
การทดลองที่ 2 การใช้งาน 7-segment, Interrupts และ Timers

วัตถุประสงค์

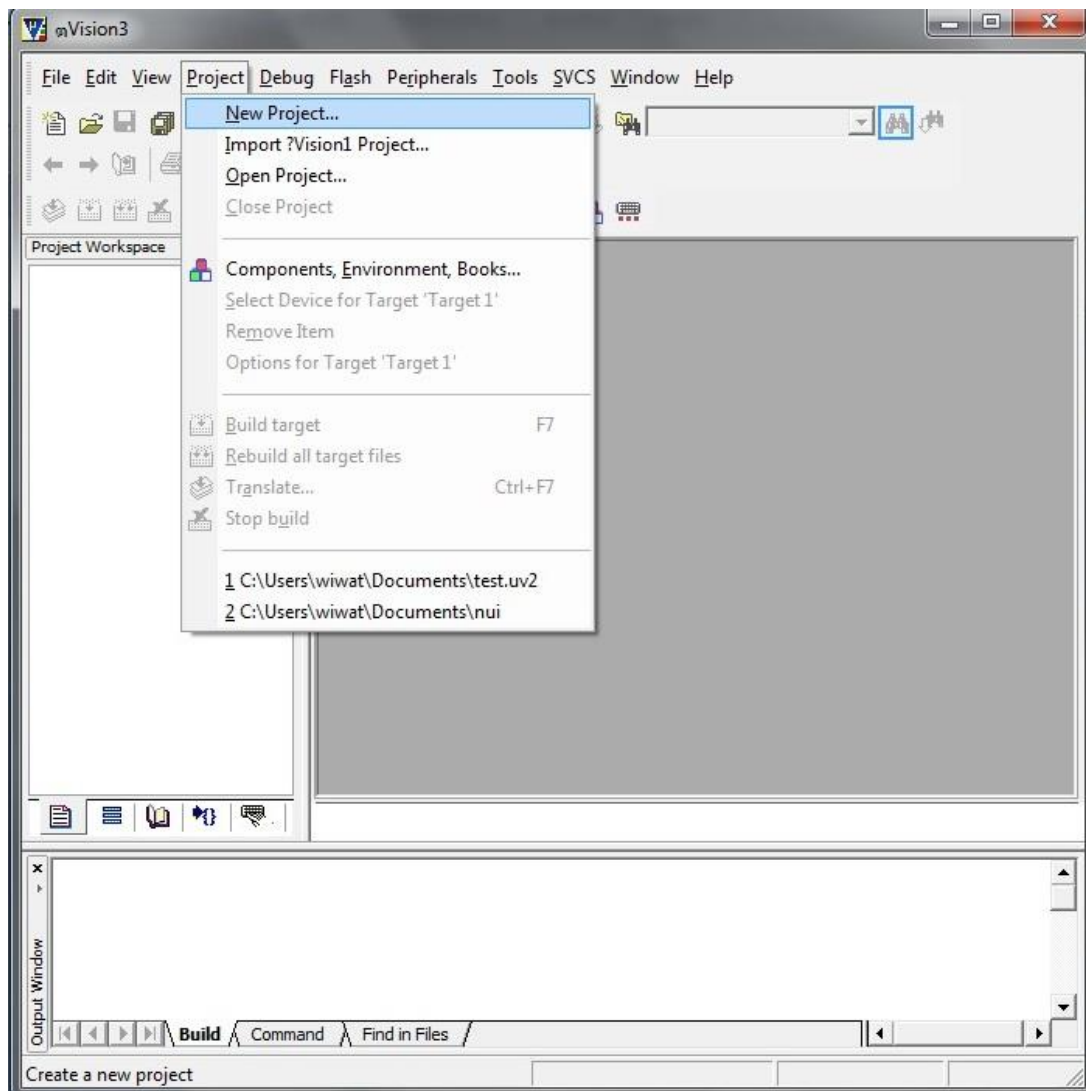
1. เพื่อให้นิสิตสามารถเขียนโปรแกรมภาษา C อย่างง่ายในการควบคุมไมโครคอนโทรลเลอร์ได้
2. เพื่อให้นิสิตเข้าใจการต่อใช้งาน 7-segment
3. เพื่อให้นิสิตเข้าใจการทำงานของ Interrupts และ Timers

อุปกรณ์ในการทดลอง

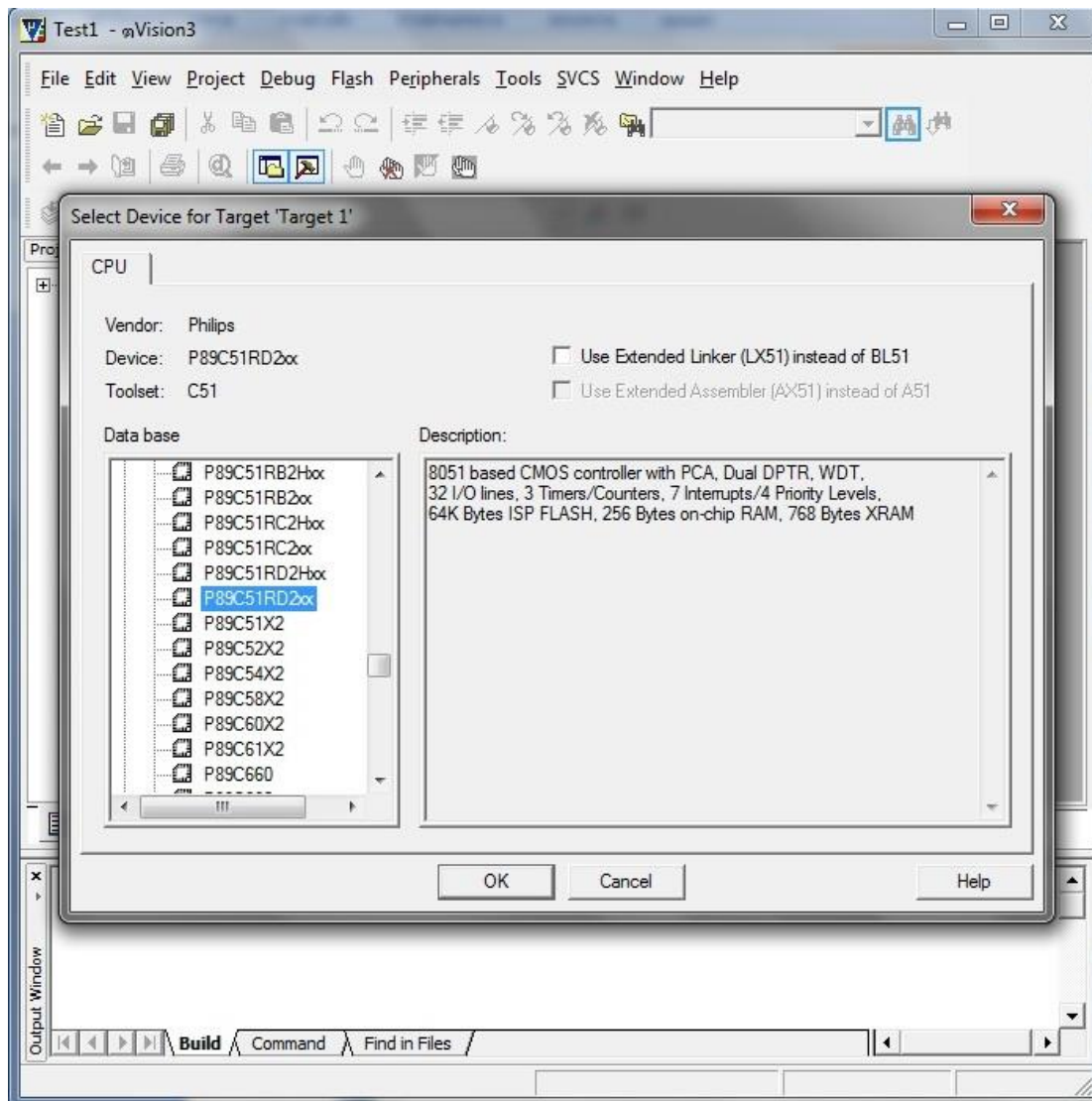
1. เครื่องคอมพิวเตอร์ PC พร้อมโปรแกรมสำหรับการเขียนและคอมไพล์ภาษา C - โปรแกรม Keil51 v.xx
2. บอร์ดไมโครคอนโทรลเลอร์ MCS-51
3. สายต่อพอร์ตอนุกรม
4. บอร์ด 7- segment display

วิธีการทดลอง

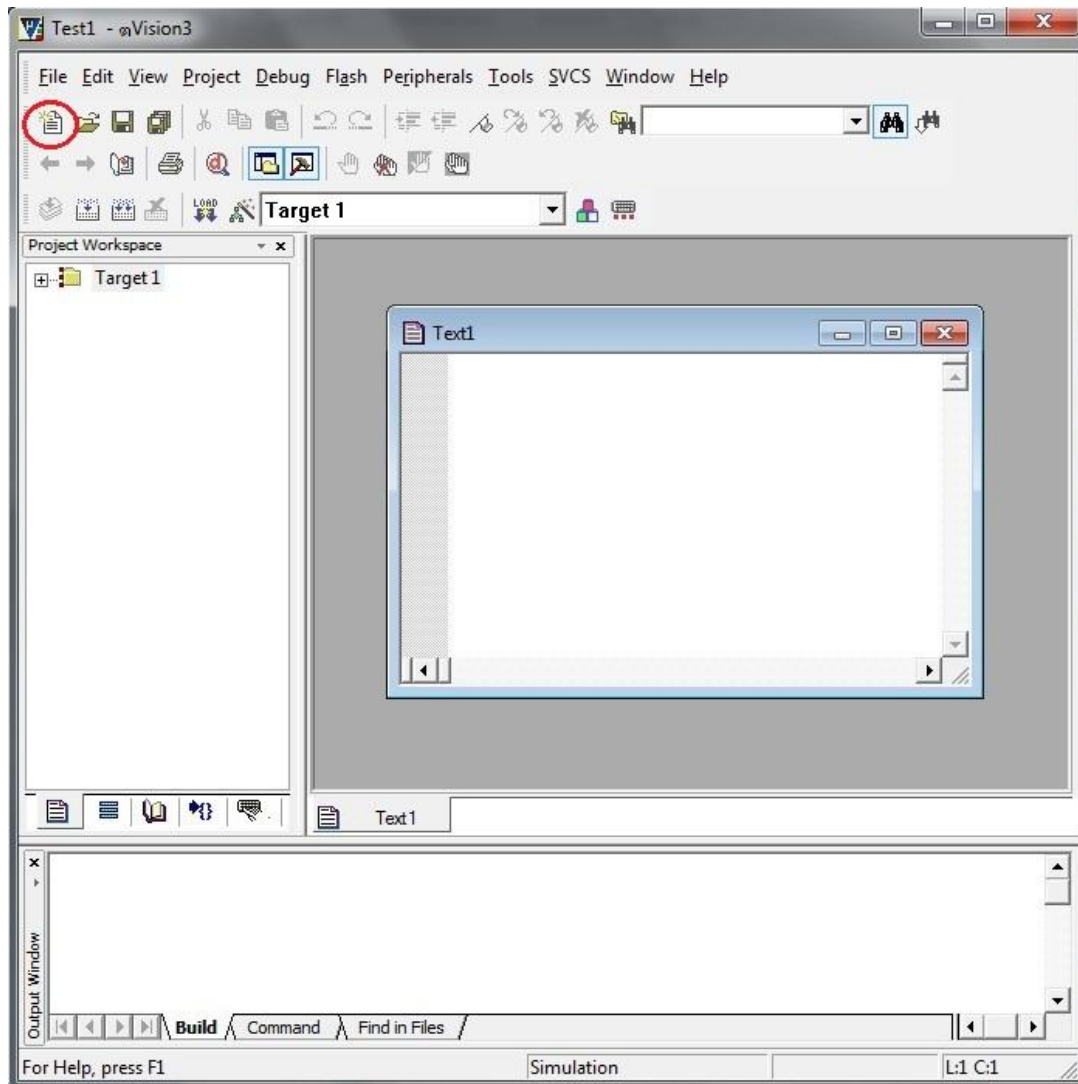
1. ใช้ IDE สำหรับพัฒนาชุดคำสั่งของ MCS-51 ด้วยภาษา C โดยใช้ Keil51 เพื่อเขียนโปรแกรม Lab02_x.c
2. สร้างโปรเจกต์ใหม่โดยเลือก New project ตั้งชื่อเป็น Lab02 แล้วกด ok



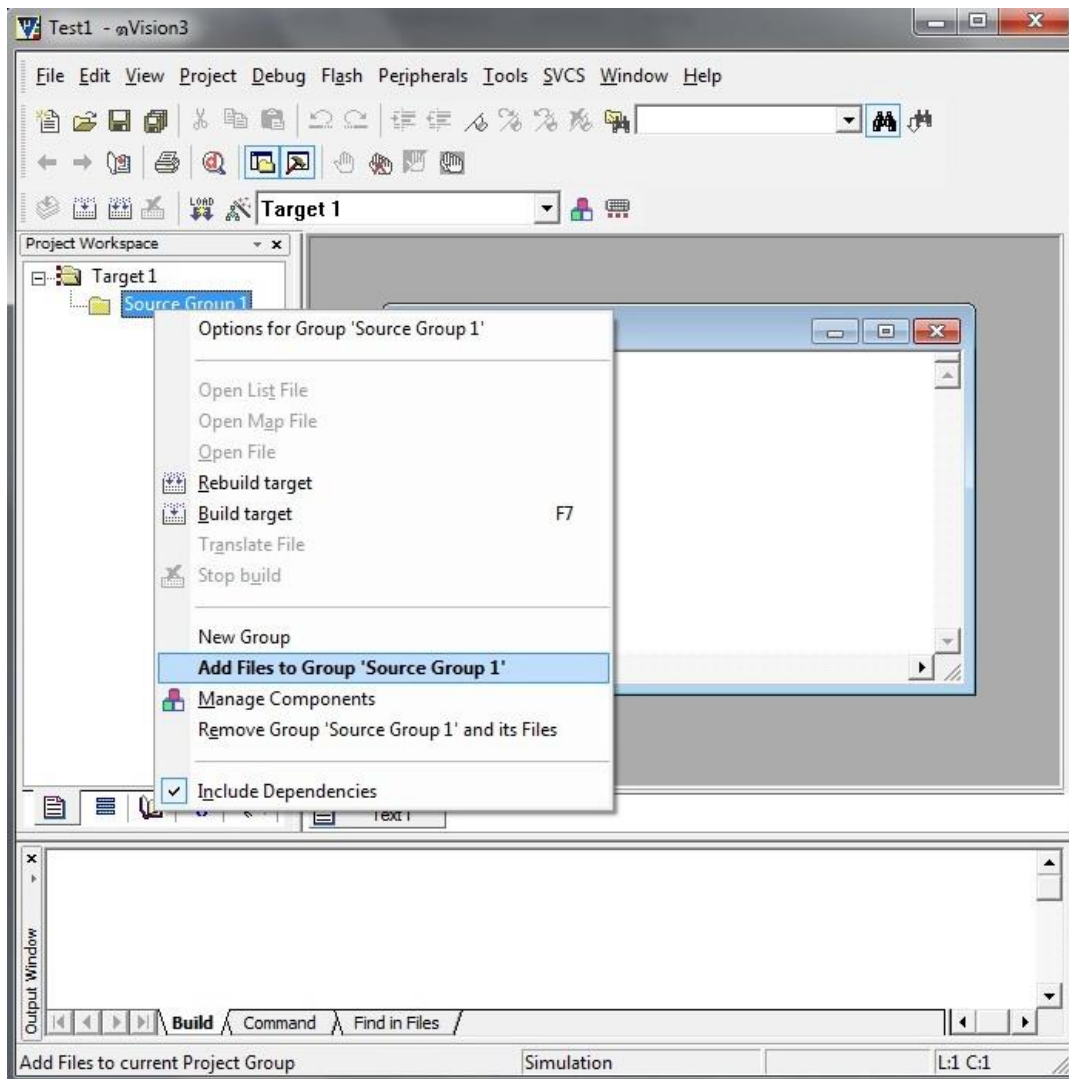
3. เลือก CPU : Philips P89C51RD2xx



4. จากนั้นคลิกที่ปุ่มวงกลมสีแดง เพื่อสร้างหน้าต่างสำหรับเขียนโปรแกรม(หน้าต่าง Text1)
สร้างไฟล์ใหม่ เขียนโปรแกรมและ save โดยใช้ชื่อ Lab02_x.c (x แทนหมายเลขการทดลอง)



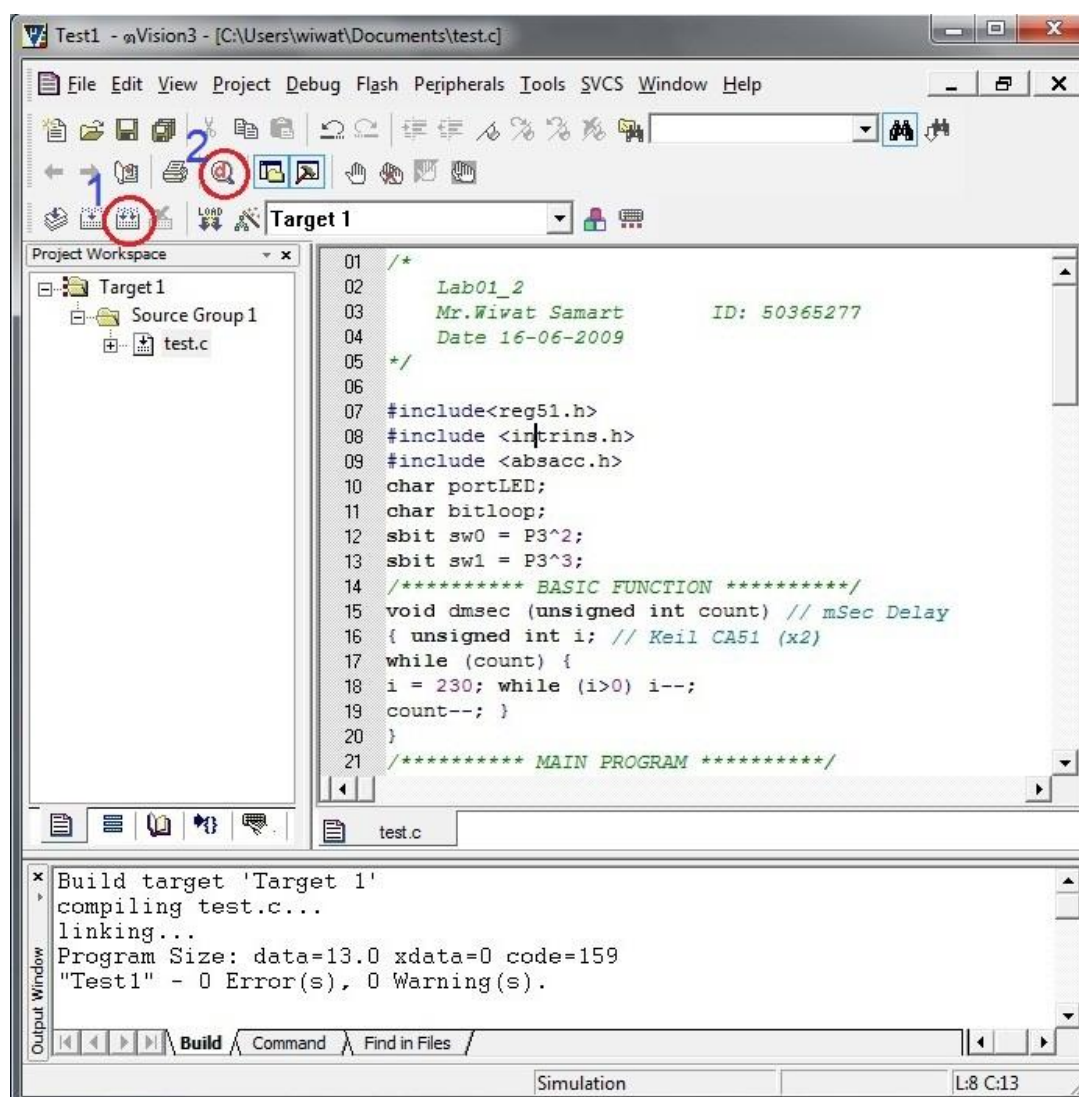
5. คลิกขวาที่ Source Group 1 แล้วเลือก Add file to group 'Source Group 1' แล้วเลือก File Lab02_x.c



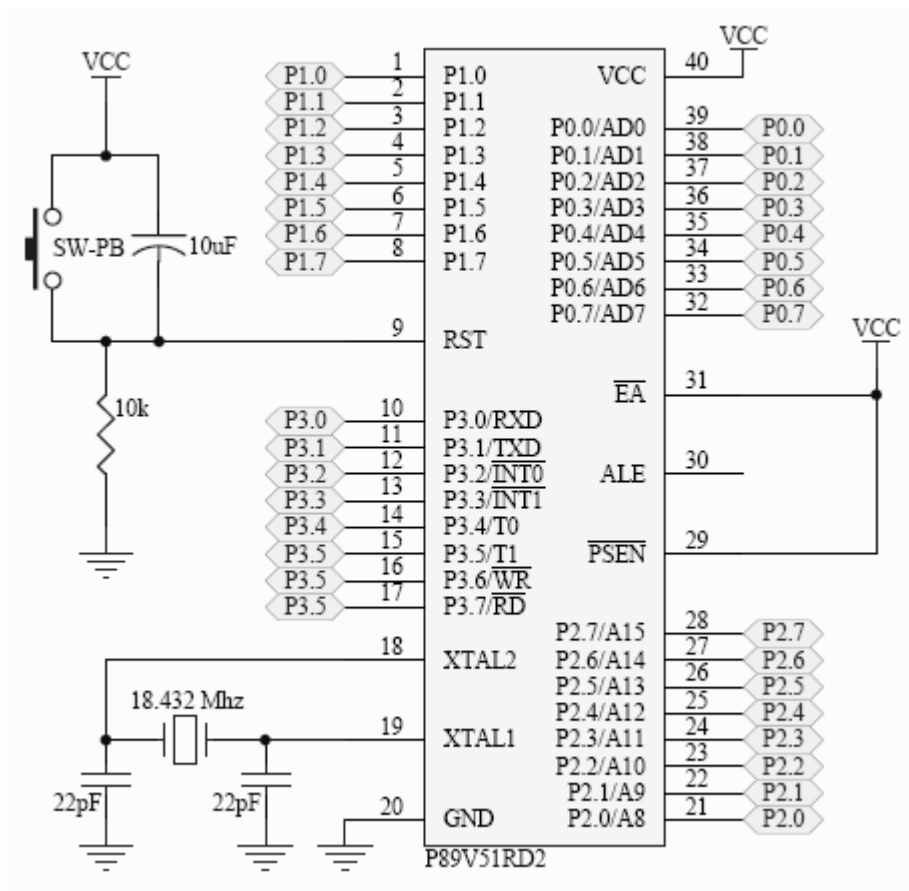
6. เขียนโปรแกรมให้เสร็จแล้วคลิกที่ วงกลมที่ 1 เพื่อทำการ compile ถ้าหากว่าไม่พบ Error โดยที่ถ้าเกิด error ขึ้น โปรแกรมจะแสดงว่า error ก็ที่ (ดู windows output ด้านล่างของโปรแกรม) คลิกที่ วงกลมที่ 2 เพื่อให้ keil ตรวจสอบ และทำการ debug พร้อมทั้งตรวจสอบขนาดของโปรแกรมที่เราสร้างขึ้นและพร้อมสำหรับการทดสอบการทำงานของโปรแกรม

7. Build โปรแกรมที่เขียน โดยก่อนหน้านั้น click ขวา ที่ Target1 เลือก Options for Target 'Target1' ที่ Tab Output เลือก create hex file โดยเลือกที่ check box 'Create Hex file'

File ที่ได้จะมีชื่อเหมือนกับชื่อโปรเจ็ค คือ Lab02.hex สามารถเปลี่ยนแปลงชื่อได้โดยแก้ไข output file ลงในช่อง 'Name of Executable'

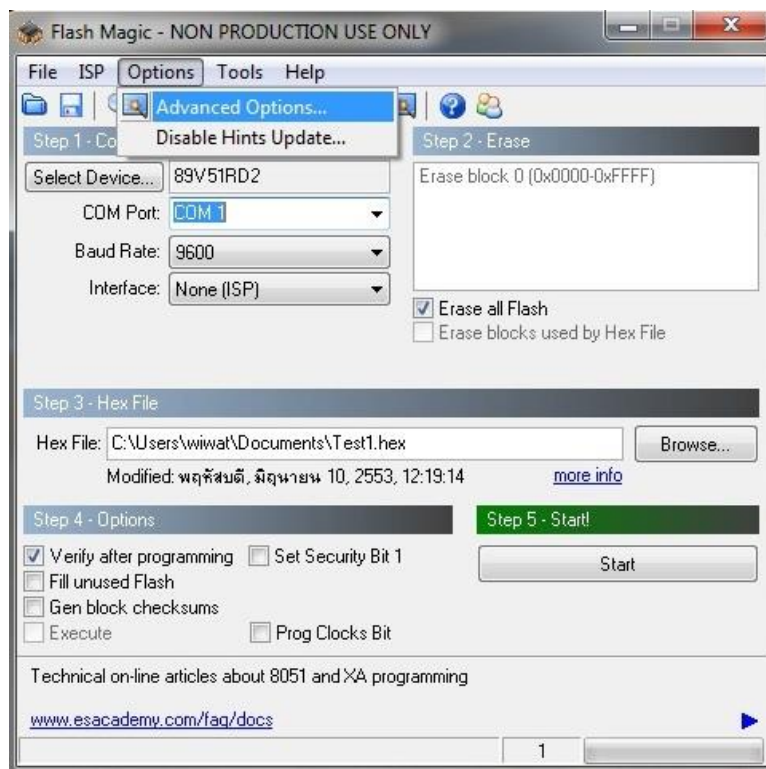


18. ทำการทดลองโดยโปรแกรมลงบนบอร์ด MCS-51 โดยใช้บอร์ดไมโครคอนโทรลเลอร์ซึ่งมีลักษณะวงจรดังนี้

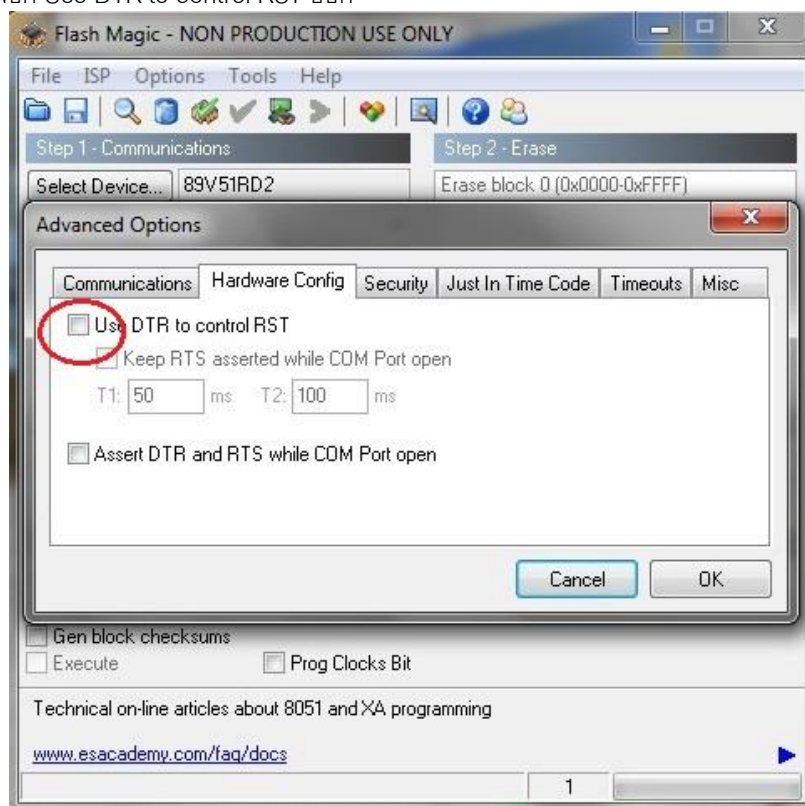


- a. ต่อสายจาก com1 ของ computer เข้ากับบอร์ดโดยใช้สายที่แจกให้เท่านั้น ทำการโหลดผ่านโปรแกรม Flash magic ดังนี้

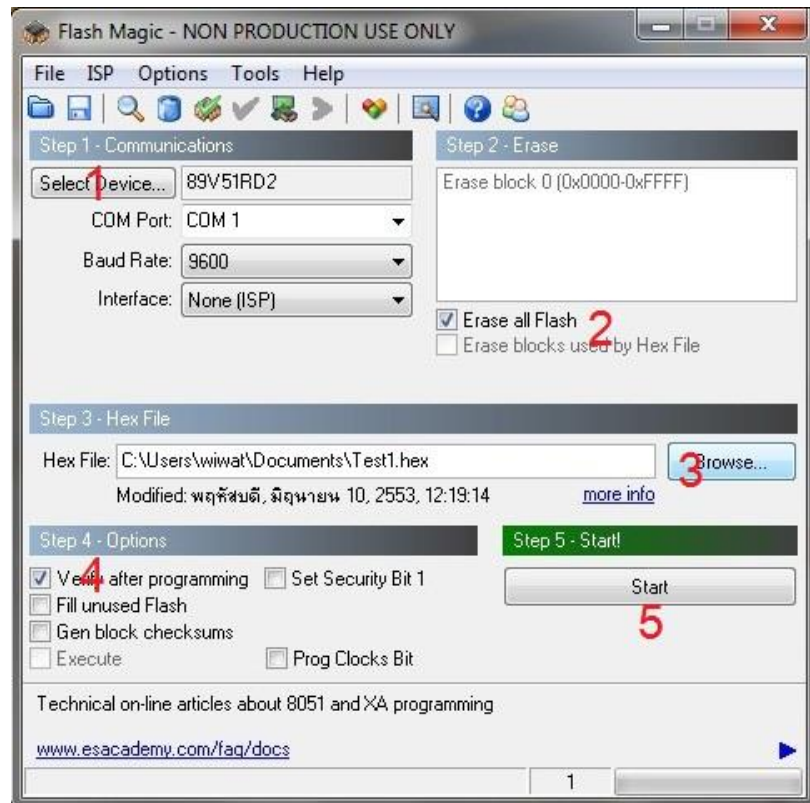
- b. เปิดโปรแกรม Flash magic โดยทำการตั้งค่าต่างๆ ดังนี้ เลือกที่ Options----> Advanced Options



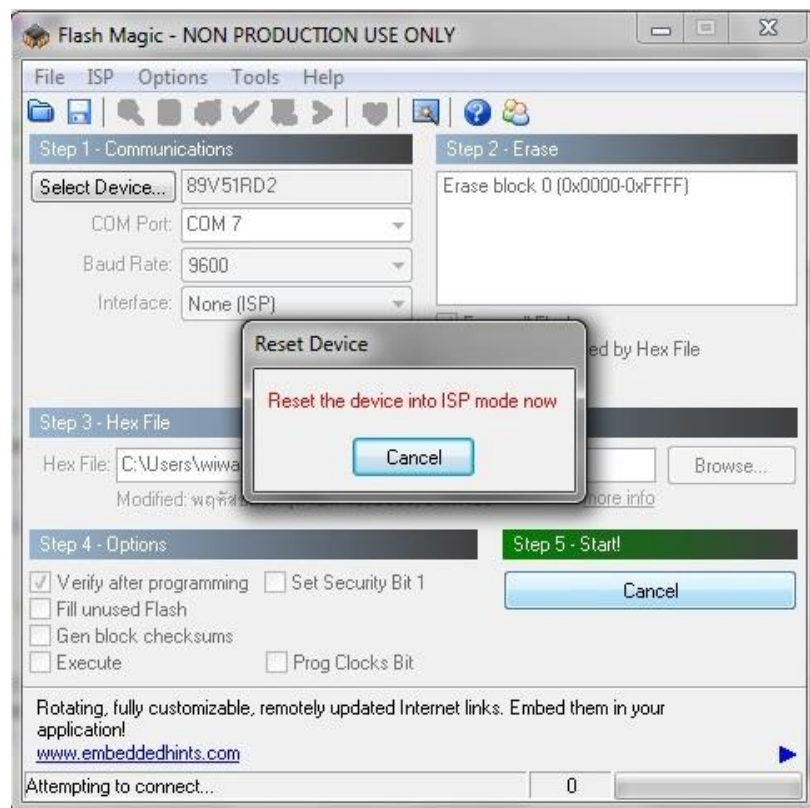
- c. คลิกเลือก Use DTR to control RST ออก



- d. ที่หมายเลข 1 เลือกรุ่นของ MCS-51 และ com port ให้ตรงกับที่เราใช้ ตั้งค่าอื่นๆ ตามหมายเลข 2 และ 4 ตามรูปพร้อมทั้ง Browse ไฟล์ที่หมายเลข 3 เลือก Hex file จากนั้นคลิกเลือก Start ที่หมายเลข 5 แล้ว โปรแกรมจะให้ Reset MCS-51 เพื่อโหลด Hex file ลงอุปกรณ์



- e. กดปุ่ม reset ที่บอร์ดไมโครคอนโทรลเลอร์



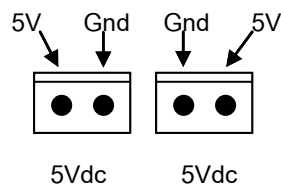
19. สำหรับการทดลองที่ 2.1 และ 2.2 ต่อ Port 0 เข้ากับ connector D0-D7 และ Port 1 เข้ากับ connector A0-A3 ของบอร์ด 7-segment และต่อ switch 2 ตัวเข้ากับ P3.2 และ P3.3

20. สำหรับการทดลองที่ 2.3 และ 2.4

a. เอาสาย serial ออกแล้วต่อสาย จากบอร์ด MCS-51 Port 1.0 กับ output ขั้ว – (สายสีดำ) จากบอร์ด Solid-state switch และ ต่อ output ขั้ว + (สายสีแดง) กับ Vcc ของบอร์ด MCS-51

b. ต่อสายไฟเข้ากับ 220 VAC output ground ของบอร์ด Solid-state switch แล้วเสียบปลั๊ก

c. ต่อขั้ว AIN ของบอร์ดลำโพงเข้ากับ บอร์ด MCS-51 port 3.0 พร้อมทั้งต่อ Vcc กับ ground ของทั้งสองบอร์ด เข้าด้วยกัน (บอร์ด MCS-51จะมีขั้ว ground อยู่ 3 อัน)



21. เมื่อทำการทดลองเสร็จแล้วทำแบบฝึกหัดส่งให้ครูช่างตรวจให้คะแนน

การทดลองที่ 2.1

/* Filename lab02_1.C	unsigned char code data7[] =
Description Test Output 7 Segments	{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x40};
Hardware P89v52RD2	/****** BASIC FUNCTION *****/
Clock 11.0592 MHz	void dmsec (unsigned int count) // mSec Delay
Number of oscillations per instruction 12	{ unsigned int i; // Keil CA51 (x2)
Compiler Keil C51 v8	while (count) {
*/	i = 230; while (i>0) i--;
#include<reg51.h>	count--; }
#include <intrins.h>	}
#include <absacc.h>	/****** MAIN PROGRAM *****/
unsigned char Select;	void main()

<pre>{ unsigned int j; while(1) { for (j=0;j<=7;j++) { Select = j; } } }</pre>	<pre>P0 = data7[j]; P1=Select; dmsec(1000); dmsec(500); } }</pre>
---	--

การทดลองที่ 2.2

<pre>/* Filename lab02_2.C Description Test external interrupt 0/1 Hardware P89v52RD2 Clock 11.0592 MHz Number of oscillations per instruction 12 Compiler Keil C51 v8 */ #include<reg51.h> #include <intrins.h> char Outdigit; bit bdata LeftFlag; unsigned char code data7[] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x 40}; void ex0_isr (void) interrupt 0</pre>	<pre>{ LeftFlag=1; } void ex1_isr (void) interrupt 2 { LeftFlag=0; } void dmsec (unsigned int count) { // mSec Delay unsigned int i; // Keil CA51 (x2) while (count) { i = 230; while (i>0) i--; count--; } } void main (void) { IT0 = 1; // Configure interrupt 0 for falling edge</pre>
---	---

EX0 = 1; // Enable EX0 Interrupt	{ Outdigit = Outdigit+1;
IT1 = 1; // Configure interrupt 1 for falling edge	if (Outdigit > 8)
EX1 = 1; // Enable EX1 Interrupt	{Outdigit=0;}
EA = 1; // Enable Global Interrupt Flag	}//end if
Outdigit = 0x00;	if(!LeftFlag)
LeftFlag=1;	{ Outdigit = Outdigit-1;
while (1)	if (Outdigit < 0)
{	{Outdigit=7;}
P1 = Outdigit;	} //end if
P0 = data7[Outdigit];	}//End of while
dmsec(500);	}//End of program
if(LeftFlag)	

การทดลองที่ 2.3

/* Filename lab02_3.C	#include<intrins.h>
Description Timer Software delay	#include <absacc.h>
Hardware P89v52RD2	unsigned int count1;
Clock 11.0592 MHz	sbit SOUND = P3^0;
Number of oscillations per instruction 12	sbit light = P1^0;
Compiler Keil C51 v8	
*/	/****** BASIC FUNCTION *****/
#include<reg51.h>	void delaysms(count1) //delay 1ms

```
{ unsigned int count2;

/* Set T/C0 Mode */

TMOD = (TMOD & 0xF0) | 0x01;

/* Load Timer with FFFFH- 922D=FC65H */

TH0 = 0xFC;

TL0 = 0x65;

TR0=1;

count2=0;

while (count2<=count1)

if (TF0==1)

{

TR0=0; /* Stop Timer 0*/

TF0=0;

count2++;

TH0 = 0xFC;

TL0 = 0x65;

TR0 = 1; /* Start Timer 0*/

}

}

/*****/

void main (void)
```

```
{ unsigned int i,j;

while (1)

{

/*****job 1 light*****/

light=0; //light on

delayms(1000);

light=1;//light off

/*****job 2 sound*****/

for(i=0;i<=2000;i++)

{

SOUND = 0;

for(j=1;j<=50;j++);

SOUND = 1;

for(j=1;j<=50;j++);

}

/*****/

}

}
```

การทดลองที่ 2.4

<pre>/* Filename lab02_4.C Description Timer Hardware Delay Hardware P89v52RD2 Clock 11.0592 MHz Number of oscillations per instruction 12 Compiler Keil C51 v8 */ #include<reg51.h> #include<intrins.h> #include<absacc.h> sbit SOUND = P3^0; sbit light = P1^0; unsigned int count; void timer0_ISR (void) interrupt 1 { TR0 = 0; /* Stop Timer 0*/ /* Load Timer with FFFFH- 922D=FC65H */ TH0 = 0xFC; TL0 = 0x65; TR0 = 1; /* Start Timer 0*/ count++;</pre>	<pre>/******job 1 light******/ if (count>=1000) {light=!light; count=0;} } // end Timer0 ISR /*******/ void main (void) {unsigned int i,j; count=0; TMOD = (TMOD & 0xF0) 0x01; /* Set T/C0 Mode */ /* Load Timer with FFFFH- 922D=FC65H */ TH0 = 0xFC; TL0 = 0x65; EA=1; ET0=1; TR0 = 1; /* Start Timer 0*/ while (1) { /******job 2 sound******/ for(i=0;i<=2000;i++) { SOUND = 0; for(j=1;j<=50;j++); SOUND = 1; for(j=1;j<=50;j++); }</pre>
--	---

```
/*******/
```

```
    }//end while
```

```
}// end main
```

แบบฝึกหัด

1. เขียนโปรแกรมเพื่อเป็นระบบกันขโมย โดยที่สัญญาณกันขโมยจะดังพร้อมดวงไฟติดเมื่อมีคนบุกรุก (มีสัญญาณ interrupt 0 จากภายนอก) และเมื่อมีคนเข้ามาตรวจสอบจะสามารถหยุดการเตือนได้ด้วยการกดรหัส แสดงรหัสที่กดบน 7-segment และวาดรูปวงจรและจำลองการทำงาน

การบ้าน

1. เขียน Flowchart อธิบายการทำงานของโปรแกรมที่ 2.1, 2.2, 2.3, 2.4 และแบบฝึกหัด

Reference

http://www.keil.com/support/man/docs/c166/c166_libref.htm

