ET-AVR JTAG (RS232) V1.0

ET-AVR JTAG (RS232) V1.0 เป็นบอร์ดที่ออกแบบมาเพื่อใช้ในการดาวน์โหลด Hex File และ ดีบัก ให้กับ MCU ตระกูล AVR ของ Atmel โดยผ่านทาง JTAG Interface ซึ่งสามารถใช้ได้กับ MCU ที่มีโมดูล JTAG Interface เท่านั้น โดยต้องใช้ร่วมกับโปรแกรม AVR Studio 4.XX

คุณสมบัติของ ET-AVR JTAG (RS232) V1.0

- 1. มีคุณสมบัติเทียบเท่า AVR JTAG ICE ของ ATMEL
- 2. สนับสนุนการดีบักแบบเรียลไทม์
- 3. โปรแกรมและดีบัก MCU ที่มีโมดูล JTAG Interface
- สามารถอัพเกรด Firmware โดยตรงผ่านโปรแกรม AVR Studio 4 เพื่อให้สามารถใช้กับ MCU เบอร์ใหม่ ๆ ได้ ซึ่ง Firmware จะติดมากับโปรแกรม AVR studio 4
- 5. สามารถใช้ได้กับระบบไฟเลี้ยงตั้งแต่ 2.7V 5.5V
- 6. ใช้แรงดันจากบอร์ด Target
- 7. การติดต่อสื่อสารผ่านพอร์ตอนุกรม (RS232)
- 8. มี LED แสดงสถานการณ์ทำงาน Power , Activity

เบอร์ของไมโครคอนโทรลเลอร์ AVR ที่สามารถใช้ได้กับ ET-AVR JTAG (AVR studio 4.12)

- ATmega16, ATmega16L
- ATmega162, ATmega162L
- ATmega169, ATmega169L, ATmega169V
- ATmega32, ATmega32L
- ATmega323, ATmega323L
- ATmega64 , ATmega64L
- ATmega128, ATmega128L
- AT90CAN128



การเชื่อมต่อ ET-JTAG AVR กับไมโครคอนโทรลเลอร์ AVR

แสดงการเชื่อมต่อ ET-AVR JTAG กับ ไมโครคอนโทรลเลอร์ AVR

การต่อใช้งานจะใช้สายสัญญาณ TCK , TDO , TMS , TDI , VCC , GND เท่านั้นส่วนขาสัญญาณ VTref ไม่จำเป็นต้องต่อก็ได้เนื่องจากวงจรของ ET-JTAG AVR ขานี้จะต่อกับ VCC อยู่แล้ว



แสดงต่ำแหน่งขาสัญญาณของ ET-AVR JTAG





แสดงการเชื่อมต่อ ET-AVR JTAG กับ ET-AVR START KIT V1.0/EXP

การใช้งาน ET-AVR JTAG ในโหมดโปรแกรม

การในงานในโหมดโปรแกรมนี้สามารถกำหนด Security Bits และ Configuration Bits เช่นเดียวกับโปรแกรม PonyProg2000 ซึ่งวิธีการใช้งานมีดังนี้

 ทำการเปิดโปรแกรม AVR Studio จากนั้นจะปรากฏหน้าต่าง Welcome to AVR Studio ให้คลิกที่ Cancel เพื่อปิดหน้าต่างนี้





2.ทำการเลือกที่เมนู Tools → Program → AVR Connect... ดังรูป

AVR Studio				
Eile Project Build <u>V</u> iew	Too	ols <u>D</u> ebug <u>H</u> elp		
D 🚅 🖬 🕼 🗸 🖻		<u>A</u> VR Prog	1.	%%% ###
Trace Disabled		ICE50 Upgrade	1111 - 1	42 5 ×2 AUTO
I/O View		ICE50 <u>S</u> elftest		
Name		JTAGICE mkII Upgrade		
		<u>C</u> ustomize		
		Options		
		Show Key Assignments		
		<u>P</u> lug-in Manager		
	AVR	Program AVR 🔹 🕨		Connect
	_			Auto Connect ^{KS}
				Write Flash
				Write EEPROM
				Read EEPROM
				Start Auto

2. จากนั้นจะปรากฏหน้าต่าง Select AVR Programmer ดังรูป ให้ทำการเลือก Platform เป็น **JTAG** ICE และ Port ที่ต่อ ET-AVR JTAG ต่ออยู่ และคลิกปุ่ม Connect...

Select AVR Programmer	
Platform: STK500 or AVRISP JTAG ICE JTAGICE mkli AVRISP mkli	Port: COM5 CDM6 COM7 COM8 COM9 Cancel
Tip: To auto-connect to the program button on the toolbar. Note that the JTAGICE cannot be u connected in a debugging session. I Disconnected Mode	nmer used last time, press the 'Programmer' ised for programming as long as it is In that case, select 'Stop Debugging' first.

 ถ้าโปรแกรม AVR Studio สามารถติดต่อกับ ET-JTAG AVR ได้จะปรากฏหน้าต่าง JTAG ICE ดัง รูป ทำการเลือกเบอร์ MCU จากช่อง Device ในที่นี้เลือกเป็น ATmega64

JTAG ICE
Program Fuses LockBits Advanced Board Auto
- Device
ATmega64 Erase Device
Programming mode
C ISP IV Erase Device Before Programming Parallel/High Voltage Serial Verify Device After Programming
Flash
Use Current Simulator/Emulator FLASH Memory Input HFX File D:\AVB-STAMP\Examples\ASSEMBLY\LED
Brazen
EEPROM
Use Current Simulator/Emulator/EEPHUM Memory Input HEX File D:\AVR-STAMP_EXAMPLE\BASIC\LED_BLI
Program Verify Read
Detecting on 'COM7'
JTAG ICĒ found on COM7 Getting revisions HW: 0xc1, SW Major: 0x7f, SW Minor: 0x00 OK

 สั่งเปิดไฟล์สำหรับที่จะใช้โปรแกรมให้กับ CPU โดยเลือกจากช่อง Input HEX File พร้อมทั้งระบุชื่อ และที่อยู่ของ HEX File ที่จะใช้โปรแกรมให้เรียบร้อย ส่วนในช่อง Programming mode ให้เลือก
 Erase Device Before Programming เพื่อทำการลบข้อมูลก่อนการโปรแกรม และ Verify Device
 After Programming เพื่อทำการตรวจสอบความถูกต้องของข้อมูลหลังจากโปรแกรม

JTAG ICE						
Program Fuses LockBits	Advanced Board A	uto				
ATmega64	•	Erase Device				
Programming mode C ISP C Parallel/High Voltage	Serial Verify Dev	rice Before Programming rice After Programming				
Flash	r/Emulator ELASH Memor					
Input HEX File D:V4	VR-STAMP\Examples\A					
Program	Verify	Read				
C Use Current Simulator/Emulator EEPROM Memory C Input HEX File D:VAVR-STAMP_EXAMPLE\BASIC\LED_BLI						
Program	Verify	Read				
Setting device parameters for Entering programming mode Reading fuse bits 0xFF, 0x0F Leaving programming mode (itag programming OK OK FEF OK OK					



5. ในกรณีที่ไม่ต้องการแก้ไขค่า Fuses และ LockBits ก็สามารถกดปุ่ม Program เพื่อโปรแกรม Hex Files ให้กับ MCU ได้เลย เนื่องจากค่า Fuses และค่า LockBits เมื่อโปรแกรมไปครั้งหนึ่งแล้วค่าจะยัง เหมือนเดิม ไม่ถูกลบไปพร้อมกับส่วนของโปรแกรม ถ้าจะแก้ไขค่าก็สามารถโปรแกรมเข้าไปทีหลังโดย ที่ไม่กระทบต่อส่วนของโปรแกรม

JTAG ICE					
Program Fuses LockBits Advanced Board Auto					
Device					
ATmega64 Erase Device					
Programming mode © ISP Image: Constraint of the second					
Flash Use Current Simulator/Emulator FLASH Memory Input HEX File es\ASSEMBLY\LED_BLINK\LED_BLINK.hex					
Program Verify Read					
C Use Current Simulator/Emulator EEPROM Memory Input HEX File D:\AVR-STAMP_EXAMPLE\BASIC\LED_BLI					
Program Verify Read					
Setting device parameters for itag programmingDK Entering programming mode OK Reading fuse bits 0xFF, 0x0FEF OK Leaving programming mode OK					

6. เมื่อการโปรแกรม Hex Files ให้กับ MCU ไม่มีข้อผิดพลาดจะได้ข้อความดังรูป

JTAG ICE							
Program Fuses LockBits Advance	ced Board Auto						
Device							
ATmega64	Erase Device						
Programming mode © ISP © Parallel/High Voltage Serial	 Erase Device Before Programming Verify Device After Programming 						
Flash	Venily Device Arter Hogramming						
C Use Current Simulator/Emulator	Use Current Simulator/Emulator FLASH Memory Consult HEX File es\ASSEMBLYVED_BLINKVED_BLINK hex						
Program	Verify Read						
EEPROM							
O Use Current Simulator/Emulator O Input HEX File D:\AVR-STAI	MP_EXAMPLE\BASIC\LED_BLI						
Program	Verify Read						
Environdenian OK							
Programming FLASH OK Beading FLASH OK							
FLASH contents is equal to file OK Leaving programming mode OK	I						

7. ส่วนถ้าต้องการแก้ไขค่า Fuses และค่า LockBits ก็สามารถเลือกไปที่ Fuses และ Lockbits ทำการ ตั้งค่าและโปรแกรมไปทีหลังได้ตามรูป

JTAG ICE						
JTAG ICE Image: Comparison of the section of the s						
CKOPT fuse (operation dependent of CKSEL fuses); [CKOPT=0] Brown-out detection level at VCC=4.0 V; [BODLEVEL=0] Brown-out detection level at VCC=2.7 V; [BODLEVEL=1] Brown-out detection enabled; [BODEN=0]						
Auto Verify Program Verify Read Smart Warnings						
Setting device parameters for itag programmingOK Entering programming mode OK Reading fuse bits 0xFF, 0x0FEF OK Leaving programming mode OK						

JTAG ICE	×
Program Fuses LockBits Advanced Board Auto	
 Mode 1: No memory lock features enabled Mode 2: Further programming disabled Mode 3: Further programming and verification disabled Application Protection Mode 1: No lock on SPM and LPM in Application Se Application Protection Mode 2: SPM prohibited in Application Section Application Protection Mode 3: LPM and SPM prohibited in Application Section Application Protection Mode 4: LPM prohibited in Application Section Boot Loader Protection Mode 1: No lock on SPM and LPM in Boot Loader Boot Loader Protection Mode 1: No lock on SPM and LPM in Boot Loader Boot Loader Protection Mode 2: SPM prohibited in Boot Loader Section Boot Loader Protection Mode 3: LPM and SPM prohibited in Boot Loader Section Boot Loader Protection Mode 4: LPM prohibited in Boot Loader Section 	
Image: Auto Verify Program ↓ Verify Read Image: Smart Warnings	
Setting device parameters for itag programmingOK Entering programming modeOK Reading lock bits0xFFOK Leaving programming modeOK	

<u>หมายเหตุ</u>

ถ้าในการใช้งานบอร์ดเพื่อทดลองต่างๆ ค่า Lockbits นี้ไม่จำเป็นต้องโปรแกรม ซึ่งค่านี้จะใช้ เพื่อป้องกันการอ่านและเขียนข้อมูลทับ MCU

การใช้งาน ET-AVR JTAG ในโหมดดีบัก

การทำงานในโหมดนี้จะใช้เพื่อดีบัก ดูสถานะการทำงานของ MCU โดยสามารถดีบักทีละ Step หรือแบบอัตโนมัติ โดยระหว่างการดีบักค่าต่าง ๆ ของ MCU จะเปลี่ยนแปลงตามโปรแกรมทำให้ สามารถดูค่าผลลัพธ์ต่างๆ ได้ทันทีซึ่งการดีบักนี้สามารถทำได้ทั้งภาษาแอสเซมบลีและภาษาซี เช่น ถ้า เขียนโปรแกรมไฟวิ่งก็จะเห็นไฟวิ่งตามสถานะการดีบักเป็นต้น ซึ่งขั้นตอนการใช้งานในโหมดดีบักจะ เป็นดังนี้ โดยจะเป็นตัวอย่างไฟกระพริบ 1 ดวง เพื่อให้ง่ายต่อการสังเกต

ตัวอย่างการดีบักโดยใช้ภาษาแอสเซมบลี

 ทำการเปิดโปรแกรม AVR Studio จากนั้นจะปรากฏหน้าต่าง Welcome to AVR Studio ให้คลิกที่ New Project เพื่อสร้างโปรเจกต์ใหม่ดังรูป



2. เลือก Project type เป็น Atmel AVR Assembler เพื่อเขียนโปรแกรมเป็นภาษาแอสเซมบลี ทำการ ตั้งชื่อโปรเจกต์ในช่อง Project name เลือกที่ช่อง Create initial file เพื่อสร้างไฟล์แอสเซมบลีพร้อมกับ สร้างไฟล์โปรเจคต์ เลือกที่ช่อง Create folder เพื่อสร้างโฟลเดอร์สำหรับเก็บไฟล์โปรเจกต์ จากนั้นทำ การเลือกไดเร็กทอรี่ที่จะเก็บไฟล์โปรเจคและคลิกปุ่ม Next ดังรูป

Welcome to AVR Stud	lip 4 Create new project Project type: ▲ Atmel AVR Assembler ▲ AVR GCC	Project name: Blinky I⊄ Create initial file I⊄ Create folder Initial file: Blinky .asm		
Strue	Location:			
Ver 4.12.452	<< Back Next :	>> Finish Cancel Help		



3. เลือก Debug platform เป็น JTAG ICE และ Device เป็น ATmega64 และคลิกปุ่ม Finish ดังรูป

	JTAGICE mkll ICE40 ICE50	AT90CAN128 ATmega128 ATmega16	
nalla	AVH Simulator ICE200	Almega162 ATmega169 ATmega32 ATmega323 ATmega64	
S S		A186HF401 AT89S51 AT89S52	5
	Port: Auto 💌 🔽	Open platform options	

4. จากนั้นจะปรากฏหน้าต่าง Text Editor สำหรับเขียนโปรแกรมดังรูป



5. ทำการพิมพ์โปรแกรมตัวอย่างภาษาแอสเซมบลี ดังตัวอย่าง จากตัวอย่างไม่ได้ใช้โปรแกรมหน่วง เวลา เพื่อว่าที่จะสังเกตเห็นการดีบักอย่างทันทีไม่มีการหน่วงเวลา

;* Examples Program For "ET-AVR STAMP ATmega64" Board * * ;* Target MCU : Atmel ATmega16 * ;* Frequency : X-TAL : 16 MHz ;* Compiler : AVR Studio 4.12 (AVR Assembler 2) * ;* Create By : ADISAK CHOOCHAN (WWW.ETT.CO.TH) * * ;* Last Update : 1/September/2005 ;* Description : Example LED Blink on Portb.0 * ;Connect PB0 to LED1 .include "m64def.inc" ; Define Register .def TEMP = R16 ; Define I/O Port,Pin LED 0 .equ = ; Main Program .CSEG 0 .ORG RJMP RESET ;Reset Handle LDI TEMP,LOW(RAMEND) ;Initial Stack Pointer **RESET:** OUT SPL, TEMP LDI TEMP,HIGH(RAMEND) OUT SPH, TEMP SBI DDRB,LED ;Config Portb.0 as output MAIN: PORTB,LED ;LED Off SBI CBI PORTB,LED ;LED On RJMP MAIN ;Loop

 6. ให้ทำการสั่งแปลโปรแกรมที่เราเขียนขึ้น โดยการคลิกเมาส์ที่เมนูคำสั่ง Build → Build ซึ่งหลังจาก แปลโปรแกรมแล้วได้ผลถูกต้องและไม่เกิดข้อผิดพลาดใด ๆ จะปรากฏข้อความ 0 errors 0 warnings ดังรูป





AVR Studio - [D:\Blinky\Blinky.asm							_ @ 🛛
Eile Project Build Edit View To	ols <u>D</u> ebug <u>W</u> indow	Help					- 8×
🗅 🚅 🖬 🕼 🥌 🌜 🧐	🖻 😤 🖬 🛛 🖊 🦽	2. 2. 1	6 住 岸	▶ = E () + ?}	라 아 *0 🗄 🐮 🚸 松 🛛 🖾 🖻		
Trace Disabled 🚽 🛠 🛠 🕂 🔺		×r≥ ~r≥ RUT	- en es				
Project ×	include "	m64def	inc"				_
Binky Binky Binky.sm Binky.sm Binky.sm Binky.asm Binky.as	:********** : Define R :***********	****** egiste ****** TEMP	****** r ******	** = R16			1
🗄 🔄 Object File	<pre> B</pre>						
	;********** ; Main Pro ;********* .CSEG	***** gram ******	**** **** .ORG	0			
			RJMP	RESET	;Reset Handle		
	RESET:		LDI OUT LDI OUT	TEMP,LOW(RAMEND) SPL,TEMP TEMP,HIGH(RAMEND) SPH,TEMP	;Initial Stack	Pointer	
			SBI	DDRB,LED	;Config Portb.0	as output	-
Project 10 View 10 Info	D·\ Blinky \ Blinky	asm					<u>ت</u> ۱۵
Puld	_ Dripminty (Dimity			1			~
Segment Begin End Coo	ie Data Used	Size	Use%				^
							_
[.cseg] 0x000000 0x000012] [.dseg] 0x000100 0x000100 [.eseg] 0x000000 0x000000 • Assembly complete, 0 errors. 0	18 0 18 0 0 0 0 0 0 warnings	65536 4096 2048	0.0% 0.0% 0.0%				=
Puild American Rind in Silar M.	real-mainte and Tracanai	ake					×
wessage Find in Files big Bi	rearpoints and tracepoi	ics		ATmega64	JTAG ICE COM7	Ln 18. Col 30	CAP NUM SCRI

7. คลิกที่เมนูคำสั่ง Debug → Start Debugging ดังรูป

😻 AVR Studio - [D:\Blinky\Blinky.asm]								
Eile Project Build Edit View Tools	Det	oug <u>W</u> indow <u>H</u> elp		_				
D 🚅 🖶 🗿 🗗 🐰 🖻 🛍 🎒 의 ≌		Start <u>D</u> ebugging	Ctrl+Shift+Alt+F5					
Trace Disabled 🔹 🗞 🛠 🕂 🔽	=	Stop Debugging	Ctrl+Shift+F5					
Project X		Run	F5	*****				
🖃 🐌 Blinky	Ш	Break	Ctrl+F5	-AVR S				
⊡ - 🔄 Source Files	G.	Reset	Shift+F5	ga16 MHz				
	{ }	Step <u>I</u> nto	F11	4.12				
E-Cabels	8	Step <u>O</u> ver	F10	DCHAN				
⊕ - Gupat	{}	Step O <u>u</u> t	Shift+F11	D Blin				
	*{}	Run to <u>C</u> ursor	Ctrl+F10	*****				
	E\$	<u>A</u> uto Step	Alt+F5					
	۵.	<u>N</u> ew Breakpoint	,	·				

 จากนั้นโปรแกรมจะทำการโหลดข้อมูลต่างลง MCU และเข้าสู่การดีบัก โดยจะมีเครื่องหมายแสดง จุดเริ่มต้นของโปรแกรมดังรูป โดยทางด้านขวาจะปรากฏหน้าต่าง I/O View แสดงค่ารีจิสเตอร์ต่าง ๆ ของ MCU

AVR Studio - [D:\Blinky\Blinky.asm] Bile Project Build Edit View Iools Debug Window	Help			_ @ X
□ ☞ 🖬 🕼 ७ % 🖻 🛍 쓸 요 요 😤 🐂 А 🔺	%%%健症 ▶ •	ET II G		66° 🗾 🔤 🔲 👦 💷
Trace Disabled 🔹 🛠 🕀 上 不 🛛 🗰 🗰	YE YE HITO 🌆 🚯			
I/O View ×	;*********	*****		-
Name Value Bits Address	; Main Program			
E	CSEG	*****		
H - En Processor		.ORG	0	
E Stack Monitor	⇒	RJMP	RESET	;Reset Handle
🗄 🎆 I/O ATMEGA64	DEGET	TDT	TEND LOUIDENNEND)	Toiticl Check Deints
	RESEI:	OUT	SPI TEMP	;Initial Stack Pointe
		LDI	TEMP .HIGH (RAMEND)	
		OUT	SPH.TEMP	
EEPROM				
EXTERNAL_INTERRUPT		SBI	DDRB,LED	;Config Portb.U as or
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	MAIN:	SBI	PORTB LED	LED Off
		CBI	PORTB,LED	;LED On
PORTB		RJMP	MAIN	;Loop
🚬 🔁 PORTB 0x00 🗌 🗌 🔤 🔤 0X18 (C				
🕂 DRB 0x00 🔲 🔤 0x17 (C				
🕀 🔁 PORTF				-
E PORTG				Þ
Project View Dinfo	D:\Blinky\Blinky.asm			4 b
- · · ·				×
Duid				^
Tracepoi	nts			
		ATm	ega64 JTAG ICE COM7	Stopped () Ln 37, Col 13 CAP NUM SCRL

 เราสามารถที่จะปรับความเร็วในการดีบักได้ โดยการเลือกที่เมนูคำสั่ง Debug → JTAG ICE
 Options โดยที่สามารถปรับความถี่ของพอร์ต JTAG โดยค่าความถี่นี้ต้องไม่เกิน 1/4 ของความถี่ที่เรา ใช้งาน

JTAG ICE 🛛 🛛 🔀
Connection Debug Breakpoints Status
JTAG Port JTAG port frequency: 4.00 MHz For safe connection between the JTAG ICE and the target device, the JTAG port frequency must be equal to or less than 1/4 of the target device frequency Target device is part of a JTAG daisy chain Devices before: Devices after: Devices
Baud rate: 19200 (default)



10. ทำการเลือกรูปแบบของการดีบักซึ่งสามารถเลือกได้ทั้งแบบอัตโนมัติ (Auto Step)หรือจะเลือกทีละ ขั้นตอน โดยเลือกที่แถบเครื่องมือดังรูป ในที่นี้ทดลองเลือกแบบ Step

11. จะสังเกตเห็นว่าเมื่อโปรแกรมดีบักผ่านคำสั่ง SBI PORTB, LED ซึ่งเป็นคำสั่งให้ PORTB.0 เป็นลอจิก 1 LED ที่อยู่กับ PORTB.0 จะยังไม่สว่างเนื่องจากวงจรของบอร์ดต่อวงจรให้ LED ทำงานที่ ลอจิก 0 ทำการกดปุ่ม Step Info อีกครั้งเมื่อผ่านคำสั่ง CBI PORTB, LED จะสังเกตว่า LED จะ สว่าง ซึ่งค่าต่าง ๆ ในหน้าต่าง I/O View ก็จะเปลี่ยนแปลงตามโปรแกรมด้วย

AVR Studio - [D:\Blinky\Blinky.asm]				
Eile Project Build Edit View Tools Debug Window	Help			_ 8×
D 😅 🖬 🕼 🥑 👗 陷 🛍 🖨 🗠 🗠 😤 🙀 🗛 🦽 🥬	4.3% 住宅 []) = EL II () =	ን 🚸 🕀 🕄 🕄 🕈 🕀 😚	
Trace Disabled 🚽 🗞 🛠 🖯 业 木 🛚 🛲 🗰 🦮	2 🕎 AUTO 🛛 🎬 🎒			
I/O View ×	;********	• * * * * * * * * * * * * * * * * * * *	(*	
Name Value Bits Address	; Define I/	0 Port,Pin	**	
E S I/O AIMEGA64	.equ	LED = 0		
E BOOT_LOAD	: Main Pros	iram		
	********	*****		
E STERNAL_INTERRUPT	.CSEG	0.0.0		
		. URG R.TMP	U RFSFT	·Reset Handle
		110112	10001	, incore inducto
PORTB	RESET:	LDI	TEMP,LOW(RAMEND)	;Initial Stack Point
		LDI	TEMP_HIGH(RAMEND)	
PINB 0xFE 0001 0x17 (0x37)		OUT	SPH,TEMP	
		CDT	DDDD IED	Genfin Denth Cons.
		DI	DDRD,LED	;config Portb.0 as o
	MAIN:	SBI	PORTB, LED	;LED Off
🕀 🔁 PORTG		CBI	PORTB, LED	;LED On
		RJMP	MAIN	;Loop
TIMER_COUNTER_0				- 1
E S TIMER_COUNTER_2				v
Brudet WIGHTER 3				
	D: (biiriky (biiriky.)	dsill		4 P
Build				×
D:\Blinky\Blinky.asm(49): No EEPROM data, deleti:	ng D:\Blinky\Blink	y.eep		
Segment Begin End Code Data Used	Size Use%			
[.cseg] 0x000000 0x000012 18 0 18	65536 0.0%			
[.aseg] 0x000000 0x000000 0 0 0	4095 0.0% 2048 0.0%			
Build Message AFFING in Files Preakpoints and Tracement	5			
		ATmega	a64 JTAG ICE COM7 Stoppe	ed 👄 Lo 44. Col 1 CAP NUM SCRU



ตัวอย่างการดีบักโดยใช้ภาษาซี

นอกจากภาษาแอสเซมบลีแล้วโปรแกรม AVR Studio ยังสามารถดีบักซอฟต์แวร์ที่เขียนด้วย ภาษาซีได้ด้วย ซึ่งตัวอย่างนี้จะเสนอวิธีการดีบักโดยใช้ภาษาซี โดยใช้โปรแกรม CodeVisionAVR C Compiler ร่วมกับโปรแกรม AVRStudio ในการดีบัก

1. เปิดโปรแกรม CodeVisionAVR C Compiler และคลิกเลือกที่เมนูคำสั่ง File → New ดังรูป



2. เลือก File Type เป็น Source เพื่อสร้างไฟล์ภาษาซีใหม่และคลิกปุ่ม OK ดังรูป

🗄 Create New File 🛛 🔀			
File Type			
⊙ Source			
C <u>P</u> roject	X <u>C</u> ancel		



3. จากนั้นจะปรากฏหน้าต่าง Editor ให้ทำการเขียนโปรแกรมดังตัวอย่าง

```
//*Hardware : ET-AVR STAMP (ATmega64)
                                                    */;
//*CPU
               : ATMEL-ATmega64
                                                    */;
//*X-TAL : 16.00 MHz
//*Filename : Main.C
//*Complier : CodeVisionAVR V1.24.7d
                                                    */;
                                                    */;
                                                    */;
//*Last Update : 9-12-2005(ETT CO.,LTD)
                                                    */;
//* : WWW.ETT.CO.TH
                                                    */;
//*Description : Example LED Blink on Portb.0 */;
//*CodeVisionAVR Complier Option Setting
                                                   */;
//* Chip type : ATmega64
//* Program type : Application
//*Clock frequency : 16.000000MHz
//* Memory model : Small
//*External SRAM size : 0
//* Data Stack size : 1024
                                                   */;
                                                   */;
                                                   */;
                                                    */;
                                                    */;
                                                    */;
#include <mega64.h>
                               // ATmega64 MCU
                                // Delay functions
#include <delay.h>
void main(void)
{
  PORTB=0x00;
                               // PB7..0 = 0
                                // PB0 = Output
  DDRB=0x01;
//Loop Blink LED on PB0
  while (1)
  {

      PORTB |= 0x01;
      // PB0 = 1 (OFF LED)

      PORTB &= 0XFE;
      // PB0 = 0 (ON LED)

    PORTB |= 0 \times 01;
  }
}
```



 ทำการบันทึกโปรแกรมภาษาซีที่เขียนโดยเลือกเมนู File → Save ทำการตั้งซื่อไฟล์และกดปุ่ม Save ดังรูป

Save C:\cvavre	val\bin\untitle	d.c As			? 🗙
Save in:	Blinky_C		•	(† 🖻 🖆 🗐	
CO Recent					
Desktop					
My Documents					
My Computer					
S					
My Network Places	File name: Save as type:	Blinky C Compiler sourc	e file (*.c)	•	Save Cancel

5. เลือกที่เมนู File → New และเลือก File Type เป็น Project เพื่อสร้างโปรเจกต์ใหม่และคลิกปุ่ม OK ดังรูป

🗄 Create New File 💦 🔀		
File Type		
C <u>S</u> ource	✓ <u>о</u> к	
• Project	X <u>C</u> ancel	

6. คลิกปุ่ม No เพื่อไม่ใช้ตัวช่วยในการสร้างโปรเจกต์ (CodeWizard)





7. ทำการตั้งชื่อโปรเจกต์ตามต้องการและคลิกปุ่ม Save ดังรูป

Create New Pro	ject				? 🗙
Save in:	🗀 Blinky_C		•	+ 🗈 📸 📰 -	
📁 Recent					
Desktop					
My Documents					
My Computer					
S					
My Network Places	File name:	Blinky		_	Save
	Save as type:	Project files (*.prj)		•	Cancel

8. ทำการเพิ่มไฟล์ภาษาซีที่เขียนไว้ก่อนหน้านี้เข้ามาในโปรเจกต์โดยการคลิกปุ่ม Add ดังรูป

🚯 Configure Project Blinky.prj	
Files C Compiler After Make	
💾 D:\Blinky_C\Blinky.prj	Zi Add
	<u> ti R</u> emove
	TÌ <u>E</u> dit File Name
	☆ Move∐p
	ncel <u>7 H</u> elp



Add File To Pr	oject				? 🔀
Look in:	🗀 Blinky_C		-	🗕 🗈 💣 📰	•
67	🗒 Blinky				
Recent					
Desktop					
My Documents					
My Computer					
					
My Network Places	File name:	Blinky		•	Open
	Files of type:	C Compiler files (*.	c)	•	Cancel

9. เมื่อทุกอย่างเรียบร้อยคลิกปุ่ม OK ดังรูป

🚯 Configure Project Blinky. prj	
Files C Compiler After Make	1
⊡ <mark>1</mark> D:\Blinky_C\Blinky.prj <mark>1</mark> Blinky.c	Add 🚼
	± i <u>R</u> emove
	TI Edit File Name
	☆ Move <u>U</u> p
	✤ Move Down
	el ? <u>H</u> elp



 ทำการตั้งค่าต่างของโปรเจกต์โดยการคลิกเมาส์ที่เมนูคำสั่ง Project → Configure จากนั้น ทำการ กำหนดเบอร์ MCU เป็น ATmega64 ค่าคริสตอลเท่ากับ 16.000000 MHz และ File Output Format(s) เป็น COF ROM HEX EEP

🗰 CodeVisionAVR - Blinky.prj - [D:\Blinky_	_C\Blinky.c]
File Edit View Project Tools Settings Wir	ndows Help
Navigator Code Ten Code VisionAl Code VisionAl Project B Notes Information	M D T T 11 //* CodeV 12 //* Chip 13 //* Progr 14 //* Clock
Pointy Notes Ctrl+N Ary G Ary Ary G Ary Ary Ary Ary Ary Ary Ary Ary Ary	15 //* Memor 16 //* Exter 17 //* Data
Dther Files	<pre>18 //******* 19 20 #include 21 #include</pre>

😽 Configure Project Blinky	.prj	×
Files C Compiler After Make	e	
Code Generation Messages	Globally #define Paths	
Chip: ATmega64 Clock: 16,000000 MHz Memory Model: Small Optimize for: Size Optimization Level: Low Program Type: Application (s)printf Features: int, width (s)scanf Features: int, width	SRAM Data Stack Size: 1024 bytes Heap Size: 0 bytes Internal SRAM size: 4096 bytes External SRAM size: 0 bytes External SRAM Wait State Code Generation Bit Variables size: 16 ▼ Promote char to int ♥ char is unsigned 8 bit enums♥ Enhanced Core Instructions ♥ Automatic Register Allocation ₩ Word Align FLASH Struct Members Use an External Startup Initialization File Stack End Markers File Output Format(s): COF ROM HEX EEP ▼	
[V OK Cancel ? Help	

 ให้ทำการสั่งแปลโปรแกรมที่เราเขียนขึ้น โดยการคลิกเมาส์ที่เมนูคำสั่ง Project → Make ซึ่ง หลังจากแปลโปรแกรมแล้วได้ผลถูกต้องและไม่เกิดข้อผิดพลาดใด ๆ จะปรากฏข้อความ No errors, No warnings ดังรูป

Information
Compiler Assembler
Chip: ATmega64 Program type: Application Memory model: Small Optimize for: Size (s)printf features: int, width (s)scanf features: int, width Promote char to int: No char is unsigned: Yes 8 bit enums: No Enhanced core instructions: On Automatic register allocation: On
209 line(s) compiled No errors No warnings
Bit variables size: 0 byte(s) Data Stack area: 100h to 4FFh Data Stack size: 1024 byte(s) Estimated Data Stack usage: 0 byte(s) Global variables size: 0 byte(s)
Hardware Stack area: 500h to 10FFh Hardware Stack size: 3072 byte(s)
Heap size: 0 byte(s)
EEPROM usage: 0 byte(s) (0.0% of EEPROM) Program size: 129 words (0.4% of FLASH)



คลิกเมาส์ที่เมนูคำสั่ง Tools → Debugger เพื่อเข้าสู่การดีบัก จากนั้นโปรแกรม CodeVisionAVR
 จะเปิดโปรแกรม AVR Studio ขึ้นมาดังรูป ให้คลิกที่ปุ่ม Open

👫 CodeVisionAVR - Blink	y.prj	D:\Blin	ky_C\BI	inky.c]	
File Edit View Project	Tools	Settings	Windows	Help	
🏗 🖸 🕒 🔙 🗗 🖍	🌼 Co	ideWizardA	VR	Shift+F2	2
Navigator Code Templates (厳 De	bugger	- b	Shift+F3	de∖
	🥸 Ch	iip Program	mer	Shift+F4	lip.
Project: Blinky	💻 Те	rminal		Shift+F5	ogr
💦 Notes	📣 Co	nfigure			.ock
	-	-	15	77 ~ 196	emor
E Included Fil	es - Li		16	//* E3	kter
Eurotions	apies		17	//* Da	ata
			18	//****	****
			19		
			20	#inclu	ıde
			21	#inclu	ıde

Welcome to AVR Stud	io 4	
1.5	New Project Open	§
	Recent projects	Modified
	0 D:\Test_ICC\test.	19-Dec-2005 10:39:05
	🧕 D:\Test_Code vision\Blinky.	23-Sep-2005 10:56:26
	0 D:\Test_Code vision\Blinky.	19-Dec-2005 10:37:51
	D:\Blinky\Blinky	19-Dec-2005 10:11:16
	D:\AVR-STAMP\Examples\ASSEMBLY\LED_BLINK\	17-Dec-2005 14:06:42
	D:\AVR-STAMP\\ASSEMBLY\LED_MOVING\LED.	17-Dec-2005 14:05:19
		15-Dec-2005 12:28:08
	D:VAVR-STAMPVV4HL595_UUT_75EG.	28-Nov-2005 16:18:50
	D:VAV R-STAMPY	28-NOV-2005 16:18:38
	▼ D. WYR-STAMENN2C_NTC_PCF0003N2C_NTC.	201007-2003 16:16:20
	,	
Ver / 12 / 52 🔽 Show d	ialog at startup	
V 61 4.12.402 14 0110W 0		
	<< Back Next>> Finish	Cancel Help



12. เลือกไฟล์ COFF symbolic debug file ที่ได้จากการการสั่งแปลโปรแกรมซึ่งอยู่ในโปรเจกต์ที่ได้ สร้างไว้ดังรูป

Open SaveFile	or ObjectFile	? 🗙
Look in:	🔁 Blinky_C 💌 🗢 🖆 🖽 🗸	
Recent Desktop My Documents My Computer	BI BI Date Modified: 19/12/2548 12:57 Size: 722 bytes	
My Network Places	File name: Blinky Files of type: Project Files, Object Files (".aps,".hex;".d90;".at	Open Cancel

13. จากนั้นโปรแกรมจะให้ทำการบันทึกโปรเจกต์ ให้คลิกปุ่ม Save เพื่อบันทึก

Save AVR Studi	o Project File			? 🛛
Save in:	🗀 Blinky_C	•	· 🗕 📥 🖶	
Recent Desktop My Documents My Computer	 Blinky Blinky Blinky Blinky Blinky Blinky Blinky Blinky Blinky Blinky.pr~ Blinky.vec Blinky_ 			
Mu Network	File name:	Blinky, cof	•	Save .
Places	Save as type:	AVR Studio Project Files *.aps	•	Cancel
AVR Studio will cr file should be loca save the project fi	eate a project for de ted at the root of the le at the same place	bugging this object file. To ensure (e original source file project. If the or e as the object file	optimal debugging, AVR riginal source is not avail	Studio's project able, you can



14. เลือก Debug platform เป็น JTAG ICE และ Device เป็น ATmega64 และคลิกปุ่ม Finish ดังรูป

Welcome to AVR Stud	io 4 Select debug platform and device Debug platform: JTAGICE mkII ICE 40 ICE 50 JTAG ICE AVR Simulator ICE 200	Device: ATmega128 ATmega16 ATmega162 ATmega169 ATmega32 ATmega323 ATmega64 ATmega64 ATmega64 ATmega51	
48	Port: Auto 🔽 🗖 Open platfor	AT89552 AT90CAN32 m options	~
Ver 4.12.452	<< Back Next>>	Finish Cancel	Help

15. เมื่อทุกอย่างเรียบร้อยโปรแกรม AVR Studio จะทำการโหลดโปรแกรมที่เขียนด้วยภาษาซีเข้ามา หลังจากนี้ก็สามารถทำการดีบักค่าต่าง ๆ เหมือนตัวอย่างการดีบักด้วยภาษาแอสเซมบลี





ขั้นตอนการอัพเกรด Firmware ของ ET-JTAG AVR

การอัพเกรด Firmware จะทำให้ ET-JTAG AVR สามารถใช้ได้กับ MCU เบอร์ใหม่ ๆ โดย Firmware นี้จะมาพร้อมกับโปรแกรม AVR Studio ซึ่งเมื่อทำการเชื่อมต่อ ET-JTAG AVR เข้าโปรแกรม AVR Studio แล้วถ้ามี Firmware ตัวใหม่จะมีข้อความเตือนให้ทำการอัพเกรด Firmware ซึ่งวิธีการ อัพเกรด Firmware มีดังนี้

เมื่อทำการติดต่อกับ ET-JTAG AVR และมี Firmware ตัวใหม่จะมีข้อเตือนดังรูป ให้คลิกปุ่ม OK เพื่อทำการอัพเกรด Firmware

Stk500D	n 🛛 🔀
1	AVR Studio has detected a JTAG ICE with an old firmware version. The firmware needs to be upgraded to the latest version. By clicking OK, AvrProg will be launced with the proper upgrade file path selected. Click "Program FLASH" to start upgrading the firmware. After upgrade is complete, press EXIT in the AvrProg dialog. Then close AvrProg and reopen the STK500/AVRISP/AVR JTAG dialog. Select CANCEL to abort upgrade. You will not be able to continue without a firmware upgrade.

 จากนั้นโปรแกรมจะทำการเรียกโปรแกรม AVR Prog ขึ้นมาให้ทำการคลิกที่ปุ่ม Browse เพื่อเลือก ไฟล์ Upgrade.ebn ซึ่งปกติจะอยู่ที่ไดเร็กทอรี่ C:\Program Files\Atmel\AVR Tools\JTAGICE (โดย ปกติโปรแกรมจะเลือกให้อัตโนมัติแล้ว) จากนั้นคลิกที่ปุ่ม Program เพื่อเริ่มทำการอัพเกรดดังรูป

AVR AVRprog	
Hex file C:\Program Files\\Upgrade.ebn	
Browse	Exit
Flash Program Verify	Read
EEPROM	
Program Verify	Read
ATmega16 BOOT	Advanced

AVR AVRprog	
Hex file C:\Program Files\\Upgrade.ebn Browse	Exit
Erasing DeviceProgramming	
Flash Program Verify	Read
EEPROM Program Verify	Read
AT mega16 BOOT	Advanced

ปิดโปรแกรม AVR Prog เมื่อการโปรแกรมเรียบร้อย จากนั้นปลดแหล่งจ่ายไฟที่เลี้ยง ET-JTAG
 AVR ซึ่งตอนนี้ Firmware ตัวใหม่ก็ได้อัพเกรดเรียบร้อยแล้ว

AXVR AVRprog	
Hex file C:\Program Files\\Upgrade.ebn	
Browse	E xit
Erasing DeviceProgrammingVerify	yingOK
Flash Program Verify EEPBOM	Read
Program Verify	Read
Device	Advanced



<u>หมายเหตุ</u> ในกรณีที่ไม่สามารถอัพเกรด Firmware แบบอัตโนมัติได้ ซึ่งส่วนมากจะมีปัญหากับชุด แปลงสัญญาณ USB TO SERIAL ก็สามารถอัพเกรดได้อีกวิธีหนึ่ง โดยต้องใช้ร่วมกับ ET-JTAG UPGRADE ADAPTER และ ET-ISP AVR ซึ่งมีวิธีการดังนี้

1. ต่อ ET-JTAG AVR เข้ากับ ET-JTAG UPGRADE ADAPTER และ ET-ISP AVR โดยที่ต่อ ET-ISP AVR เข้ากับพอร์ตขนานของคอมพิวเตอร์ โดยใช้ ET-CAP10P ดังรูป ด้วยเหตุที่ ET-JTAG AVR ต้องการไฟเลี้ยงจากภายนอก ดังนั้นจึงต้องรับไฟเลี้ยงผ่านทาง ET-JTAG UPGRADE ADAPTER โดย ที่ไฟเลี้ยง 5V นี้สามารถต่อมาจากบอร์ด ET-AVR START KIT V1 เนื่องจาก ได้เตรียมขั้วต่อไว้แล้ว





2. เปิดโปรแกรม PonyProg2000 เลือกกำหนดเบอร์ CPU จาก Device → AVR Micro →เป็น

ATmega16

PonyProg20	00 - Serial Devi	ce Programmer				
File Edit Device	Command Script	: Utility Setup ?	Window			
1	😹 🔒 😫		ड ि 🔊 🕈	AVR micro	▼ ATmega16	•
60 69 60	2 P3 P3	28	🗾			
🏂 No Name						E^
						=
						~
<						>
PonyProg2000	ATmega16	Size 16896 Bytes	CRC 0000h			

3. เลือกที่เมนู Command → Security and Configuration Bits ทำการเลือก Fuse Bit ดังรูป จากนั้น คลิกปุ่ม Write

Configuration and Security bits
7 6 BootLock12 BootLock11 BootLock02 BootLock01 Lock2 Lock1
🔽 OCDEN 🗹 JTAGEN 🕼 SPIEN 🔲 CKOPT 🗐 EESAVE 🕼 BOOTSZ1 🕼 BOOTSZ0 🕼 BOOTRST
🗆 BODLEVEL 🗖 BODEN 🔽 SUT1 🔽 SUT0 🧮 CKSEL3 🧮 CKSEL2 🧮 CKSEL1 🔲 CKSEL0
Checked items means programmed (bit = 0)
Hefer to device datasheet, please
Cancel OK Clear All Set All Write Read



4. ปลดแหล่งจ่ายไฟ 5V ออกจากบอร์ด ET-JTAG UPGRADE ADAPTER และถอด ET-ISP AVR ออกจาก ET-JTAG UPGRADE ADAPTER ทำการเชื่อมต่อ ET-JTAG AVR เข้ากับคอมพิวเตอร์ทาง พอร์ต RS232 และก็ต่อไฟเลี้ยง 5V กลับไปตามเดิม ดังรูป



5. เปิดโปรแกรม AVR Studio 4 เลือกที่เมนู Tools → AVR Prog... จากนั้นให้ทำการคลิกที่ปุ่ม
 Browse เพื่อเลือกไฟล์ Upgrade.ebn ซึ่งปกติจะอยู่ที่ไดเร็กทอรี่ C:\Program Files\Atmel\AVR
 Tools\JTAGICE (โดยปกติโปรแกรมจะเลือกให้อัตโนมัติแล้ว) จากนั้นคลิกที่ปุ่ม Program เพื่อเริ่มทำ
 การอัพเกรดดังรูป

AVR HHIII: AVRprog	
Hex file	
C:\Program Files\\Upgrade.ebn	
Browse	E sit
- Flash	
Program Verify	Read
EEPROM	
Program Verify	Read
Device	
ATmega16 BOOT 🗾	Advanced

AXX HAIII: AVRprog	
⊢ Hex file	
C:\Program Files\\Upgrade.ebn	
Browse	Exit
Erasing DeviceProgramming	
Flash	
Program Verify	Read
EEPROM-	
Program Verify	Read
Device	
ATmega16 BOOT	Advanced

6. ปิดโปรแกรม AVR Prog เมื่อการโปรแกรมเรียบร้อย จากนั้นปลดแหล่งจ่ายไฟ 5V ออกจากบอร์ด ET-JTAG UPGRADE ADAPTER

AVR AVRprog		
Hex file C:\Program Files\	\Upgrade.ebn	
Browse		Exit
Erasing DevicePro	ogrammingVerify	vingOK
Flash Program	Verify	Read
EEPROM Program	Verify	Read
Device ATmega16 BOO1	Ţ	Advanced



7. ทำการต่อวงจรเหมือนข้อที่ 1 โดยถอด ET-JTAG AVR ออกจากคอมพิวเตอร์ และต่อไฟเลี้ยง 5V เข้าที่บอร์ด ET-JTAG UPGRADE ADAPTER



8. เปิดโปรแกรม PonyProg2000 และทำการเลือก Fuse Bit ดังรูปจากนั้นคลิกปุ่ม Write ซึ่งตอนนี้ Firmware ตัวใหม่ก็ได้อัพเกรดเรียบร้อยแล้ว

Configuration and Security bits			
7 🗖 6 🗖 BootLock12 🗖 BootLock11 🗖 BootLock02 🗖 BootLock01 🗖 Lock2 🗖 Lock1			
🔽 OCDEN 🗹 JTAGEN 🖉 SPIEN 🧮 CKOPT 🗐 EESAVE 🔽 BOOTSZ1 🔽 BOOTSZ0 🗐 BOOTRST			
🗆 BODLEVEL 🗆 BODEN 🔽 SUT1 🔽 SUT0 🗖 CKSEL3 🗐 CKSEL2 🗐 CKSEL1 🗐 CKSEL0			
Checked items means programmed (bit = 0)			
Refer to device datasheet, please			
Cancel OK Clear All Set All Write Read			