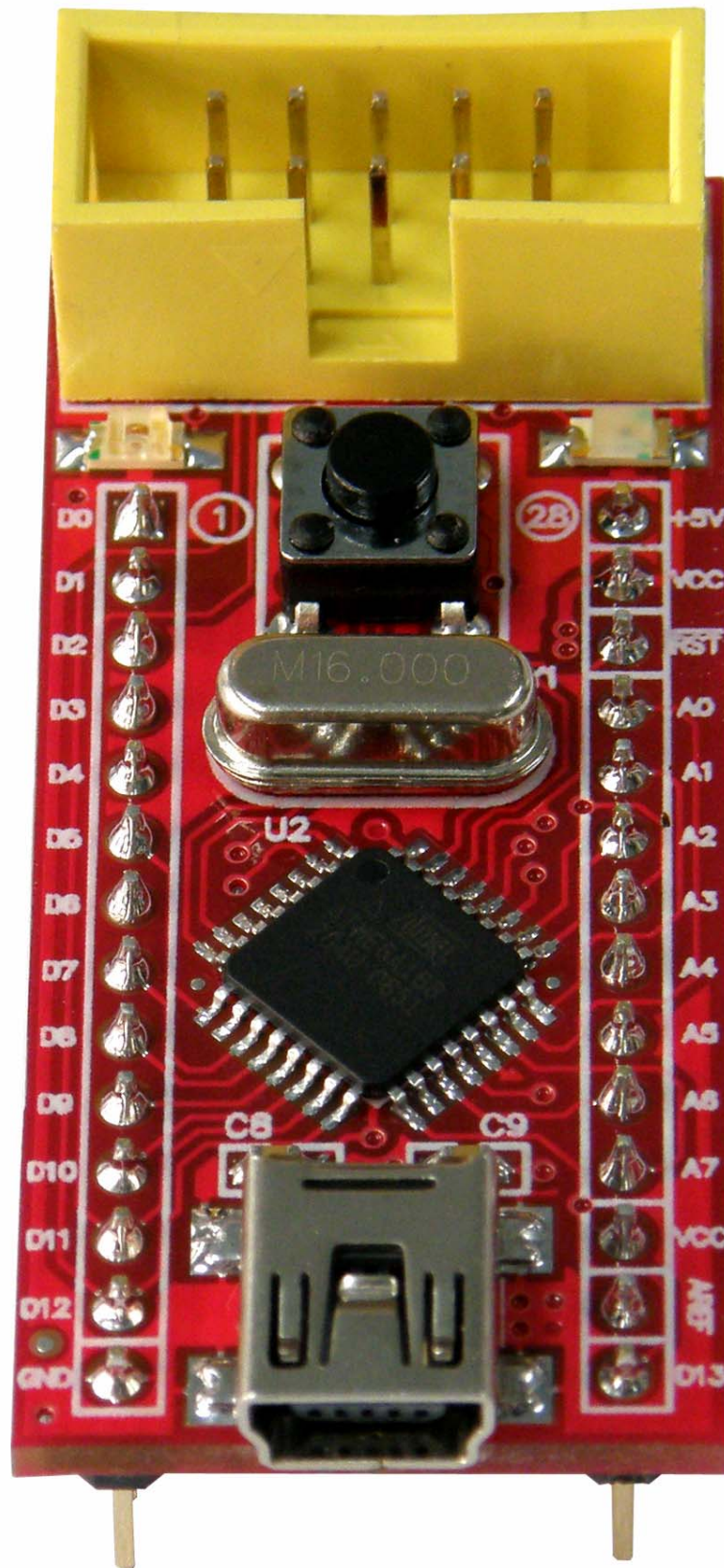


ET-EASY168 STAMP



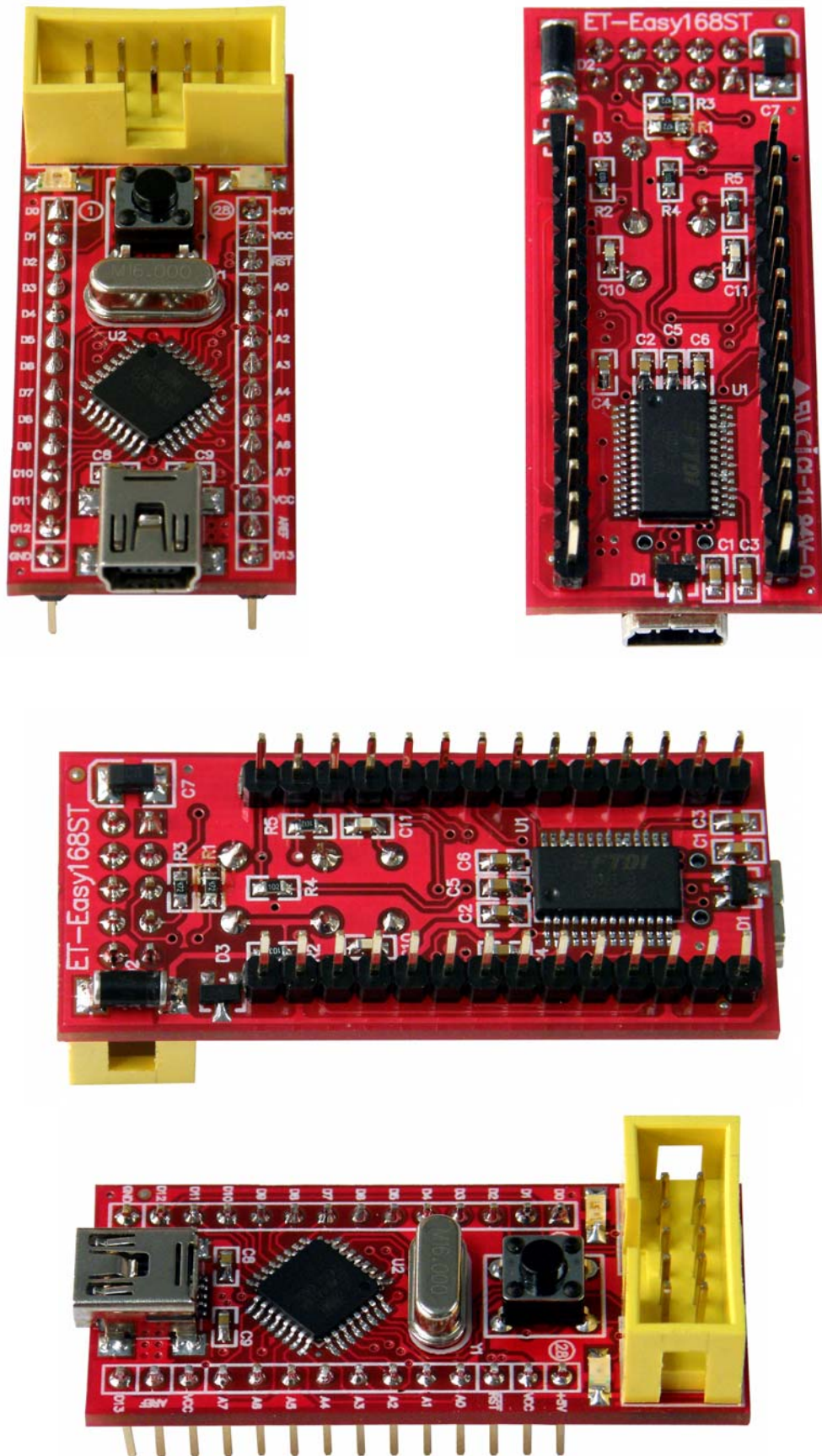
รูปแสดงโครงสร้างของบอร์ด ET-EASY168 STAMP

ET-EASY168 STAMP เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล AVR8 ขนาดเล็กจิ๋ว โดยมีขนาดของบอร์ดเพียง 2cm x 5cm เท่านั้น ซึ่งขนาดบอร์ด ประมาณเท่ากับตัวถังของไอซี 28 DIP 300 โดยเลือกใช้ไมโครคอนโทรลเลอร์ตระกูล AVR8 เบอร์ ATmega168 ของ ATMEL เป็น MCU ประจำบอร์ด โดยเลือกใช้ MCU ที่มีรูปร่างตัวถังแบบ 32 TQFP พร้อมวงจรรอบนอกที่จำเป็นอย่าง Oscillator และ Reset รวมไว้ด้วยภายในบอร์ด นอกจากนี้แล้วภายในตัวบอร์ดยังได้รวมเอาไอซี USB Bridge ของ FTDI เบอร์ FT232R เพื่อใช้ติดต่อสื่อสารแบบอนุกรมด้วย RS232 กับคอมพิวเตอร์ PC ผ่านทางพอร์ต USB ได้โดยตรง

ทำให้บอร์ด ET-EASY168 STAMP เป็นบอร์ดทดลองขนาดเล็กที่เพียบพร้อมไปด้วยวงจรพื้นฐานที่จำเป็นต่อการใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR8 อย่างแท้จริง เพียงแค่เสียบสาย USB จากพอร์ต USB ของเครื่องคอมพิวเตอร์ PC เข้ากับหัว USB ของบอร์ด ET-EASY168 STAMP ก็สามารถทำการเขียนโปรแกรม และ Download Code ให้กับ MCU เพื่อทำการทดลองได้ทันที

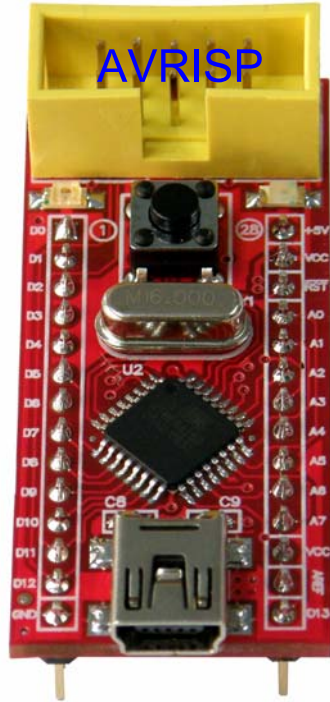
### คุณสมบัติของบอร์ด

- เลือกใช้ MCU ตระกูล AVR8 เบอร์ ATmega168 ของ ATMEL Run ความถี่ 16.00 MHz
  - มีหน่วยความจำ Flash สำหรับเขียนโปรแกรม 16KByte ถ้าใช้การพัฒนาโปรแกรมผ่านระบบ AVRISP หรือ 14Kbyte เมื่อใช้การพัฒนาโปรแกรมผ่านระบบ Boot Loader RS232
  - มี SRAM ใช้งานขนาด 1KByte และ EEPROM ใช้งานขนาด 512 Byte
  - มี GPIO ใช้งานจำนวน 22 บิต
    - Digital GPIO จำนวน 14 บิต
    - Analog Input (ADC) ขนาดความละเอียด 10บิต จำนวน 8 ช่อง
- ใช้งานกับแรงดันไฟตรงขนาด +5VDC โดยใช้ได้ทั้งกับแหล่งจ่าย +5VDC/500mA จากพอร์ต USB และจากแหล่งจ่าย +5VDC จากภายนอกได้ด้วย พร้อม LED Power แสดงสถานะของแหล่งจ่าย
- มีวงจร External Reset แบบ RC Reset และ Switch Reset พร้อมภายในบอร์ด
- หัวต่อใช้งานวางตัวบน Pin Header ระยะห่าง 2.54mm(100mil) ขนาด 28 Pin (ด้านละ14Pin) ระยะห่าง 600mil(1.5cm) ง่ายต่อการนำไปต่อประยุกต์ใช้งาน และ ขยายวงจร I/O สามารถใช้กับ Project Board และ PCB เอนกประสงค์ได้โดยง่าย
- มีหัวต่อ USB สำหรับเชื่อมต่อสื่อสารกับคอมพิวเตอร์ PC ผ่าน USB Bridge ของ FTDI ในรูปแบบของการสื่อสารอนุกรม RS232 สำหรับใช้งานสื่อสารและ Download Code ให้กับ MCU ในบอร์ด
- มีหัว AVRISP แบบ IDE 10PIN สำหรับใช้ Download โปรแกรมให้กับ MCU ภายในบอร์ดในกรณีไม่ต้องการใช้การพัฒนาโปรแกรมผ่านทาง Boot Loader
- มี LED แสดงสถานะ โดยต่อกับ PB5 ของ AVR (Digital-13 ของ Arduino Project) สำหรับใช้เป็นอุปกรณ์ทดลองการทำงานอย่างง่าย

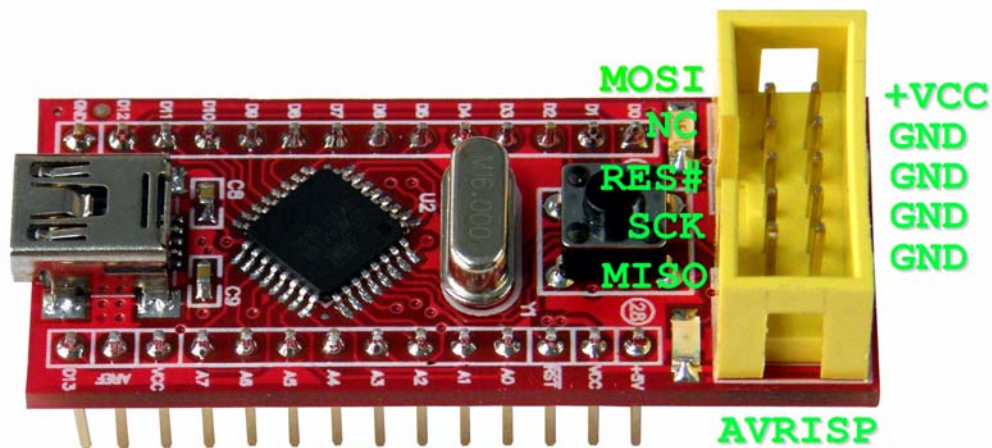


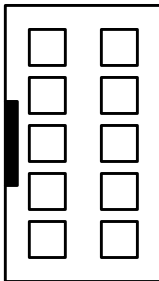
รูปแสดง ลักษณะของบอร์ด ET-EASY168 STAMP



AVR	Arduino	Pin	ET-EASY168 STAMP	Pin	Arduino	AVR
PD0	Digital-0	1		28	+5V(+Vin)	+5V(+Vin)
PD1	Digital-1	2		27	+VCC(+5V)	+VCC(+5V)
PD2	Digital-2	3		26	RESET#	RESET(PC6)
PD3	Digital-3	4		25	Analog-0	PC0/ADC0
PD4	Digital-4	5		24	Analog-1	PC1/ADC1
PD5	Digital-5	6		23	Analog-2	PC2/ADC2
PD6	Digital-6	7		22	Analog-3	PC3/ADC3
PD7	Digital-7	8		21	Analog-4	PC4/ADC4
PB0	Digital-8	9		20	Analog-5	PC5/ADC5
PB1	Digital-9	10		19	Analog-6	ADC6
PB2	Digital-10	11		18	Analog-7	ADC7
PB3	Digital-11	12		17	+VCC(+5V)	+VCC(+5V)
PB4	Digital-12	13		16	+AREF	+AREF
GND	GND	14		15	Digital-13	PB5

ตารางแสดง การจัดสรรขาสัญญาณของบอร์ด ET-EASY168 STAMP



AVR	Arduino	Pin	AVRISP	Pin	Arduino	AVR
PB3	Digital-11	MOSI		+VCC	+VCC	+VCC
-	-	NC		GND	GND	GND
RES#	RES#	RES#		GND	GND	GND
PB5	Digital-13	SCK		GND	GND	GND
PB4	Digital-12	MISO		GND	GND	GND

### หน้าที่ของขาสัญญาณในการใช้งานแบบ “Arduino Project”

- +5V(+Vin) เป็นขาสำหรับใช้เป็นจุดรับแรงดันขนาด +5VDC จากภายนอกเพื่อใช้เป็นแหล่งจ่ายไฟเลี้ยงให้กับบอร์ด
- +VCC(+5V) เป็นขาแหล่งจ่ายไฟจุดเดียวกันกับที่ป้อนให้กับ +VCC ของ MCU ซึ่งจุดนี้จะรับแรงดันมาจาก 2 แหล่ง ด้วยกันคือ ขารับแรงดัน +5V(+Vin) จากขา 28 ของบอร์ด และ จากขา +VUSB(+5V) จากขั้ว USB ของบอร์ด โดยมี Diode ป้องกันการย้อนกลับของแรงดันไว้แล้ว
- +AREF เป็นขาสำหรับรับสัญญาณแรงดันอ้างอิง (Analog Reference) ให้กับวงจร Analog Input ในกรณีที่ต้องการใช้แรงดันอ้างอิงจากภายนอก
- RESET# เป็นขาสัญญาณ RESET ของ CPU ทำงานที่ Logic “0”
- Digital[0..13] เป็นขา I/O แบบ Digital สามารถใช้งานเชื่อมต่อกับสัญญาณ Logic TTL (5V) ต่างๆ
- Analog[0..7] เป็นขา Input แบบ Analog สามารถรับ Input แบบ Analog 0..+5V

### หน้าที่ของขาสัญญาณในการใช้งานแบบ “AVR Micro Controller”

- +5V(+Vin) เป็นขาสำหรับใช้เป็นจุดรับแรงดันขนาด +5VDC จากภายนอกเพื่อใช้เป็นแหล่งจ่ายไฟเลี้ยงให้กับบอร์ด
- +VCC(+5V) เป็นขาแหล่งจ่ายไฟจุดเดียวกันกับที่ป้อนให้กับ +VCC ของ MCU ซึ่งจุดนี้จะรับแรงดันมาจาก 2 แหล่ง ด้วยกันคือ ขารับแรงดัน +5V(+Vin) จากขา 28 ของบอร์ด และ จากขา +VUSB(+5V) จากขั้ว USB ของบอร์ด โดยมี Diode ป้องกันการย้อนกลับของแรงดันไว้แล้ว
- +AREF เป็นขาสำหรับรับสัญญาณแรงดันอ้างอิง (Analog Reference) ให้กับวงจร Analog Input ในกรณีที่ต้องการใช้แรงดันอ้างอิงจากภายนอก
- RESET# เป็นขาสัญญาณ RESET ของ CPU ทำงานที่ Logic “0”
- PB[0..5] เป็นขา I/O แบบ Digital สามารถใช้งานเชื่อมต่อกับสัญญาณ Logic TTL (5V) ต่างๆ
- PD[0..7] เป็นขา I/O แบบ Digital สามารถใช้งานเชื่อมต่อกับสัญญาณ Logic TTL (5V) ต่างๆ
- PC[0..5] เป็นขา I/O ซึ่งสามารถกำหนดให้เป็นได้ทั้ง Digital และ Analog Input
- ADC6,ADC7 เป็นขา Input แบบ Analog สามารถรับ Input แบบ Analog 0..+5V

## การพัฒนาโปรแกรมของบอร์ด ET-EASY168 STAMP

ในการพัฒนาโปรแกรมของบอร์ด ET-EASY168 STAMP นั้น ผู้ใช้สามารถเลือกใช้วิธีการพัฒนาโปรแกรมได้ 2 รูปแบบด้วยกัน คือ

- **AVR Micro Controller** เป็นการพัฒนาโปรแกรมตามรูปแบบของ AVR Micro Controller ปรกติ ซึ่งสามารถเลือกใช้โปรแกรมภาษาใดก็ได้ที่รองรับการใช้งานร่วมกับ AVR เบอร์ ATmega168 ซึ่งผู้ใช้สามารถเลือกใช้โปรแกรมในการพัฒนาได้ตามความถนัด เช่น ภาษาเบสิก BASCOM-AVR หรือ ภาษาซี เช่น Code Vision และ WinAVR เป็นต้น
- **Arduino Project** เป็นการพัฒนาโปรแกรมโดยใช้โปรแกรมและชุดคำสั่งในการเขียนโปรแกรมด้วย ภาษาซี (C++) ของ “Arduino Project” ซึ่งเป็นโครงการพัฒนาไมโครคอนโทรลเลอร์ AVR แบบ Open Source ซึ่งกำลังได้รับความนิยมอย่างแพร่หลาย เนื่องจากเป็นโครงการที่เปิดเผยทั้ง Source Code ในการพัฒนาให้ทั้งหมดและยังมีตัวอย่างโครงงานพร้อมตัวอย่างโปรแกรมการทดลองต่างๆ แจกจ่ายให้ผู้สนใจนำมาใช้ศึกษา เรียนรู้และทดลอง ได้ฟรี โดยไม่เสียค่าใช้จ่ายใดๆ ผู้สนใจสามารถเข้าไปค้นหารายละเอียดต่างๆ ของ Arduino Project นี้ได้ <http://www.arduino.cc/>

\*\*\*\*\*

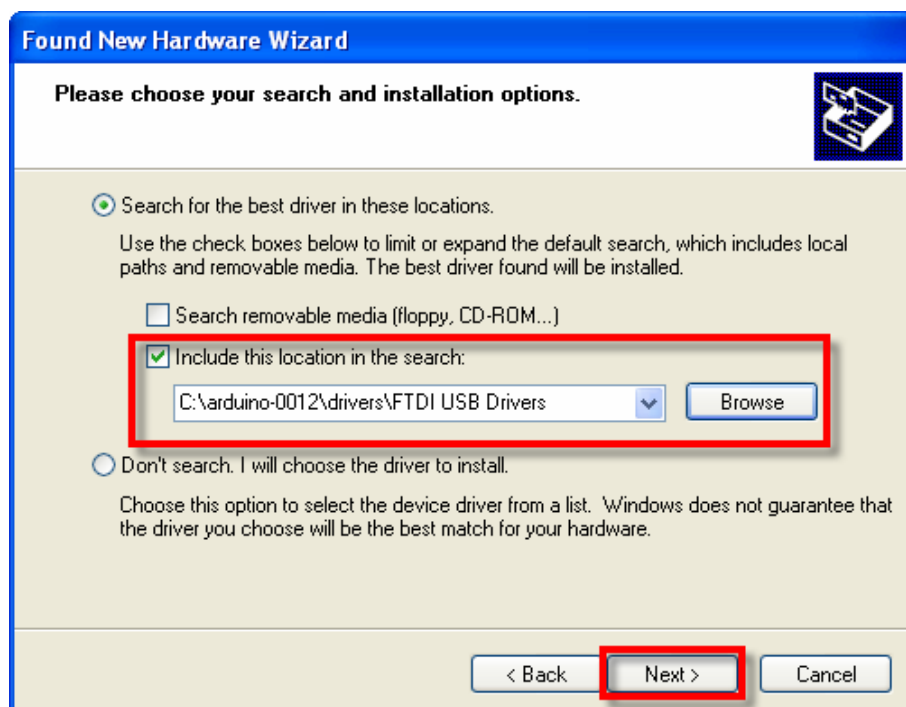
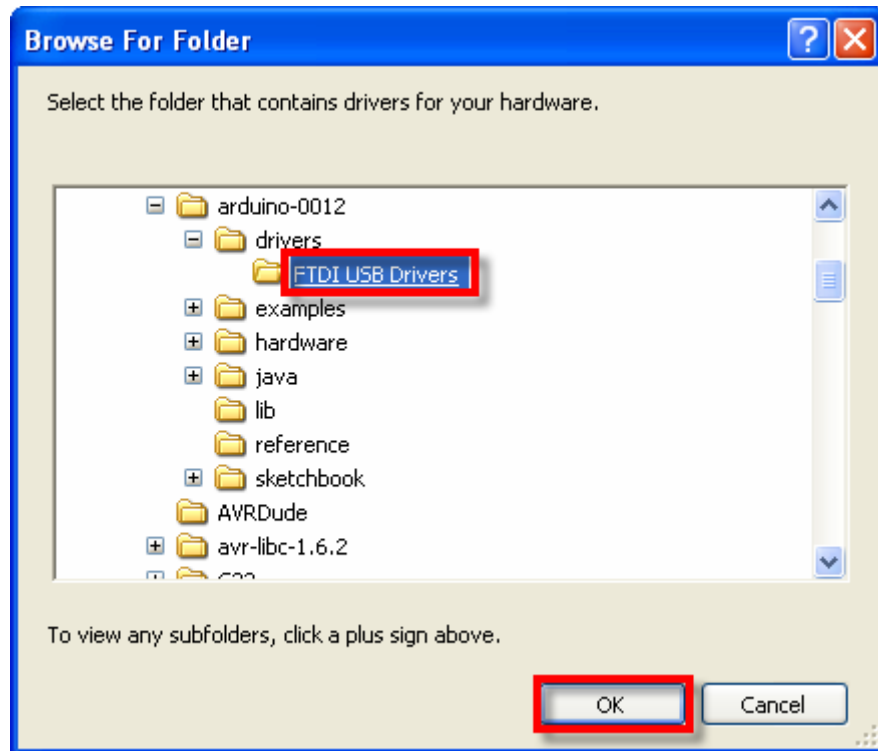
## การติดตั้ง Driver ของ USB Bridge ของบอร์ด ET-EASY168 STAMP

บอร์ด ET-EASY168 STAMP จะใช้ชิพ USB Bridge ของ FTDI เป็นตัวกลางในการเชื่อมต่อกับคอมพิวเตอร์ PC โดย USB Bridge ของ FTDI จะทำหน้าที่เป็นตัวกลางในการเชื่อมต่อและติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ PC กับ MCU ATmega168 ของบอร์ด ET-EASY168 STAMP ในรูปแบบของพอร์ตอนุกรม (Visual Com Port) โดยโปรแกรม Application ต่างๆที่ทำงานอยู่บนคอมพิวเตอร์ PC รวมทั้งโปรแกรม Arduino จะมองเห็น พอร์ต USB ที่เชื่อมต่อกับบอร์ด ET-EASY168 STAMP เป็นพอร์ตสื่อสารอนุกรม (Com Port) ช่องหนึ่งเท่านั้น ซึ่งถ้าเครื่องคอมพิวเตอร์ของผู้ใช้เคยทำการติดตั้ง Driver สำหรับ USB Bridge ของ FTDI ไว้ก่อนแล้ว เมื่อทำการเชื่อมต่อสาย USB ของบอร์ด ET-EASY168 STAMP เข้ากับ USB HUB ของเครื่องคอมพิวเตอร์ PC แล้ว Windows จะทำการติดตั้ง Driver ให้เองโดยอัตโนมัติ แต่ถ้าเครื่องคอมพิวเตอร์ PC ยังไม่เคยติดตั้ง Driver ของ FTDI ไว้ก่อนก็จะต้องทำการติดตั้ง Driver ให้กับบอร์ดให้เรียบร้อยเสียก่อนซึ่งมีลำดับขั้นตอนดังนี้

1. เตรียมแผ่น CD ROM ที่บรรจุ Driver ของ FTDI ไว้ให้พร้อม หรือ ในกรณีที่ผู้ใช้ได้ทำการติดตั้งโปรแกรมของ Arduino ไว้เรียบร้อยแล้ว ภายในโฟลเดอร์ของโปรแกรม Arduino ก็จะมี Driver ของ FTDI จัดเตรียมไว้ให้เรียบร้อยแล้ว โดยจะอยู่ที่ “C:\arduino-0012\drivers\FTDI USB Drivers”
2. ทำการเสียบสาย USB ของบอร์ด ET-EASY168 STAMP เข้ากับพอร์ต USB HUB ของเครื่องคอมพิวเตอร์ PC ซึ่ง Windows จะตรวจพบอุปกรณ์ใหม่ โดยเป็น “FT232R USB UART” และแจ้งให้ผู้ใช้ทำการติดตั้ง Driver ให้กับอุปกรณ์ ดังรูป

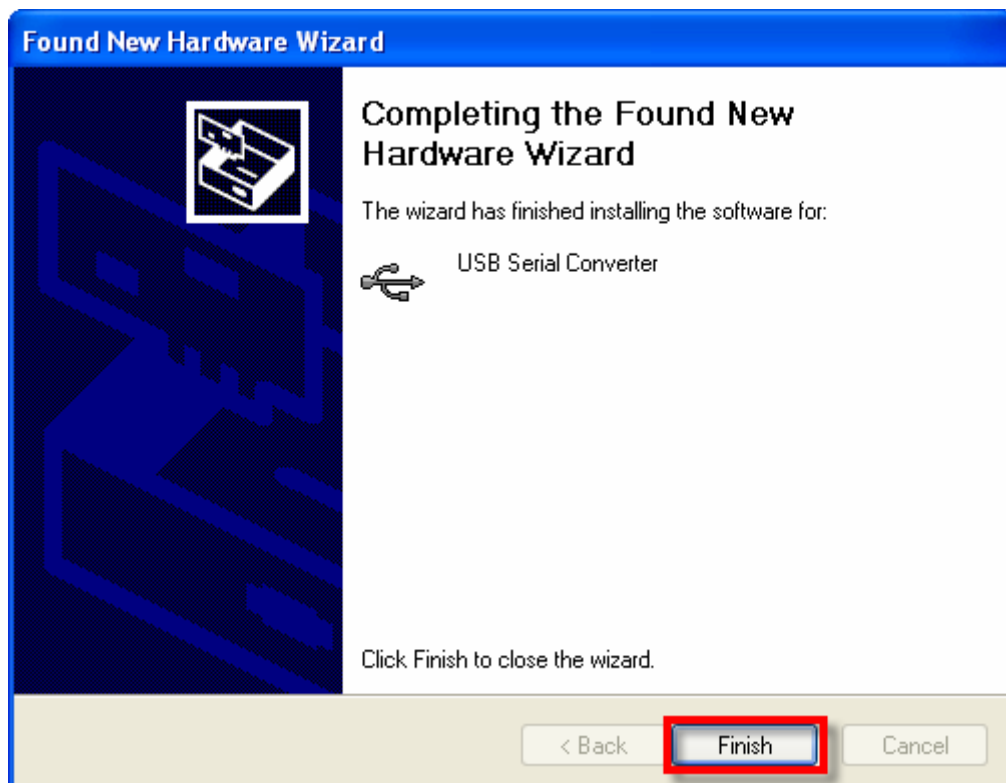


3. ให้เลือก Install from list or specific location(Advanced) แล้วเลือก Next ซึ่ง Windows ก็จะแจ้งให้ผู้ใช้ระบุตำแหน่งไฟล์ไดรเวอร์ที่บรรจุไฟล์ Driver ของ FTDI ไว้ ก็ให้เลือกที่ Browse และเลือกไปยัง Drive และ ไฟล์ไดรเวอร์ที่เก็บไฟล์ Driver ไว้ ซึ่งถ้าผู้ใช้ได้ทำการติดตั้งโปรแกรมของ Arduino ไว้แล้ว ก็ให้เลือกไปที่ "C:\arduino-0012\drivers\FTDI USB Drivers" แล้วเลือก Next ดังรูป

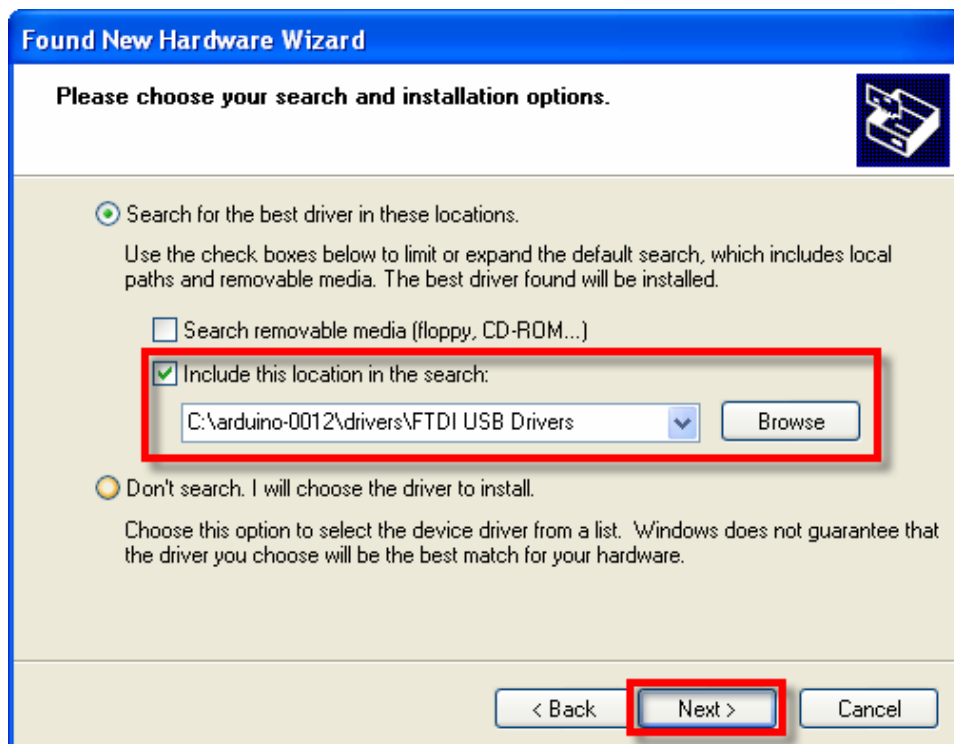
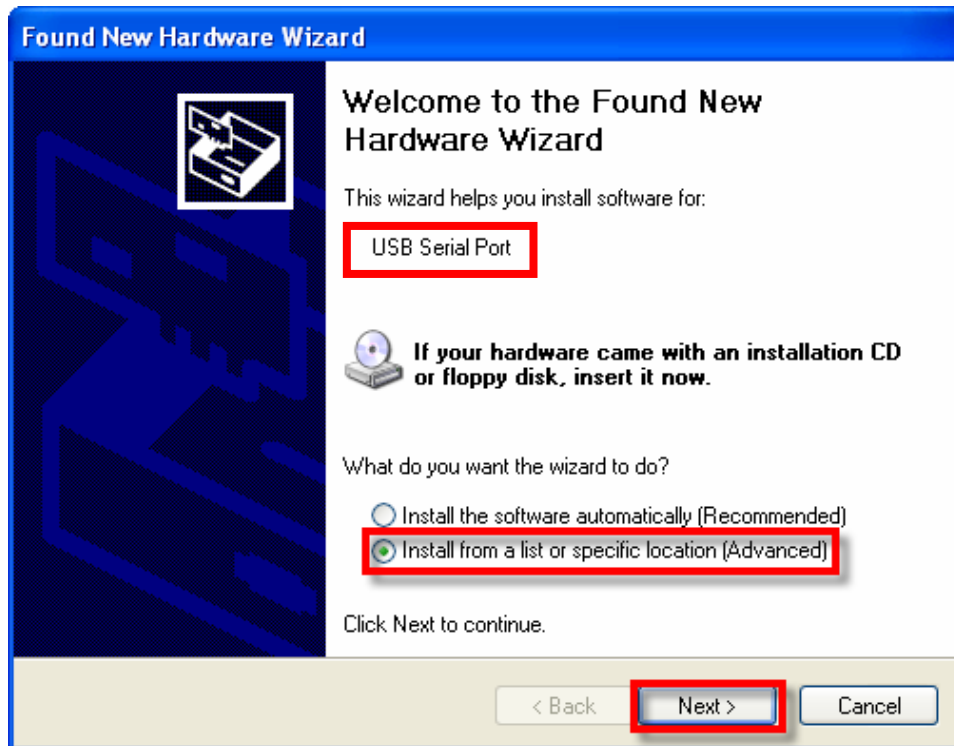




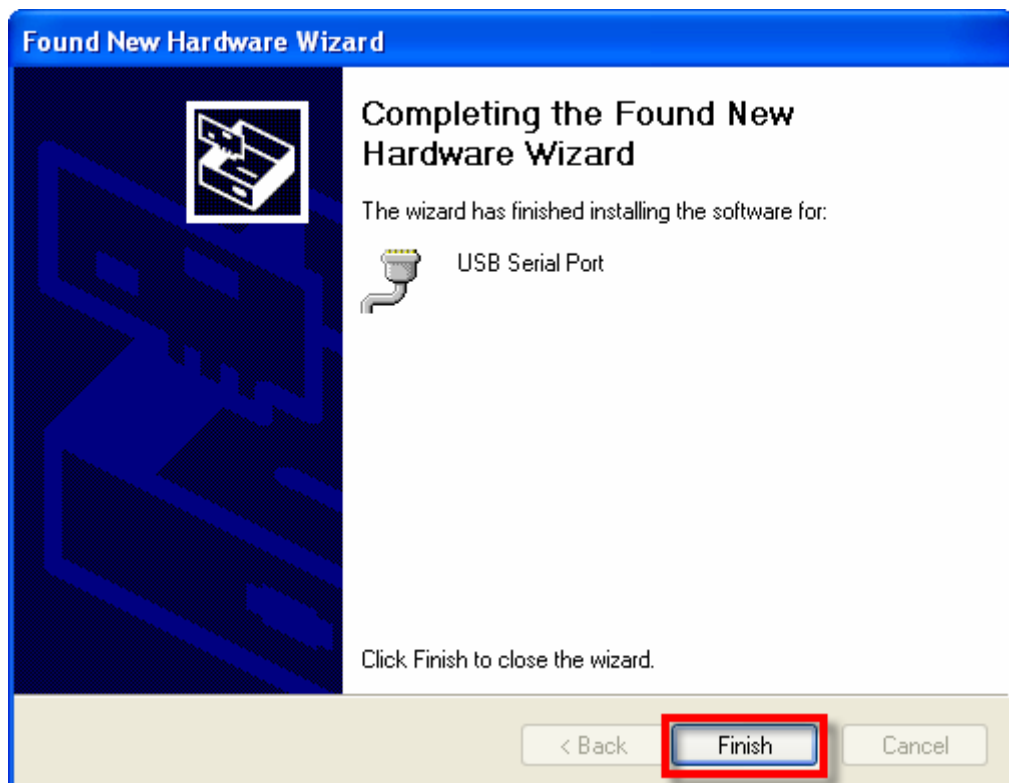
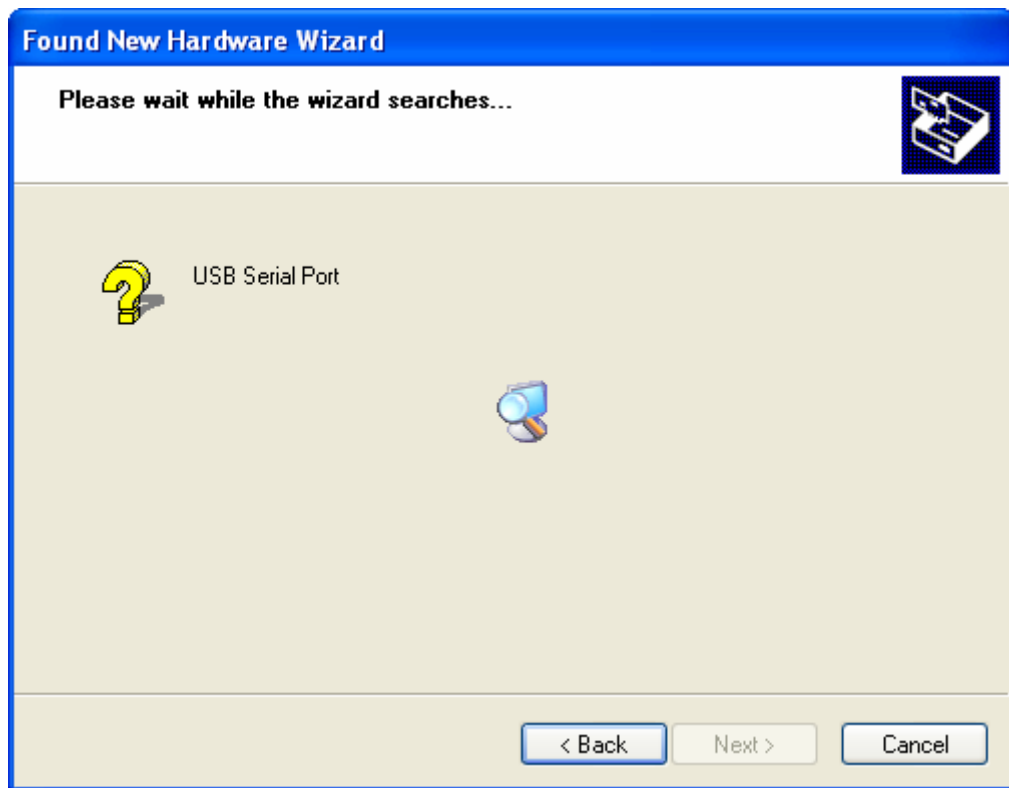
4. ในขั้นตอนโปรแกรม Windows จะทำการค้นหาและติดตั้ง Driver ให้กับอุปกรณ์ ให้รอสักครู่จนการทำงานเสร็จเรียบร้อยแล้ว แล้วเลือก Finish ดังรูป



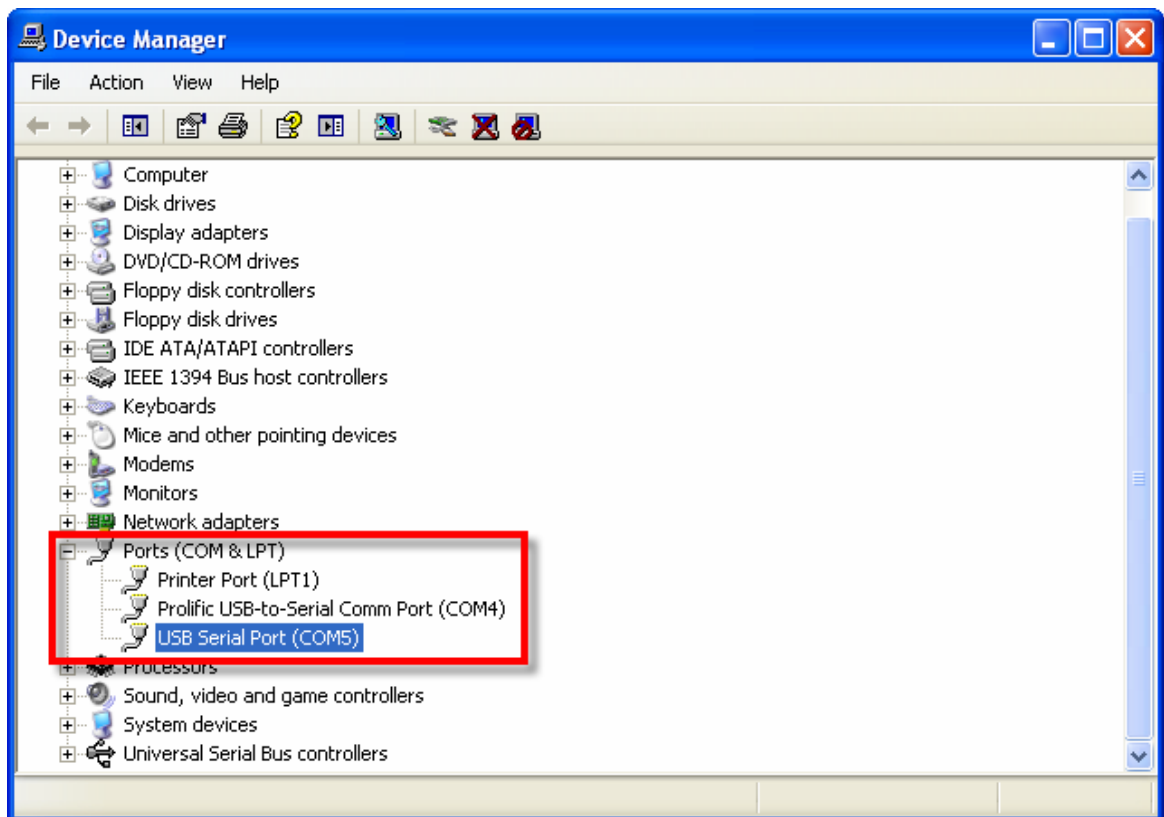
5. หลังจากทำการติดตั้ง Driver ของฮาร์ดแวร์เรียบร้อยแล้ว Windows ก็จะตรวจพบว่ามีอุปกรณ์ใหม่ ถูกเชื่อมต่ออยู่ โดยเป็นอุปกรณ์ประเภท “USB Serial Port” และแจ้งให้ผู้ใช้ทำการติดตั้ง Driver ให้กับอุปกรณ์ใหม่ที่ระบุเป็น “USB Serial Port” อีกครั้งหนึ่ง ซึ่งก็ให้เลือกระบุตำแหน่งไฟล์ไดรเวอร์ที่เก็บไฟล์ Driver ไว้ ซึ่งให้เลือกเหมือนขั้นตอนในหัวข้อที่ 3 ดังรูป



6. ในขั้นตอนนี้โปรแกรม Windows จะทำการค้นหาและติดตั้ง Driver ให้กับอุปกรณ์ ให้รอสักครู่จนการทำงานเสร็จเรียบร้อยแล้ว แล้วเลือก Finish ดังรูป

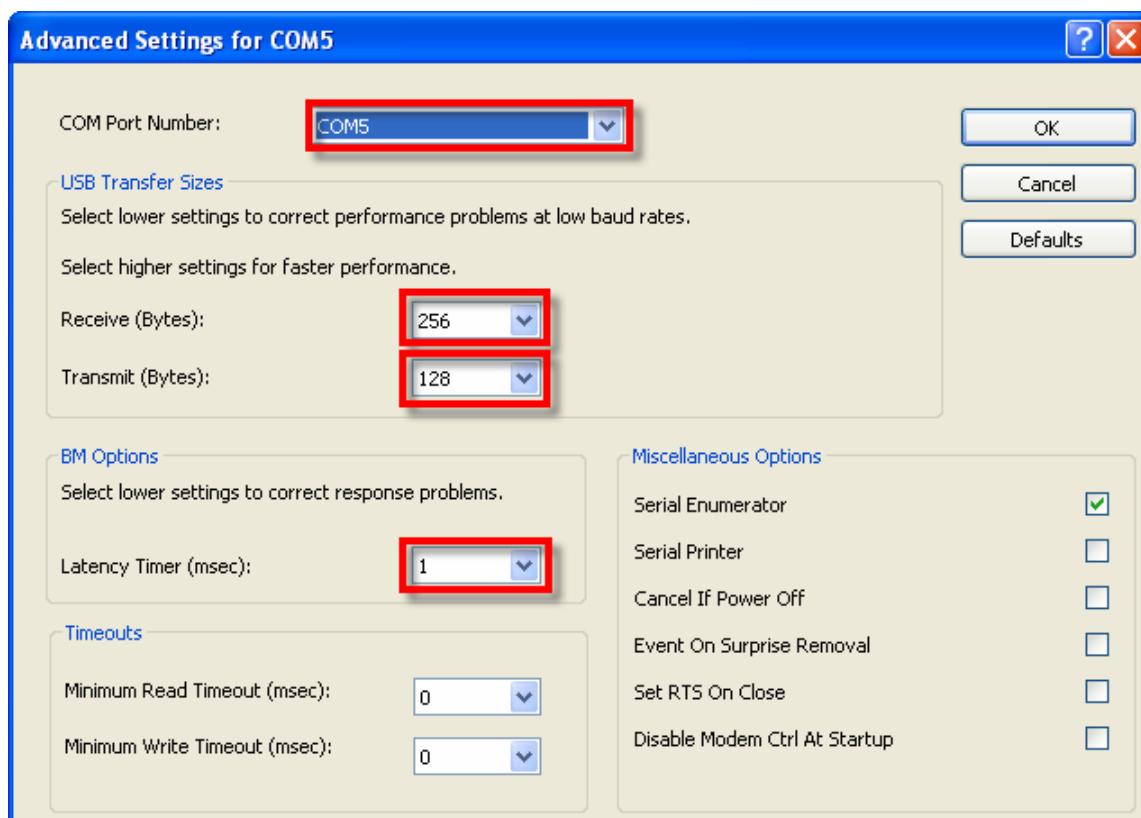
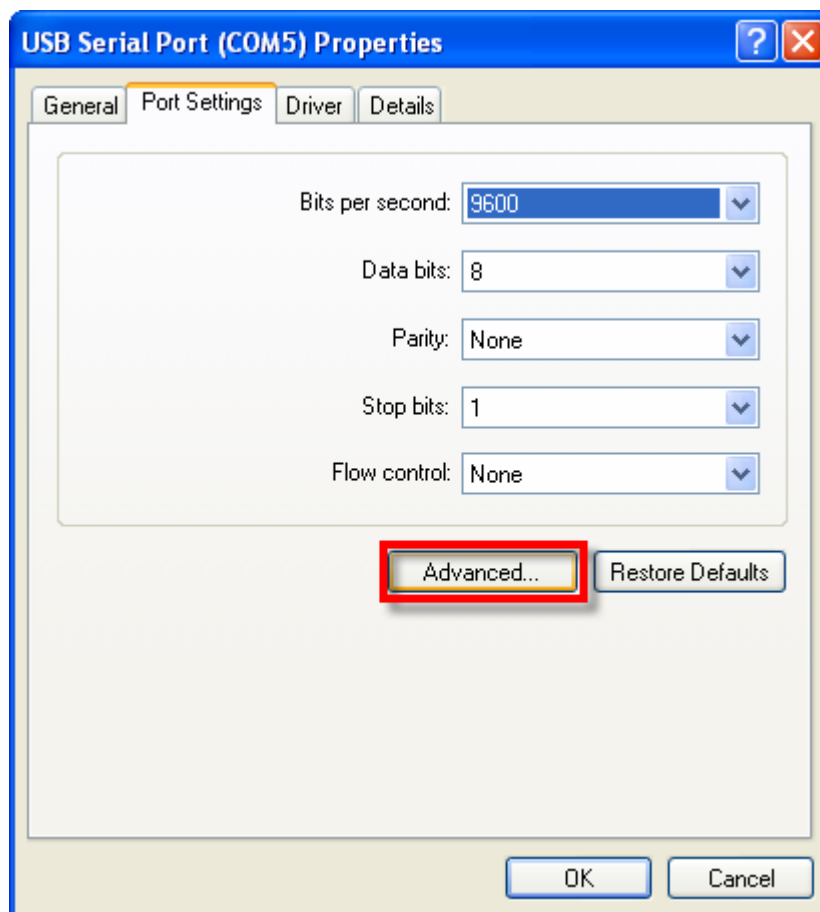


7. หลังจากทำการติดตั้ง Driver เรียบร้อยแล้ว ก็สามารถใช้งานอุปกรณ์ได้แล้ว แต่เพื่อความถูกต้องในครั้งแรกนี้ควรต้องเข้าไปทำการตรวจสอบและปรับแต่งค่าให้กับอุปกรณ์ก่อน โดยในขั้นตอนนี้ให้ไปที่ “My Computer → Control Panel → System → Hardware → Device Manager” แล้วทำการตรวจสอบที่ Ports (COM&LPT) แล้วดูที่ชื่อของ “USB Serial Port” ซึ่งให้ผู้ใช้งานจดจำหมายเลขของ Com Port ของอุปกรณ์ดังกล่าวไว้ เพื่อใช้อ้างอิงถึงในการเรียกใช้งาน ดังรูป



8. ในขั้นตอนนี้ให้คลิกเมาส์ที่เครื่องหมาย (+) หน้าหัวข้อ Ports(COM&LPT) แล้วมองหาอุปกรณ์ที่ชื่อ “USB Serial Port” ตามที่เราได้ทำการติดตั้ง Driver ไว้เรียบร้อยแล้ว หรือ ถ้าไม่แน่ใจว่าอุปกรณ์ดังกล่าวใช่อุปกรณ์ที่เป็นของบอร์ด “ET-EASY168 STAMP” หรือไม่ ให้ทดสอบด้วยการถอดสาย USB ออก รายชื่ออุปกรณ์ดังกล่าวจะต้องหายไป แต่เมื่อเสียบสาย USB กลับเข้ามาใหม่ รายชื่อของอุปกรณ์ดังกล่าวก็จะต้องปรากฏให้เห็นอีกครั้ง ถ้าทุกอย่างถูกต้อง ก็ให้ทำการคลิกเมาส์ที่ Tab รายชื่อของอุปกรณ์ดังกล่าว เมื่อปรากฏหน้าต่าง USB Serial Port Properties ขึ้นมาแล้วให้เลือกที่ Port Setting แล้วเลือก Advance เพื่อเข้าไปกำหนดค่าให้กับอุปกรณ์ให้เรียบร้อย ดังนี้
- USB Transfer Size → Receive (Bytes) ให้กำหนดเป็น 256
  - USB Transfer Size → Transmit (Bytes) ให้กำหนดเป็น 128
  - BM Option → Latency Timer (mSec) ให้กำหนดเป็น 1





## การพัฒนาโปรแกรมของ ET-EASY168 STAMP แบบ Arduino Project

ตามปกติแล้วบอร์ด ET-EASY168 STAMP จะทำการ ติดตั้งโปรแกรม Bootloader ไว้ให้เรียบร้อยแล้ว โดยใช้ Bootloader ชื่อ “BOOT\_EASY168\_AUTO\_16MHZ.HEX” ซึ่งเป็น Bootloader ที่ทางทีมงาน อีทีที ได้นำต้นฉบับจาก Arduino มาปรับปรุงเงื่อนไขการทำงานใหม่ เพื่อให้การทำงานสอดคล้องกับระบบ ฮาร์ดแวร์ของบอร์ด ET-EASY168 STAMP ได้ดียิ่งขึ้น โดยโปรแกรม Bootloader นี้จะใช้สำหรับ ติดต่อสื่อสารเพื่อส่ง Upload Code จากคอมพิวเตอร์ PC ให้กับ MCU ในบอร์ดทำงาน โดยไม่ต้องใช้เครื่อง โปรแกรมภายนอกให้ยุ่งยาก ซึ่ง คุณสมบัติของ Bootloader ที่ทาง อีทีที ปรับปรุงเพิ่มเติมขึ้น มีคุณสมบัติ การทำงานเป็นดังนี้

- สื่อสารกับโปรแกรมภายนอกด้วย Protocol แบบ STK500 (STK500V1)
- ใช้ความเร็ว Baudrate 19200 โดยใช้ความถี่ XTAL 16 MHz
- โปรแกรม Bootloader มีขนาด 2KByte ทำงานที่ตำแหน่ง 0x3800-0x3FFF
- ใช้ LED ที่ต่อกับขา Digital-13(PB5) เป็นตัวแสดงสถานะในขณะที่ Bootloader ทำงาน
- โปรแกรมใน Bootloader จะทำงานโดยอัตโนมัติทุกครั้งหลังการรีเซ็ต โดย MCU จะเริ่มต้นทำงาน ใน Bootloader นี้ก่อนเสมอ ซึ่งถ้าไม่มีการติดต่อสื่อสารจากโปรแกรม Arduino ภายในเวลา ประมาณ 3 วินาที MCU ก็จะออกจากการทำงานใน Bootloader เพื่อไปเริ่มต้นทำงานตามคำสั่งที่เป็นของผู้ใช้ โดยอัตโนมัติ โดยในขณะที่ Bootloader ทำงาน ตอนเริ่มต้นจะเห็น LED ที่ต่อไว้กับ ขาสัญญาณ Digital-13(PB5) กระพริบ 3 ครั้ง แล้วติดค้าง เพื่อบอกการติดต่อสื่อสารจากโปรแกรม สำหรับสั่งให้ทำการ Upload Code ให้กับ MCU ซึ่งถ้าไม่มีการติดต่อสื่อสารจากโปรแกรมของ Arduino ภายในระยะเวลาประมาณ 3 วินาที โปรแกรมก็จะกระโดดไปทำงานในตำแหน่งเริ่มต้นที่เป็นส่วนของโปรแกรมที่ผู้ใช้เขียนขึ้นทันที

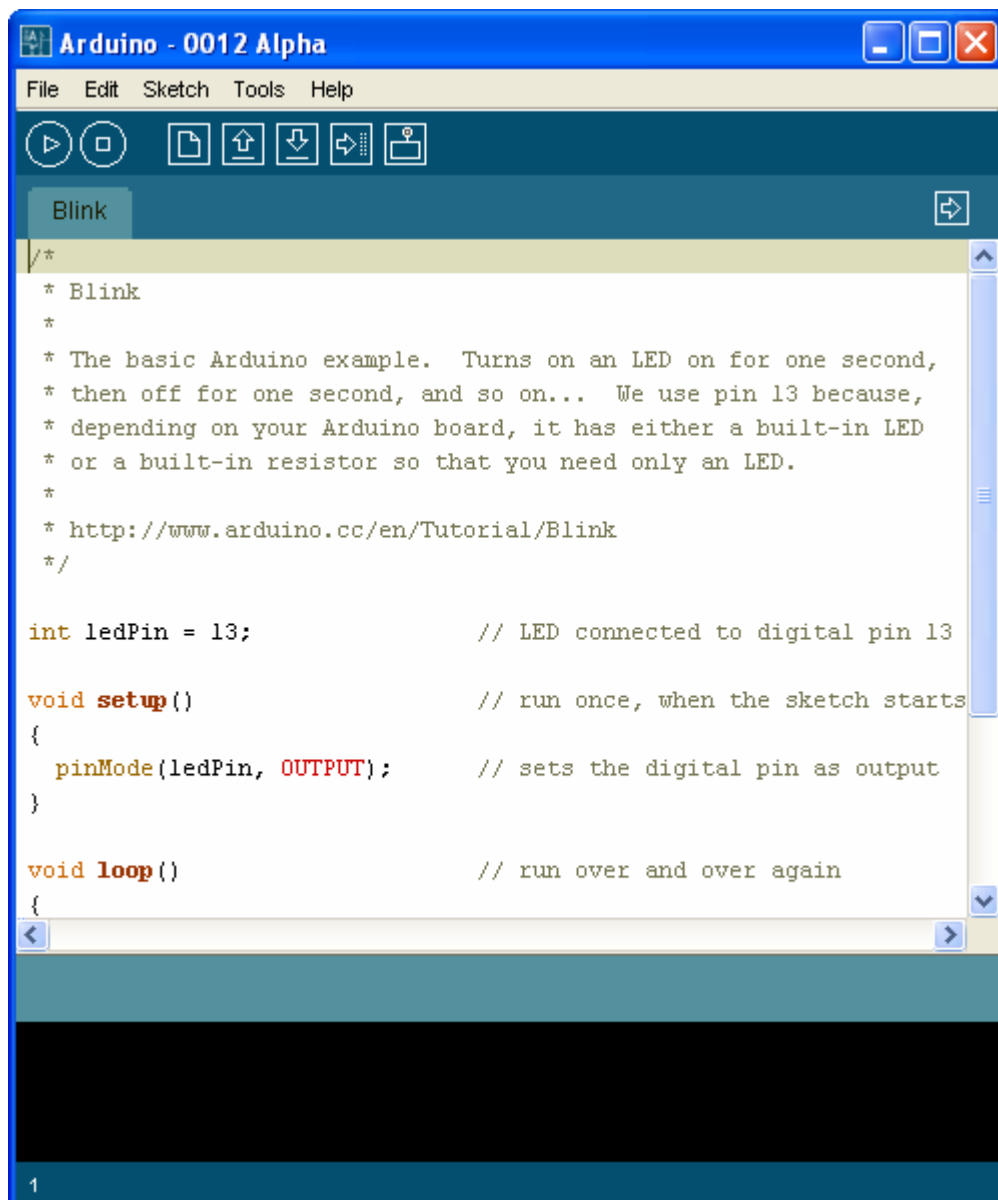
นอกจากนี้แล้วผู้ใช้อย่างสามารถเปลี่ยนไปใช้โปรแกรม Bootloader รุ่นมาตรฐาน ตามแบบโปรแกรม ของ Arduino ได้อีกด้วย โดยใช้โปรแกรม Bootloader ที่ชื่อ “ATmegaBOOT\_168\_diecimila.hex” โดยไฟล์ ดังกล่าวจะถูกบรรจุไว้ใน Drive และโฟลเดอร์เดียวกันกับที่ผู้ใช้ได้ทำการติดตั้งโปรแกรมของ Arduino ไว้ แล้วคือ “C:\arduino-0012\hardware\bootloaders\atmega168\ATmegaBOOT\_168\_diecimila.hex”

แต่อย่างไรก็ตามในการที่จะสามารถทำการติดตั้งโปรแกรม Bootloader Code ให้กับ MCU ได้นั้น ผู้ใช้จำเป็นต้องมีเครื่องมือสำหรับทำหน้าที่ Program Code ให้กับ MCU อยู่ด้วย โดยใช้เครื่องโปรแกรมที่มี ขั้วต่อตามมาตรฐานของ “AVRISP” ของ ATMEL แบบ IDE 10 PIN ได้ทันที โดยวิธีการให้ศึกษาเพิ่มเติม จากหัวข้อ **“การโปรแกรม Bootloader ให้กับบอร์ด ET-EASY168 STAMP”** ในตอนท้ายของคู่มือนี้

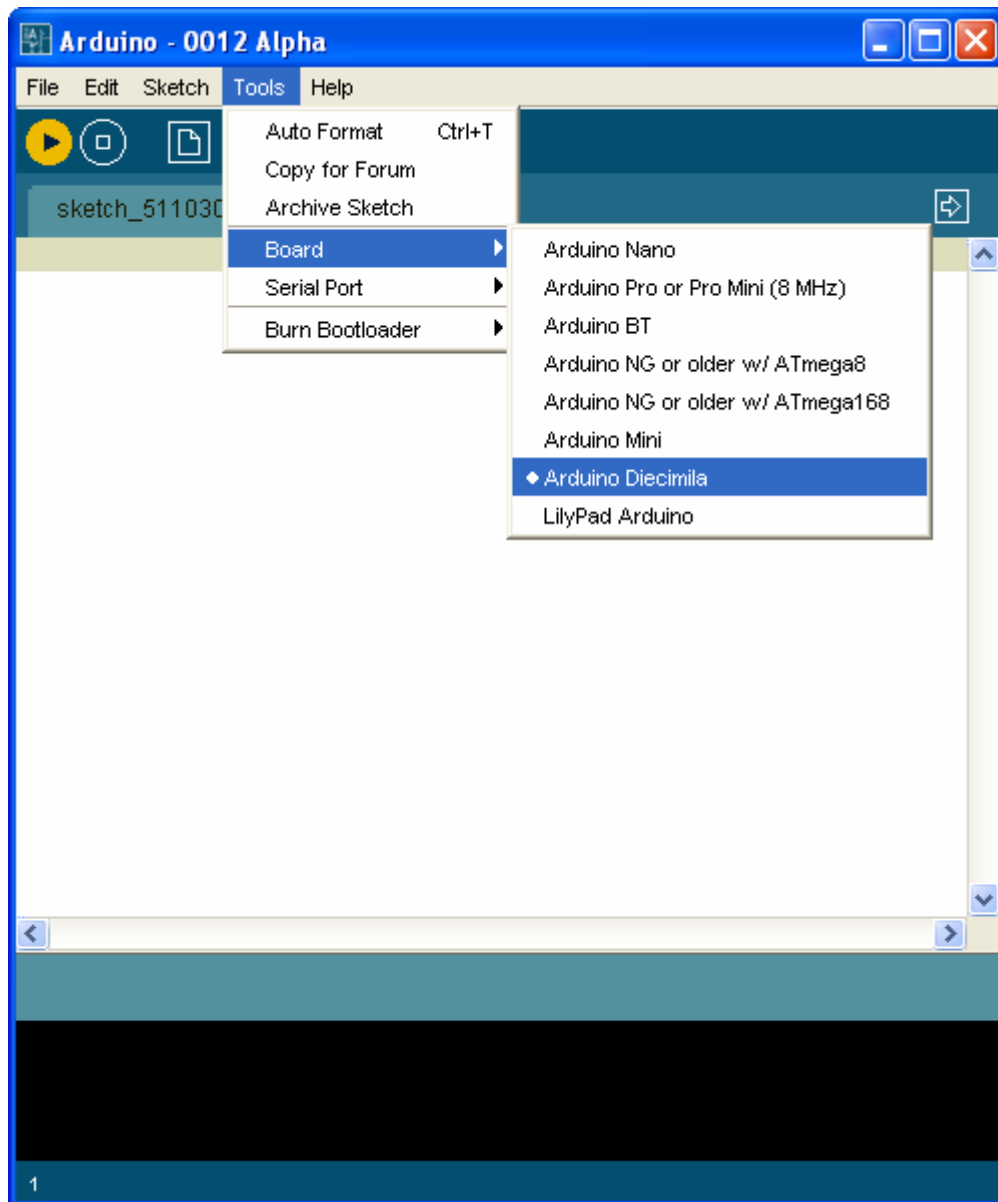
## การติดตั้งโปรแกรม Arduino

หลังจากที่เราได้ทำการติดตั้ง USB Driver ให้กับบอร์ดเป็นที่เรียบร้อยแล้ว ก็เป็นอันเสร็จสิ้นขั้นตอนของการเตรียมการแล้ว ลำดับขั้นตอนต่อจากนี้เป็นต้นไป ก็เป็นเรื่องของการใช้งาน การเขียนโปรแกรม และการศึกษาเรียนรู้ต่างๆตามความต้องการแล้ว แต่ก่อนอื่นเราจะต้องทำการติดตั้งโปรแกรมของ Arduino เพื่อใช้เป็นโปรแกรมสำหรับศึกษาเรียนรู้ ซึ่งมีลำดับขั้นตอนดังต่อไปนี้

1. ทำการ unzip ไฟล์ชื่อ “arduino-0012-win.zip” ไว้ในฮาร์ดดิสก์ ซึ่งขอแนะนำให้ทำการ unzip ไว้ที่ Root นอกสุดใน Drive C โดยหลังจากทำการ unzip เรียบร้อยแล้วจะได้โปรแกรมอยู่ที่ “c:\arduino-0012”
2. ทำการสั่ง Run โปรแกรม “arduino.exe” จะได้ผลดังรูป

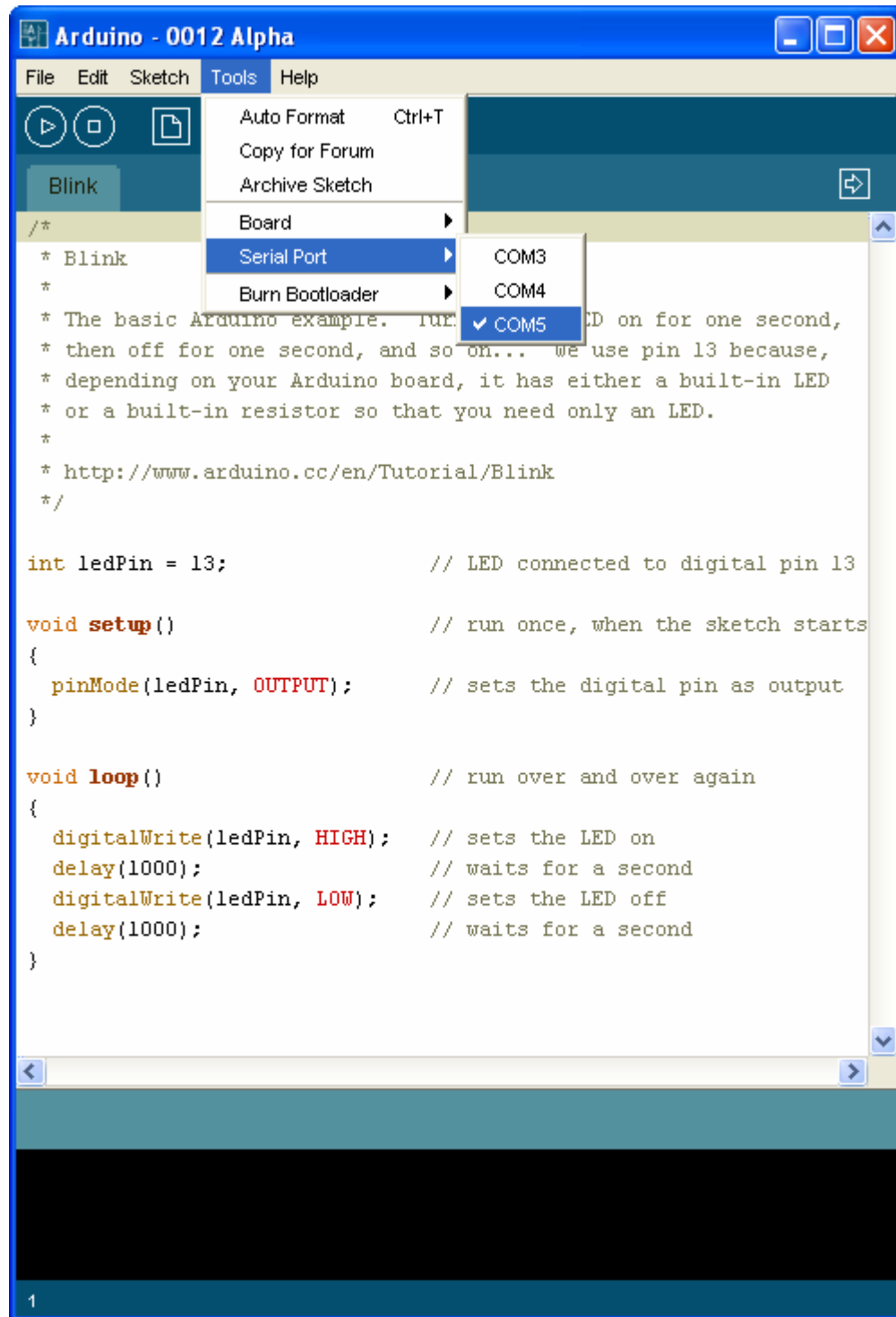


3. ในครั้งแรกของการเรียกใช้งานโปรแกรม ให้ทำการกำหนดระบบฮาร์ดแวร์ที่จะใช้งานกับโปรแกรมของ Arduino ให้เรียบร้อยเสียก่อน เนื่องจากในปัจจุบันนี้ มีการออกแบบวงจรและสร้างฮาร์ดแวร์บอร์ดแบบต่างๆสำหรับนำมาใช้งานร่วมกับโปรแกรมพัฒนาของ Arduino ไว้มากมายหลายรุ่น โดยในกรณีของบอร์ด ET-EASY168 STAMP ให้ทำการเลือกกำหนดชื่อบอร์ดเป็น “Decimila” โดยคลิกเมาส์ที่ “Tools → Board → Arduino Decimila” ดังรูป

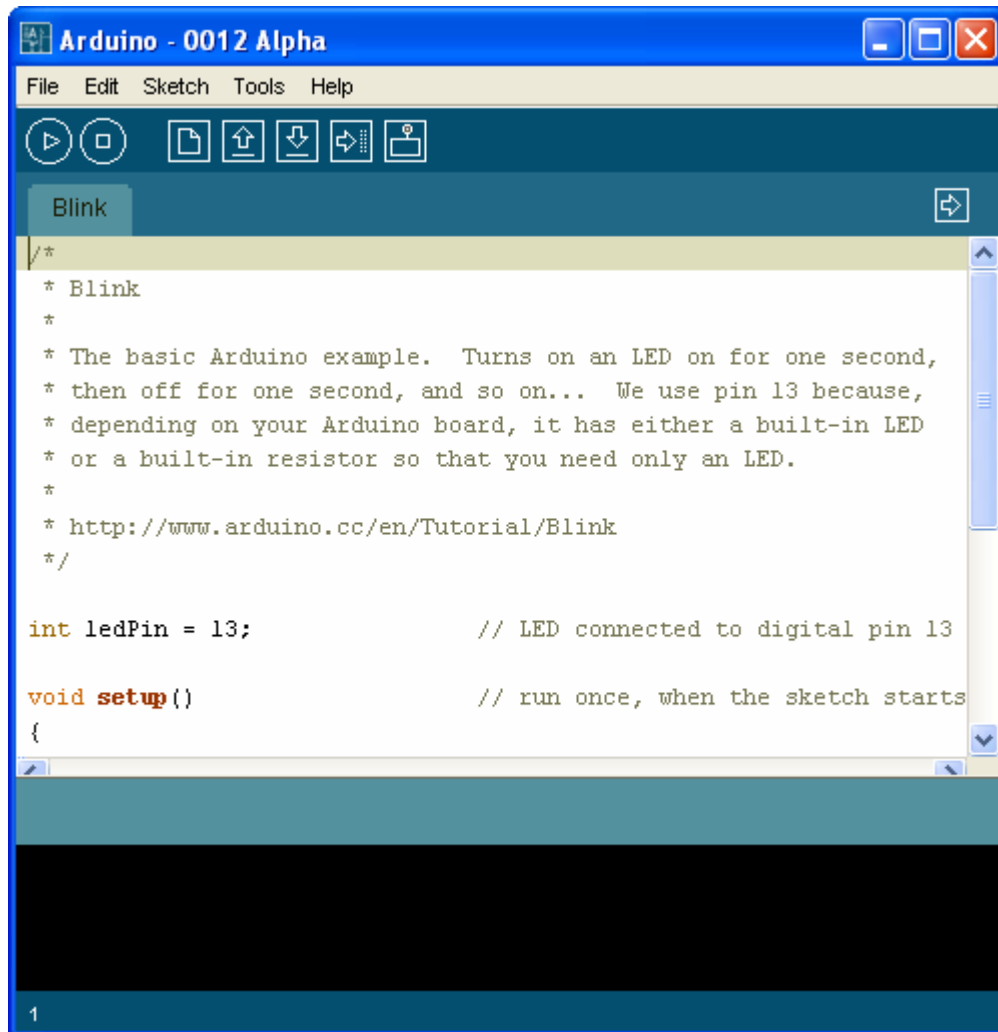




4. เลือกกำหนดหมายเลขพอร์ต สำหรับติดต่อสื่อสารกับบอร์ด ให้ตรงกับหมายเลข Comport ที่ได้ทำการติดตั้ง Driver ของ USB ไว้ในตอนแรก เช่น ถ้าตอนติดตั้ง Driver ของ USB แล้วได้หมายเลข Comport เป็น COM5 ให้คลิกเมาส์ที่ Tools → Serial Port → COM5 ดังรูป

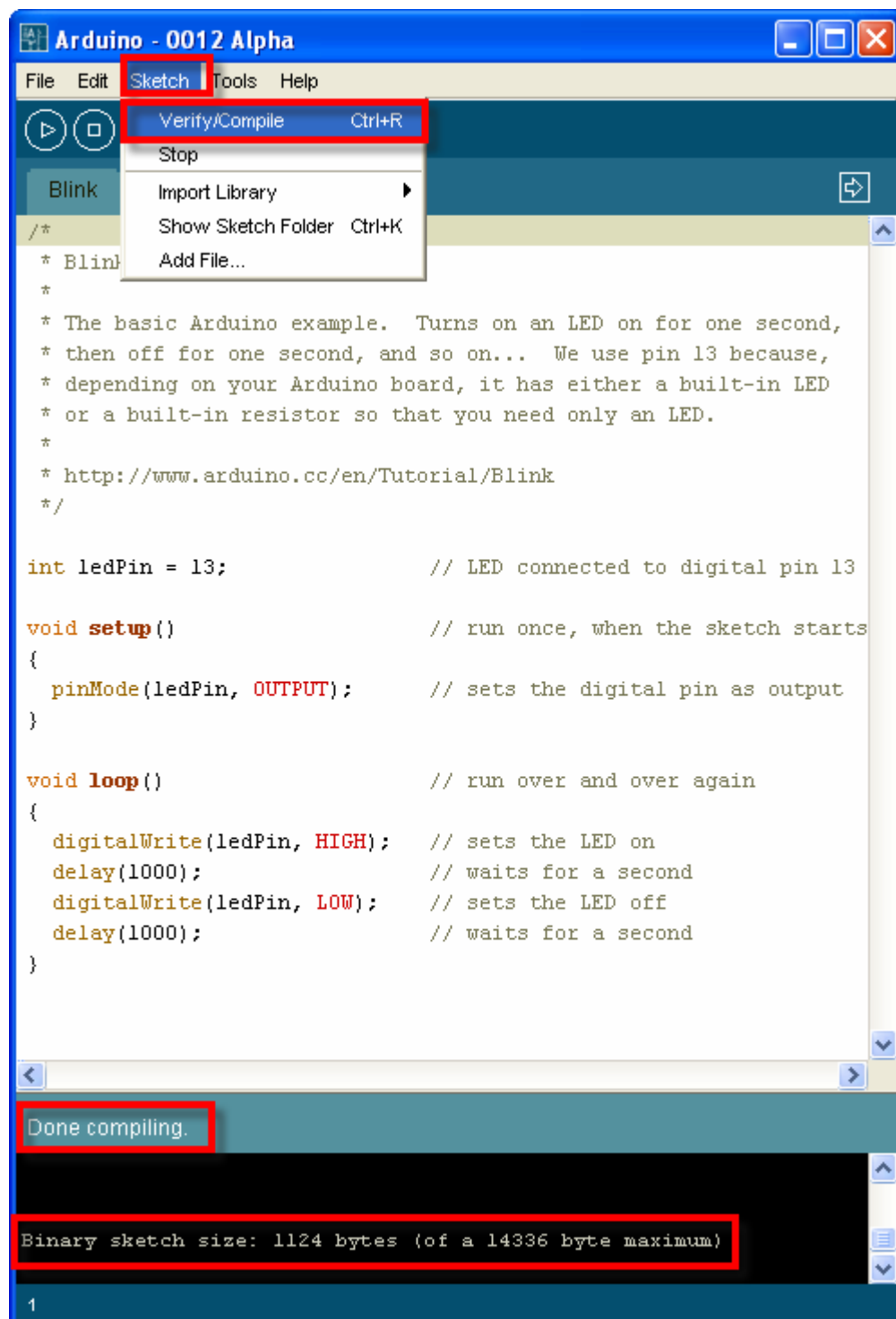


5. ทดสอบเขียนโปรแกรม โดยคลิกเมาส์ที่ File → New แล้วพิมพ์โปรแกรมทดสอบ หรืออาจใช้การ  
สั่งเปิดไฟล์ตัวอย่างที่สร้างไว้แล้วขึ้นมาแทนก็ได้ โดยในที่นี้ขอแนะนำให้ทดสอบด้วยโปรแกรมไฟ  
กระพริบ โดยให้เลือก “File → sketchbook → Examples → Digital → Blink” ซึ่งจะได้ดังรูป

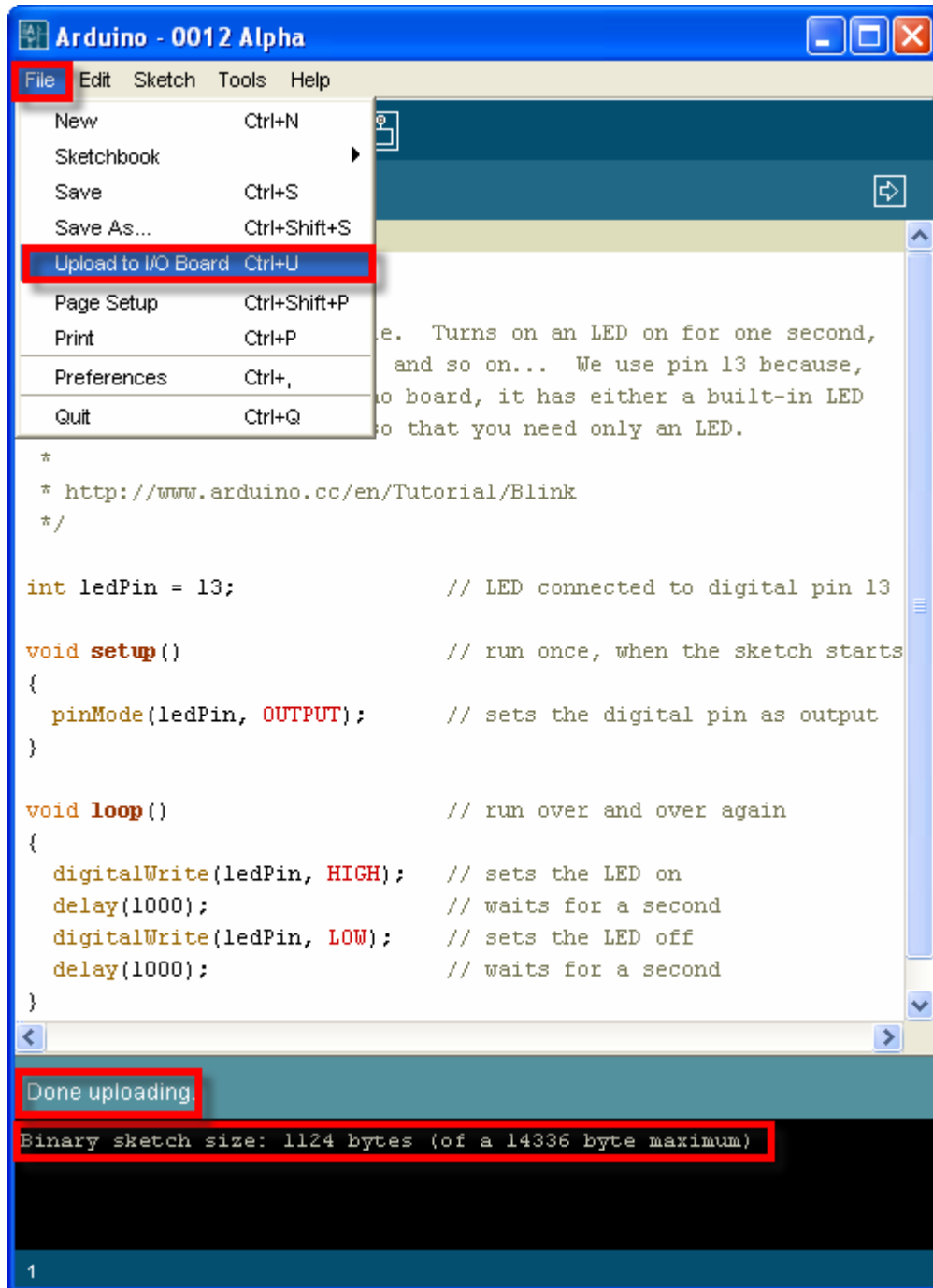


```
int ledPin = 13;  
  
void setup()  
{  
  pinMode(ledPin, OUTPUT);  
}  
  
void loop()  
{  
  digitalWrite(ledPin, HIGH);  
  delay(1000);  
  digitalWrite(ledPin, LOW);  
  delay(1000);  
}
```

6. อัปเดตโปรแกรมโดยคลิกเมาส์ที่ “Sketch → Verify/Compile” ดังตัวอย่าง



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



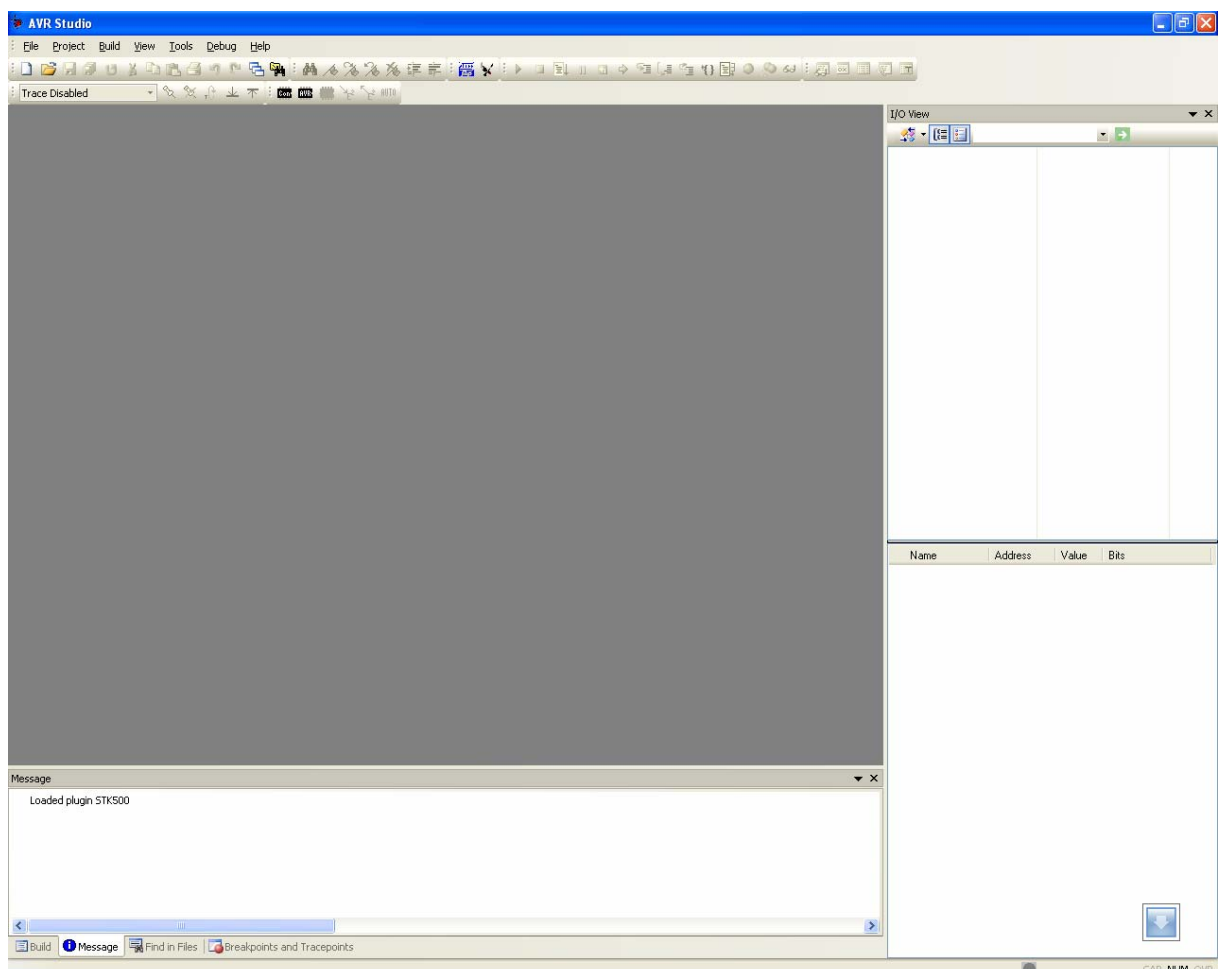
8. หลังจากที่ทำกา Upload Code ให้กับบอร์ดเป็นที่เรียบร้อยแล้ว บอร์ดก็จะเริ่มต้นทำงานตามคำสั่งที่เขียนไว้ในโปรแกรมทันที โดยจะสังเกตเห็น LED กระพริบ ติด และดับ สลับกันไปมา ด้วยความเร็วประมาณ 1 วินาที ตลอดเวลา



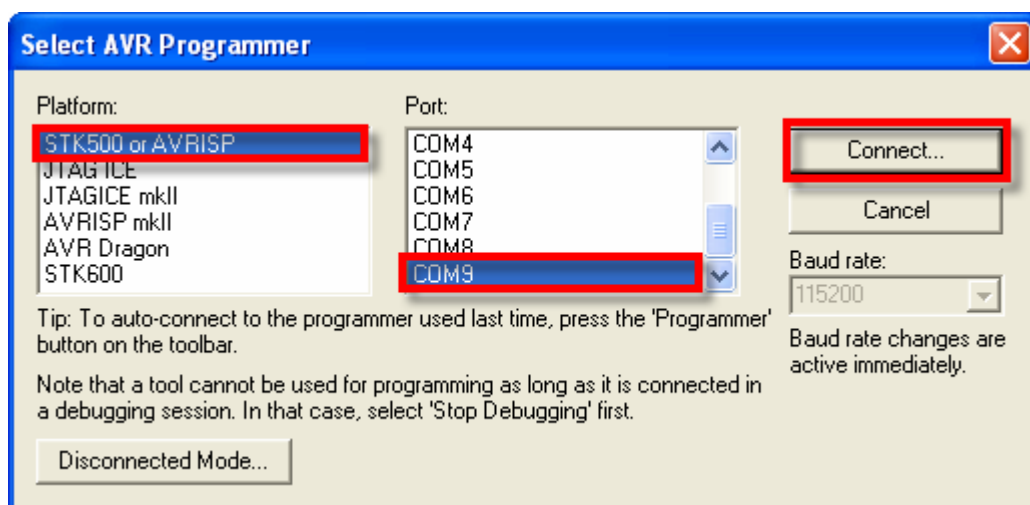
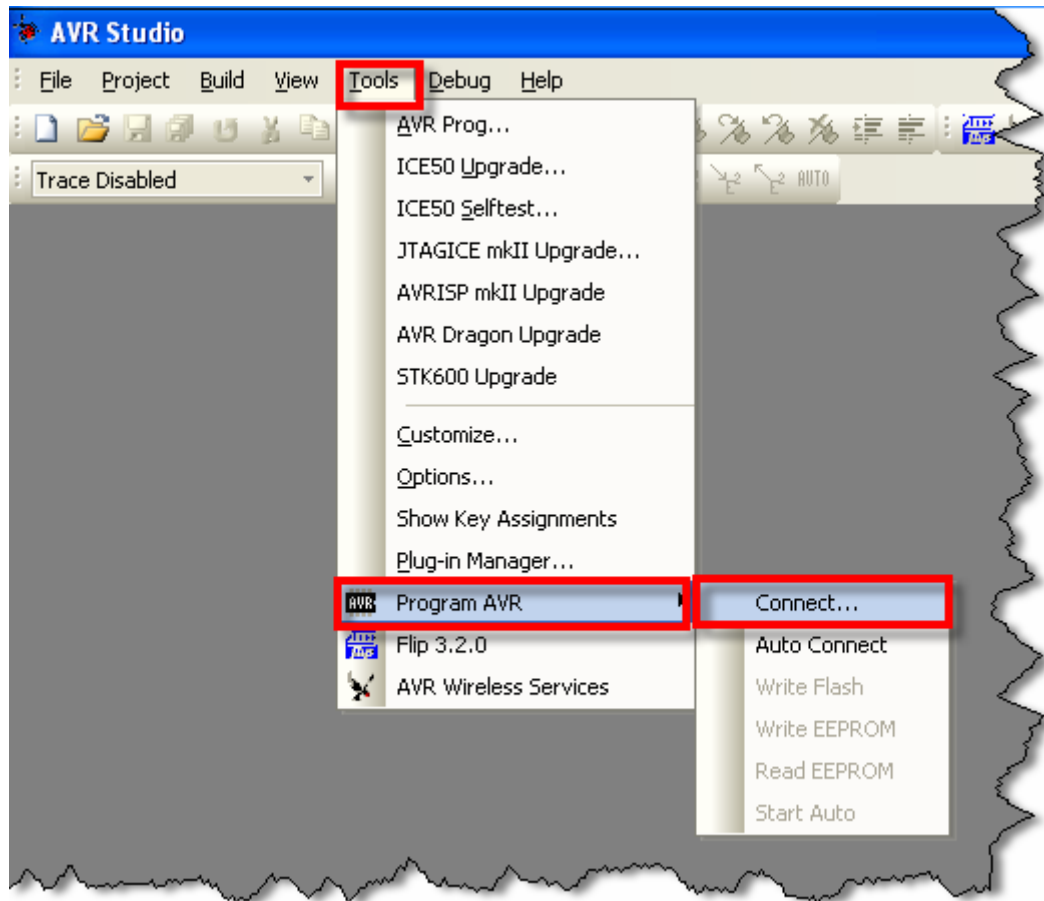
## การโปรแกรม Bootloader ให้กับบอร์ด ET-EASY168 STAMP

ตามปกติแล้วบอร์ด ET-EASY168 STAMP ได้ทำการโปรแกรม Bootloader ไว้ให้เป็นที่ยอมรับอยู่แล้ว ผู้ใช้สามารถใช้งานได้ทันที แต่อย่างไรก็ตามในกรณีที่ต้องการเปลี่ยนแปลง Bootloader หรือ เกิดความผิดพลาดในการใช้งาน จนทำให้ Bootloader เสียหายไป ผู้ใช้ก็ยังสามารถทำการ โปรแกรม Bootloader ให้กับบอร์ดได้ใหม่ โดยบอร์ด ET-EASY168 STAMP ได้ออกแบบ ให้มีขั้วต่อ AVRISP สำหรับใช้เป็นช่องทางในการโปรแกรม Code ให้กับ MCU ได้โดยตรง ด้วยเครื่องโปรแกรมทุกระุ่นที่มีขั้วต่อ 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

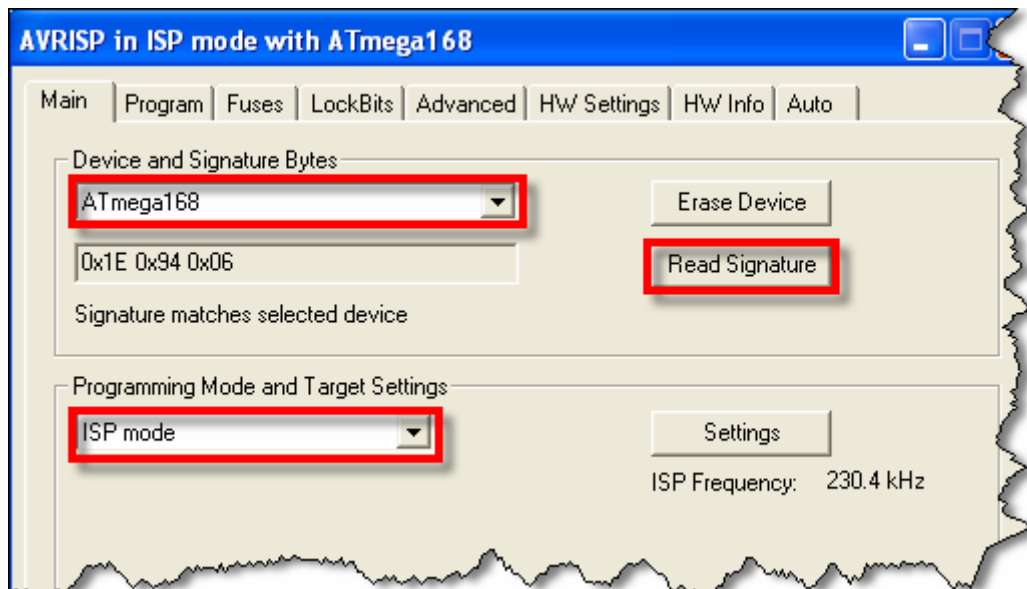
1. ต่อสาย USB ให้กับบอร์ด ET-EASY168 STAMP ในกรณีต้องการใช้ไฟเลี้ยงจากพอร์ต USB หรือ จ่ายไฟเลี้ยง +5V ให้กับบอร์ดที่ขา 28(+5V)
2. ต่อสาย USB ให้กับเครื่องโปรแกรม ET-AVR ISP USB V1 พร้อมทั้งต่อสายแพร์ 10 Pin ระหว่างขั้วต่อของ AVRISP ของทั้ง 2 บอร์ดเข้าด้วยกัน
3. สั่ง Run โปรแกรม AVR Studio 4 ดังรูป



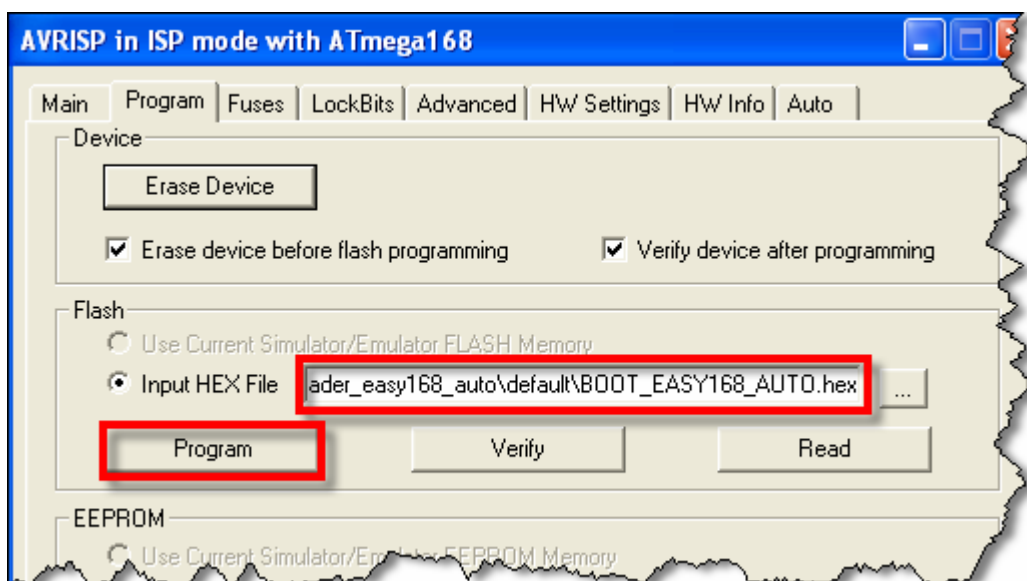
4. เลือกคลิกเมาส์ที่ Tools → Program AVR → Connect.. → STK500 or AVRISP จากนั้นก็ให้เลือกหมายเลขของ Comport ที่เป็นของเครื่องโปรแกรม ET-AVR ISP USB V1 ตามที่ลง Driver ไว้ พร้อมกับเลือก Connect (จากตัวอย่างเป็น Com9) ดังรูป



5. หลังจากที่ทำกร Connect เรียบร้อยแล้วให้ทดสอบการเชื่อมต่อโดยเลือกที่ tab ของ Main พร้อมกับเลือกเบอร์ของ MCU เป็น ATmega168 และเลือก Programming Mode and target Settings เป็น ISP mode แล้วทดสอบเลือก Read Signature ดู ซึ่งถ้าทุกอย่างถูกต้องควรได้ผลดังรูป



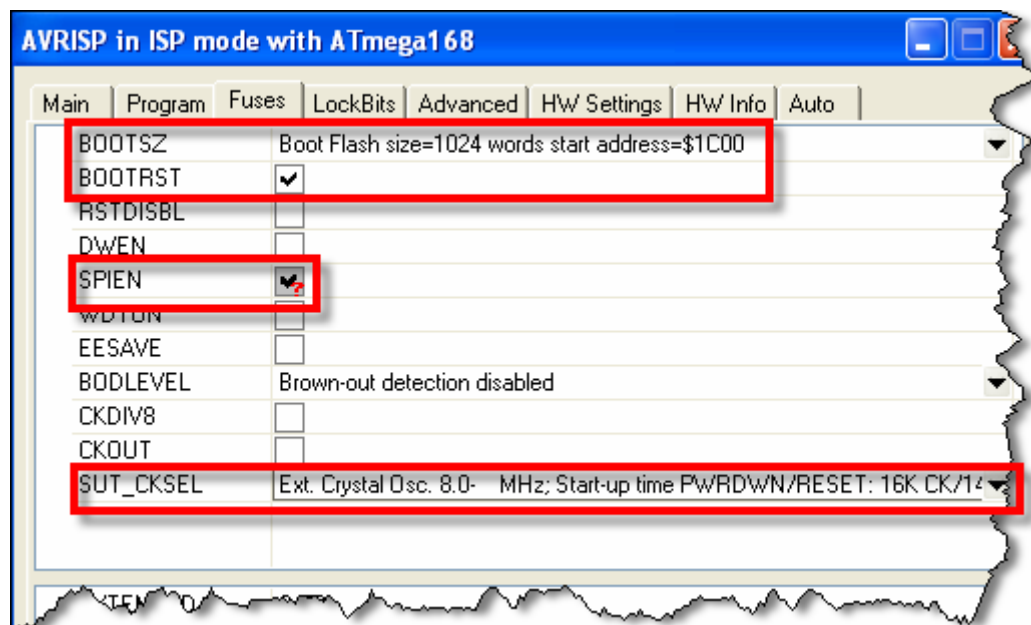
6. ให้เลือกไปที่ tab ของ Program พร้อมทั้งเลือก ตัวเลือกต่างๆดังนี้
- Device ให้เลือก Erase device before flash programming และ Verify device after programming
  - Flash ให้เลือก Input HEX File เป็น BOOT\_EASY168\_AUTO\_16MHZ.HEX จากนั้นให้เลือกที่ Program เพื่อสั่ง Program Bootloader ให้กับ MCU ซึ่งจะได้ผลดังรูป



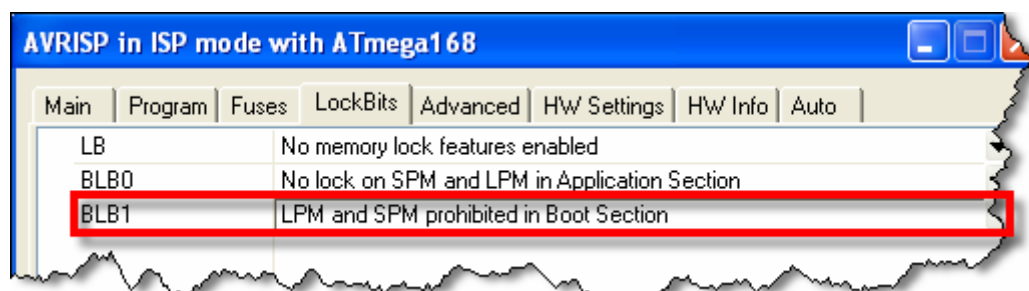
7. หลังจากทำการโปรแกรม Code ให้กับ MCU เรียบร้อยแล้ว ให้เลือกมาที่ Tab ของ Fuses เพื่อทำการสั่งโปรแกรม Fuse Bit ให้กับ MCU โดยให้เลือกกำหนดค่าตัวเลือกดังนี้

- BOOTSZ ให้เลือกเป็น Boot Flash size = 1024 word start address = \$1C00
- 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 แล้วสั่ง program เป็นอันเสร็จสิ้นกระบวนการ Program Bootloader



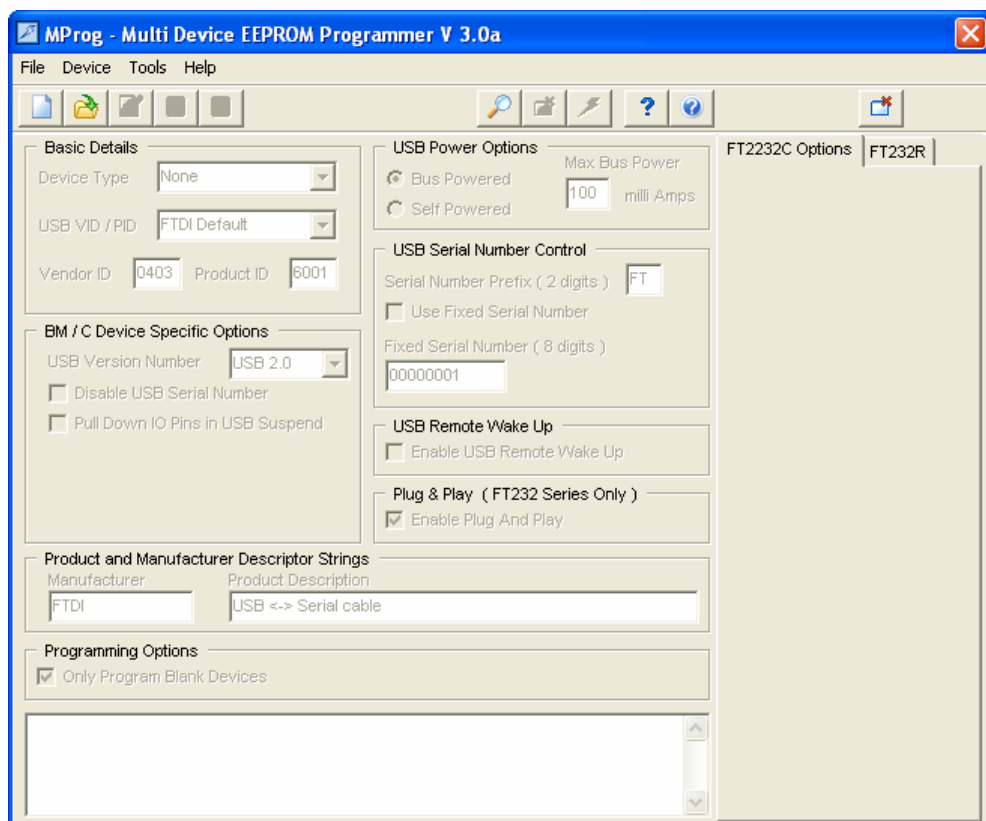


## การโปรแกรม USB Bridge ของ FTDI ให้พอร์ต USB จ่ายกระแสได้ 500mA

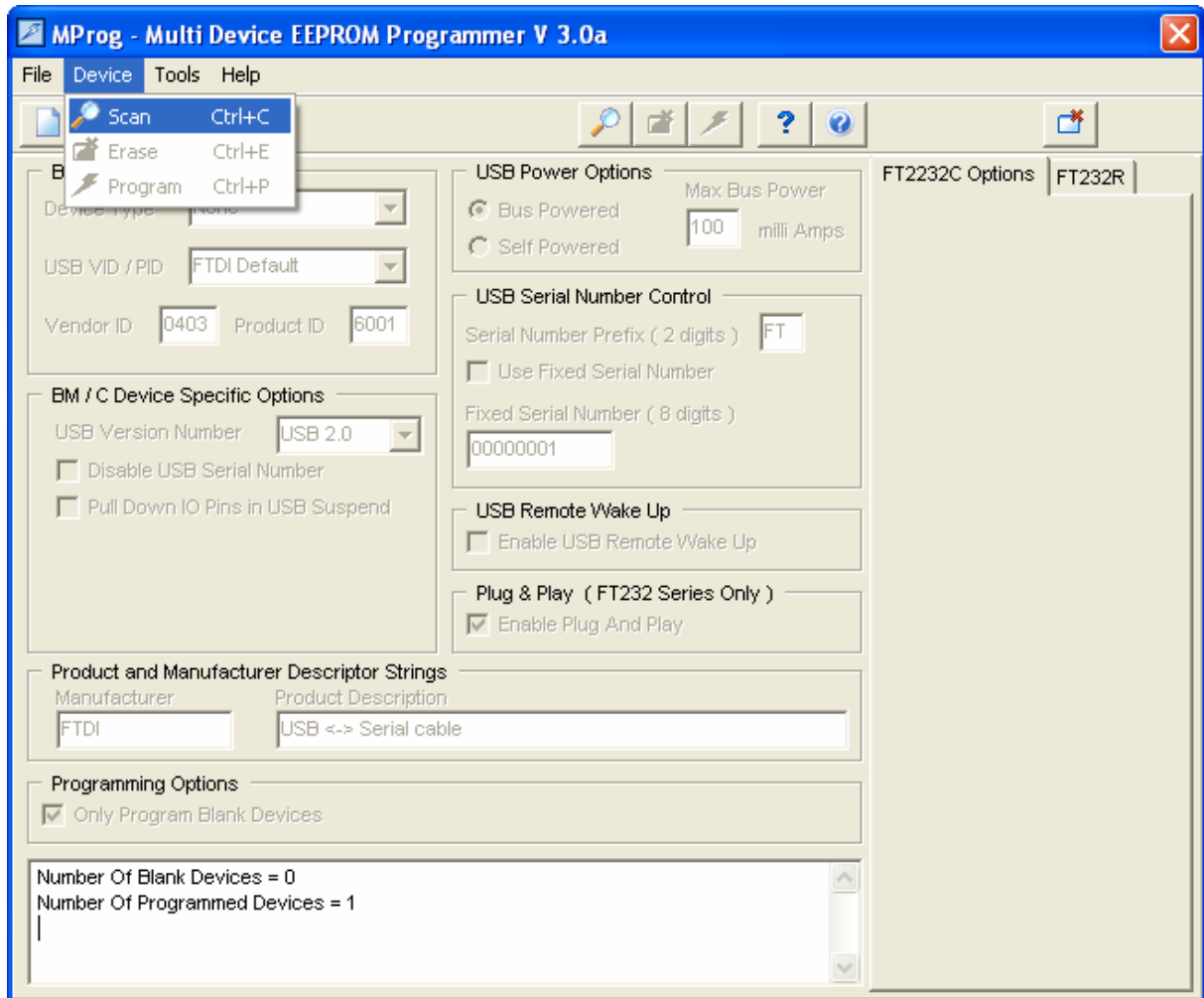
ตามปกติแล้ว USB Bridge ของ FTDI เบอร์ FT232R จะถูกกำหนด Configuration ค่าให้ใช้แหล่งจ่ายจากพอร์ต USB โดยตามค่ามาตรฐานของ USB Driver ของ FTDI นั้นจะร้องขอกระแสไฟ ไปยัง USB Host เพียงแค่ 90mA เท่านั้น ซึ่งตามมาตรฐานแล้วพอร์ต USB สามารถจ่ายกระแสไฟให้กับอุปกรณ์ที่เชื่อมต่อด้วยสูงสุดที่ 500mA ซึ่งเพื่อความสะดวกในการทดลองต่างๆ ด้วยบอร์ด ET-EASY168 STAMP ในกรณีที่ยังไม่ได้เชื่อมต่อกับอุปกรณ์ภายนอกมากมายนัก(ใช้กระแสไม่เกิน 500mA) เราก็สามารถเข้าไปสั่งแก้ไขเปลี่ยนค่า Configuration ของ USB HUB เพื่อให้จ่ายกระแสออกมาให้ครบ 500mA ได้ด้วย ซึ่งจะทำให้เกิดความสะดวกและคล่องตัวในการทดลองต่างๆมากขึ้นด้วย แต่อย่างไรก็ตามถ้าหากว่าผู้ใช้ ทำการทดลองกับ เครื่องคอมพิวเตอร์แบบ Notebook ที่ใช้กระแสไฟจาก Battery หรือมีการเชื่อมต่อบอร์ดทดลองกับอุปกรณ์ที่ใช้กระแสไฟมากๆ ก็ควรจัดหาแหล่งจ่ายไฟภายนอกเพื่อจ่ายให้กับบอร์ดเองจะเหมาะสมกว่า ข้อแนะนำนี้เป็นการแนะนำสำหรับผู้ที่ต้องการทดสอบการทำงานหรือทดลองเขียนโปรแกรมกับบอร์ด แบบที่ไม่ได้เน้นการเชื่อมต่อกับอุปกรณ์ภายนอกมากมายนัก เช่น ทดสอบไฟกระพริบ โดยใช้ LED เป็นอุปกรณ์การทดลอง หรือ การรับส่งข้อมูลกับพอร์ตสื่อสารอนุกรม เป็นต้น

โดยการแก้ไขค่า Configuration ของ USB Bridge ของ FTDI จะใช้โปรแกรม “MProg.exe” เป็นตัวจัดการโดยมีลำดับขั้นตอนดังต่อไปนี้

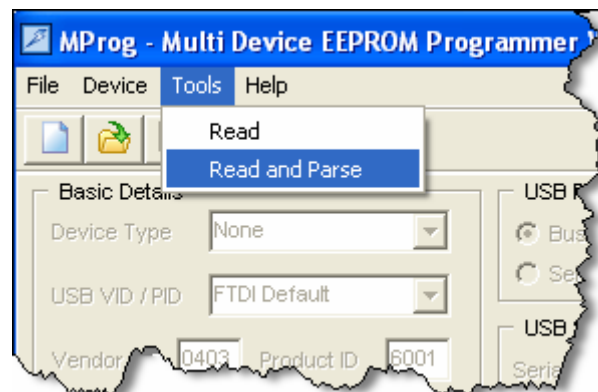
1. ต่อสาย USB ของบอร์ด ET-EASY168 STAMP เข้ากับพอร์ต USB แล้วสั่ง Run Program MProg ซึ่งจะได้ผลดังรูป

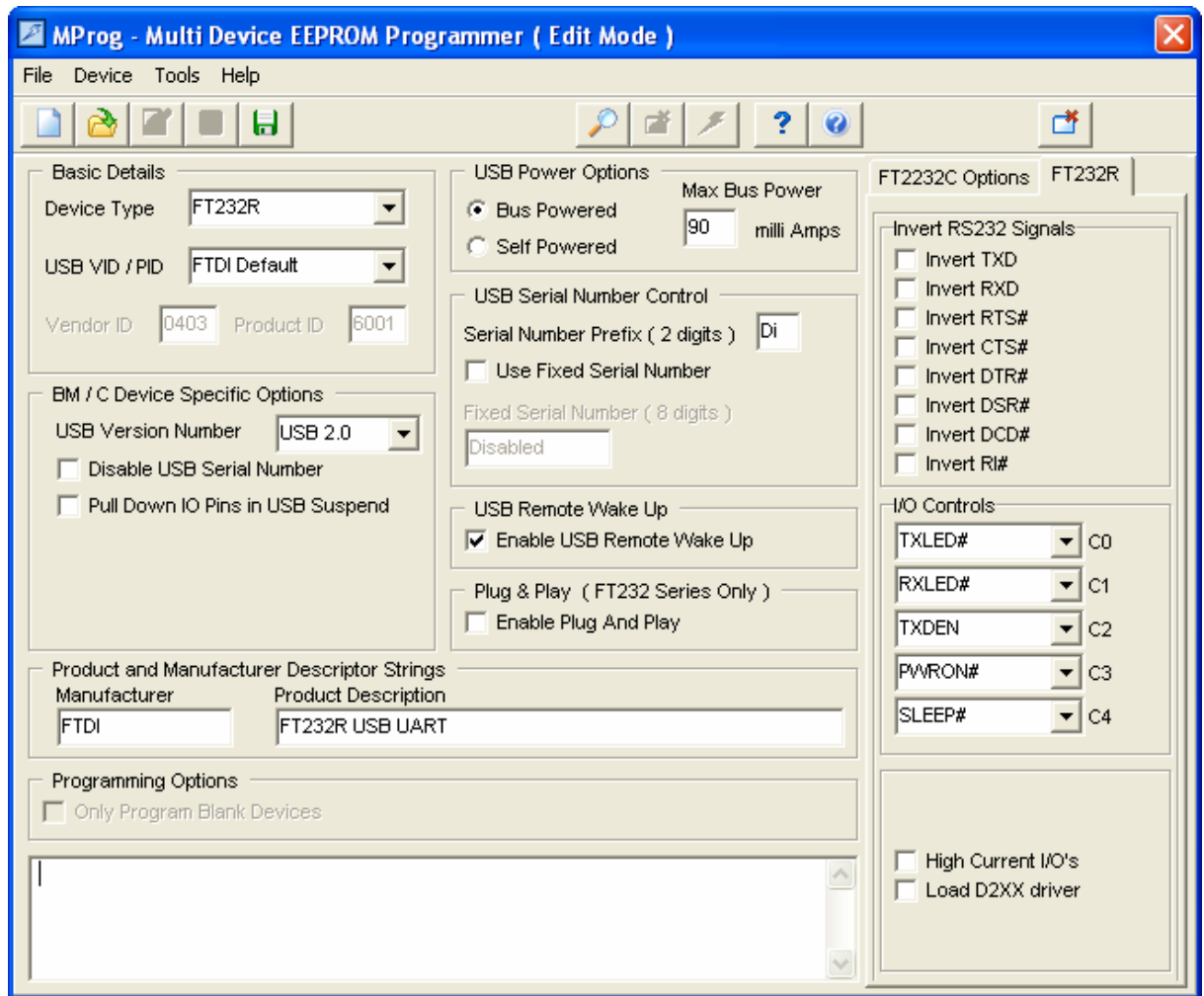


2. สั่งโปรแกรม “mProg” ค้นหา USB Bridge ที่ติดตั้งไว้ในเครื่อง โดยเลือกที่เมนู Device → Scan ซึ่งควรได้ผลดังรูป

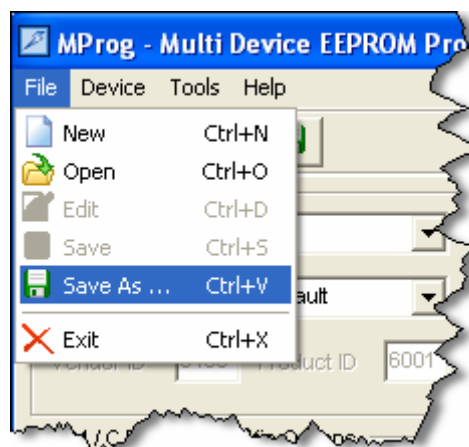


3. สั่งอ่านค่า Configuration เดิมของ FTDI ออกมา โดยเลือกที่ Tools → Read and parse ซึ่งควรได้ผลดังรูป

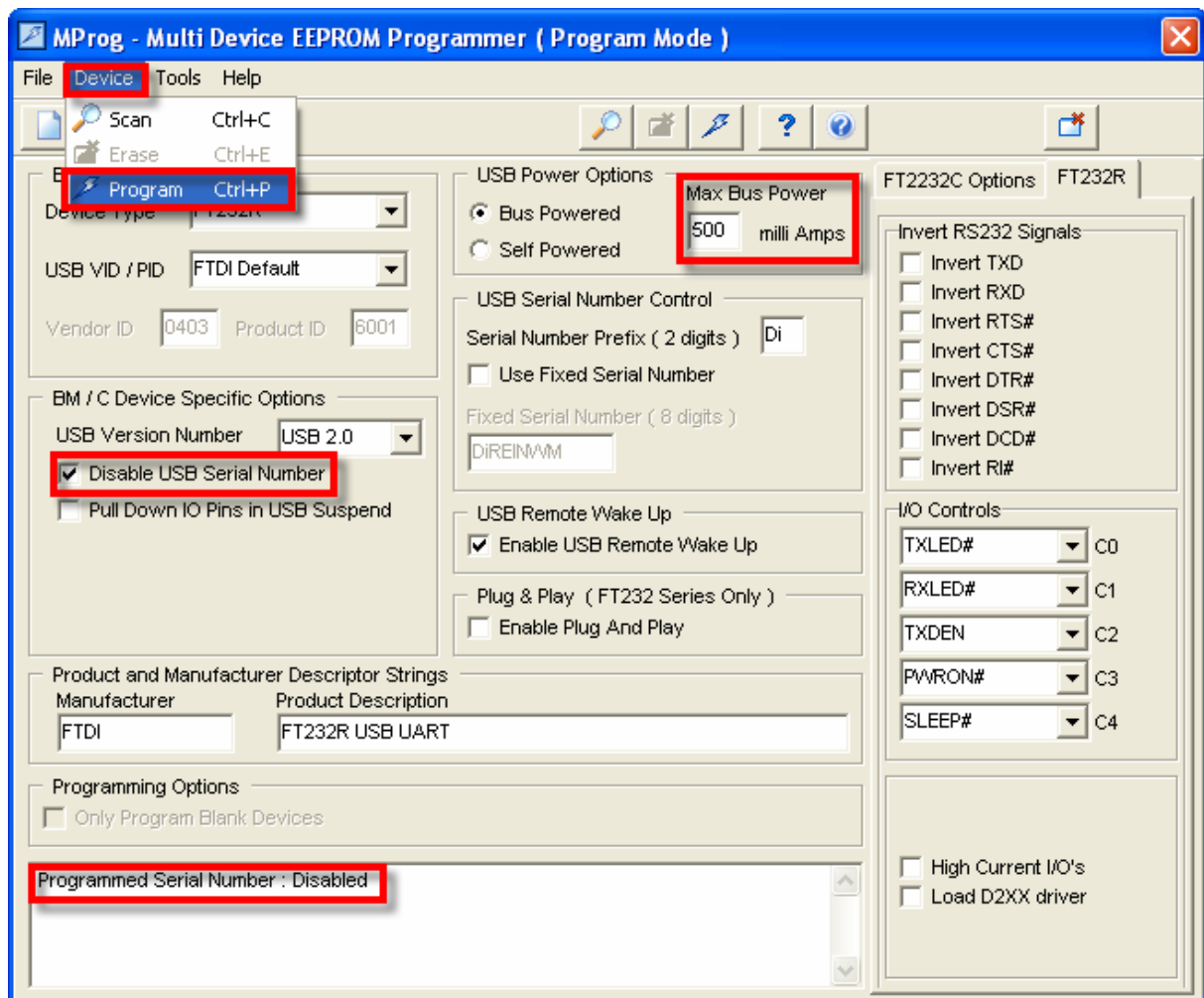
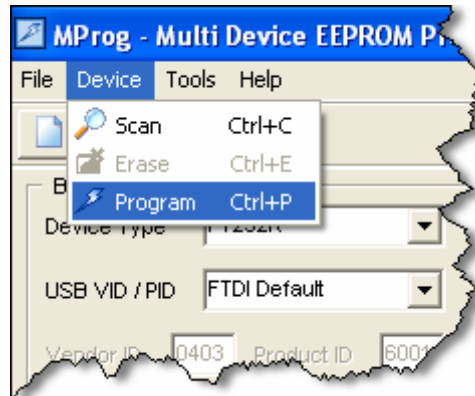




4. ให้ทำการแก้ไขค่า Configuration ในส่วนของ USB Power Options ให้เลือกเป็น Bus Powered และกำหนดค่า Max Bus Power จากเดิมที่กำหนดไว้ที่ 90 milli Amps เป็น 500 milli Amps ส่วนค่า Configuration อื่นๆไม่ต้องเปลี่ยนแปลง จากนั้นให้ทำการบันทึกค่า Configuration ไว้ในเครื่องก่อนโดยให้เลือกที่ File → Save As.. ดังรูป



5. ตั้งโปรแกรมค่า Configuration คำนึงให้กับ FTDI โดยให้เลือกที่ Device → Program แล้วรอนโปรแกรมทำงานเสร็จ หลังจากนั้นให้ ถอดสาย USB ออกแล้วเสียบกลับเข้าไปใหม่ USB ก็จะทำตามค่า Configuration ใหม่ได้แล้ว



**การพัฒนาโปรแกรมของ ET-EASY168 STAMP แบบ AVR Microcontroller**

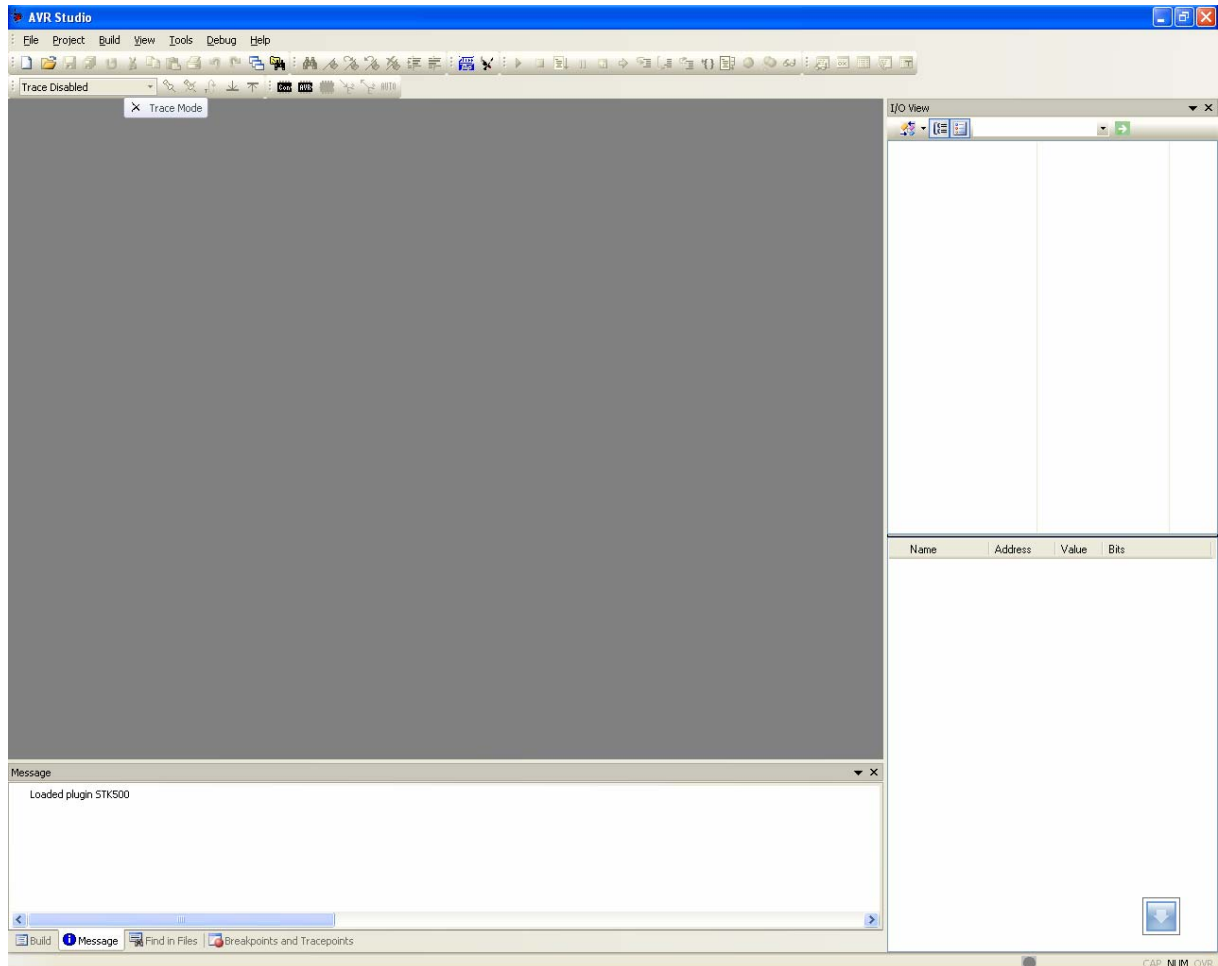
ในกรณีที่ผู้ใช้ต้องการที่จะพัฒนาโปรแกรมให้กับบอร์ด ET-EASY168 STAMP แบบปกติ ในรูปแบบของไมโครคอนโทรลเลอร์ AVR ก็สามารถทำได้ โดยสามารถเลือกภาษาในการเขียนโปรแกรมได้เอง ซึ่งในกรณีนี้ผู้ใช้สามารถเลือกใช้ภาษาใดๆก็ได้ที่รองรับการใช้งานกับ MCU ตระกูล AVR เบอร์ ATmega168 โดย ผู้ใช้งานสามารถบริหารจัดการระบบทรัพยากรต่างๆที่อยู่ภายในตัว ATmega168 ได้เองทั้งหมด โดยสามารถทำได้ 2 แนวทางคือ

- **การพัฒนาโปรแกรมโดยใช้เครื่องโปรแกรมภายนอก** ในกรณีนี้จะมีข้อดีคือ ไม่สูญเสียทรัพยากรใดๆเลย ผู้ใช้สามารถใช้งานและกำหนดคุณสมบัติการทำงานของทรัพยากรต่างๆที่มีอยู่ใน MCU ได้เอง ตามต้องการ แต่มีข้อเสียคือ ต้องมีเครื่องโปรแกรมภายนอก สำหรับทำหน้าที่โปรแกรม Code ให้กับ MCU โดยสามารถใช้ได้กับเครื่องโปรแกรมทุกรุ่นที่ รองรับการใช้งานกับ MCU เบอร์ ATmega168 และมีข้อจำกัดตามมาตรฐาน AVRISP ของ ATMEL
- **การพัฒนาโปรแกรมโดยใช้งานร่วมกับ Bootloader** ในกรณีนี้จะมีข้อดี คือ เมื่อทำการเขียนโปรแกรมเสร็จแล้วสามารถส่ง Program Code ให้กับ MCU ผ่านทางโปรแกรม Bootloader ที่ติดตั้งไว้แล้วได้ทันที โดยไม่ต้องใช้เครื่องโปรแกรมภายนอก แต่มีข้อจำกัด คือ ต้องเสียพื้นที่หน่วยความจำ Flash สำหรับเก็บ Code โปรแกรมไป 2KByte (0x3800 -0x3FFF) เพื่อให้ติดตั้งโปรแกรม Bootloader ซึ่งตามปกติแล้วบอร์ด ET-EASY168 STAMP จะทำการ ติดตั้งโปรแกรม Bootloader ไปให้เรียบร้อยแล้ว โดยในกรณีนี้ผู้ใช้จะสามารถเขียนโปรแกรมใช้งานได้ทั้งหมดจำนวน 14 Kbyte จากที่มีอยู่ทั้งหมด 16Kbyte โดยต้องเขียนโปรแกรมให้มีตำแหน่งการทำงานของ Code อยู่ระหว่างตำแหน่งแอดเดรส 0x0000 ถึง 0x37FF (0x1C00 - 0x1FFF K Word) ส่วน SRAM,EEPROM และทรัพยากรอื่นๆที่มีอยู่ใน MCU ผู้ใช้ยังสามารถใช้งานได้ครบตามจำนวนที่มีอยู่ในตัว MCU ทุกประการ

## ตัวอย่างการพัฒนาโปรแกรมด้วย WinAVR โดยใช้งานร่วมกับ Bootloader

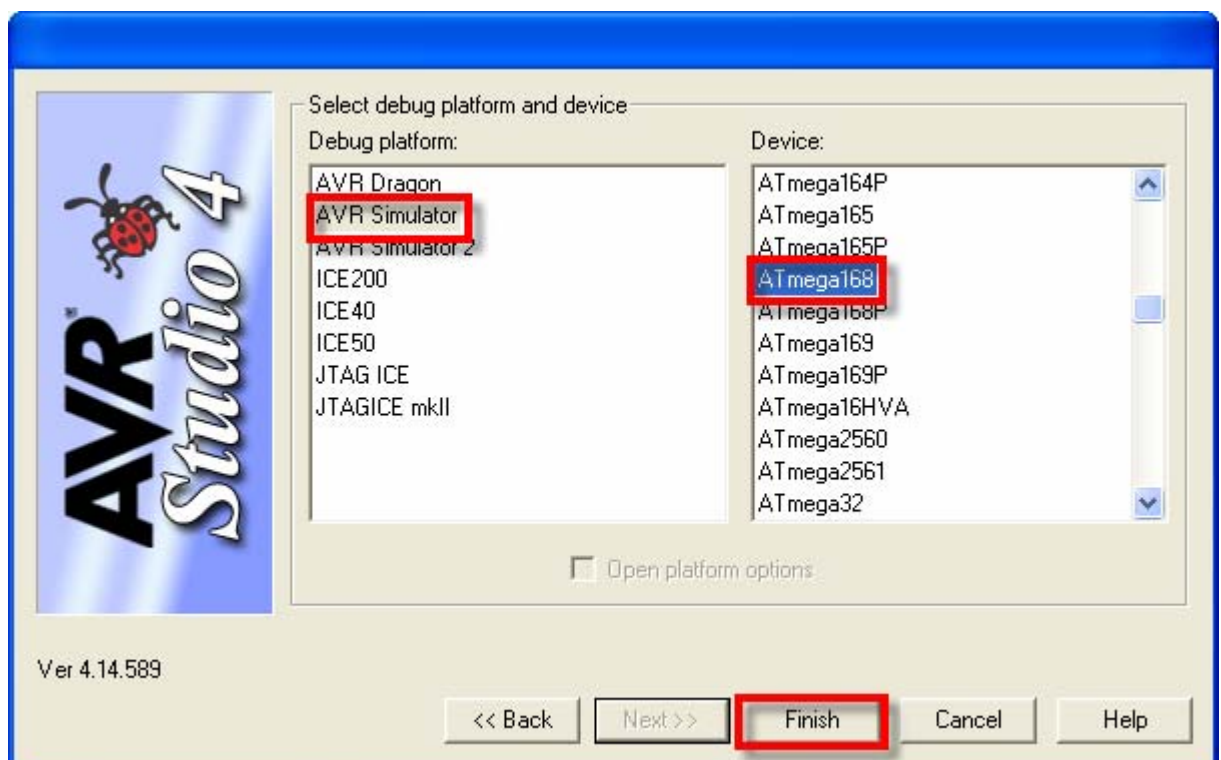
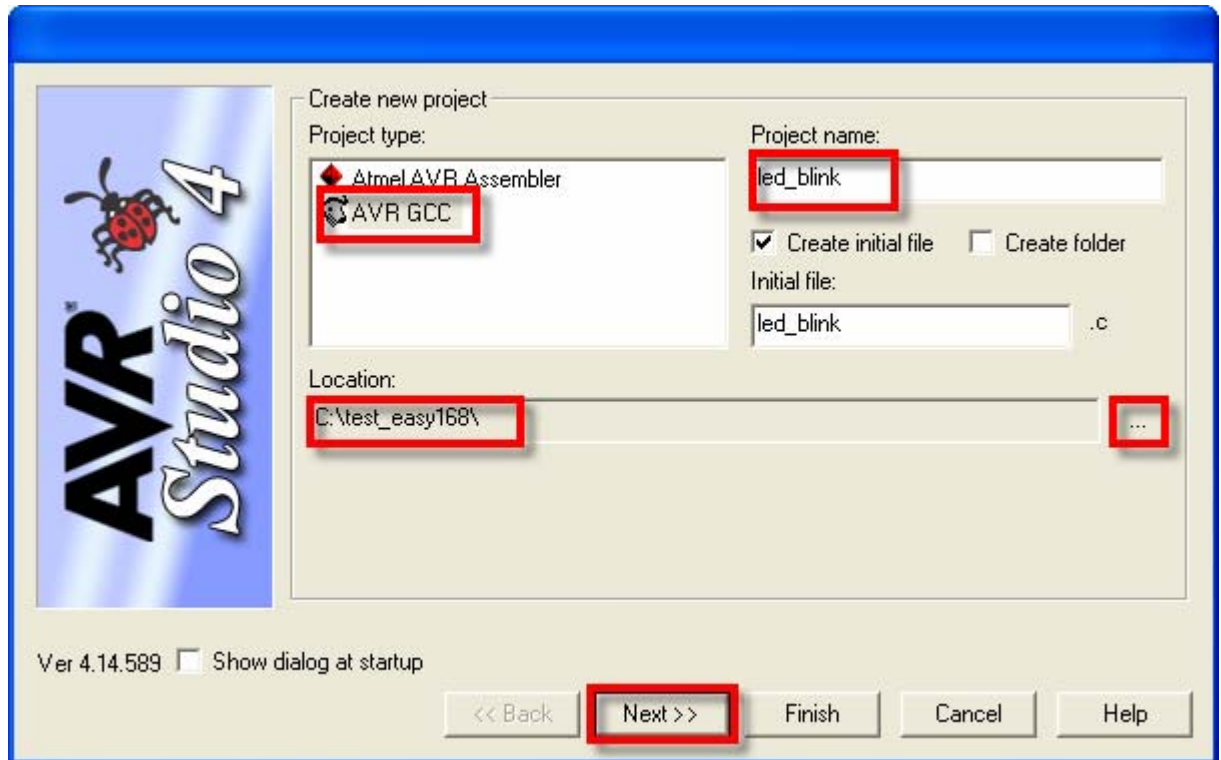
ตัวอย่างที่จะแสดงต่อไปนี้จะแสดงให้เห็นวิธีการพัฒนาโปรแกรมให้กับ ATmega168 ด้วยภาษาซี โดยใช้โปรแกรม AVR Studio4 ของ ATMEL ร่วมกับ Compiler ภาษาซีของ WinAVR

### 1. สั่ง Run Program AVR Studio4



2. สร้าง 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\_easy168\”
  - 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 ATmega168 ดังรูป





4. ให้พิมพ์คำสั่งของโปรแกรมสำหรับทดสอบการทำงาน ในหน้าต่าง Text Editor ของโปรแกรม โดยในที่นี้จะทดสอบด้วย Code โปรแกรม สำหรับทำหน้าที่เป็นไฟกะพริบที่ PB5 ซึ่งเป็น LED ที่ติดตั้งไว้บนบอร์ด ET-EASY168 STAMP อยู่แล้ว ดังตัวอย่าง

```
/* **** */
/* Program Test LED Blinking */
/* Board : ET-EASY168 STAMP */
/* MCU : ATmega168 */
/* X-TAL : 16.00MHz */
/* **** */

#include <avr/io.h>
#define F_CPU 16000000UL // X-TAL = 16 MHz
#include <util/delay.h>

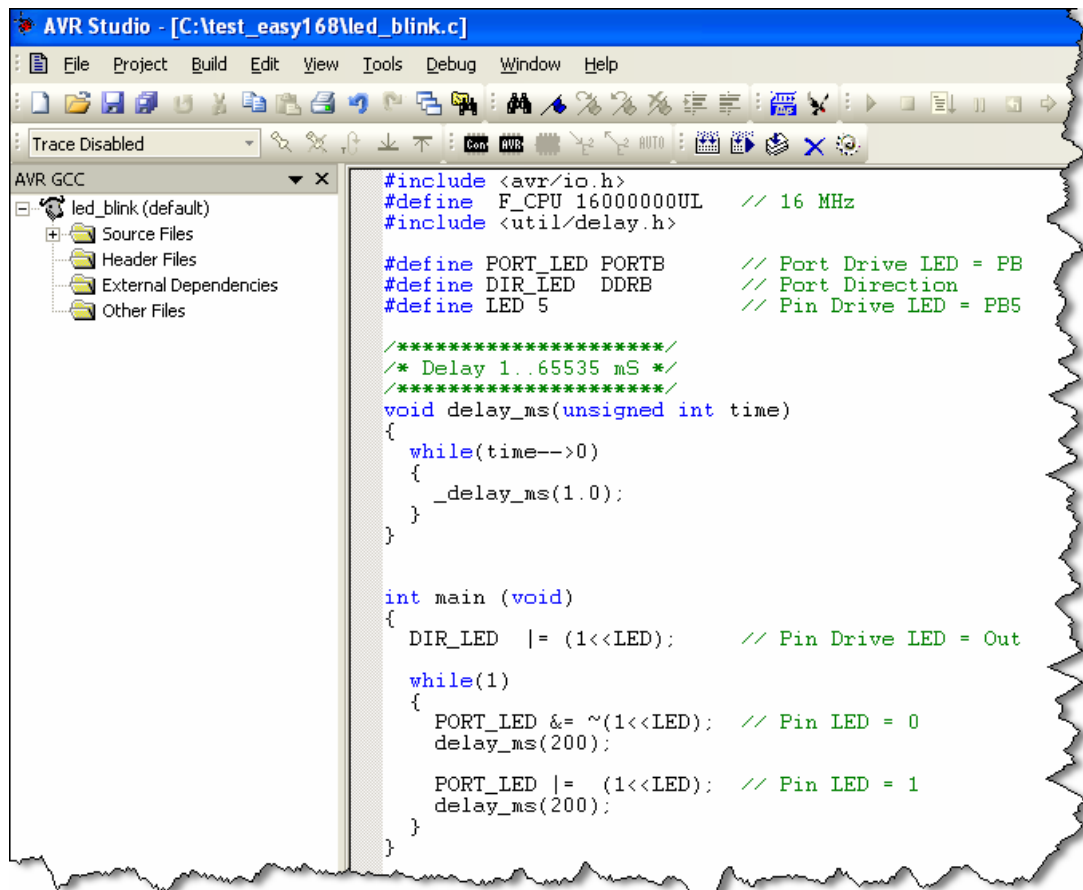
#define PORT_LED PORTB // Port Drive LED = PB
#define DIR_LED DDRB // Port Direction
#define LED 5 // Pin Drive LED = PB5

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

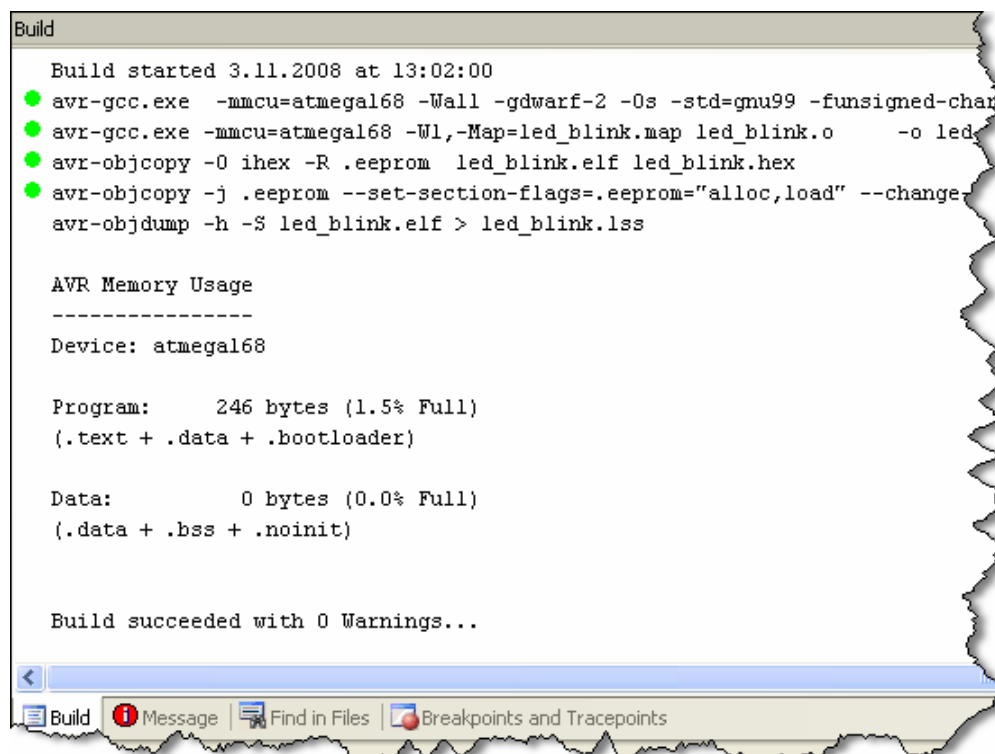
int main (void)
{
    DIR_LED |= (1<<LED); // Pin Drive LED = Out

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

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



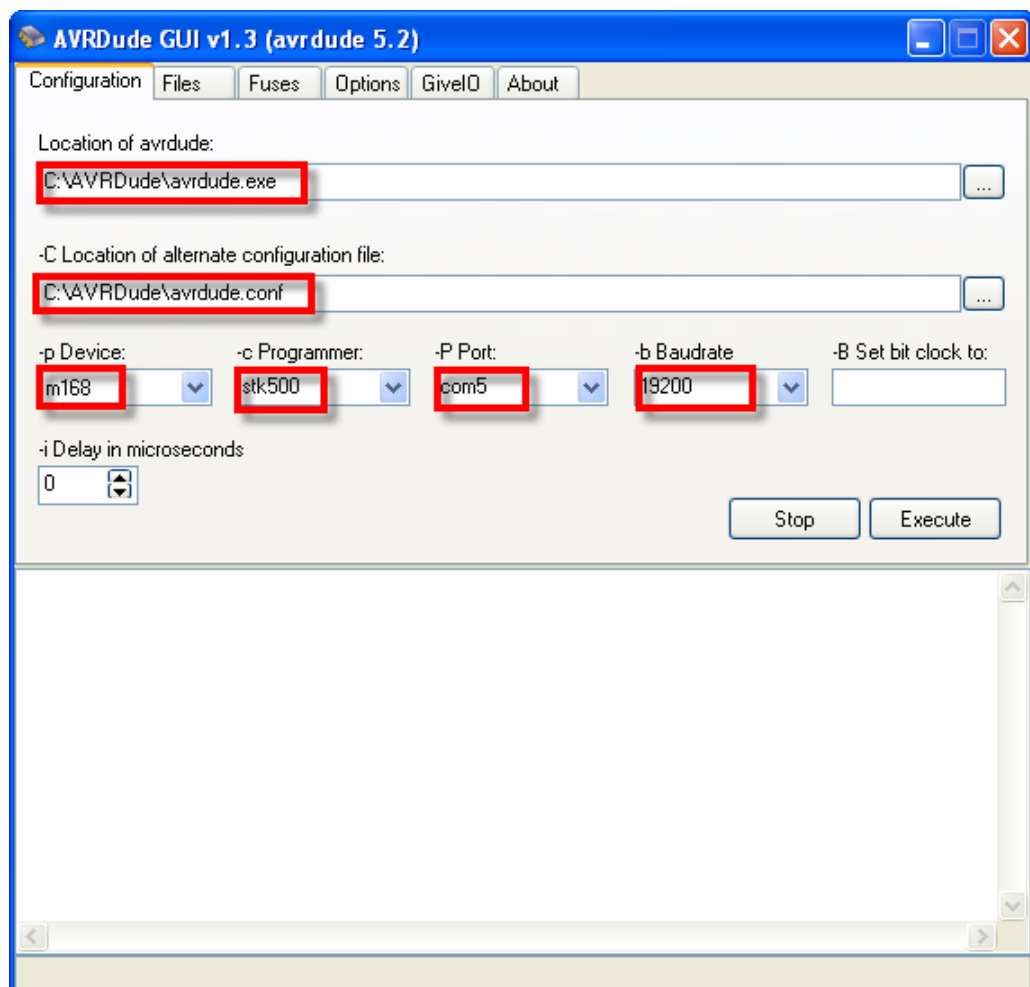
5. หลังจากพิมพ์ Code โปรแกรมเสร็จแล้วให้สั่งแปลโปรแกรม โดยเลือกที่ build → rebuild all ซึ่งผลการแปลคำสั่งจะได้เป็น HEX File ที่มีชื่อเดียวกันกับ project ที่สร้างไว้ ดังรูป



## การส่งโปรแกรม Hex Code ให้กับบอร์ด ET-EASY168 STAMP ผ่าน Bootloader

ในการพัฒนาโปรแกรมของ บอร์ด ET-EASY168 STAMP ในรูปแบบของ AVR Microcontroller นั้น หลังจากที่ทำกรเขียนโปรแกรม และแปลงคำสั่งจนได้ HEX File เรียบร้อยแล้วในการส่ง Program HEX Code ที่ได้จากการแปลคำสั่งของโปรแกรมนี้นี้ ผ่านทางโปรแกรม Bootloader ของบอร์ดนั้น ขอแนะนำ ให้ใช้โปรแกรม AVRdude โดยใช้งานผ่าน AVRdudeGUI ซึ่งมีขั้นตอนดังนี้

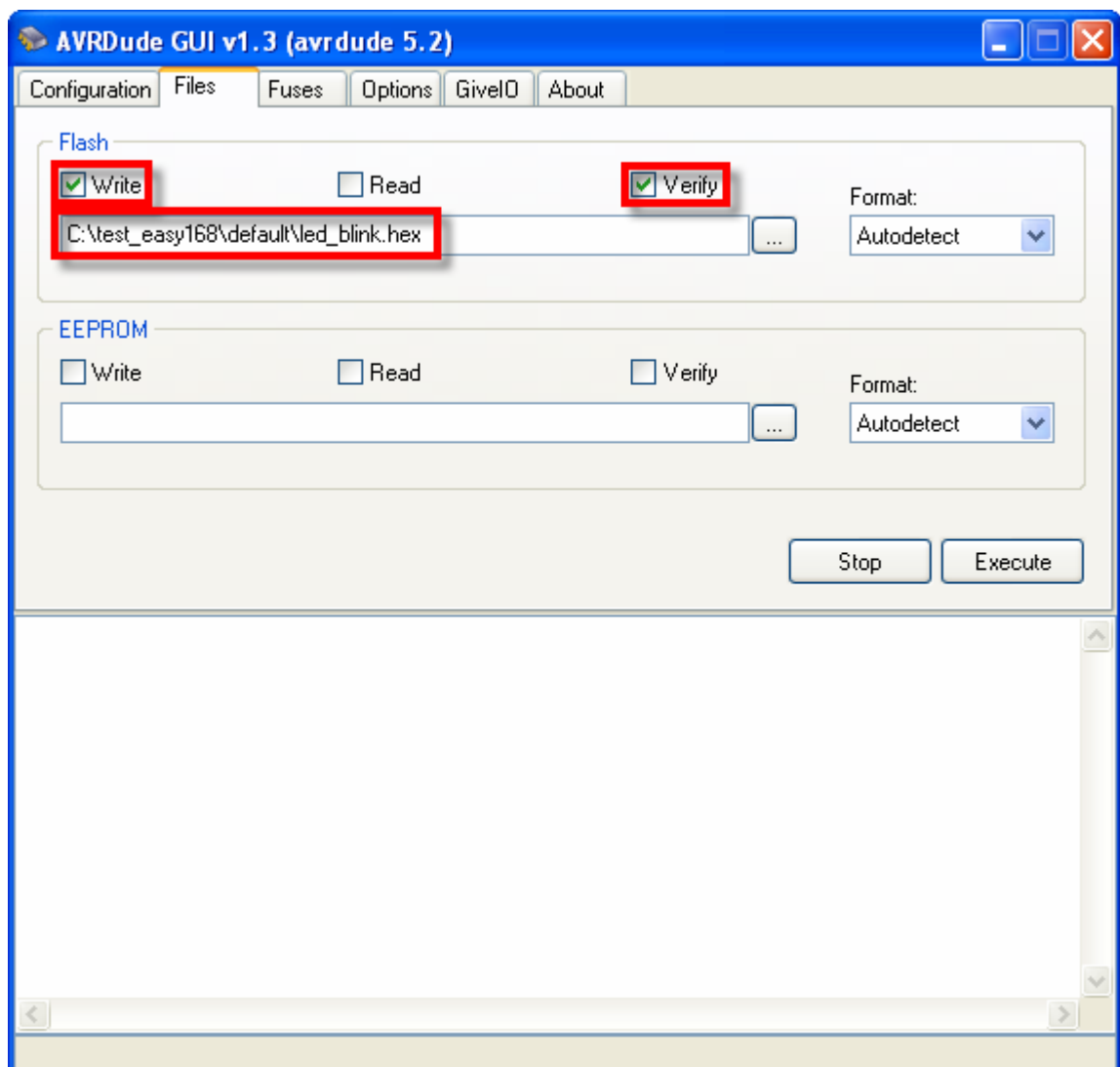
1. ทำการสร้างโฟลเดอร์ชื่อ AVRDUDE ไว้ใน Drive C ("C:\AVRDUDE\") จากนั้นให้ทำการ Copy โปรแกรมของ avrdude และ avrdudeGUI ไว้ใน "C:\AVRDUDE\" โดยจะมีทั้งหมด 3 ไฟล์ด้วยกัน คือ avrdude.exe, avrdude.conf และ avrdudegui.exe
2. สั่ง Run Program AVRdude.EXE แล้วเลือกที่ Tab ของ Configuration ให้เลือกกำหนดค่าต่างๆ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



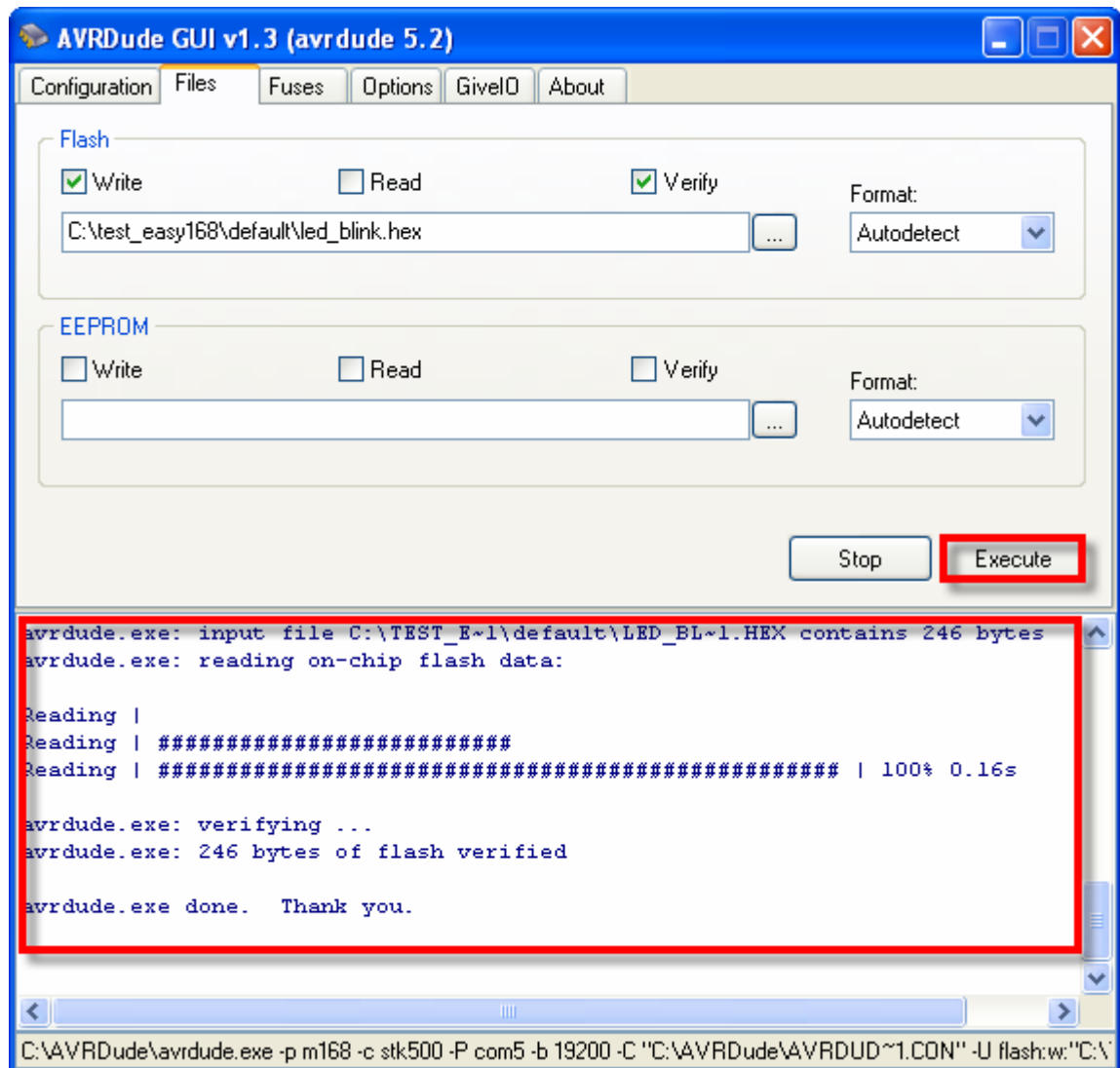
หมายเหตุ ถ้าหมายเลข Comport มากกว่า com8 ให้พิมพ์หมายเลข Comport ลงไปในช่องรับค่าของ -P Port ได้เอง เช่น com9 แต่ถ้าหมายเลข Comport มีค่าเป็น 2 หลักให้เพิ่มเครื่องหมาย "\\." นำหน้าชื่อหมายเลข Comport ลงไปด้วย เช่น ถ้าใช้กับ Com13 ก็ให้กำหนดเป็น "\\com13" เป็นต้น

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 m168
  - -c Programmer ให้เลือกเป็น stk500
  - -p Port ให้เลือกเป็นหมายเลข Comport ตามที่ได้ติดตั้ง Drive ของ USB(FTDI) ไว้
  - -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:\test\_easy\default\led\_blink.hex" ดังรูป



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



5. หลังจากส่งโปรแกรมเสร็จให้รอสักครู่ประมาณ 3 วินาที บอร์ดจะเริ่มทำงานทันที โดยจะเห็น LED กระพริบ ติด และ ดับ สลับกันไปมาไม่รู้จบ ซึ่งถ้าต้องการทดสอบการทำงานใหม่ ก็ให้ลอง กด สวิตช์ Reset ที่บอร์ด ซึ่งตอนเริ่มต้นจะเห็น LED กระพริบอย่างเร็วจำนวน 3 ครั้งแล้วติดค้าง ซึ่งช่วงนี้เป็นการทำงานของ MCU ตามโปรแกรมใน Bootloader ซึ่งเมื่อไม่มีการติดต่อสื่อสารเพื่อสั่งให้โปรแกรมเข้าทำงานในโหมด Program Code ของ Bootloader ภายในเวลา 3 วินาที MCU ก็ จะจบการทำงานจากส่วนของ Bootloader แล้วกระโดดไปทำงานในส่วนที่เป็น Code ของผู้ใช้งานทันที ซึ่งหลังการรีเซ็ตทุกครั้งจะเป็นเช่นนี้ไปตลอด โดยโปรแกรมที่เป็นส่วนของผู้ใช้จะเริ่มทำงาน หลังจากพ้นสภาวะรีเซ็ตไปแล้วประมาณ 3 วินาที

