

ตัวอย่างการพัฒนาโปรแกรม ET-AVR STAMP ATmega64 ด้วย CodeVisionAVR

สำหรับโปรแกรมที่จะใช้ในการเขียนโปรแกรมภาษาซี ก็คือโปรแกรม CodeVisionAVR ซึ่ง ซอฟต์แวร์ตัวนี้ทางบริษัท HP InfoTech มีเวอร์ชั่นทดลองใช้ซึ่งผู้ทดลองสามารถเข้าไปดาวน์โหลด โปรแกรมได้ที่เว็บไซต์ <u>www.hpinfotech.com</u> แต่อย่างไรก็ตามทางทีมงานได้รวบรวมโปรแกรมนี้ไว้ใน แผ่น CD-ROM แล้ว ซึ่งเป็นโปรแกรม CodeVisionAVR เวอร์ชั่น 1.24.7e โดยข้อจำกัดของเวอร์ชั่นทดลอง คือสามารถคอมไพล์ Source code ได้ไม่เกิน **2kbytes** ซึ่งขั้นตอนการใช้งานโปรแกรมมีดังนี้

1. เปิดโปรแกรม CodeVisionAVR C Compiler และคลิกเลือกที่เมนูคำสั่ง File → New ดังรูป



2. เลือก File Type เป็น Source เพื่อสร้างไฟล์ภาษาซีใหม่และคลิกปุ่ม OK ดังรูป

🗳 Create Ne	ew File 🛛 🔀
File Type	
⊙ Source	
○ <u>P</u> roject	X <u>C</u> ancel



3. จากนั้นจะปรากฏหน้าต่าง Editor ให้ทำการเขียนโปรแกรมดังตัวอย่าง ซึ่งเป็นตัวอย่างโปรแกรมไฟ กระพริบที่ PORTB.0

```
//*Hardware : ET-AVR STAMP (ATmega64)
                                       */;
//*CPU
           : ATMEL-ATmega64
                                       */;
//*X-TAL : 16.00 MHz
                                       */;
//*Filename
           : Main.C
                                       */;
//*Complier : CodeVisionAVR V1.24.7e
                                       */;
//*Last Update : 9-02-2006(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 : 16.00000MHz
//* Memory model : Small
                                       */;
                                       */;
                                       */;
                                        */;
//*External SRAM size : 0
//*Data Stack size : 1024
                                        */;
                                        */;
#include <mega64.h>
                       // ATmega64 MCU
#include <delay.h>
                        // Delay functions
void main(void)
{
 PORTB=0x00;
                        // PB7..0 = 0
                        // PB0 = Output
 DDRB=0x01;
//Loop Blink LED on PB0
 while (1)
 {
   delay_ms(200);
                        // Display LED Delay
 }
}
```



 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:	Blinky		▼	Save
	Save as type:	C Compiler source f	ile (*.c)	•	Cancel

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

🕂 Create Ne	w File 🛛 🔀
File Type	
C <u>S</u> ource	🗸 <u>о</u> к
• Project	X Cancel

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)		<u> </u>	Cancel N

8. ทำการเพิ่มไฟล์ภาษาซีที่เขียนไว้ก่อนหน้านี้เข้ามาในโปรเจกต์โดยการคลิกปุ่ม Add ดังรูป

🚯 Configure Project Blinky. prj	
Files C Compiler After Make	
🛱 D:\Blinky_C\Blinky.prj	Zi Add
	±i <u>R</u> emove
	TÌ Edit File Name
	슈 Move <u>U</u> p
	✤ Move Down
<u> </u>	
	el ? <u>H</u> elp



Add F	File To Pro	oject				? 🗙
	Look in:	🗀 Blinky_C		•	+ 🗈 💣 🖩	
	<u>e</u> [🗒 Blinky				
F	Recent					
	B					
D	esktop					
1						
Му С	Documents					
Į						
МуТ	Computer					
	i					
My F	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	
D:\Blinky.c\Blinky.prj Blinky.c	★ Add ★ Bemove ★ Bemove ↓ Edit File Name ↓ Move Up ↓ Move Down
	я <u>7 Н</u> ер



 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:\Blink	y_C\Blinky.c]
💾 File Edit View	Project Tools Settings	Windows Help
<u>e tee</u>	Check Syntax	M 触 🖻 🔧 🔀
Navigator Code Ten	Lompile F9 Make Shift⊥E9	11 //* CodeV
🕞 🕵 CodeVisionA	Stop Compilation	12 //* Chip
Project: B	1 Information	13 //* Plogi 14 //* Clock
Blinky	🕅 Notes Ctrl+N	15 //* Memor
ternet in		- 16 //* Exter
	ancions	17 //* Data
💾 🛄 Other File	s	18 //*******
		19 20 #inglude
		21 #include

🚮 Configure Project Blinky. prj	×
Files C Compiler After Make	
Code Generation Messages Globally #define Paths	
Chip: ATmega64 Chip: ATmega64 Clock: 16,000000 MHz Data Stack Size: 1024 bytes Heap Size: 0 Memory Model: Internal SRAM size: Small Internal SRAM size: Optimize for: External SRAM wait State Size Internal SRAM wait State Optimization Level: Code Generation Low Internal SRAM wait State Program Type: Promote char to int Application Internal Statup Initialization File int, width Istack End Markers File Output Format(s): COF ROM HEX EEP	
V OK Zancel ? Hel	р

 ให้ทำการสั่งแปลโปรแกรมที่เราเขียนขึ้น โดยการคลิกเมาส์ที่เมนูคำสั่ง Project → Make ซึ่ง หลังจากแปลโปรแกรมแล้วได้ผลถูกต้องและไม่เกิดข้อผิดพลาดใด ๆ จะปรากฏข้อความ No errors, No warnings ต่อจากนี้ผู้ใช้ก็สามารถนำ Hex File ที่ได้จากสั่งแปลโปรแกรมนี้ไปทำการ Download ลง MCU ได้ทันที

1 Information
Compiler Assembler
Chip: AT mega64 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)
<u> </u>