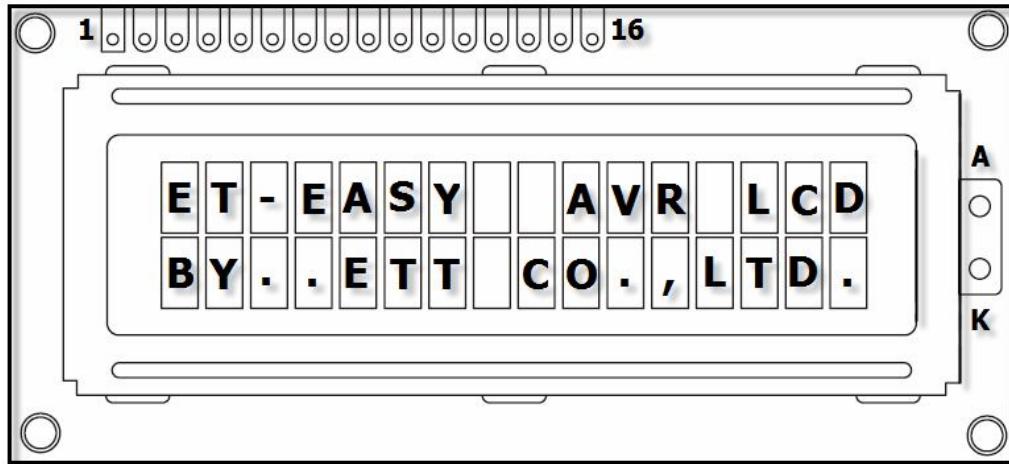
RS-232



รูปแสดง วงจรการต่อสายของ RS232 สำหรับใช้งานกับบอร์ด

- หมายเลข 9 เป็นขั้วต่อ LCD ขนาด 16 ตัวอักษร 2 บรรทัด ที่มีการจัดเรียงสัญญาณเป็นแบบแถวเดี่ยวขนาด 14 ถึง 16 Pin โดยมีขาสัญญาณอยู่ด้านบนของหน้าจอ ดังตัวอย่าง

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
GND	+5V	VO	RS	RW	EN	D0	D1	D2	D3	D4	D4	D6	D7	A	K



รูปแสดง ลักษณะของ LCD ที่จะนำมาติดตั้งใช้งานกับบอร์ด ET-EASY AVR LCD

- หมายเลข 10 เป็น MCU ประจำบอร์ด ซึ่งสามารถใช้ได้กับ AVR ขนาด 28ขา ได้หลายเบอร์ เช่น ATMEGA8,ATMEGA48,ATMEGA88,ATMEGA168 และ ATMEGA328
- หมายเลข 11 เป็น Crystal Oscillator ค่าความถี่ 19.6608 MHz

การพัฒนาโปรแกรมของบอร์ดด้วย Boot Loader

ในกรณีที่ต้องการจะพัฒนาโปรแกรมของบอร์ด ET-EASY AVR LCD ด้วย Boot Loader นั้น ก็สามารถทำได้โดยการต่อสวิตช์แบบ กดติดปล่อยดับ (Push Button Switch) ระหว่าง PD2 กับ GND เพื่อใช้สำหรับทำหน้าที่ Reset MCU ให้เข้าทำงานใน Boot Loader ซึ่งตามปกติแล้ว MCU ที่ติดตั้งไปกับบอร์ดนั้นทาง อีทีที ได้ทำการบรรจุโปรแกรม Boot Loader ไว้ให้เป็นที่ยอมรับแล้ว ผู้ใช้สามารถทดสอบการทำงานของ Boot Loader ได้ทันที แต่ถ้ามีการใช้เครื่องโปรแกรมแบบ AVRISP ส่งโปรแกรมข้อมูลอื่นเข้าไปแทนที่โปรแกรม Boot Loader ไปแล้วก็ต้องทำการสั่ง Program Boot Loader ใหม่ให้เรียบร้อยเสียก่อน ซึ่งการทดสอบการทำงานของโปรแกรม Boot Loader สามารถทำได้โดยการกดสวิตช์ PD2 ค้างไว้แล้วกดสวิตช์ Reset ดู โดย PD2 ต้องถูกกดค้างไว้จนกว่าสวิตช์ Reset จะถูกกด และ ปล่อยเรียบร้อยก่อน จึงค่อยปล่อยสวิตช์ของ PD2 เป็นลำดับสุดท้าย โดยถ้าทุกอย่างถูกต้องจะต้องเห็น LED กระพริบ 3 ครั้งแล้วติดค้าง เพื่อแสดงว่า MCU เข้าสู่งานทำงานใน Boot Loader เรียบร้อยแล้ว แต่เมื่อกดสวิตช์ Reset ตามปกติโดยไม่มีการกดสวิตช์ PD2 ร่วมด้วย MCU จะทำงานตาม Code คำสั่งที่เขียนขึ้นตามปกติ

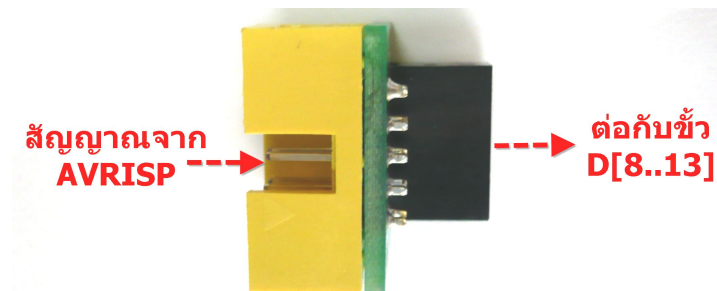
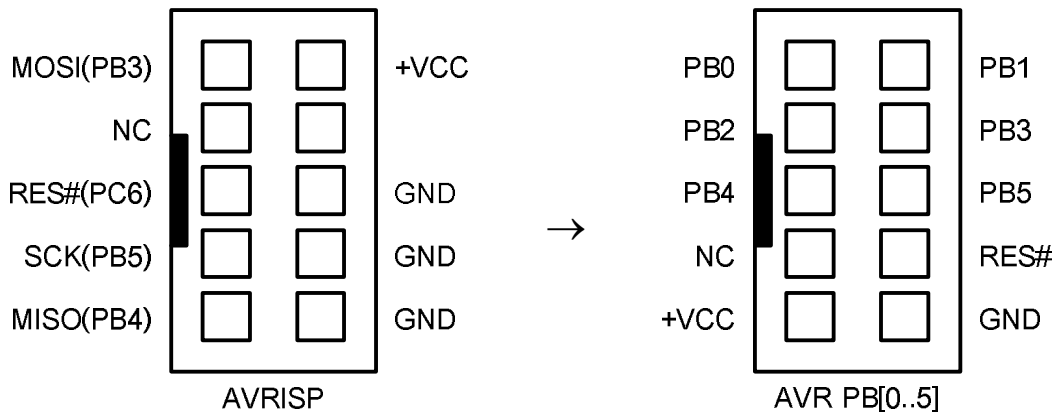
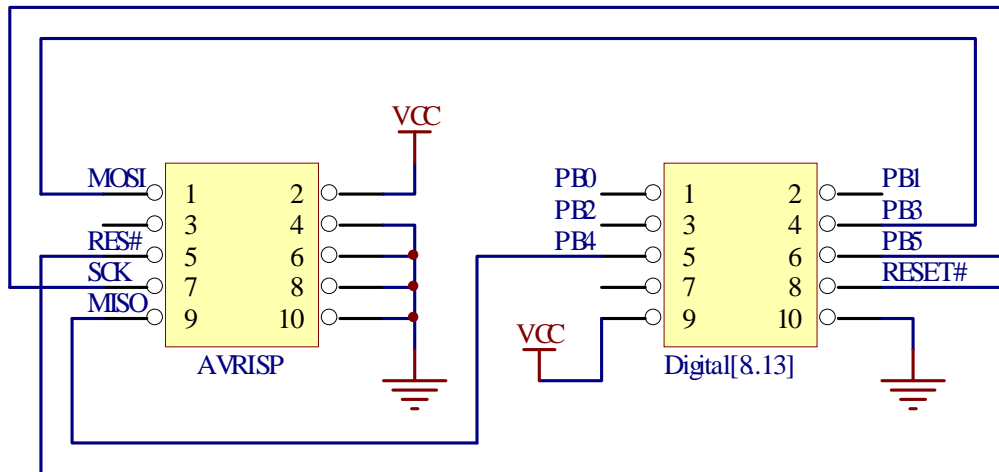


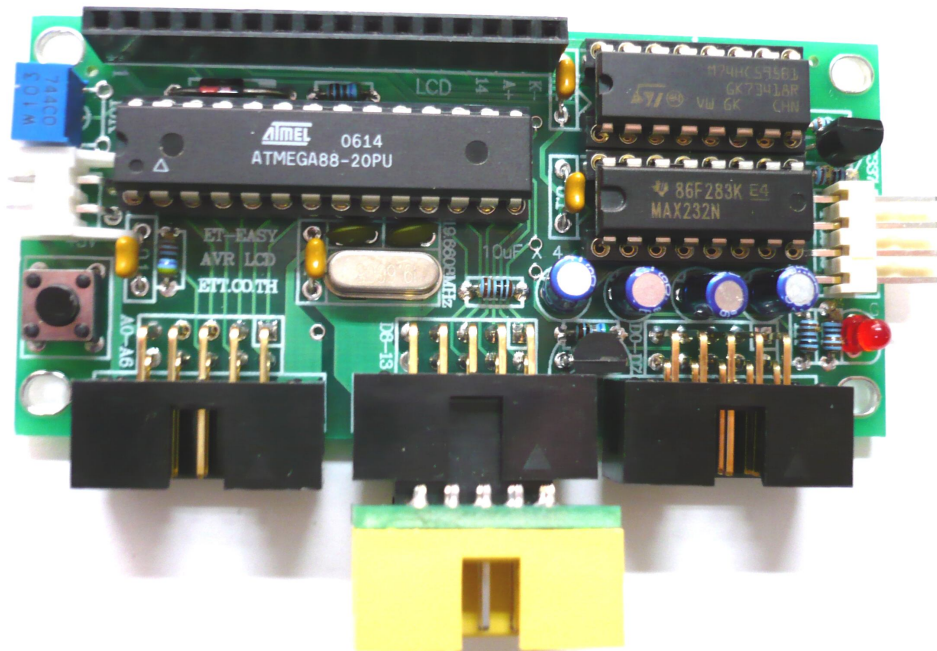
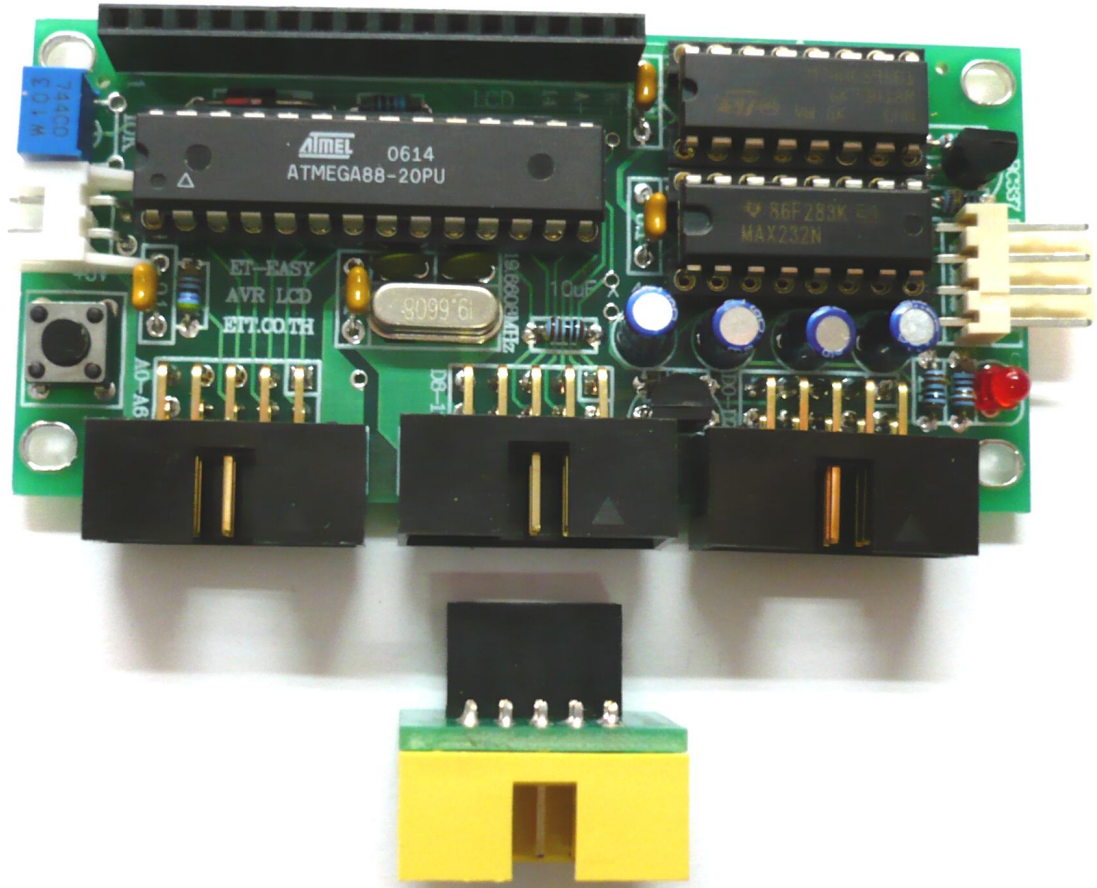
รูปแสดง การต่อสวิตช์ สำหรับใช้ทำงานกับ Boot Loader

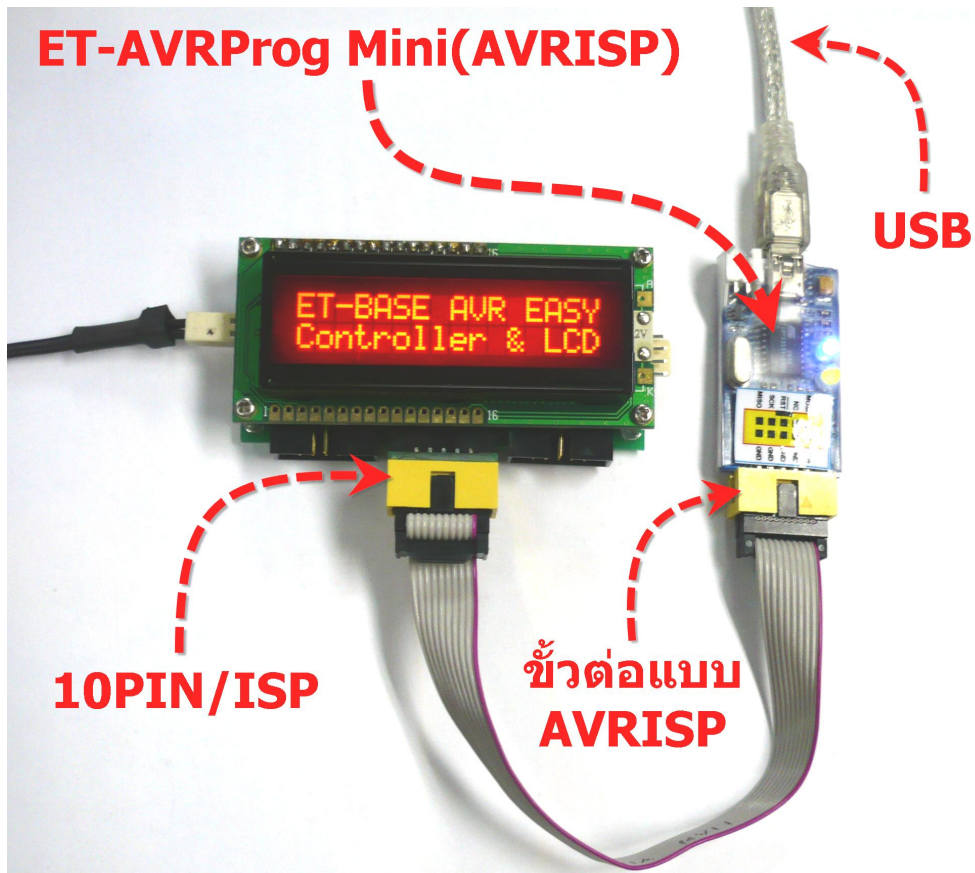
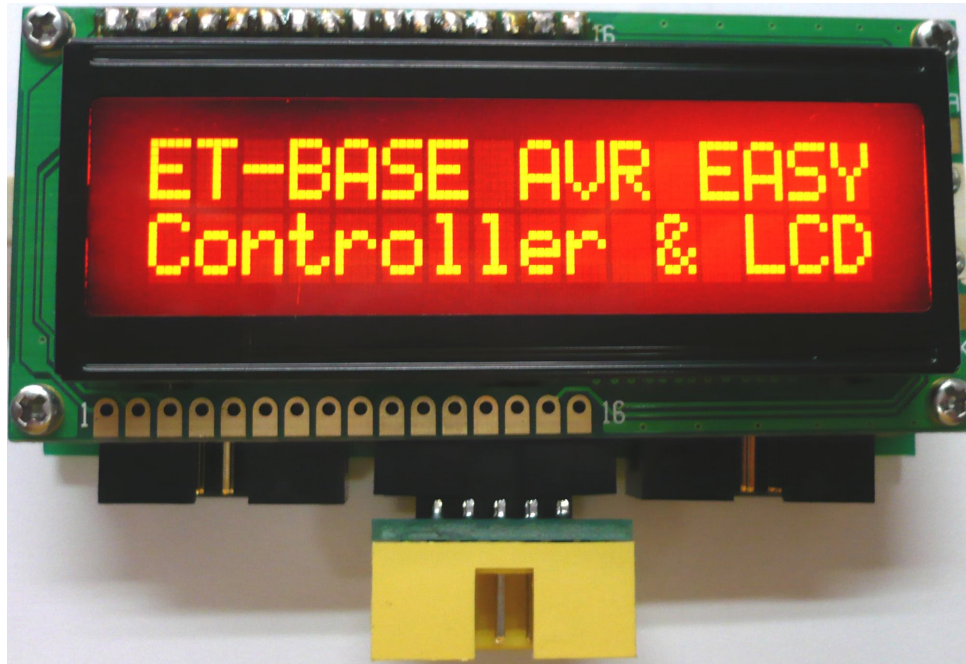
ซึ่งโปรแกรม Boot Loader สามารถใช้สำหรับการพัฒนาโปรแกรมของบอร์ดด้วย Arduino และการ Program Hex File จากภายนอกให้กับบอร์ดผ่านทาง RS232 ก็ได้เช่นเดียวกัน

การพัฒนาโปรแกรมของบอร์ดด้วย AVRISP

สำหรับแนวทางในการพัฒนาโปรแกรมของบอร์ด ET-EASY AVR LCD ด้วย AVRISP นั้น เนื่องจากบอร์ดไม่ได้จัดทำขั้วต่อสำหรับใช้ในการโปรแกรมบอร์ดด้วย "AVRISP" ไว้ภายในบอร์ดโดยตรงแต่ก็สามารถดัดแปลงต่อขาสัญญาณของ AVR เพื่อใช้สำหรับการโปรแกรมด้วย AVRISP ได้ โดยใช้ชุดสลับสัญญาณ รุ่น "10PIN/ISP" สำหรับแปลงสัญญาณของ AVRISP ให้ตรงกับขั้วต่อ I/O ของ AVR PB[0..5] ซึ่งก็จะทำให้สามารถนำขั้วต่อ I/O PB[0..5] หรือ D[8..13] ของบอร์ด ไปต่อกับเครื่องโปรแกรมที่จัดขั้วตรงตามมาตรฐานของ AVRISP ได้ทันที ดังตัวอย่าง





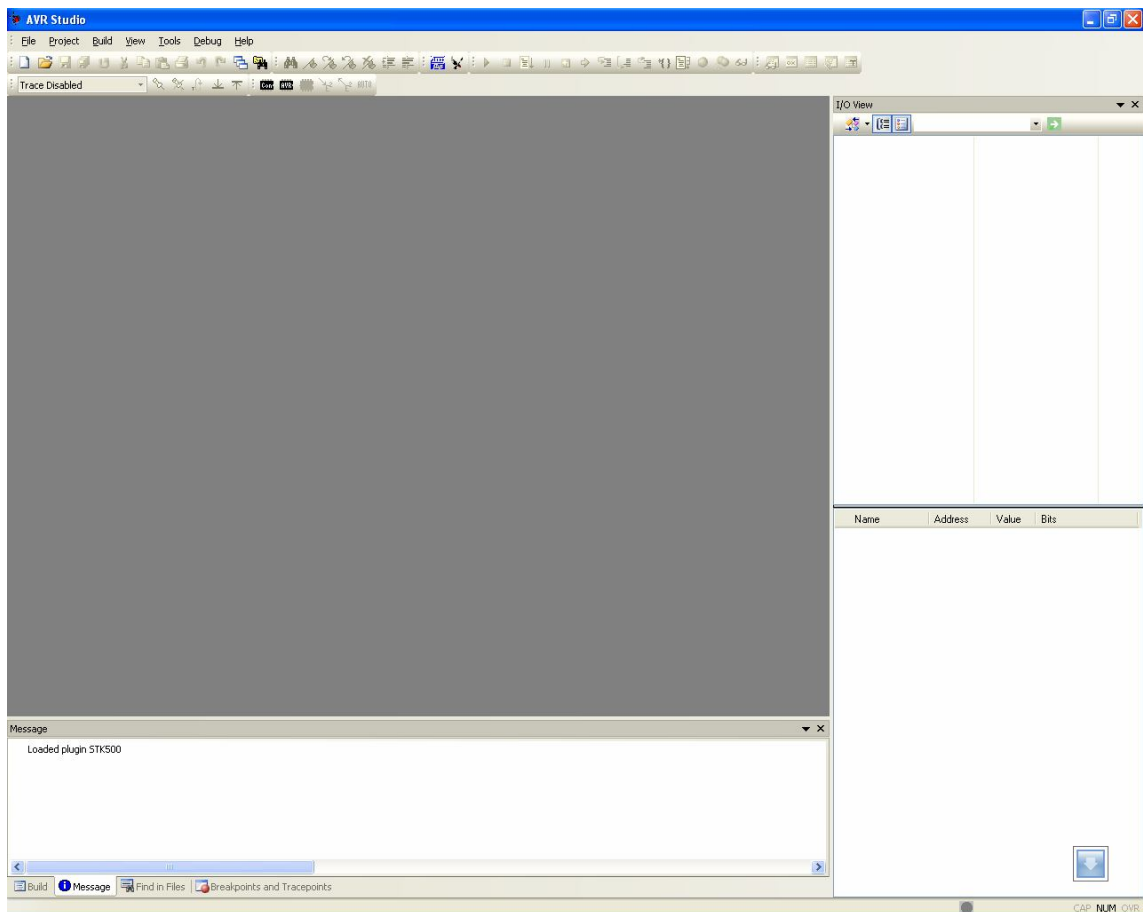


ตัวอย่าง การโปรแกรมกับเครื่องโปรแกรมแบบ AVRISP รุ่น "ET-AVRProg Mini"

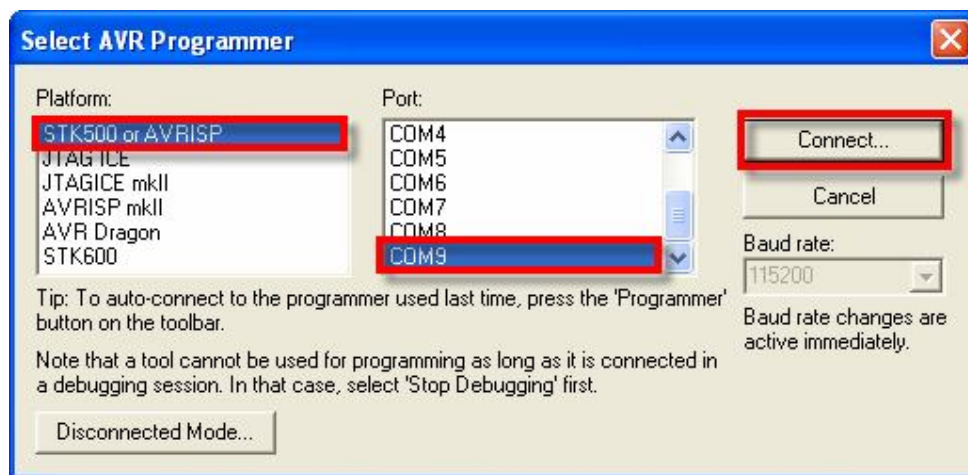
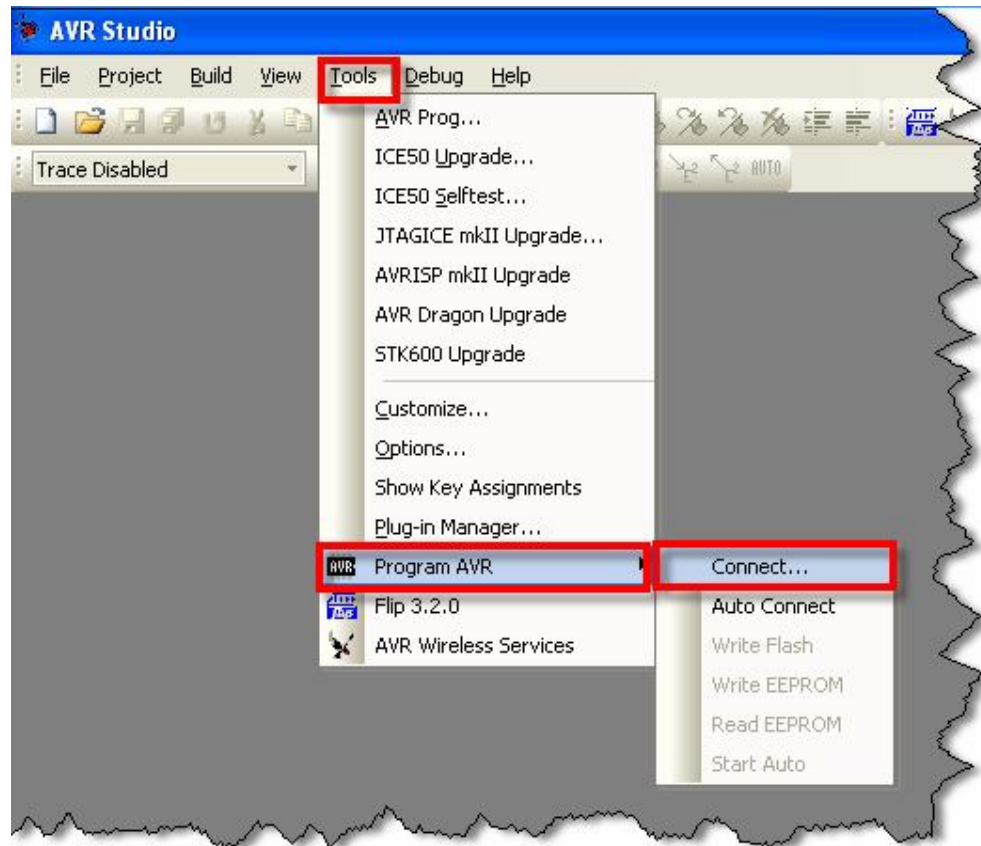
การโปรแกรม **Bootloader** ให้กับบอร์ด **ET-EASY AVR LCD**

ตามปกติแล้วบอร์ด ET-EASY AVR LCD ได้ทำการโปรแกรม **Bootloader** ไว้ให้เป็นที่เรียบร้อยแล้ว ผู้ใช้สามารถใช้งานได้ทันที แต่อย่างไรก็ตามในกรณีที่ต้องการเปลี่ยนแปลง **Bootloader** หรือ เกิดความผิดพลาดในการใช้งาน จนทำให้ **Bootloader** เสียหายไป ผู้ใช้ก็ยังสามารถทำการ โปรแกรม **Bootloader** ให้กับบอร์ดได้ใหม่ โดยบอร์ด ET-EASY AVR LCD ได้ออกแบบ ให้มีขั้วต่อ **AVRISP** สำหรับใช้เป็นช่องทางในการโปรแกรม **Code** ให้กับ **MCU** ได้โดยตรง ด้วยเครื่องโปรแกรมทุกรุ่นที่มีขั้วต่อ ตรงตามมาตรฐาน **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

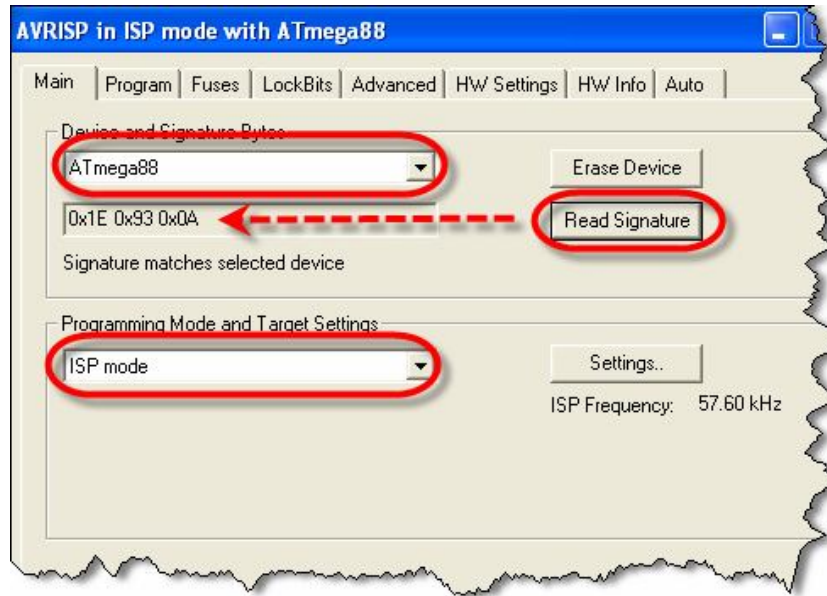
1. ต่อสาย **RS232** จากคอมพิวเตอร์ ให้กับบอร์ด **ET-EASY AVR LCD** และจ่ายไฟให้บอร์ด
2. ต่อสาย **USB** ให้กับเครื่องโปรแกรม **ET-AVR ISP USB V1** พร้อมทั้งต่อสายแพร์ **10 Pin** ระหว่างขั้วต่อของ **AVRISP** ของทั้ง 2 บอร์ดเข้าด้วยกัน
3. สั่ง **Run** โปรแกรม **AVR Studio 4** ดังรูป



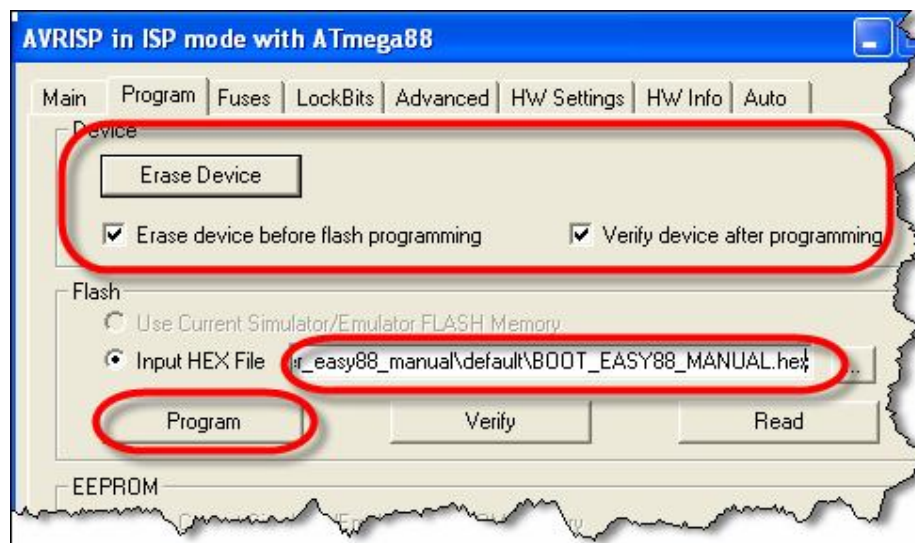
4. เลือกคลิกเมาส์ที่ **Tools** → **Program AVR** → **Connect..** → **STK500 or AVRISP** จากนั้นก็ให้เลือกหมายเลขของ **Comport** ที่เป็นของเครื่องโปรแกรม **ET-AVR ISP USB V1** ตามที่ลง **Driver** ไว้พร้อมกับเลือก **Connect** (จากตัวอย่างเป็น **Com9**) ดังรูป



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



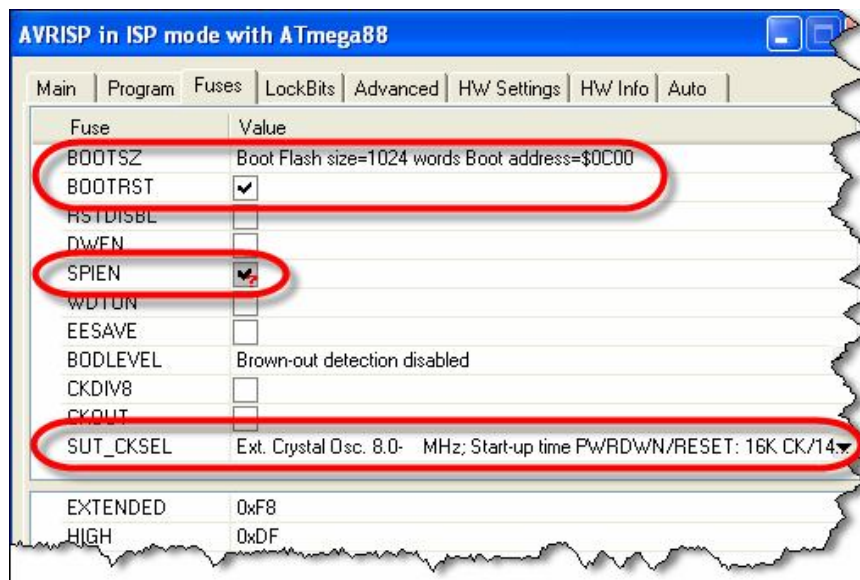
6. ให้เลือกไปที่ **tab** ของ **Program** พร้อมทั้งเลือก ตัวเลือกต่างๆดังนี้
- **Device** ให้เลือก **Erase device before flash programming** และ **Verify device after programming**
 - **Flash** ให้เลือก **Input HEX File** เป็น **BOOT_EASY88_MANUAL_19_6608MHZ.HEX** จากนั้นให้เลือกที่ **Program** เพื่อสั่ง **Program Bootloader** ให้กับ **MCU** ซึ่งจะได้ผลดังรูป



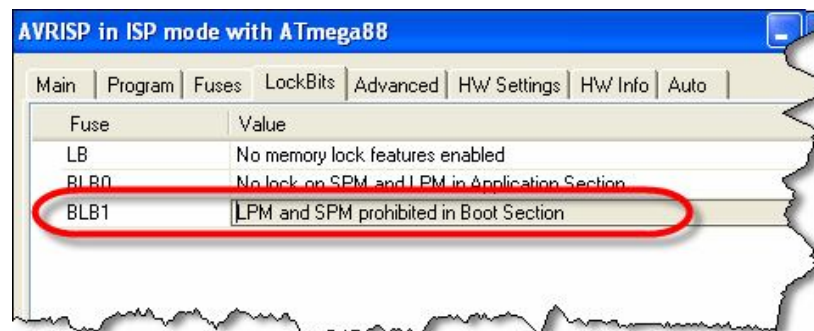
7. หลังจากทำการโปรแกรม Code ให้กับ MCU เรียบร้อยแล้ว ให้เลือกมาที่ Tab ของ Fuses เพื่อทำการตั้งโปรแกรม Fuse Bit ให้กับ MCU โดยให้เลือกกำหนดค่าตัวเลือกดังนี้

- 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



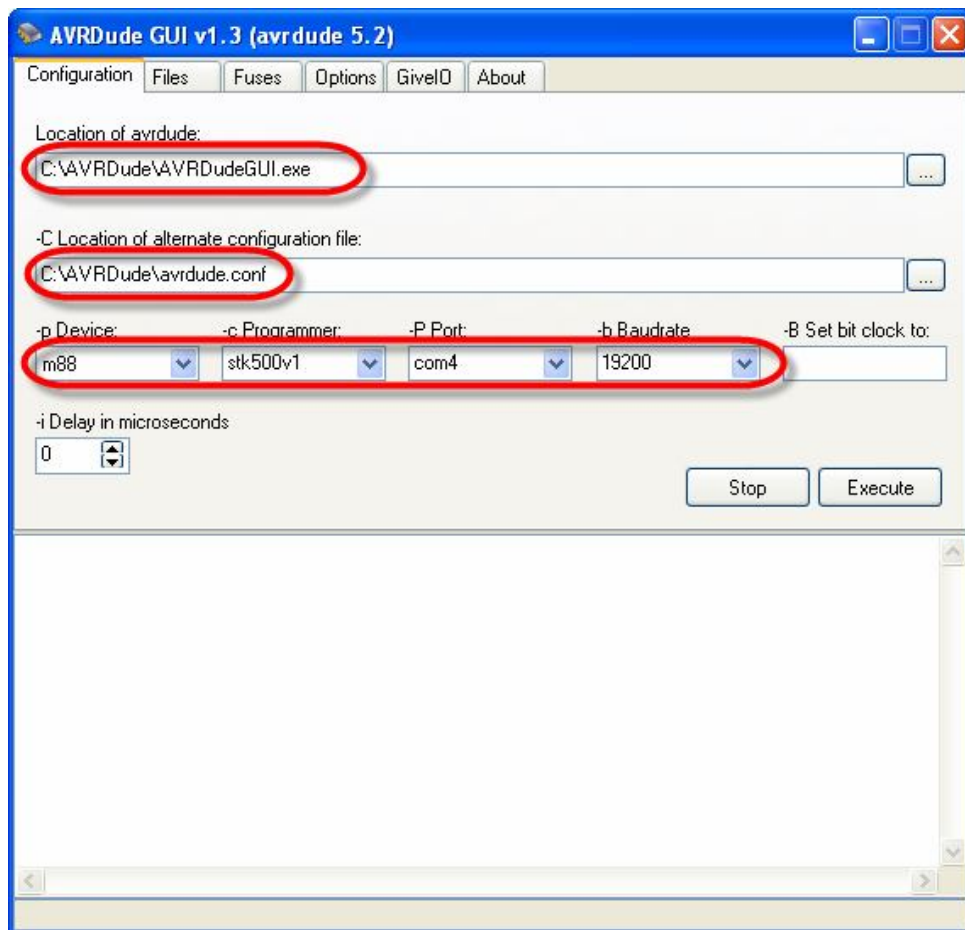
8. หลังจากทำการตั้ง program Fuse Bit เรียบร้อยแล้ว ให้เลือก Tab มาที่ LockBits แล้วเลือกการ Protect เฉพาะ Bootloader โดยเลือก BLB1 เป็น LPM and SPM prohibited in Boot Section แล้วสั่ง program เป็นอันเสร็จสิ้นกระบวนการ Program Bootloader



การส่งโปรแกรม Hex File ให้กับบอร์ด ET-EASY AVR LCD ผ่าน Bootloader

ในการพัฒนาโปรแกรมของ บอร์ด ET-EASY AVR LCD ในรูปแบบของ AVR Microcontroller นั้น หลังจากที่ทำกรเขียนโปรแกรม และแปลงคำสั่งจนได้ HEX File เรียบร้อยแล้วในการส่ง Program HEX Code ที่ได้จากการแปลงคำสั่งของโปรแกรมนี้นี้ ผ่านทางโปรแกรม Bootloader ของบอร์ดนั้น ขอแนะนำให้ใช้โปรแกรม AVRdude โดยใช้งานผ่าน AVRdudeGUI ซึ่งมีขั้นตอนดังนี้

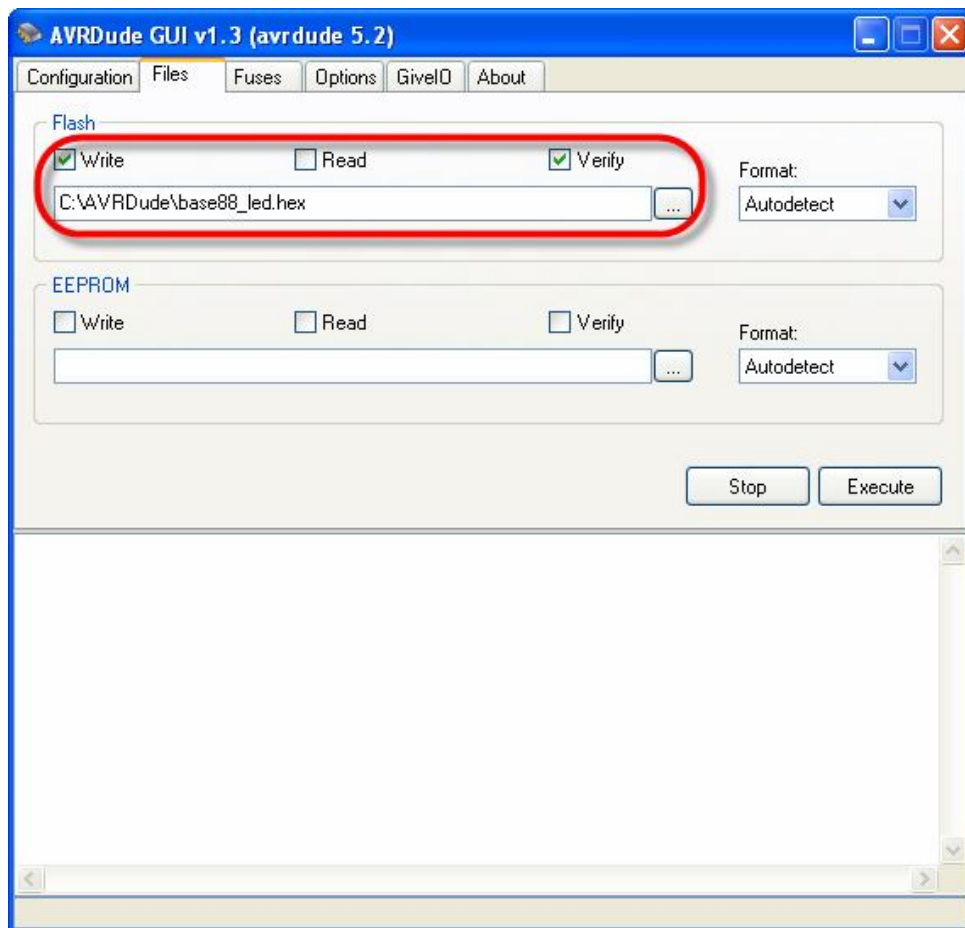
1. ทำการสร้างโฟลเดอร์ชื่อ AVRDUDE ไว้ใน Drive C ("C:\AVRDUDE\") จากนั้นให้ทำการ Copy โปรแกรมของ avrdude และ avrdudeGUI ไว้ใน "C:\AVRDUDE\" โดยจะมีทั้งหมด 3 ไฟล์ด้วยกัน คือ avrdude.exe, avrdude.conf และ avrdudegui.exe
2. สั่ง Run Program AVRdude.EXE แล้วเลือกที่ Tab ของ Configuration ให้เลือกกำหนดค่าต่างๆ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



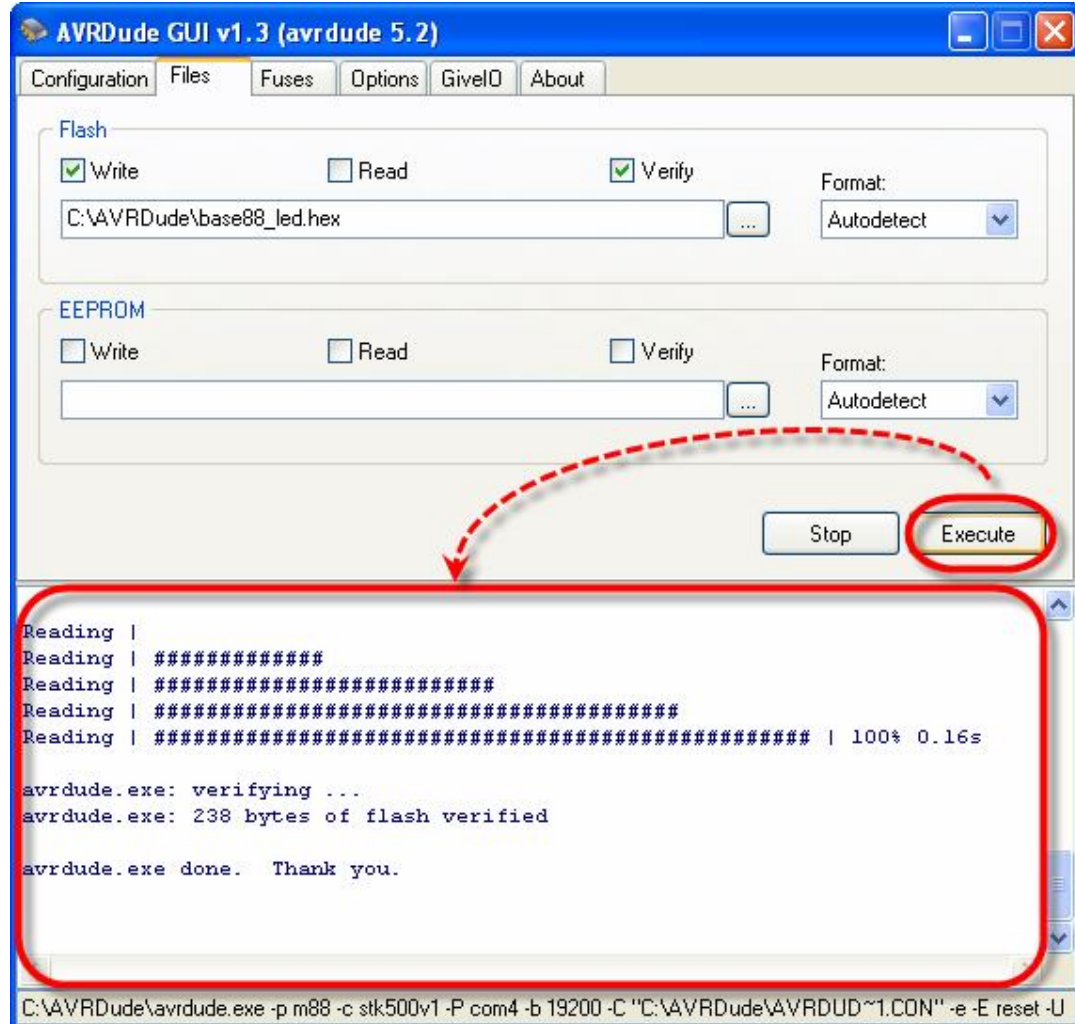
หมายเหตุ ถ้าหมายเลข Comport มากกว่า com8 ให้พิมพ์หมายเลข Comport ลงไปในช่องรับค่าของ -P Port ได้เอง เช่น com9 แต่ถ้าหมายเลข Comport มีค่าเป็น 2 หลักให้ เพิ่มเครื่องหมาย "\\\" นำหน้าชื่อหมายเลข Comport ลงไปด้วย เช่น ถ้าใช้กับ Com13 ก็ให้กำหนดเป็น "\\com13" เป็นต้น

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 พร้อมทำการกำหนดชื่อ 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. หลังจากสั่งไปแกรมเสร็จให้กดสวิทซ์ **RESET**บอร์ดจะเริ่มต้นทำงานทันที โดยจะเห็น **LED** กระพริบติด และ ดับ สลับกันไปมาไม่รู้จบ

การพัฒนาโปรแกรมของ ET-EASY AVR LCD ด้วย Arduino

ตามปกติแล้วบอร์ด ET-EASY AVR LCD จะทำการ ติดตั้งโปรแกรม Bootloader ไว้ให้เรียบร้อยแล้ว โดยใช้ Bootloader ชื่อ "BOOT_EASY88_MANUAL_19_6608MHZ.HEX" ซึ่งเป็น Bootloader ที่ทางทีมงาน อีทีที ได้นำต้นฉบับจาก Arduino มาปรับปรุงเงื่อนไขการทำงานใหม่ เพื่อให้การทำงานสอดคล้องกับระบบฮาร์ดแวร์ของบอร์ด ET-EASY AVR LCD ได้ดียิ่งขึ้น โดยโปรแกรม Bootloader นี้จะใช้สำหรับติดต่อสื่อสารเพื่อส่ง 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 ก็จะทำออกจากการทำงานใน Bootloader เพื่อไปเริ่มต้นทำงานตามคำสั่งที่เป็นของผู้ใช้โดยอัตโนมัติ แต่ถ้าพบว่าขา PD2 เป็น LOW จะเข้าทำงานใน Bootloader Programming Mode โดยในขณะที่ Bootloader Programming Mode ทำงาน ตอนเริ่มต้นจะเห็น LED ที่ต่อกับขาสัญญาณ Digital-13(PB5) กระพริบ 3 ครั้ง แล้วติดค้าง เพื่อบอกการติดต่อสื่อสารจากโปรแกรมสำหรับสั่งให้ทำการ Upload Code ให้กับ MCU จนกว่าจะถูกสั่งรีเซ็ต

สำหรับบอร์ด ET-EASY AVR LCD นั้น จะใช้สวิทช์ BL(PD2) และสวิทช์ RESET เป็นตัวเลือกการทำงานของ Bootloader ซึ่งถ้าต้องการให้ MCU ทำงานจาก Code ที่ผู้ใช้เขียนขึ้นก็ให้ใช้การกดสวิทช์ RESET เพียงอย่างเดียว แต่ถ้าต้องการให้ MCU เข้าสู่การทำงานใน Bootloader Programming Mode ก็ให้กดสวิทช์ BL(PD2) ค้างไว้แล้วกดสวิทช์ RESET โดยต้องกดสวิทช์ BL(PD2) ค้างไว้ก่อนจนกว่าจะปล่อยสวิทช์ RESET แล้วจึงค่อยปล่อยสวิทช์ BL(PD2) เป็นลำดับสุดท้าย โดยจะสังเกตเห็น LED ที่ต่อไว้ที่ขา PB5 กระพริบ 3 ครั้งแล้วติดค้างรอไว้ เพื่อบอกให้ทราบว่า MCU เข้าทำงานใน Bootloader แล้ว

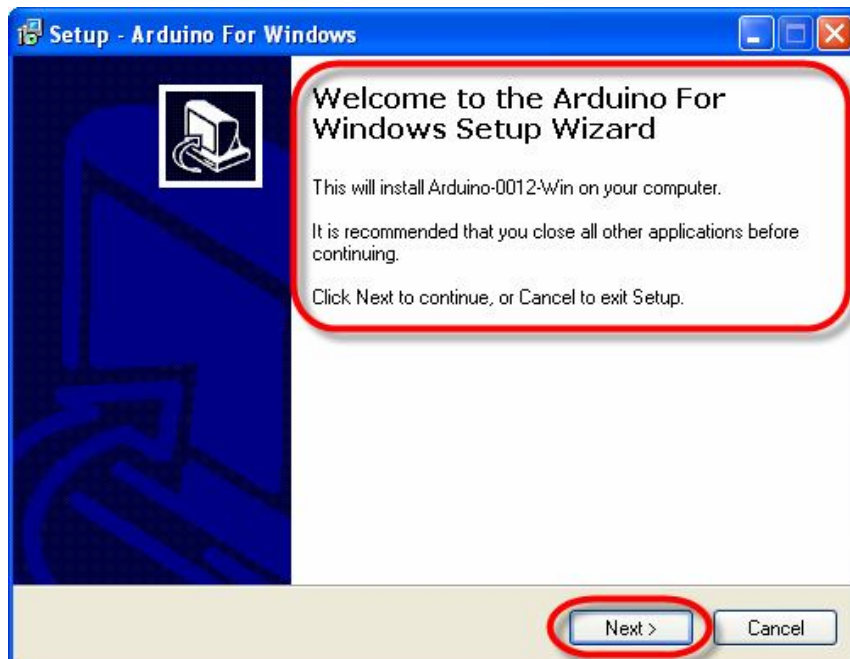
การติดตั้งโปรแกรม Arduino

สำหรับโปรแกรม Arduino นั้น ได้รับการพัฒนาขึ้นมาให้สามารถใช้งานกับระบบปฏิบัติการแบบต่างๆ ได้หลาย Platform ซึ่งปัจจุบัน (เดือน ธันวาคม พศ.2551) โปรแกรมของ Arduino ได้รับการปรับปรุงเป็นรุ่น เวอร์ชัน "Arduino-0012" แล้ว โดยมีโปรแกรมให้เลือกใช้งาน 4 Platform ทั้ง Windows, Mac OSx และ Linux โดยผู้อ่านสามารถเข้าไป ตรวจสอบ หรือ Download โปรแกรมรุ่นใหม่ๆของ Arduino มาใช้งาน ได้ฟรีโดยไม่เสียค่าใช้จ่ายใดๆจาก "<http://arduino.cc/>" หรือ "<http://arduino.cc/en/Main/Software>" ซึ่งเป็นเว็บไซต์ที่ได้รวบรวมรายละเอียดและข่าวคราวความเคลื่อนไหวต่างๆ เกี่ยวกับ Arduino มากมาย ซึ่ง ข้อมูลต่างๆจะได้รับการปรับปรุงอย่างต่อเนื่องเป็นประจำ

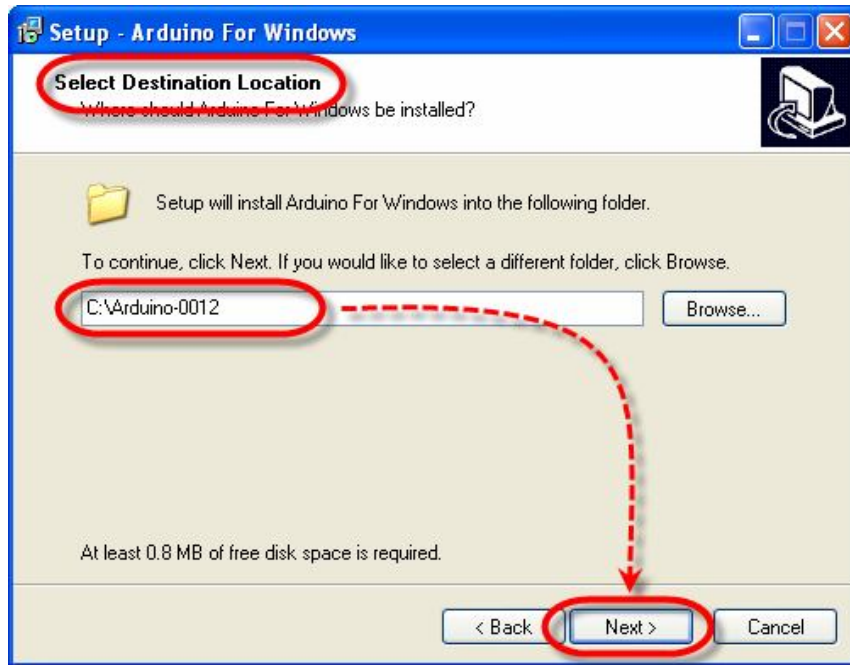
สำหรับกรณี ที่ผู้อ่านใช้งานกับบอร์ด Arduino ของ บริษัท อีทีที นั้น โปรแกรมต่างๆจะถูกรวบรวมไว้ใน แผ่น CD ROM เป็นที่เรียบร้อยแล้ว โดยโปรแกรมหดงกล่าวจะเป็น รุ่นที่ได้รับการปรับแต่งรายละเอียดให้สามารถใช้งานร่วมกันกับบอร์ดรุ่นต่างๆที่ทางบริษัท อีทีที ผลิตขึ้นมาใหม่ได้ด้วย นอกจากนี้แล้วทางบริษัท อีทีที ยังได้ทำการเพิ่มเติม Library ส่วนที่ทาง อีทีที พัฒนาปรับปรุงขึ้นมาใหม่รวมไว้ในชุดโปรแกรมหดงกล่าวด้วยแล้ว พร้อมกับทำการเพิ่ม Install Shield เข้าไปด้วยเพื่อให้ผู้ใช้สามารถทำการติดตั้งใช้งานโปรแกรมได้โดยง่ายเช่นเดียวกันกับการ Install โปรแกรมทั่วไป

สำหรับขั้นตอนการติดตั้งโปรแกรมนั้น สามารถทำได้ตามขั้นตอนของ Wizard ในการติดตั้งของโปรแกรมได้ทันที โดยมีลำดับขั้นตอนดังนี้

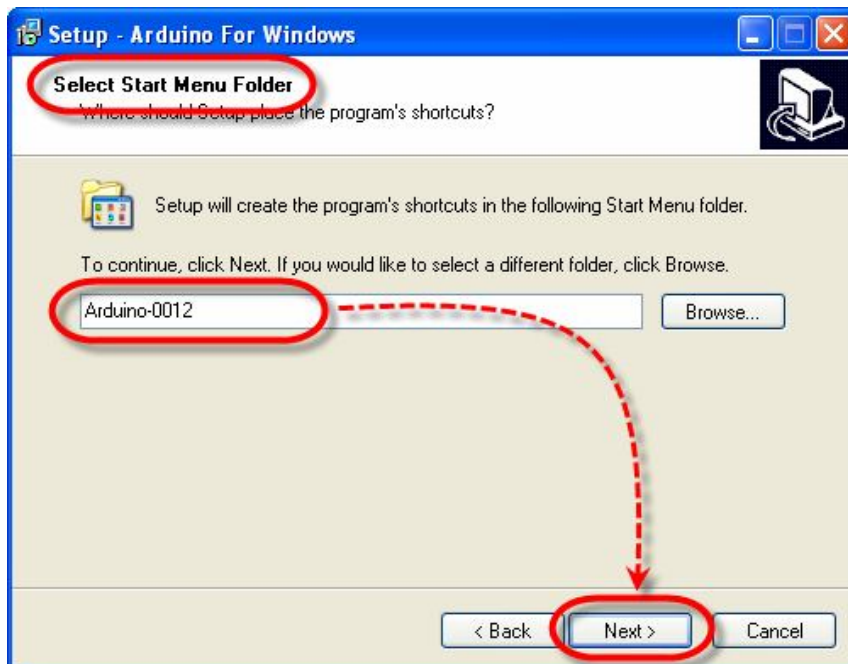
1. สั่ง Run ไฟล์ "ET-ARDUINO-0012-WIN.EXE" ซึ่งจะได้ผลดังรูป



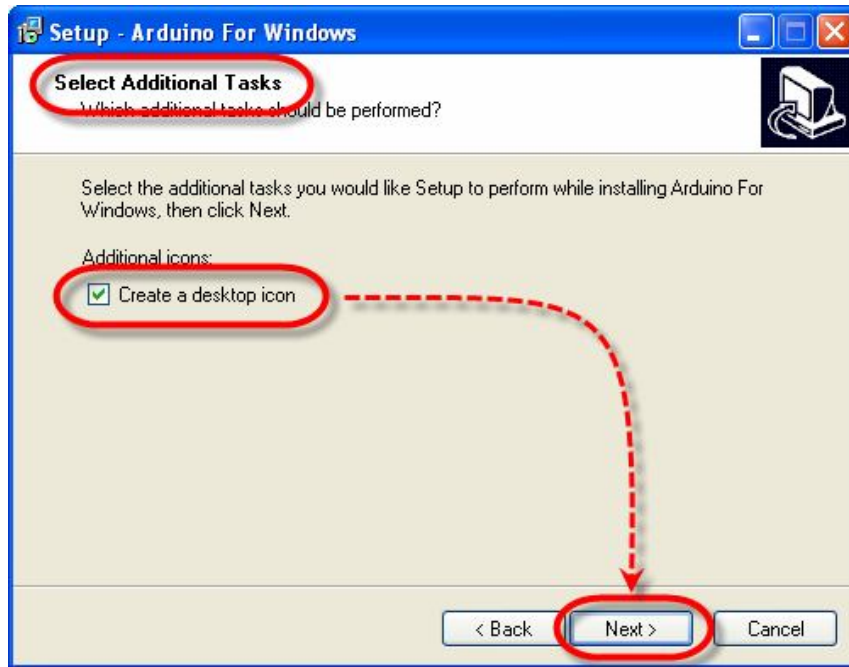
2. ในขั้นตอนนี้โปรแกรมจะให้กำหนด ตำแหน่งโฟลเดอร์ที่จะใช้สำหรับติดตั้งโปรแกรม ซึ่งให้เลือก กำหนดตามค่า Default ของโปรแกรมการติดตั้ง คือ C:\Arduino-0012 แล้วเลือก Next ดังรูป



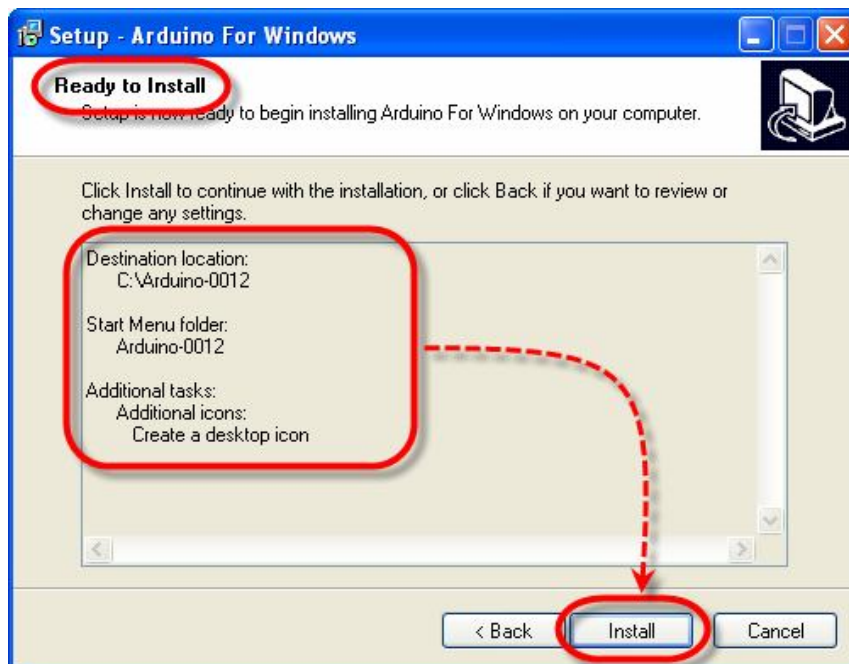
3. ในขั้นตอนนี้โปรแกรมจะให้กำหนด ชื่อโฟลเดอร์ที่จะใช้สำหรับเรียกใช้โปรแกรมผ่านทางเมนูคำสั่งของ Windows ซึ่งในขั้นตอนนี้นแนะนำให้เลือกกำหนดตามค่า Default ของโปรแกรมการติดตั้ง คือ C:\Arduino-0012 แล้วเลือก Next ดังรูป



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



5. เมื่อถึงขั้นตอนนี้ โปรแกรมก็พร้อมทำการติดตั้งแล้ว โดยโปรแกรมจะแสดงค่าตัวเลือกต่างๆ ที่ถูกกำหนดไว้ในขั้นตอนนี้ก่อนหน้านี้นี้ให้ทราบ เมื่อทุกอย่างถูกต้องให้เลือก **Install** ซึ่งโปรแกรมก็จะเริ่มทำการติดตั้งทันที



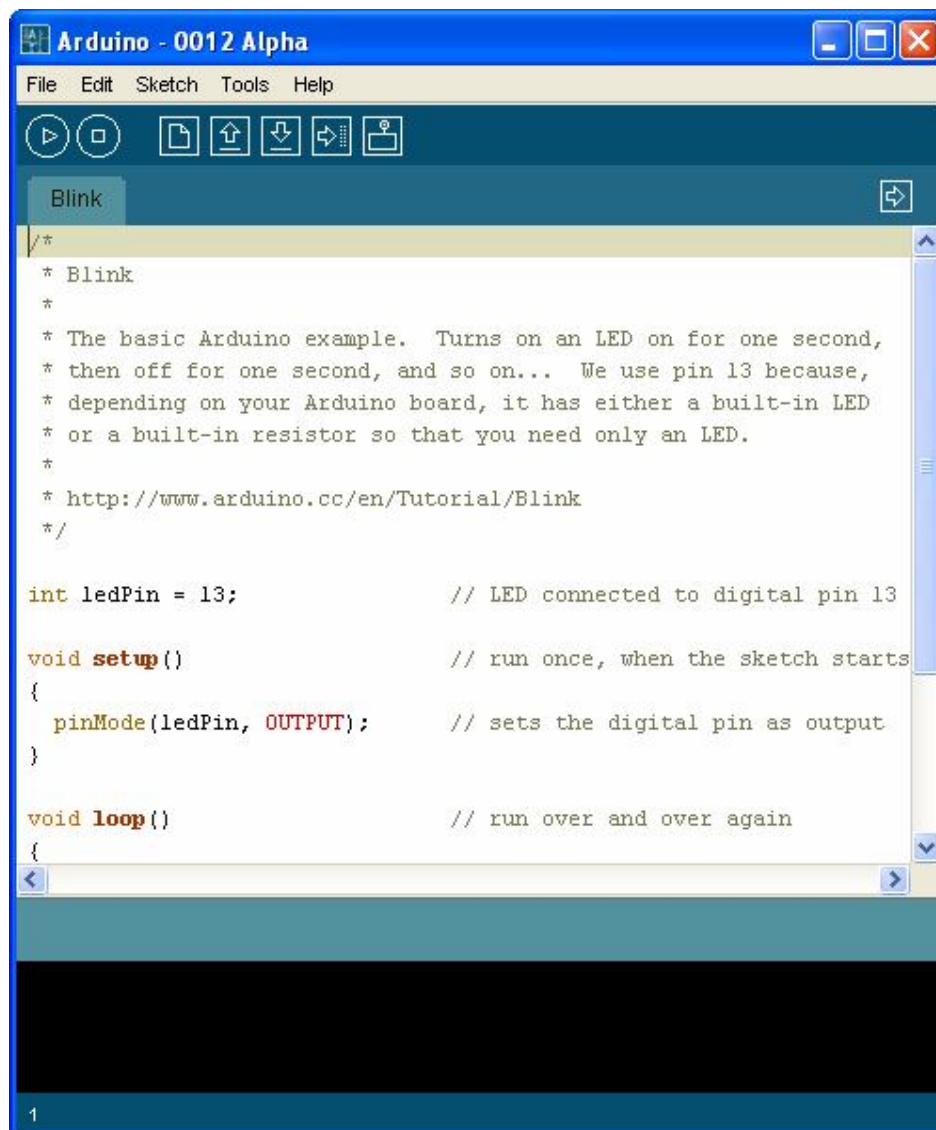
6. ให้รอจนกระทั่งขั้นตอนการติดตั้งเรียบร้อยแล้วเลือก **Finish** ดังรูป



ทดสอบเขียนโปรแกรมใช้งานด้วย Arduino

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

1. ทำการสั่ง Run โปรแกรม "arduino.exe" จะได้ผลดังรูป



The screenshot shows the Arduino IDE interface with the 'Blink' sketch loaded. The code in the editor is as follows:

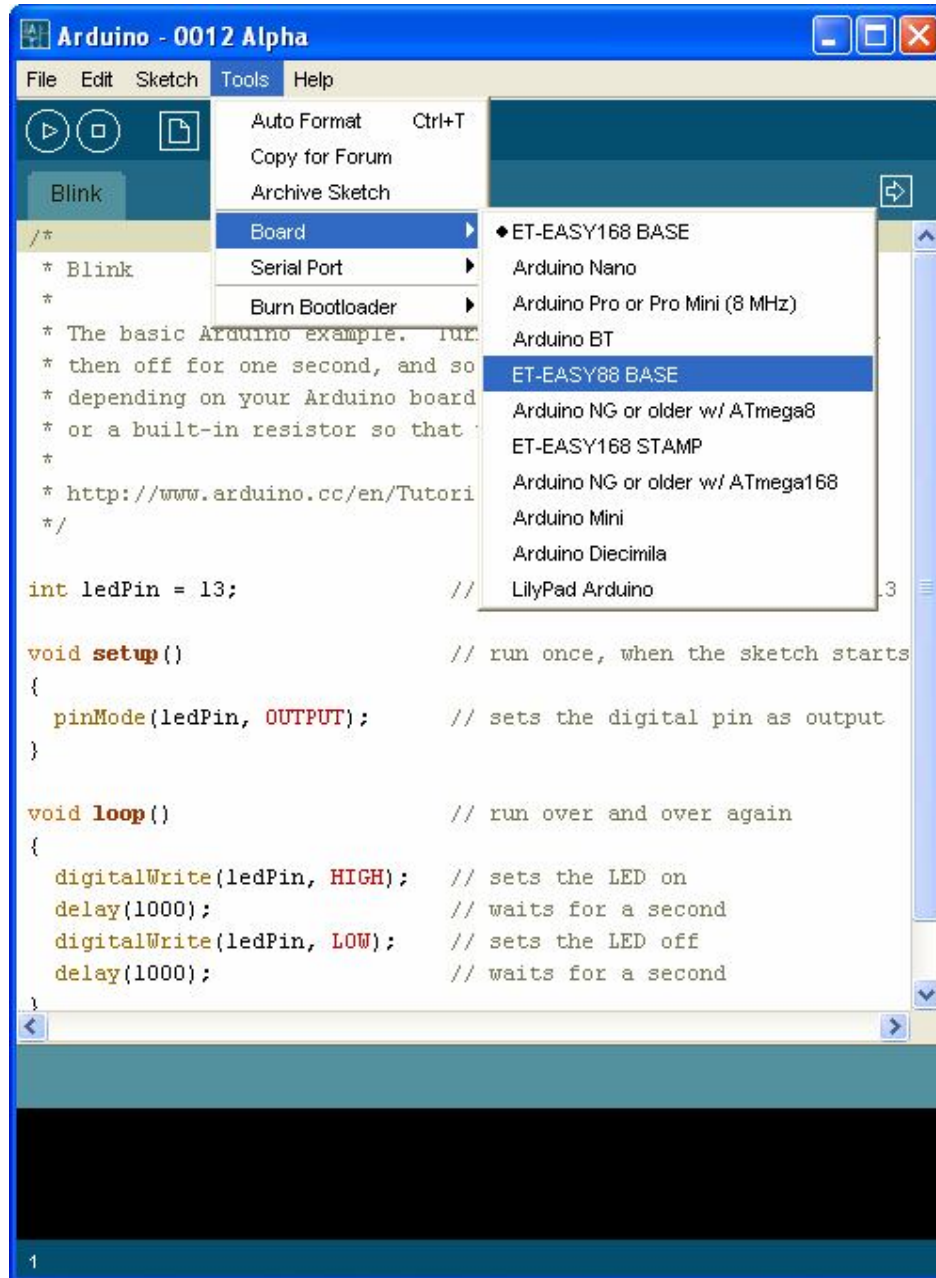
```
Arduino - 0012 Alpha
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13

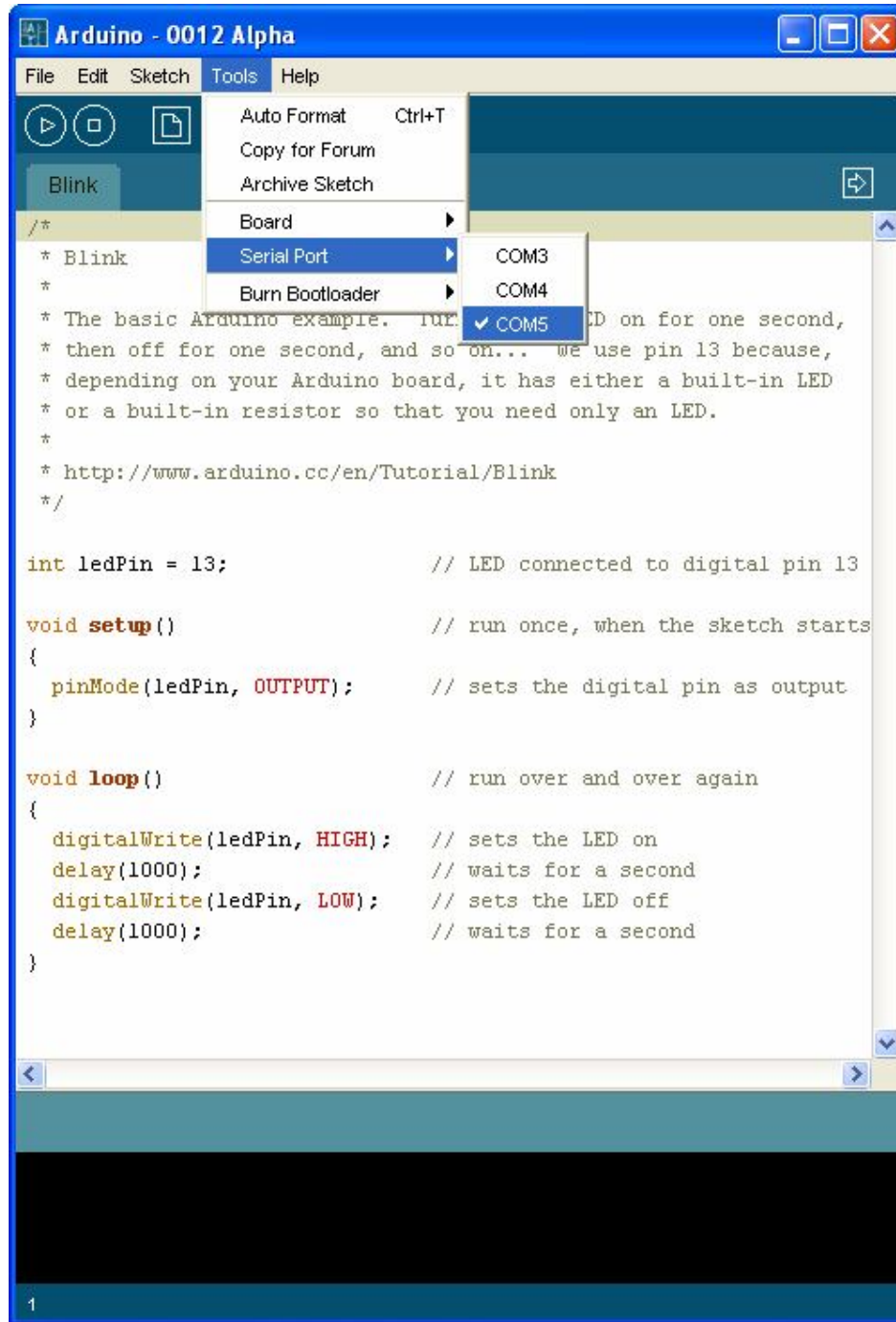
void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the positive voltage)
  delay(1000);                // wait for a second
  digitalWrite(ledPin, LOW);  // turn the LED off by making the pin LOW (no voltage)
  delay(1000);                // wait for a second
}
```

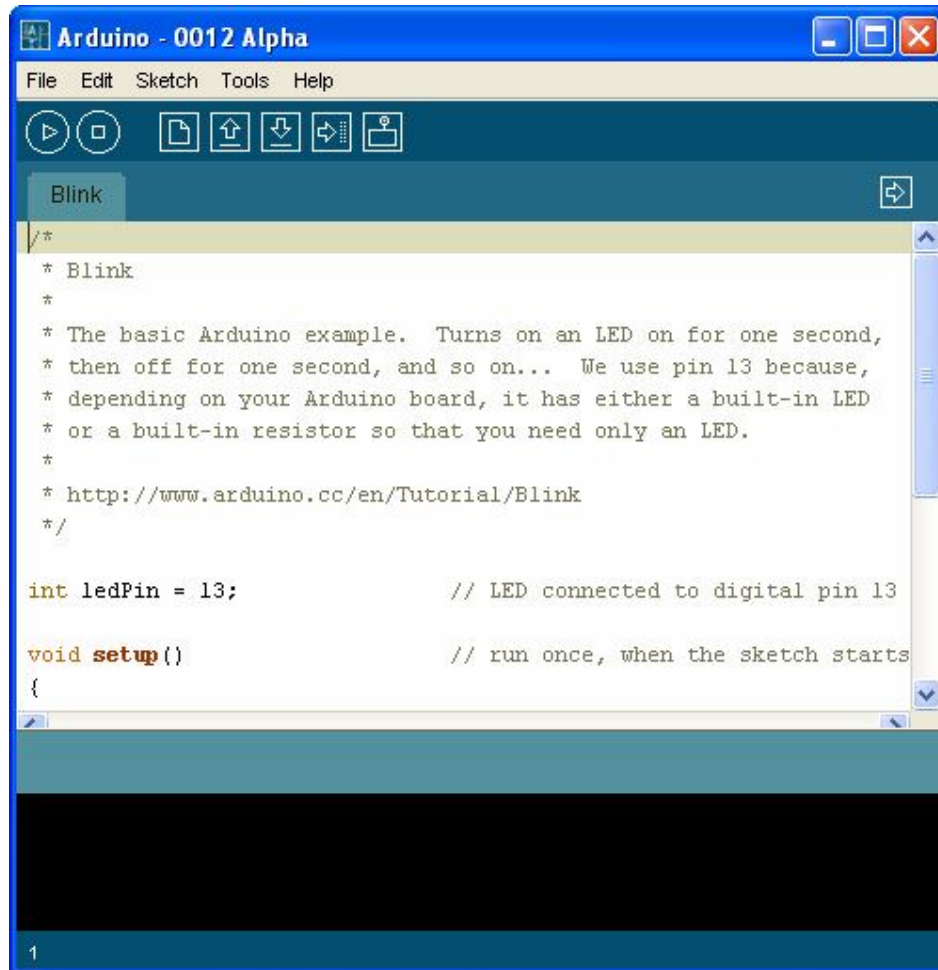
2. ในครั้งแรกของการเรียกใช้งานโปรแกรม ให้ทำการกำหนดระบบฮาร์ดแวร์ที่จะใช้งานกับโปรแกรมของ **Arduino** ให้เรียบร้อยเสียก่อน เนื่องจากในปัจจุบันนี้ มีการออกแบบวงจรและสร้างฮาร์ดแวร์บอร์ดแบบต่างๆสำหรับนำมาใช้งานร่วมกับโปรแกรมพัฒนาของ **Arduino** อยู่มากมายหลายรุ่น โดยในกรณีของบอร์ด **ET-EASY AVR LCD** ให้ทำการเลือกกำหนดชื่อบอร์ดเป็น **"EASY88 BASE"** โดยคลิกเมาส์ที่ **"Tools → Board → "ET-EASY88 BASE"** ดังรูป



- 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 หรืออาจใช้คำสั่งเปิดไฟล์ตัวอย่างที่สร้างไว้แล้วขึ้นมาแทนก็ได้ โดยในที่นี้ขอแนะนำให้ทดสอบด้วยโปรแกรมไฟกระพริบ โดยให้เลือก "File → sketchbook → Examples → Digital → Blink" ซึ่งจะได้ดังรูป

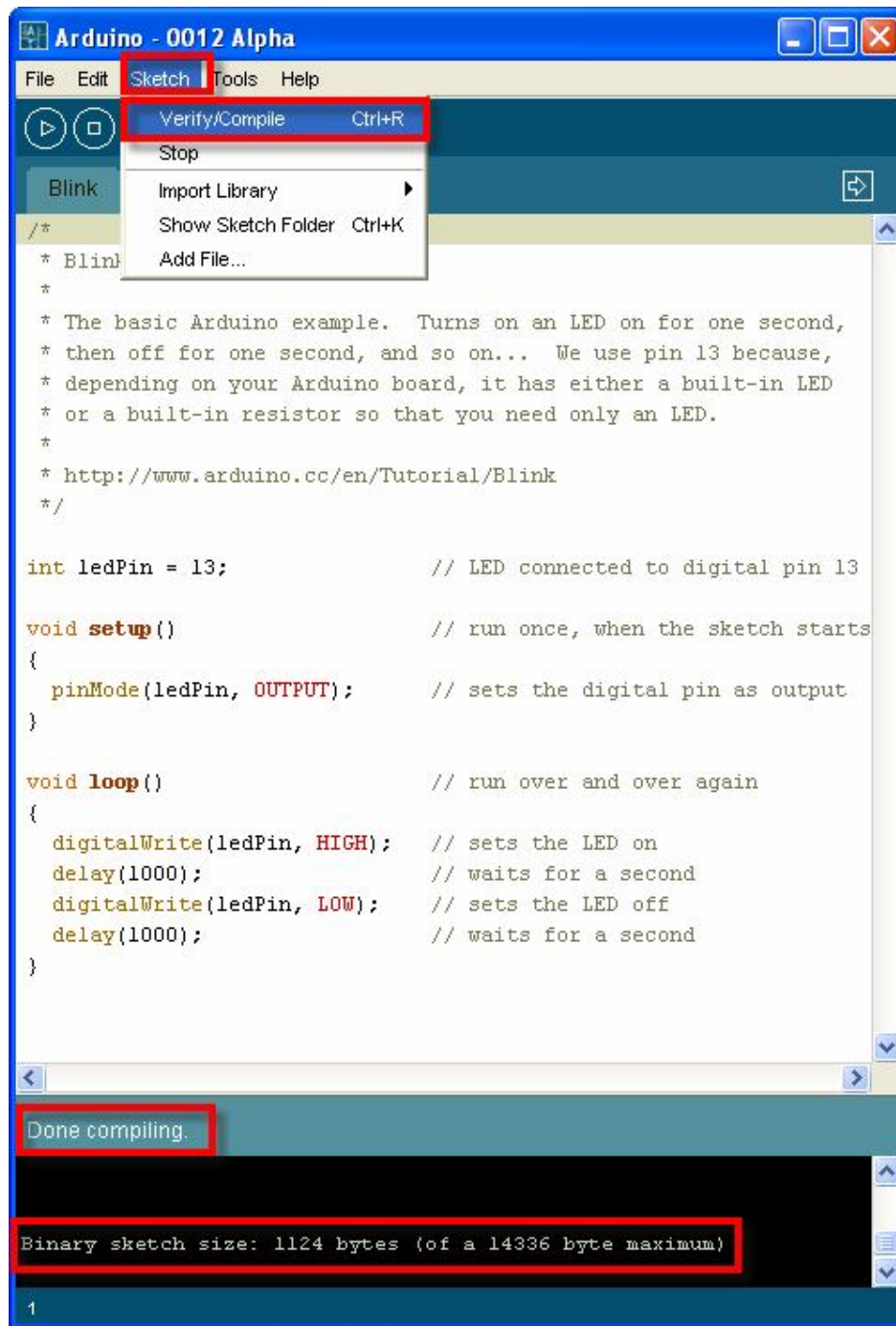


```
int ledPin = 13;

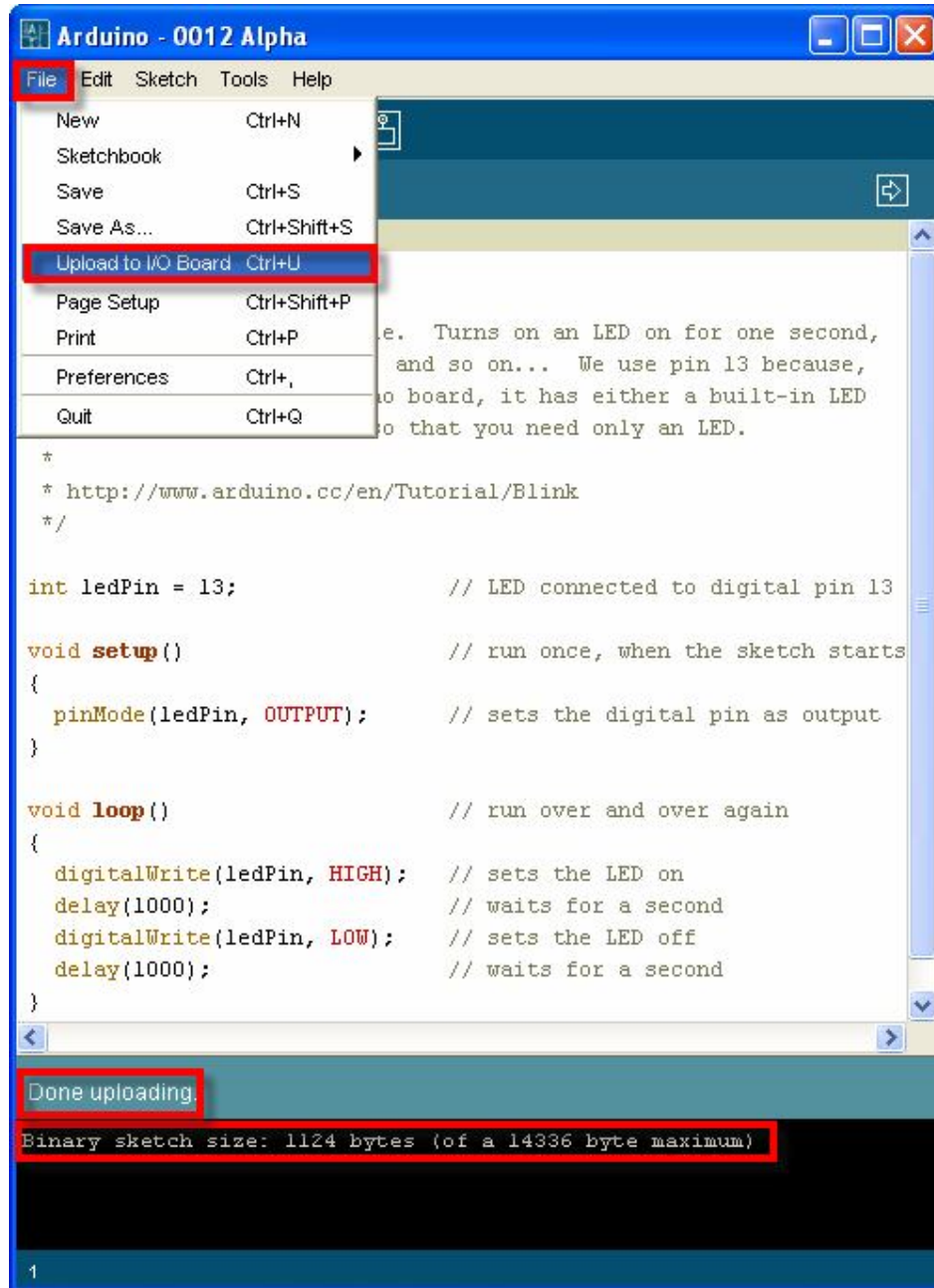
void setup()
{
  pinMode(ledPin, OUTPUT);
}

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

การพัฒนาโปรแกรมของ **ET-EASY AVR LCD** แบบ **AVR Microcontroller**

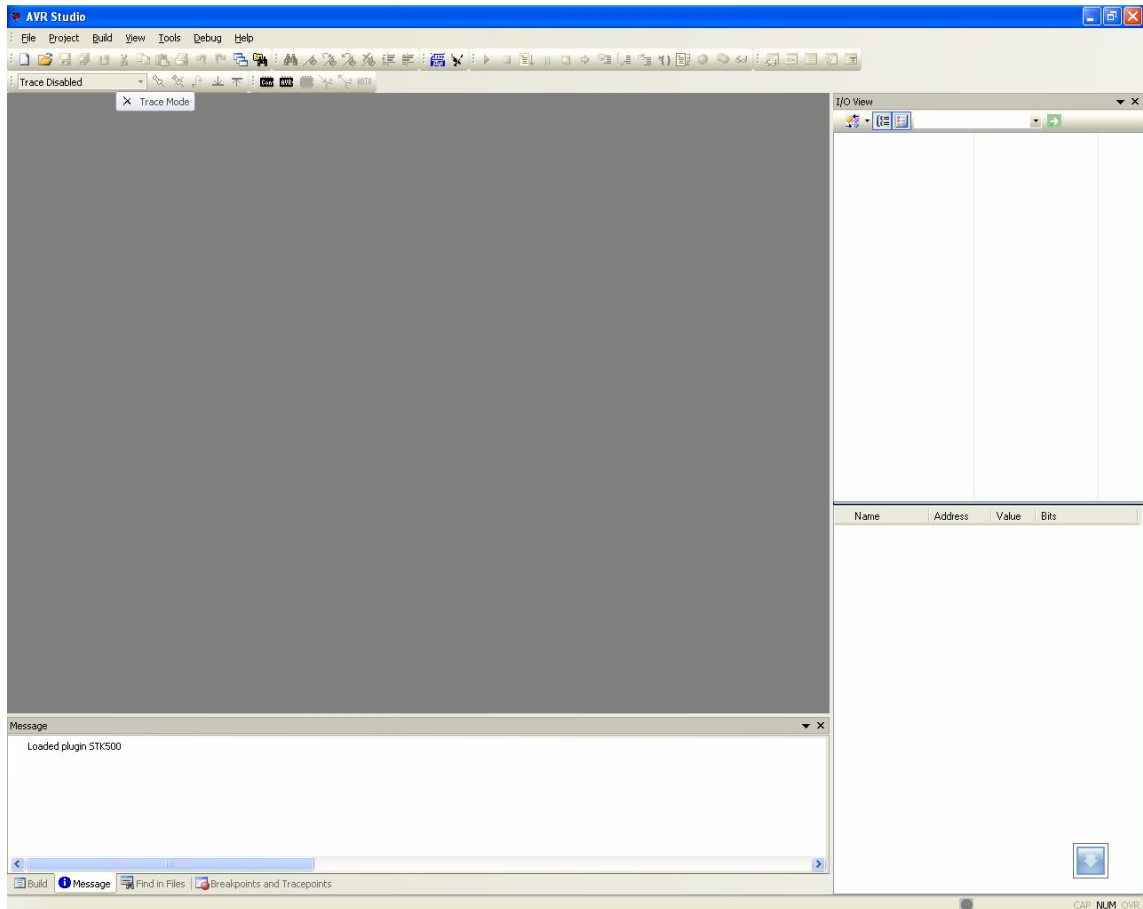
ในกรณีที่ผู้ใช้ต้องการที่จะพัฒนาโปรแกรมให้กับบอร์ด **ET-EASY AVR LCD** แบบปรกติ ในรูปแบบของไมโครคอนโทรลเลอร์ **AVR** ก็สามารถทำได้ โดยสามารถเลือกภาษาในการเขียนโปรแกรมได้เอง ซึ่งในกรณีนี้ผู้ใช้สามารถเลือกใช้ภาษาใดก็ได้ที่รองรับการใช้งานกับ **MCU** ตระกูล **AVR** เบอร์ **ATmega88** โดยผู้ใช้งานสามารถบริหารจัดการระบบทรัพยากรต่างๆที่อยู่ภายในตัว **ATmega88** ได้เองทั้งหมด โดยสามารถทำได้ 2 แนวทางคือ

- การพัฒนาโปรแกรมโดยใช้เครื่องโปรแกรมภายนอก ในกรณีนี้จะมีข้อดีคือ ไม่สูญเสียทรัพยากรใดๆเลย ผู้ใช้สามารถใช้งานและกำหนดคุณสมบัติการทำงานของทรัพยากรต่างๆที่มีอยู่ใน **MCU** ได้เอง ตามต้องการ แต่มีข้อเสียคือ ต้องมีเครื่องโปรแกรมภายนอก สำหรับทำหน้าที่โปรแกรม **Code** ให้กับ **MCU** โดยสามารถใช้ได้กับเครื่องโปรแกรมทุกรุ่นที่รองรับการใช้งานกับ **MCU** เบอร์ **ATmega88** และมีข้อต่อตามมาตรฐาน **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-EASY AVR LCD** 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

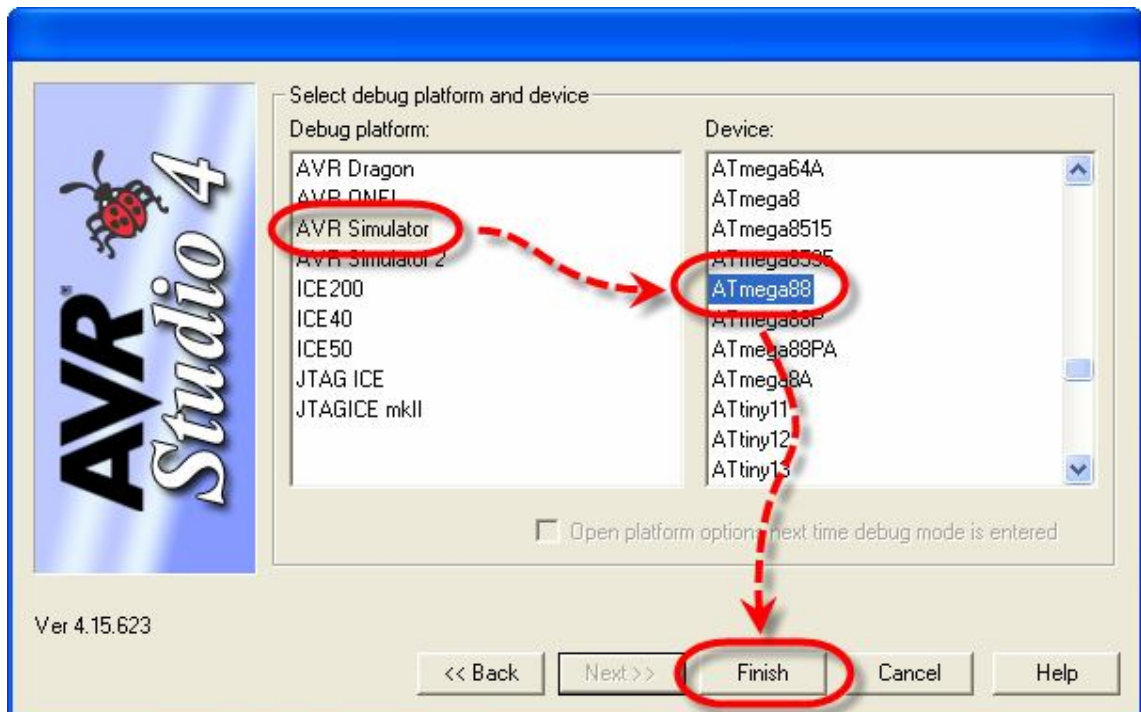
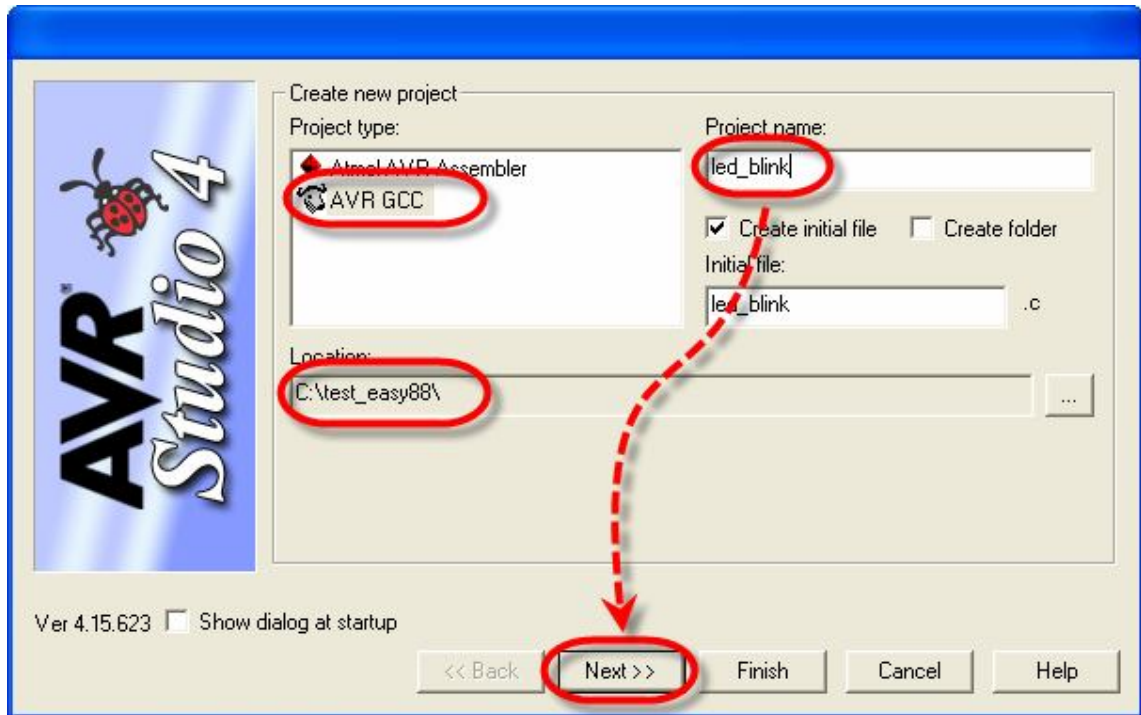
ตัวอย่างที่จะแสดงต่อไปนี้จะแสดงให้เห็นวิธีการพัฒนาโปรแกรมให้กับ ATmega88 ด้วยภาษาซี โดยใช้โปรแกรม AVR Studio4 ของ ATMEL ร่วมกับ Compiler ภาษาซีของ WinAVR

1. สั่ง Run Program AVR Studio4



- สั่งสร้าง project ใหม่ โดยเลือกที่ project → New project จากนั้นเลือกกำหนดตัวเลือกต่างๆ ให้กับโปรแกรดังนี้
 - Project type เลือกกำหนดเป็น AVR GCC
 - Location สำหรับบันทึก project ให้ระบุตำแหน่ง Folder ที่ต้องการใช้บันทึกไฟล์ และ Code ต่างของ project ในตัวอย่างกำหนดไว้ที่ "C:\test_easy88"
 - Project 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 Code โปรแกรม สำหรับทำหน้าที่เป็นไฟกระพริบที่ PB5 ซึ่งเป็น LED ที่ติดตั้งไว้บนบอร์ด ET-EASY AVR LCD อยู่แล้ว ดังตัวอย่าง

```

/*****/
/* Program Test LED Blinking */
/* Board : ET-EASY AVR LCD */
/* 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);
    }
}

```



```

AVR Studio - [C:\test_easy88\led_blink.c *]
File Project Build Edit View Tools Debug Window Help
Trace Disabled
AVR GCC
led_blink (default)
  Source Files
  Header Files
  External Dependencies
  Other Files

/* **** */
/* 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 = PB

/* **** */
/* 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 = O

```

- หลังจากพิมพ์ Code โปรแกรมเสร็จแล้วให้สั่งแปลโปรแกรม โดยเลือกที่ **build → rebuild all** ซึ่งผลจากการแปลคำสั่งจะได้เป็น **HEX File** ที่มีชื่อเดียวกันกับ **project** ที่สร้างไว้ ดังรูป

```

Build
Build started 29.12.2008 at 14:41:38
● avr-gcc -mmcu=atmega88 -Wall -gdwarf-2 -Os -std=gnu99 -funsigned-char -funsigned-bitfields -fp
● avr-gcc -mmcu=atmega88 -WL,-Map=led_blink.map led_blink.o -o led_blink.elf
● avr-objcopy -O ihex -R .eeprom -R .fuse -R .lock -R .signature led_blink.elf led_blink.hex
● avr-objcopy -j .eeprom --set-section-flags=.eeprom="alloc,load" --change-section-lma .eeprom=0
avr-objdump -h -S led_blink.elf > led_blink.lss

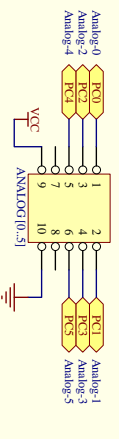
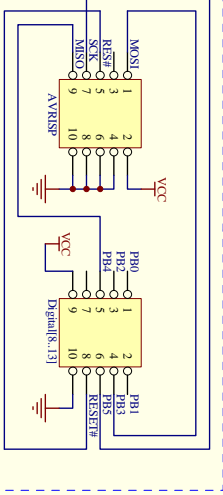
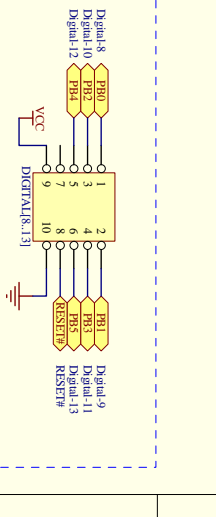
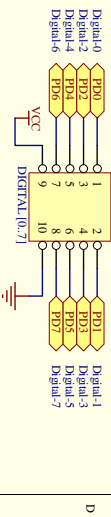
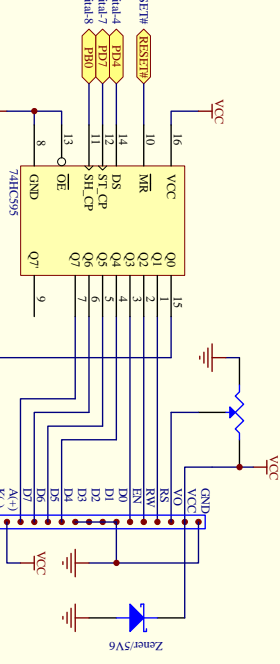
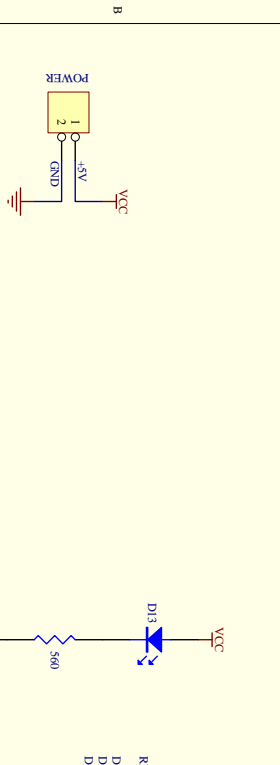
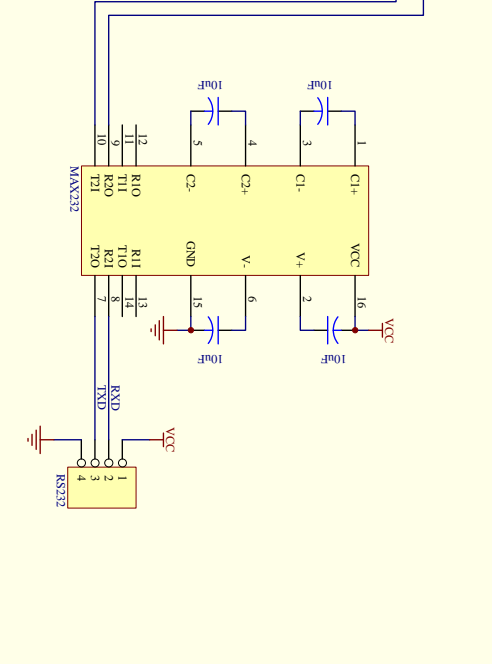
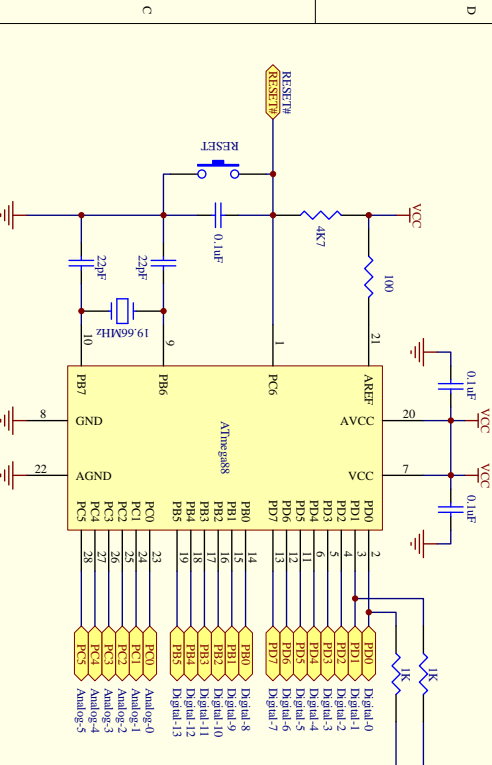
AVR Memory Usage
-----
Device: atmega88

Program: 188 bytes (2.3% Full)
(.text + .data + .bootloader)

Data: 0 bytes (0.0% Full)
(.data + .bss + .noinit)

Build succeeded with 0 Warnings...

```



Title: ETE-EASY AVR LCD/ATMEGA88/168/328 : 28PIN DIP

Number: ETT.CO.LTD./WWW.ETT.CO.TH

Size: B

Date: 2-Feb-2009

File: D:\eweb\88.rtdb