

ET-EASY168 STAMP



รูปแสดงโครงสร้างของบอร์ด ET-EASY168 STAMP

ET-EASY168 STAMP เป็นบอร์ดไมโครคอนโทรเลอร์ในตระกูล AVR8 ขนาดเล็กจิ๋ว โดยมีขนาด ของบอร์ดเพียง 2cm x 5cm เท่านั้น ซึ่งขนาดบอร์ด ประมาณเท่ากับตัวถังของไอซี 28 DIP 300 โดยเลือกใช้ ไมโครคอนโทรเลอร์ตระกูล AVR8 เบอร์ ATmega168 ของ ATMEL เป็น MCU ประจำบอร์ด โดยเลือกใช้ MCU ที่มีรูปร่างตัวถังแบบ 32 TQFP พร้อมวงจรรอบนอกที่จำเป็นอย่าง Oscillator และ Reset รวมไว้ด้วย ภายในบอร์ด นอกจากนี้แล้วภายในตัวบอร์ดยังได้รวมเอาไอซี USB Bridge ของ FTDI เบอร์ FT232R เพื่อ ใช้ติดต่อสื่อสารแบบอนุกรมด้วย RS232 กับคอมพิวเตอร์ PC ผ่านทางพอร์ต USB ได้โดยตรง

ทำให้บอร์ด ET-EASY168 STAMP เป็นบอร์ดทดลองขนาดเล็กที่เพียบพร้อมไปด้วยวงจรพื้นฐานที่ จำเป็นต่อการใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR8 อย่างแท้จริง เพียงแต่เสียบสาย USB จากพอร์ต USB ของเครื่องคอมพิวเตอร์ PC เข้ากับขั้ว USB ของบอร์ด ET-EASY168 STAMP ก็สามารถทำการเขียน โปรแกรม และ Download Code ให้กับ MCU เพื่อทำการทดลองได้ทันที

คุณสมบัติของบอร์ด

- เลือกใช้ MCU ตระกูล AVR8 เบอร์ ATmega168 ของ ATMEL Run ความถี่ 16.00 MHz
 - O มีหน่วยความจำ Flash สำหรับเขียนโปรแกรม 16KByte ถ้าใช้การพัฒนาโปรแกรมผ่าน ระบบ AVRISP หรือ 14Kbyte เมื่อใช้การพัฒนาโปรแกรมผ่านระบบ Boot Loader RS232
 - o มี SRAM ใช้งานขนาด 1KByte และ EEPROM ใช้งานขนาด 512 Byte
 - o 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
- ขั้วต่อใช้งานวางตัวบน 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
- มีขั้วต่อ USB สำหรับเชื่อมต่อสื่อสารกับคอมพิวเตอร์ PC ผ่าน USB Bridge ของ FTDI ในรูปแบบ ของการสื่อสารอนุกรม RS232 สำหรับใช้งานสื่อสารและ Download Code ให้กับ MCU ในบอร์ด
- มีขั้ว AVRISP แบบ IDE 10PIN สำหรับใช้ Download โปรแกรมให้กับ MCU ภายในบอร์ดในกรณี ไม่ต้องการใช้การพัฒนาโปรแกรมผ่านทาง Boot Loader
- มี LED แสดงสถานะ โดยต่อกับ PB5 ของ AVR (Digital-13 ของ Arduino Project) สำหรับใช้เป็น อุปกรณ์ทดลองการทำงานอย่างง่าย



ET







รูปแสดง ลักษณะของบอร์ด ET-EASY168 STAMP

AVR	Arduino	Pin	ET-EASY168 STAMP	Pin	Arduino	AVR
/////	7 (1001110				711001110	7.010
PD0	Digital-0	1		28	+5V(+Vin)	+5V(+Vin)
PD1	Digital-1	2	AVRISP	27	+VCC(+5V)	+VCC(+5V)
PD2	Digital-2	3		26	RESET#	RESET(PC6)
PD3	Digital-3	4	E States	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 จากภายนอกเพื่อใช้เป็นแหล่งจ่าย ไฟเลี้ยงให้กับบอร์ด
- +VCC(+5V) เป็นขาแหล่งจ่ายไฟจุดเดียวกันกับที่ป้อนให้กับ +VCC ของ MCU ซึ่งจุดนี้จะรับ แรงดันมาจาก 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 ในกรณีต้องการใช้แรงดันอ้างอิงจากภายนอก
- 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 จากภายนอกเพื่อใช้เป็นแหล่งจ่าย ไฟเลี้ยงให้กับบอร์ด
- +VCC(+5V) เป็นขาแหล่งจ่ายไฟจุดเดียวกันกับที่ป้อนให้กับ +VCC ของ MCU ซึ่งจุดนี้จะรับ แรงดันมาจาก 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 ในกรณีต้องการใช้แรงดันอ้างอิงจากภายนอก
- 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

ETT

การพัฒนาโปรแกรมของบอร์ด ET-EASY168 STAMP

ในการพัฒนาโปรแกรมของบอร์ด ET-EASY168 STAMP นั้น ผู้ใช้สามารถเลือกใช้วิธีการพัฒนา โปรแกรมได้ 2 รูปแบบด้วยกัน คือ

- AVR Micro Controller เป็นการพัฒนาโปรแกรมตามรูปแบบของ AVR Micro Controller ปรกติ ซึ่งสามารถเลือกใช้โปรแกรมภาษาใดๆก็ได้ที่รองรับการใช้งานร่วมกับ AVR เบอร์ ATmega168 ซึ่ง ผู้ใช้สามารถเลือกใช้โปรแกรมในการพัฒนาได้ตามความถนัด 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 ในการพัฒนาให้ทั้งหมดและยังมีตัวอย่างโครงงานพร้อมตัวอย่างโปรแกรมการ ทดลองต่างๆแจกจ่ายให้ผู้สนใจนำมาใช้ศึกษา เรียนรู้และทดลอง ได้ฟรี โดยไม่เสียค่าใช้จ่ายใดๆ ผู้สนใจสามารถเข้าไปค้นหารายละเอียดต่างๆของ 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 เป็นตัวกลางในการเชื่อมต่อกับ คอมพิวเตอร์ PC โดย USB Bridge ของ FTDI จะทำหน้าที่เป็นตัวกลางในการเชื่อมต่อและติดต่อสื่อสาร ระหว่างเครื่องคอมพิวเตอร์ PC กับ MCU ATmega168 ของบอร์ด ET-EASY168 STAMP ในรูปแบบของ พอร์ตอนุกรม (Visual Com Port) โดยโปรแกรม Application ต่างๆที่ทำงานอยู่บนคอมพิวเตอร์ PC รวมทั้ง โปรแกรม Arduino จะมองเห็น พอร์ต USB ที่เชื่อมต่อกับบอร์ด ET-EASY168 STAMP เป็นพอร์ตสื่อสาร อนุกรม (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 แต่ถ้าเครื่อง คอมพิวเตอร์ PC ยังไม่เคยติดตั้ง Driver ของ FTDI ไว้ก่อนก็จะต้องทำการติดตั้ง Driver ให้กับบอร์ดให้ เรียบร้อยเสียก่อนซึ่งมีลำดับขั้นตอนดังนี้

- เตรียมแผ่น CD ROM ที่บรรจุ Driver ของ FTDI ไว้ให้พร้อม หรือ ในกรณีที่ผู้ใช้ได้ทำการติดตั้ง โปรแกรมของ Arduino ไว้เรียบร้อยแล้ว ภายในโฟล์เดอร์ของโปรแกรม Arduino ก็จะมี Driver ของ FTDI จัดเตรียมไว้ให้เรียบร้อบแล้ว โดยจะอยู่ที่ "C:\arduino-0012\drivers\FTDI USB Drivers\"
- ทำการเสียบสาย USB ของบอร์ด ET-EASY168 STAMP เข้ากับพอร์ต USB HUB ของเครื่อง คอมพิวเตอร์ PC ซึ่ง Windows จะตรวจพบอุปกรณ์ใหม่ โดยเป็น "FT232R USB UART" และ แจ้งให้ผู้ใช้ทำการติดตั้ง Driver ให้กับอุปกรณ์ ดังรูป

Found New Hardware Wizard					
	Welcome to the Found New Hardware Wizard				
	This wizard helps you install software for: FT232R USB UART				
	If your hardware came with an installation CD or floppy disk, insert it now.				
	What do you want the wizard to do?				
	 Install the software automatically (Recommended) Install from a list or specific location (Advanced) 				
	Click Next to continue.				
	< Back Next > Cancel				

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

Br	owse For Folder		? 🔀
9	elect the folder that contains drivers for your ha	ardware.	
	🖃 🚞 arduino-0012		~
	 Control drivers Control drivers FTDI USB Drivers Control drivers 		
	🗉 🧰 hardware		
	🗉 🚞 java		
	ib		
			×
۱	o view any subfolders, click a plus sign above.		
		ок	Cancel



Found New Hardware Wizard
Please choose your search and installation options.
 Search for the best driver in these locations. Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed. Search removable media (floppy, CD-ROM) Include this location in the search: C:\arduino-0012\drivers\FTDI USB Drivers Browse On't search. I will choose the driver to install. Choose this option to select the device driver form a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.
< Back Next > Cancel

4. ในขั้นตอนนี้โปรแกรม Windows จะทำการค้นหาและติดตั้ง Driver ให้กับอุปกรณ์ ให้รอสักครู่จน การทำงานเสร็จเรียบร้อย แล้วเลือก Finish ดังรูป







5. หลังจากทำการติดตั้ง Driver ของฮาร์ดแวร์เรียบร้อยแล้ว Windows ก็จะตรวจพบว่ามีอุปกรณ์ใหม่ ถูกเชื่อมต่ออยู่ โดยเป็นอุปกรณ์ประเภท "USB Serial Port" และแจ้งให้ผู้ใช้ทำการติดตั้ง Driver ให้กับอุปกรณ์ใหม่ที่ระบุเป็น "USB Serial Port" อีกครั้งหนึ่ง ซึ่งก็ให้เลือกระบุตำแหน่งโฟลเดอร์ที่ เก็บไฟล์ Driver ไว้ ซึ่งให้เลือกเหมือนขั้นตอนในหัวข้อที่ 3 ดังรูป

Found New Hardware Wiz	ard
	Welcome to the Found New Hardware Wizard This wizard helps you install software for: USB Serial Port If your hardware came with an installation CD or floppy disk, insert it now. What do you want the wizard to do? Install the software automatically (Recommended) Install from a list or specific location (Advanced)
	Click Next to continue.



Found New Hardware Wizard
Please choose your search and installation options.
 Search for the best driver in these locations.
Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.
Search removable media (floppy, CD-ROM)
 Include this location in the search: C:\arduino-0012\drivers\FTDI USB Drivers Browse
O Don't search. I will choose the driver to install.
Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.
K Back Next > Cancel

6. ในขั้นตอนนี้โปรแกรม Windows จะทำการค้นหาและติดตั้ง Driver ให้กับอุปกรณ์ ให้รอสักครู่จน การทำงานเสร็จเรียบร้อย แล้วเลือก Finish ดังรูป







7. หลังจากทำการติดตั้ง Driver เรียบร้อยแล้ว ก็สามารถใช้งานอุปกรณ์ได้แล้ว แต่เพื่อความถูกต้องใน ครั้งแรกนี้ควรต้องเข้าไปทำการตรวจสอบและปรับแต่งค่าให้กับอุปกรณ์ก่อน โดยในขั้นตอนนี้ให้ไป ที่ "My Computer → Control Panel → System → Hardware → Device Manager" แล้ว ทำการตรวจสอบที่ Ports (COM&LPT) แล้วดูที่ชื่อของ "USB Serial Port" ซึ่งให้ผู้ใช้จดจำ หมายเลขของ Com Port ของอุปกรณ์ดังกล่าวไว้ เพื่อใช้อ้างอิงถึงในการเรียกใช้งาน ดังรูป



- 8. ในขั้นตอนนี้ให้คลิกเมาส์ที่เครื่องหมาย (+) หน้าหัวข้อ Ports(COM&LPT) แล้วมองหาอุปกรณ์ที่ชื่อ "USB Serial Port" ตามที่เราได้ทำการติดตั้ง Driver ไว้เรียบร้อยแล้ว หรือ ถ้าไม่แน่ใจว่าอุปกรณ์ ดังกล่าวใช่อุปกรณ์ที่เป็นของบอร์ด "ET-EASY168 STAMP" หรือไม่ ให้ทดสอบด้วยการถอดสาย USB ออก รายชื่ออุปกรณ์ดังกล่าวจะต้องหายไป แต่เมื่อเสียบสาย USB กลับเข้ามาใหม่ รายชื่อ ของอุปกรณ์ดังกล่าวก็จะต้องปรากฏให้เห็นอีกครั้ง ถ้าทุกอย่างถูกต้อง ก็ให้ทำการคลิกเมาส์ที่ Tab รายชื่อของอุปกรณ์ดังกล่าว เมื่อปรากฏหน้าต่าง USB Serial Port Properties ขึ้นมาแล้วให้เลือก ที่ Port Setting แล้วเลือก Advance เพื่อเข้าไปกำหนดค่าให้กับอุปกรณ์ให้เรียบร้อย ดังนี้

 - b. USB Transfer Size Transmit (Bytes) ให้กำหนดเป็น 128
 - c. BM Option → Latency Timer (mSec) ให้กำหนดเป็น 1

USB Serial Port (COM5) Properties	
General Port Settings Driver Details	
Bits per second: 9600	
Data bits: 8	
Parity: None	
Stop bits: 1	
Flow control: None	
Advanced	
OK Cancel	

 \downarrow



การพัฒนาโปรแกรมของ ET-EASY168 STAMP แบบ Arduino Project

ตามปรกติแล้วบอร์ด ET-EASY168 STAMP จะทำการ ติดตั้งโปรแกรม Bootloader ไว้ให้เรียบร้อย แล้ว โดยใช้ Bootloader ชื่อ "BOOT_EASY168_AUTO_16MHZ.HEX" ซึ่งเป็น Bootloader ที่ทางทีมงาน อีทีที ได้นำต้นฉบับจาก Arduino มาปรับปรุงเงื่อนไขการทำงานใหม่ เพื่อให้การทำงานสอดคล้องกับระบบ ฮาร์ดแวร์ของบอร์ด ET-EASY168 STAMP ได้ดียิ่งขึ้น โดยโปรแกรม Bootloader นี้จะใช้สำหรับ ติดต่อสื่อสารเพื่อสั่ง 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 เพื่อไปเริ่มต้นทำงานตามคำสั่งที่ เป็นของผู้ใช้ โดยอัตโนมัติ โดยในขณะที่ Bootloader ทำงาน ตอนเริ่มต้นจะเห็น LED ที่ต่อไว้กับ ขาสัญญาณ Digital-13(PB5) กระพริบ 3 ครั้ง แล้วติดค้าง เพื่อรอการติดต่อสื่อสารจากโปรแกรม สำหรับสั่งให้ทำการ Upload Code ให้กับ MCU ซึ่งถ้าไม่มีการติดต่อสื่อสารจากโปรแกรมของ Arduino ภายในระยะเวลาประมาณ 3 วินาที โปรแกรมก็จะกระโดดไปทำงานในตำแหน่งเริ่มต้นที่ เป็นส่วนของโปรแกรมที่ผู้ใช้เขียนขึ้นทันที

นอกจากนี้แล้วผู้ใช้ยังสามารถเปลี่ยนไปใช้โปรแกรม Bootloader รุ่นมาตรฐาน ตามแบบโปรแกรม ของ Arduino ได้อีกด้วย โดยใช้โปแกรม Bootloader ที่ชื่อ "ATmegaBOOT_168_diecimila.hex" โดยไฟล์ ดังกล่าวจะถูกบรรจุไว้ใน Drive และโฟลเดอร์เดียวกันกับที่ผู้ใช้ได้ทำการติดตั้งโปรแกรมของ Arduino ไว้ แล้วคือ <u>"C:\arduino-0012\hardware\bootloaders\atmega168\ATmegaBOOT_168_diecimila.hex"</u>

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

การติดตั้งโปรแกรม Arduino

หลังจากที่เราได้ทำการติดตั้ง USB Driver ให้กับบอร์ดเป็นที่เรียบร้อยแล้ว ก็เป็นอันเสร็จสิ้นขั้นตอน ของการเตรียมการแล้ว ลำดับขั้นตอนต่อจากนี้เป็นต้นไป ก็เป็นเรื่องของการใช้งาน การเขียนโปรแกรม และ การศึกษาเรียนรู้ต่างๆตามความต้องการแล้ว แต่ก่อนอื่นเราจะต้องทำการติดตั้งโปรแกรมของ Arduino เพื่อ ใช้เป็นโปรแกรมสำหรับศึกษาเรียนรู้ ซึ่งมีลำดับขั้นตอนดังต่อไปนี้

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



 ในครั้งแรกของการเรียกใช้งานโปรแกรม ให้ทำการกำหนดระบบฮาร์ดแวร์ที่จะใช้งานกับโปรแกรม ของ Arduino ให้เรียบร้อยเสียก่อน เนื่องจากในปัจจุบันนี้ มีการออกแบบวงจรและสร้างฮาร์ดแวร์ บอร์ดแบบต่างๆสำหรับนำมาใช้งานร่วมกับโปรแกรมพัฒนาของ Arduino ไว้มากมายหลายรุ่น โดย ในกรณีของบอร์ด ET-EASY168 STAMP ให้ทำการเลือกกำหนดชื่อบอร์ดเป็น "Diecimila" โดย คลิกเมาส์ที่ "Tools → Board → Arduino Diecimila" ดังรูป

Arduino - 00	12 Alpha	
File Edit Sketch	Tools Help	
<u> </u>	Auto Format Ctrl+T	
	Copy for Forum	
sketch_511030	Archive Sketch	Ð
	Board 🕨 🕨	Arduino Nano 📃 🔼
	Serial Port 🔹 🕨	Arduino Pro or Pro Mini (8 MHz)
	Burn Bootloader 🔹 🕨	Arduino BT
	1	Arduino NG or older w/ ATmega8
		Arduino NG or older w/ATmega168
		Arduino Mini
		◆ Arduino Diecimila
		LilyPad Arduino
		×
		<u> </u>
1		

 เลือกกำหนดหมายเลขพอร์ต สำหรับติดต่อสื่อสารกับบอร์ด ให้ตรงกับหมายเลข Comport ที่ได้ทำ การติดตั้ง Driver ของ USB ไว้ในตอนแรก เช่น ถ้าตอนติดตั้ง Driver ของ USB แล้วได้หมายเลข Comport เป็น COM5 ให้คลิกเมาส์ที่ Tools → Serial Port → COM5 ดังรูป

🗿 Arduino - 0012 Alpha				
File Edit Sketch Tools Hel	p			
Auto Fo	rmat Ctrl+T r Forum			
Blink Archive	Sketch			₽
/* Board	•			
* Blink Serial Po	ort 🕨	COM3		
* Burn Bo	otioader 🕨 🕨	COM4		
* The basic A rduino e s	ampie, iur	✓ COM5	D on for one second	1
* then off for one set	cond, and so	on we	use pin 13 because,	
* depending on your An * or a built-in region	cduino board,	it has e	ither a built-in LED	
* OF A DUIIC-IN LESIS	cor so chac y	iou neeu o.	niy an LED.	
* http://www.arduino.(cc/en/Tutoria	al/Blink		
*/				
<pre>int ledPin = 13;</pre>	11	LED conne	cted to digital pin .	13
<pre>void setum()</pre>	11	run once.	when the sketch sta	rts
{		,		
pinMode(ledPin, OUTP	лт); //	sets the	digital pin as outpu	t
}				
maid lass ()			and amon avain	
	//	run over	and over again	
<pre>digitalWrite(ledPin,</pre>	HIGH): //	sets the	LED on	
delay(1000);	11	waits for	a second	
digitalWrite(ledPin,	LOW); //	sets the :	LED off	
<pre>delay(1000);</pre>	17	waits for	a second	
}				
				~
<				>
1				

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);
    delay(1000);
}
```

6. สั่งแปลโปรแกรมโดยคลิกเมาส์ที่ "Sketch → Verify/Compile" ดังตัวอย่าง



 สั่ง Download Code ให้กับบอร์ด โดยคลิกเมาส์เลือกที่ "File → Upload to I/O Board" แล้วรอ สักครู่จนโปรแกรมทำงานเสร็จ ซึ่งควรได้ผลดังรูป

🚰 Arduino - 0012 Alpha		<
File Edit Sketch Tools Help		
New Ctrl+N 🖳		
Sketchbook 🔹 🕨 💾		
Save Ctrl+S	₽	
Save As Ctrl+Shift+S		~
Upload to I/O Board Ctrl+U		
Page Setup Ctrl+Shift+P		
Print Ctrl+P	Turns on an LED on for one second,	
Preferences Ctrl+,	nd so on We use pin 13 because,	
Quit Ctrl+Q	board, it has either a built-in LED	
*	ande you need only an mo.	
* http://www.arduino.cc/en/T	utorial/Blink	
*/		
<pre>int ledPin = 13;</pre>	// LED connected to digital pin 13	=
void setum()	// run once when the skatch starts	
{	// fair once, when are sketch starts	
<pre>pinMode(ledPin, OUTPUT);</pre>	// sets the digital pin as output	
}		
void loop()	// run over and over again	
{ digitalWrite(ledDin_HTCH);	// sats the IFD on	
delav(1000);	// waits for a second	
digitalWrite(ledPin, LOW);	// sets the LED off	
delay(1000);	// waits for a second	
}		~
<	>	
Done uploading.		
Binary sketch size: 1124 bytes	s (of a 14336 b y te maximum)	
1		

 หลังจากที่ทำการ Upload Code ให้กับบอร์ดเป็นที่เรียบร้อยแล้ว บอร์ดก็จะเริ่มต้นทำงานตาม คำสั่งที่เขียนไว้ในโปรแกรมทันที โดยจะสังเกตเห็น LED กระพริบ ติด และ ดับ สลับกันไปมา ด้วย ความเร็วประมาณ 1 วินาที ตลอดเวลา

การโปรแกรม Bootloader ให้กับบอร์ด ET-EASY168 STAMP

ตามปรกติแล้วบอร์ด ET-EASY168 STAMP ได้ทำการโปรแกรม Bootloader ไว้ให้เป็นที่เรียบร้อย แล้ว ผู้ใช้สามารถใช้งานได้ทันที่ แต่อย่างไรก็ตามในกรณีที่ต้องการเปลี่ยนแปลง Bootloader หรือ เกิด ความผิดพลาดในการใช้งาน จนทำให้ Bootloader เสียหายไป ผู้ใช้ก็ยังสามารถทำการ โปรแกรม Bootloader ให้กับบอร์ดได้ใหม่ โดยบอร์ด ET-EASY168 STAMP ได้ออกแบบ ให้มีขั้วต่อ AVRISP สำหรับ ใช้เป็นช่องทางในการโปรแกรม Code ให้กับ MCU ได้โดยตรง ด้วยเครื่องโปรแกรมทุกรุ่นที่มีขั้วต่อ ตรงตาม มาตรฐาน AVRISP ของ ATMEL ได้ทันที โดยในที่นี้จะขอแสดงตัวอย่าง วิธีการ โปรแกรม Bootloader โดย ใช้เครื่องโปรแกรม ของ อีทีที รุ่น "ET-AVR ISP USB V1.0" โดยใช้โปรแกรม "AVR Studio 4" ของ ATMEL เป็นตัวจัดการ ซึ่งมีลำดับขั้นตอนดังต่อไปนี้

- ต่อสาย USB ให้กับบอร์ด ET-EASY168 STAMP ในกรณีต้องการใช้ไฟเลี้ยงจากพอร์ต USB หรือ จ่ายไฟเลี้ยง +5V ให้กับบอร์ดที่ขา 28(+5V)
- ต่อสาย USB ให้กับเครื่องโปรแกรม ET-AVR ISP USB V1 พร้อมทั้งต่อสายแพร์ 10 Pin ระหว่าง ขั้วต่อของ AVRISP ของทั้ง 2 บอร์ดเข้าด้วยกัน



3. สั่ง Run โปรแกรม AVR Studio 4 ดังรูป

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

File Project Build View	Debug Help AVR Prog ICE50 Upgrade ICE50 Selftest JTAGICE mkII Upgrade AVRISP mkII Upgrade AVR Dragon Upgrade STK600 Upgrade Customize Ontions	るる。 ないで してい してい してい してい してい してい してい してい
Fr	Show Key Assignments Plug-in Manager	Connect
	Flip 3.2.0	Auto Connect Write Flash Write EEPROM Read EEPROM Start Auto
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	m m m m m m m m m m m m m m m m m m m

Select AVR Programmer		
Platform: STK500 or AVRISP JTAGICE JTAGICE mkll AVRISP mkll AVR Dragon STK600	Port: COM4 COM5 COM6 COM7 COM8 COM9	Connect Cancel Baud rate:
STK600       COM9         Tip: To auto-connect to the programmer used last time, press the 'Programmer' button on the toolbar.         Note that a tool cannot be used for programming as long as it is connected in a debugging session. In that case, select 'Stop Debugging' first.         Disconnected Mode		Baud rate changes are active immediately.

 หลังจากที่ทำการ Connect เรียบร้อยแล้วให้ทดสอบการเชื่อมต่อโดยเลือกที่ tab ของ Main พร้อม กับเลือกเบอร์ของ MCU เป็น ATmega168 และเลือก Programming Mode and target Settings เป็น ISP mode แล้วทดสอบเลือก Read Signature ดู ซึ่งถ้าทุกอย่างถูกต้องควรได้ผลดังรูป

AVRISP in ISP mode with ATmega168	
Main Program Fuses LockBits Advanced HW Se	ettings   HW Info   Auto   🛛
Device and Signature Bytes	(
ATmega168	Erase Device
0x1E 0x94 0x06	Read Signature
Signature matches selected device	~ ~
Programming Mode and Target Settings	}
ISP mode	Settings
·	ISP Frequency: 230.4 kHz
	>
	man

- 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 จากนั้นให้
     เลือกที่ Program เพื่อสั่ง Program Bootloader ให้กับ MCU ซึ่งจะได้ผลดังรูป



- หลังจากที่ทำการโปรแกรม Code ให้กับ MCU เรียบร้อยแล้ว ให้เลือกมาที่ Tab ของ Fuses เพื่อทำ การสั่งโปรแกรม 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

A	AVRISP in ISP mode with ATmega168				
	Main Program	Ises LockBits Advanced HW Settings HW Info Auto	$\langle$		
	BOOTSZ	Boot Flash size=1024 words start address=\$1000	- {		
	RSTDISBL DWEN		-i		
	SPIEN				
	EESAVE		$\rightarrow$		
	CKDIV8	Brown-out detection disabled	-7		
	CKOUT SUT_CKSEL	Ext. Crystal Osc. 8.0- MHz; Start-up time PWRDWN/RESET: 16K	CK/14 式		
			5		
[	A CLEN CON		~		

 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

AVRISP in ISP mode with ATmega168		
Main Program Fus	es LockBits Advanced HW Settings HW Info Auto	
LB	No memory lock features enabled	5
BLBO	No lock on SPM and LPM in Application Section	5
BLB1	LPM and SPM prohibited in Boot Section	<
ll man	mun and	المستر

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

ตามปรกติแล้ว USB Bridge ของ FTDI เบอร์ FT232R จะถูกกำหนด Configuration ค่าให้ใช้ แหล่งจ่ายจากพอร์ต USB โดยตามค่ามาตรฐานของ USB Driver ของ FTDI นั้นจะร้องขอกระแสไฟ ไปยัง USB Host เพียงแค่ 90mA เท่านั้น ซึ่งตามมาตรฐานแล้วพอร์ต USB สามารถจ่ายกระแสไฟให้กับอุปกรณ์ที่ เชื่อมต่อด้วยสูงสุดที่ 500mA ซึ่งเพื่อความสะดวกในการทดลองต่างๆ ด้วยบอร์ด ET-EASY168 STAMP ใน กรณีที่ยังไม่ได้เชื่อมต่อกับอุปกรณ์ภายนอกมากมายนัก(ใช้กระแสไม่เกิน 500mA) เราก็สามารถเข้าไปสั่ง แก้ไขเปลี่ยนค่า Configuration ของ USB HUB เพื่อให้จ่ายกระแสออกมาให้ครบ 500mA ได้ด้วย ซึ่งจะทำ ให้เกิดความสะดวกและคล่องตัวในการทดลองต่างๆมากขึ้นด้วย แต่อย่างไรก็ตามถ้าหากว่าผู้ใช้ ทำการต่อ ทดลองกับ เครื่องคอมพิวเตอร์แบบ Notebook ที่ใช้กระแสไฟจาก Battery หรือมีการเชื่อมต่อบอร์ดทดลอง กับอุปกรณ์ที่ใช้กระแสไฟมากๆ ก็ควรจัดหาแหล่งจ่ายไฟภายนอกเพื่อจ่ายให้กับบอร์ดเองจะเหมาะสมกว่า ข้อแนะนำนี้เป็นการแนะนำสำหรับผู้ที่ต้องการทดสอบการทำงานหรือทดลองเขียนโปรแกรมกับบอร์ด แบบ ที่ไม่ได้เน้นการเชื่อมต่อกับอุปกรณ์ภายนอกมากมายนัก เช่น ทดสอบไฟกระพริบ โดยใช้ LED เป็นอุปกรณ์ การทดลอง หรือ การรับส่งข้อมูลกับพอร์ตสี่อสารอนุกรม เป็นต้น

โดยการแก้ไขค่า Configuration ของ USB Bridge ของ FTDI จะใช้โปรแกรม "MProg.exe" เป็น ตัวจัดการโดยมีลำดับขั้นตอนดังต่อไปนี้

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

MProg - Multi Device EEPROM Prog	rammer V 3.0a	×
File Device Tools Help		
	🔎 🖆 🗡 ? 🥹	<b></b>
Basic Details         Device Type         None         USB VID / PID         FTDI Default         Vendor ID         0403         Product ID         6001    BM / C Device Specific Options          USB Version Number         USB Version Number         Disable USB Serial Number         Pull Down IO Pins in USB Suspend	USB Power Options Bus Powered Self Powered USB Serial Number Control Serial Number Prefix ( 2 digits ) USB Fixed Serial Number Fixed Serial Number ( 8 digits ) D00000001 USB Remote Wake Up Enable USB Remote Wake Up Plug & Play (FT232 Series Only ) Enable Plug And Play	FT2232C Options FT232R
Product and Manufacturer Descriptor Strings         Manufacturer       Product Description         FTDI       USB <-> Serial cak         Programming Options       Image: Construction of the series of th		



 สั่งโปรแกรม "mProg" ค้นหา USB Bridge ที่ติดตั้งไว้ในเครื่อง โดยเลือกที่เมนู Device → Scan ซึ่งควรได้ผลดังรูป

MProg - Multi Device EEPROM Progr	ammer V 3.0a	
File Device Tools Help		
Scan Ctrl+C	🔎 🖆 🗡 ? 🔞	<b></b>
Ctrl+E     Program Ctrl+P     Device Type     Point     USB VID / PID FTDI Default     Vendor ID 0403 Product ID 6001     BM / C Device Specific Options     USB Version Number. USB 20	USB Power Options Max Bus Power Bus Powered Self Powered USB Serial Number Control Serial Number Prefix ( 2 digits ) Use Fixed Serial Number Fixed Serial Number ( 8 digits )	FT2232C Options FT232R
OSB 2.0      OSB 2.0	00000001 USB Remote Wake Up Enable USB Remote Wake Up Plug & Play (FT232 Series Only ) Enable Plug And Play	
Product and Manufacturer Descriptor Strings           Manufacturer         Product Description           FTDI         USB <-> Serial cabl	e	
Programming Options Only Program Blank Devices Number Of Blank Devices = 0 Number Of Programmed Devices = 1		

 สั่งอ่านค่า Configuration เดิมของ FTDI ออกมา โดยเลือกที่ Tools → Read and parse ซึ่งควร ได้ผลดังรูป



MProg - Multi Device FEPROM Prog	rammer ( Edit Mode )	X
File Device Tools Help		
	P 🛎 🗡 ? 🥝	ď
Basic Details         Device Type       FT232R         USB VID / PID       FTDI Default         Vendor ID       0403         Product ID       6001         BM / C Device Specific Options         USB Version Number         Disable USB Serial Number         Pull Down IO Pins in USB Suspend	USB Power Options Bus Powered Self Powered USB Serial Number Control USB Serial Number Control Serial Number Prefix ( 2 digits ) USB Fixed Serial Number Fixed Serial Number ( 8 digits ) Disabled USB Remote Wake Up V Enable USB Remote Wake Up Plug & Play (FT232 Series Only ) Enable Plug And Play	FT2232C Options       FT232R         Invert RS232 Signals       Invert TXD         Invert RXD       Invert RXD         Invert RTS#       Invert CTS#         Invert DTR#       Invert DCD#         Invert RI#       Invert RI#         I/O Controls       TXLED#       C0         RXLED#       C1       TXDEN
Product and Manufacturer Descriptor Strings         Manufacturer         Product Description         FTDI         FT232R USB UART         Programming Options         Only Program Blank Devices		PV/RON# C3 SLEEP# C4 High Current I/O's Load D2XX driver

 ให้ทำการแก้ไขค่า Configuration ในส่วนของ USB Power Options ให้เลือกเป็น Bus Powered และกำหนดค่า Max Bus Power จากเดิมที่กำหนดไว้ที่ <u>90 milli Amps</u> เป็น <u>500 milli Amps</u> ส่วน ค่า Configuration อื่นๆไม่ต้องเปลี่ยนแปลง จากนั้นให้ทำการบันทึกค่า Configuration ไว้ในเครื่อง ก่อนโดยให้เลือกที่ File → Save As.. ดังรูป



สั่งโปรแกรมค่า Configuration คืนให้กับ FTDI โดยให้เลือกที่ Device → Program แล้วรอจน
 โปรแกรมทำงานเสร็จ หลังจากนั้นให้ ถอดสาย USB ออกแล้วเสียบกลับเข้าไปใหม่ USB ก็จะ
 ทำงานตามค่า Configuration ใหม่ได้แล้ว



🖉 MProg - Multi Device EEPROM Prog	rammer ( Program Mode )	$\mathbf{X}$
File Device Tools Help		
Ctrl+C	P 🖆 🗡 ? 🎯	<u> </u>
USB VID / PID FTDI Default Vendor ID 0403 Product ID 6001 BM / C Device Specific Options USB Version Number USB 2.0 Disable USB Serial Number Pull Down IO Pins in USB Suspend	USB Power Options Bus Powered Self Powered USB Serial Number Control Serial Number Prefix ( 2 digits ) Use Fixed Serial Number Fixed Serial Number ( 8 digits ) DiREINWM USB Remote Wake Up USB Remote Wake Up Plug & Play (FT232 Series Only ) Finable Plug And Play	FT2232C Options FT232R  Invert RS232 Signals Invert TXD Invert RXD Invert RTS# Invert CTS# Invert DTR# Invert DCR# Invert DCD# Invert RI#  I/O Controls  TXLED# C0  RXLED# C1  TXDEN C2
Product and Manufacturer Descriptor Strings       Manufacturer     Product Description       FTDI     FT232R USB UART		PVVRON# C3 SLEEP# C4
FTDI     FT232R USB UART       Programming Options     Only Program Blank Devices       Programmed Serial Number : Disabled     Image: Construction of the series of t		☐ High Current I/O's ☐ Load D2XX driver

## การพัฒนาโปรแกรมของ ET-EASY168 STAMP แบบ AVR Microcontroller

ในกรณีที่ผู้ใช้ต้องการที่จะพัฒนาโปรแกรมให้กับบอร์ด ET-EASY168 STAMP แบบปรกติ ใน รูปแบบของไมโครคอนโทรลเลอร์ AVR ก็สามารถทำได้ โดยสามารถเลือกภาษาในการเขียนโปรแกรม ได้เอง ซึ่งในกรณีนี้ผู้ใช้สามารถเลือกใช้ภาษาใดๆก็ได้ที่รองรับการใช้งานกับ MCU ตระกูล AVR เบอร์ ATmega168 โดย ผู้ใช้งานสามารถบริหารจัดการระบบทรัพยากรต่างๆที่อยู่ภายในตัว ATmega168 ได้เอง ทั้งหมด โดยสามารถทำได้ 2 แนวทางคือ

- การพัฒนาโปรแกรมโดยใช้เครื่องโปรแกรมภายนอก ในกรณีนี้จะมีข้อดีคือ ไม่สูญเสีย ทรัพยากรใดๆเลย ผู้ใช้สามารถใช้งานและกำหนดคุณสมบัติการทำงานของทรัพยากรต่างๆที่มีอยู่ ใน MCU ได้เอง ตามต้องการ แต่มีข้อเสียคือ ต้องมีเครื่องโปรแกรมภายนอก สำหรับทำหน้าที่ โปรแกรม Code ให้กับ MCU โดยสามารถใช้ได้กับเครื่องโปรแกรมทุกรุ่นที่ รองรับการใช้งานกับ MCU เบอร์ ATmega168 และมีขั้วต่อตามมาตรฐาน AVRISP ของ ATMEL
- การพัฒนาโปรแกรมโดยใช้งานร่วมกับ Bootloader ในกรณีนี้จะมีข้อดี คือ เมื่อทำการเขียน โปรแกรมเสร็จแล้วสามารถสั่ง Program Code ให้กับ MCU ผ่านทางโปรแกรม Bootloader ที่ ติดตั้งไว้แล้วได้ทันที โดยไม่ต้องใช้เครื่องโปรแกรมภายนอก แต่มีข้อจำกัด คือ ต้องเสียพื้นที่ หน่วยความจำ Flash สำหรับเก็บ Code โปรแกรมไป 2KByte (0x3800 -0x3FFF) เพื่อใช้ติดตั้ง โปรแกรม Bootloader ซึ่งตามปรกติแล้วบอร์ด ET-EASY168 STAMP จะทำการ ติดตั้งโปรแกรม Bootloader ไปให้เรียบร้อยแล้ว โดยในกรณีนี้ผู้ใช้จะสามารถเขียนโปรแกรมใช้งานได้ทั้งหมด จำนวน 14 Kbyte จากที่มีอยู่ทั้งหมด 16Kbyte โดยต้องเขียนโปรแกรมให้มีตำแหน่งการทำงานของ Code อยู่ระหว่างตำแหน่งแอดเดรส 0x0000 ถึง 0x37FF (0x1C00 - 0x1FFF K Word) ส่วน SRAM,EEPROM และทรัพยากรอื่นๆที่มีอยู่ใน MCU ผู้ใช้ยังสามารถใช้งานได้ครบตามจำนวนที่มี อยู่ในตัว MCU ทุกประการ

#### ตัวอย่างการพัฒนาโปรแกรมด้วย WinAVR โดยใช้งานร่วมกับ Bootloader

ตัวอย่างที่จะแสดงต่อไปนี้จะแสดงให้เห็นวิธีการพัฒนาโปรแกรมให้กับ ATmega168 ด้วยภาษาซี

โดยใช้โปรแกรม AVR Studio4 ของ ATMEL ร่วมกับ Compiler ภาษาซีของ WinAVR

1. สั่ง Run Program AVR Studio4



- 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_easy168\"
  - Poject name ให้กำหนดชื่อ project ตามต้องการในตัวอย่างกำหนดเป็น "led_blink" และให้ เลือก Create initial file ไว้ด้วย ซึ่งเมื่อเราทำการกำหนดชื่อ project name เสร็จแล้ว โปรแกรม จะสร้างไฟล์ ที่มีชื่อเดียวกันกับ project name ให้เองโดยอัตโนมัติ

เมื่อกำหนดค่าตัวเลือกต่างๆ ให้กับโปรแกรมเรียบร้อยแล้ว ให้เลือกที่ Next แล้วกำหนดค่าใน
 Debug platform เป็น AVR Simulator และเลือก Device เป็น ATmega168 ดังรูป

Ver 4.14.589 T Show d	Create new project Project type: Atmel AVB Assembler	Project name: led_blink Create initial file Initial file: led_blink .c
	Location: C:\test_easy168\ alog at startup	



10 A	AVR Dragon AVR Simulator AVR Simulator 2 ICE 200 ICE 40	ATmega164P ATmega165 ATmega165P ATmega168 ATmega168P	^
Sind	ICE50 JTAG ICE JTAGICE mkll	ATmega169 ATmega169P ATmega16HVA ATmega2560 ATmega2561 ATmega32	×
	FC	pen platform options	

 ให้พิมพ์คำสั่งของโปรแกรมสำหรับทดสอบการทำงาน ในหน้าต่าง Text Editor ของโปรแกรม โดย ในที่นี้จะทดสอบด้วย Codeโปรแกรม สำหรับทำหน้าที่เป็นไฟกระพริบที่ PB5 ซึ่งเป็น LED ที่ติดตั้ง ไว้บนบอร์ด 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 1600000UL // 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)
ł
 DIR_LED = (1<<LED); // Pin Drive LED = Out
 while(1)
 {
   PORT LED &= \sim (1<<LED); // Pin LED = 0
   delay_ms(200);
   PORT_LED = (1<<LED); // Pin LED = 1
    delay_ms(200);
 }
```

AVR Studio - [C:\test_easy168\	ed_blink.c]
Eile Project Build Edit View	Iools Debug Window Help
: 🗋 📂 🔲 🕼 🕫 X 🗈 🛍 🖨 🛚	9 [2] 🔁 骗 🛦 🖄 🖄 🍇 崖 韋 [1] 🎆 ❤ [1] 🕨 🖬 👔 🔹 🕹
i Trace Disabled	
AVR GCC	<pre>#include <avr io.h=""> #define F_CPU 16000000UL // 16 MHz #include <util delay.h=""></util></avr></pre>
	#define PORT_LED PORTB       // Port Drive LED = PB         #define DIR_LED DDRB       // Port Direction         #define LED 5       // Pin Drive LED = PB5
	/***********************
	/* Delay 165535 mS */
	void delay_ms(unsigned int time)
	t while(time>0)
	ms(1.0); }
	) · · · · · · · · · · · · · · · · · · ·
	l l l l l l l l l l l l l l l l l l l
	int main (void)
	DIR_LED  = (1< <led); drive="" led="Out&lt;/td" pin=""></led);>
	while(1)
	<pre>PORT_LED &amp;= ~(1&lt;<led); delay_ms(200);<="" led="0" pin="" pre=""></led);></pre>
	<pre>PORT_LED  = (1&lt;<led); delay_ms(200);="" led="1" pin="" pre="" }<=""></led);></pre>

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



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

ในการพัฒนาโปรแกรมของ บอร์ด ET-EASY168 STAMP ในรูปแบบของ AVR Microcontroller นั้น หลังจากที่ทำการเขียนโปรแกรม และสั่งแปลคำสั่งจนได้ HEX File เรียบร้อยแล้วในการสั่ง Program HEX Code ที่ได้จากการแปลคำสั่งของโปรแกรมนี้ ผ่านทางโปรแกรม Bootloader ของบอร์ดนั้น ขอแนะนำ ให้ใช้โปรแกรม AVRDude โดยใช้งานผ่าน AVRDudeGUI ซึ่งมีขั้นตอนดังนี้

- 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
- สั่ง Run Program AVRDude.EXE แล้วเลือกที่ Tab ของ Configuration ให้เลือกกำหนดค่าต่างๆ ดังต่อไปนี้

AVRDude GUI v1.3 (avrdude 5.2)	×
Configuration Files Fuses Options GivelO About	_
Location of avrdude:	
C:VAVRDude\avrdude.exe	
-C Location of alternate configuration file:	
-p Device:     -c Programmer:     -P Port:     -b Baudrate     -B Set bit clock to:       m168     stk500     com5     19200	]
-i Delay in microseconds	
Stop Execute	
	^
	v
< >>	

หมายเหตุ ถ้าหมายเลข Comport มากกว่า com8 ให้พิมพ์หมายเลข Comport ลงไปในช่องรับค่า ของ –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 ที่ได้ติดตั้งไว้ตาม ขั้นตอนที่ 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
- เปลี่ยน Tab ไปที่ File แล้วเลือกกำหนดส่วนของ Flash โดยให้เลือก Write และ Verify พร้อมทำ กำหนดชื่อ HEX File ไฟล์ ที่ต้องการจะโปรแกรม ซึ่งในที่นี้จะใช้ HEX File ที่ได้จากตัวอย่างที่เขียน ด้วยภาษาซีของ WinAVR ซึ่งจะอยู่ใน "C:\test_easy\default\led_blink.hex" ดังรูป

🏶 AVRDude GUI v	1.3 (avrdude 5.2)		
Configuration Files	Fuses Options Give	elO About	
- Flash			
Vrite	📃 Read	Verify	Format:
C:\test_easy168\d	efault\led_blink.hex		Autodetect 🔽
EEPROM			
Write	Read	Verify	Format:
			Autodetect 🔽
			Stop Execute
			~
<			>

เมื่อกำหนดค่าต่างๆให้กับโปรแกรมเรียบร้อยแล้ว ให้เลือกที่ Execute ที่อยู่ใน Tab ของ Files ซึ่ง
 โปรแกรม avrdude จะเริ่มต้นทำการ โปรแกรม HEX File ให้กับบอร์ดทันที ให้รอจนเสร็จดังรูป

🛸 AVRDude GUI v1.3 (avı	dude 5.2)				
Configuration Files Fuses	Options GivelO Abou	ıt			
- Flash					
Vrite	Read	🔽 Verify	Format:		
C:\test_easy168\default\led	_blink.hex		Autodetect 🗸		
EEPROM					
Write	Read	🔲 Verify	Format:		
			Autodetect 🗸		
			Stop Execute		
avrdude.exe: input file avrdude.exe: reading or	/ C:\TEST_E~I\defaul) -chip flash data:	f/PRD_RP~1.HRX CO	ntains 246 bytes 🛕		
Reading					
Reading   ###################################					
Reading   ###################################					
avrdude.exe: verifying avrdude.exe: 246 bytes of flash verified					
avrdude exe done Thank you					
initiade.exe done. Ind.	n you.		E		
E - M			~		
<					

5. หลังจากสั่งโปแกรมเสร็จให้รอสักครู่ประมาณ 3 วินาที บอร์ดจะเริ่มต้นทำงานทันที โดยจะเห็น LED กระพริบ ติด และ ดับ สลับกันไปมาไม่รู้จบ ซึ่งถ้าต้องการทดสอบการทำงานใหม่ ก็ให้ลอง กด สวิตช์ Reset ที่บอร์ด ซึ่งตอนเริ่มต้นจะเห็น LED กระพริบอย่างเร็วจำนวน 3 ครั้งแล้วติดค้าง ซึ่ง ช่วงนี้เป็นการทำงานของ MCU ตามโปรแกรมใน Bootloader ซึ่งเมื่อไม่มีการติดต่อสื่อสารเพื่อสั่ง ให้โปรแกรมเข้าทำงานในโหมด Program Code ของ Bootloader ภายในเวลา 3 วินาที MCU ก็จะ จบการทำงานจากส่วนของ Bootloader แล้วกระโดดไปทำงานในส่วนที่เป็น Code ของผู้ใช้เรอง ทันที ซึ่งหลังการรีเซ็ตทุกครั้งจะเป็นเช่นนี้ไปตลอด โดยโปรแกรมที่เป็นส่วนของผู้ใช้จะเริ่มต้นทำงาน หลังจากพ้นสภาวะรีเซ็ตไปแล้วประมาณ 3 วินาที

