

การทดลองที่ 6 การใช้งาน ADC และ DAC

วัตถุประสงค์

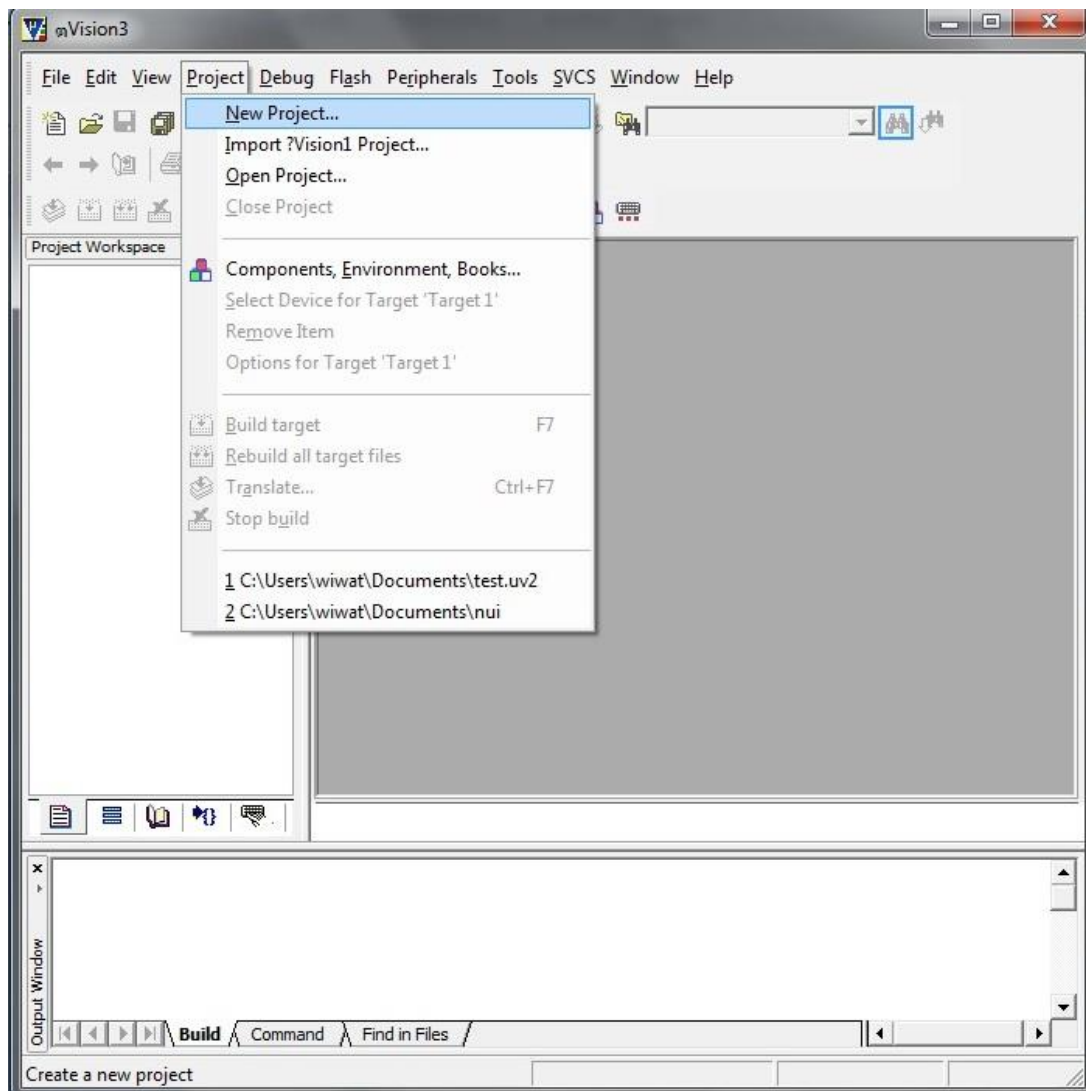
1. เพื่อให้นิสิตสามารถเขียนโปรแกรมภาษา C อย่างง่ายในการควบคุมไมโครคอนโทรลเลอร์ได้
3. เพื่อให้นิสิตสามารถใช้งาน ADC และ DAC เบอร์ PCF8591 ได้

อุปกรณ์ในการทดลอง

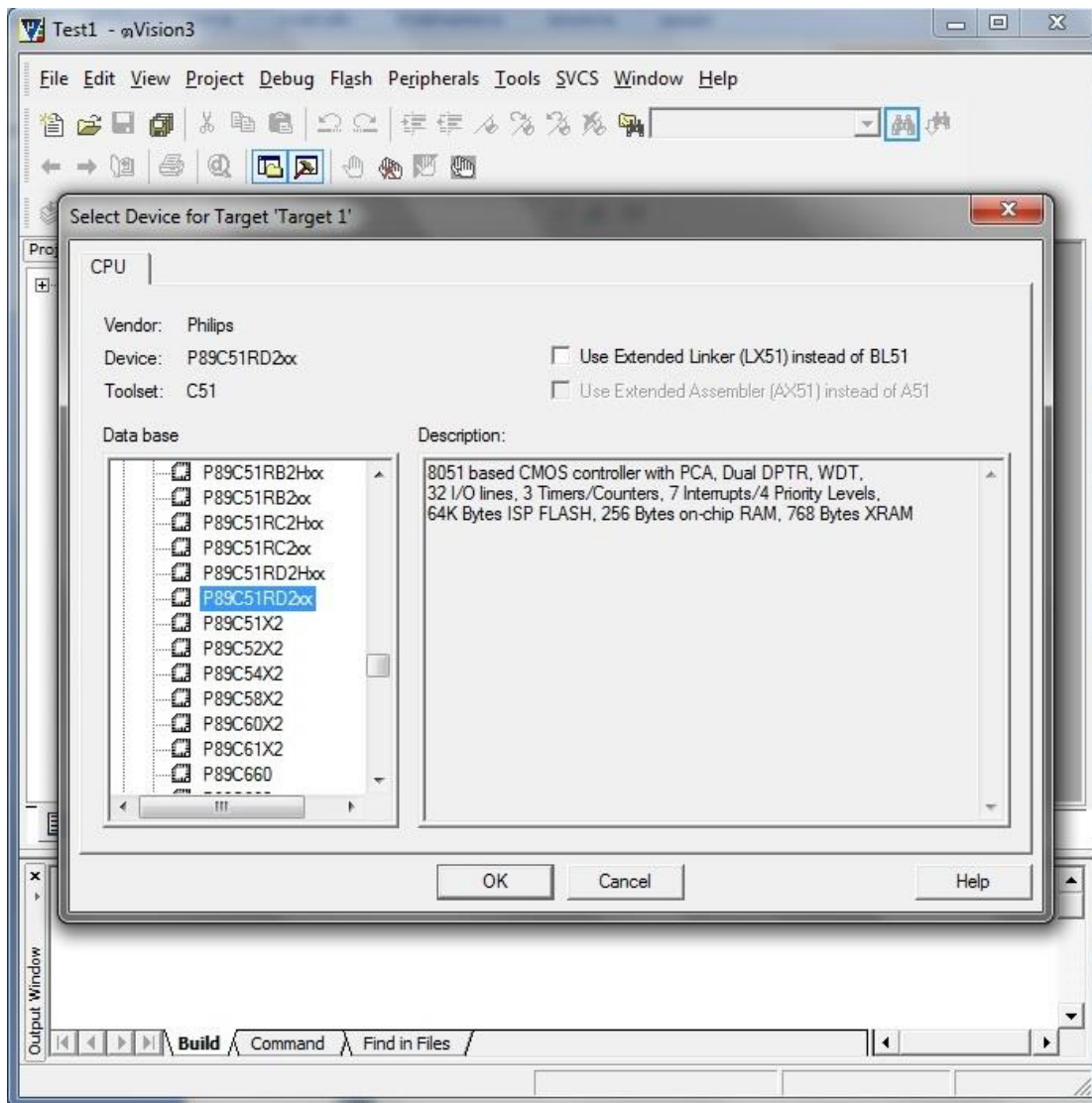
1. ชุดทดลองไมโครคอนโทรลเลอร์ SILA-START-C51
2. เครื่องคอมพิวเตอร์ PC พร้อมโปรแกรมสำหรับการเขียนและคอมไพล์ภาษา C - โปรแกรม Keil51 v.xx
3. สายต่อพอร์ตอนุกรม
4. แหล่งกำเนิดสัญญาณกระแสตรงปรับค่าได้ 0-5 V.
5. Multi-meter

การใช้งาน Keil51

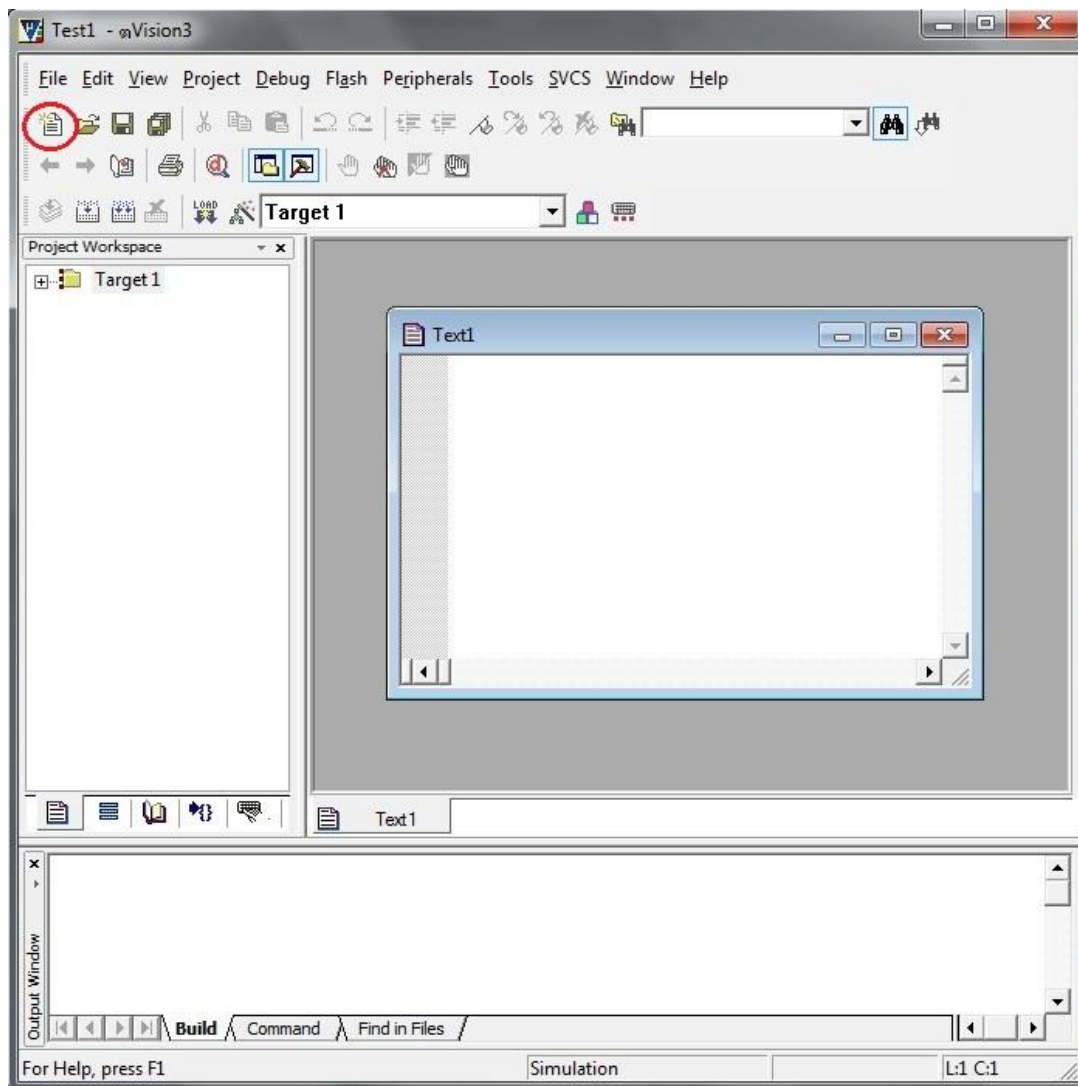
1. ใช้ IDE สำหรับพัฒนาชุดคำสั่งของ MCS-51 ด้วยภาษา C โดยใช้ Keil51 เพื่อเขียนโปรแกรม Lab06_x.c
2. สร้างโปรเจกต์ใหม่โดยเลือก New project ตั้งชื่อเป็น Lab06 แล้วกด ok



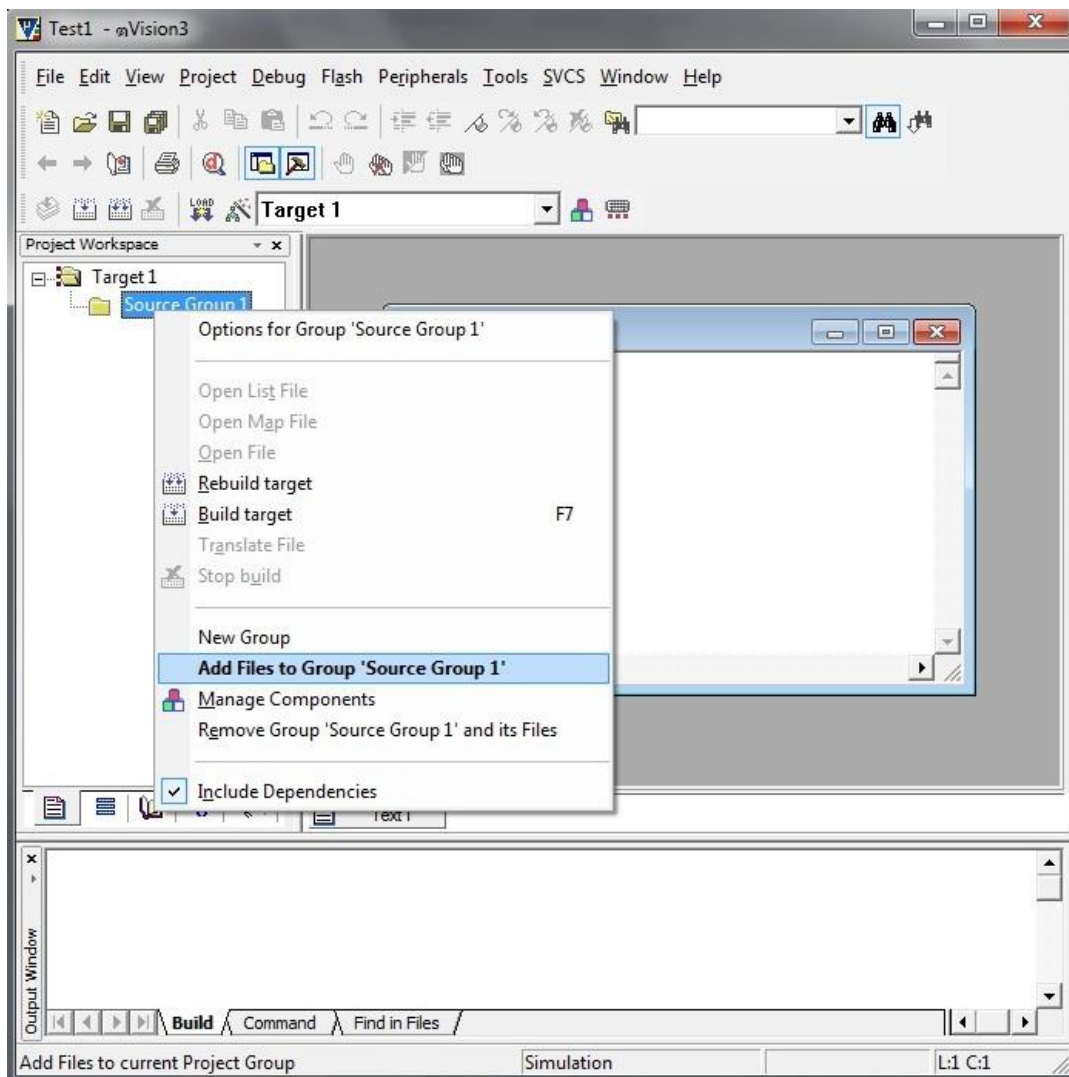
3. เลือก CPU : Philips P89C51RD2xx



4. จากนั้นคลิกที่ปุ่มวงกลมสีแดง เพื่อสร้างหน้าต่างสำหรับเขียนโปรแกรม(หน้าต่าง Text1)
สร้างไฟล์ใหม่ เขียนโปรแกรมและ save โดยใช้ชื่อ Lab06_x.c (x แทนหมายเลขการทดลอง)

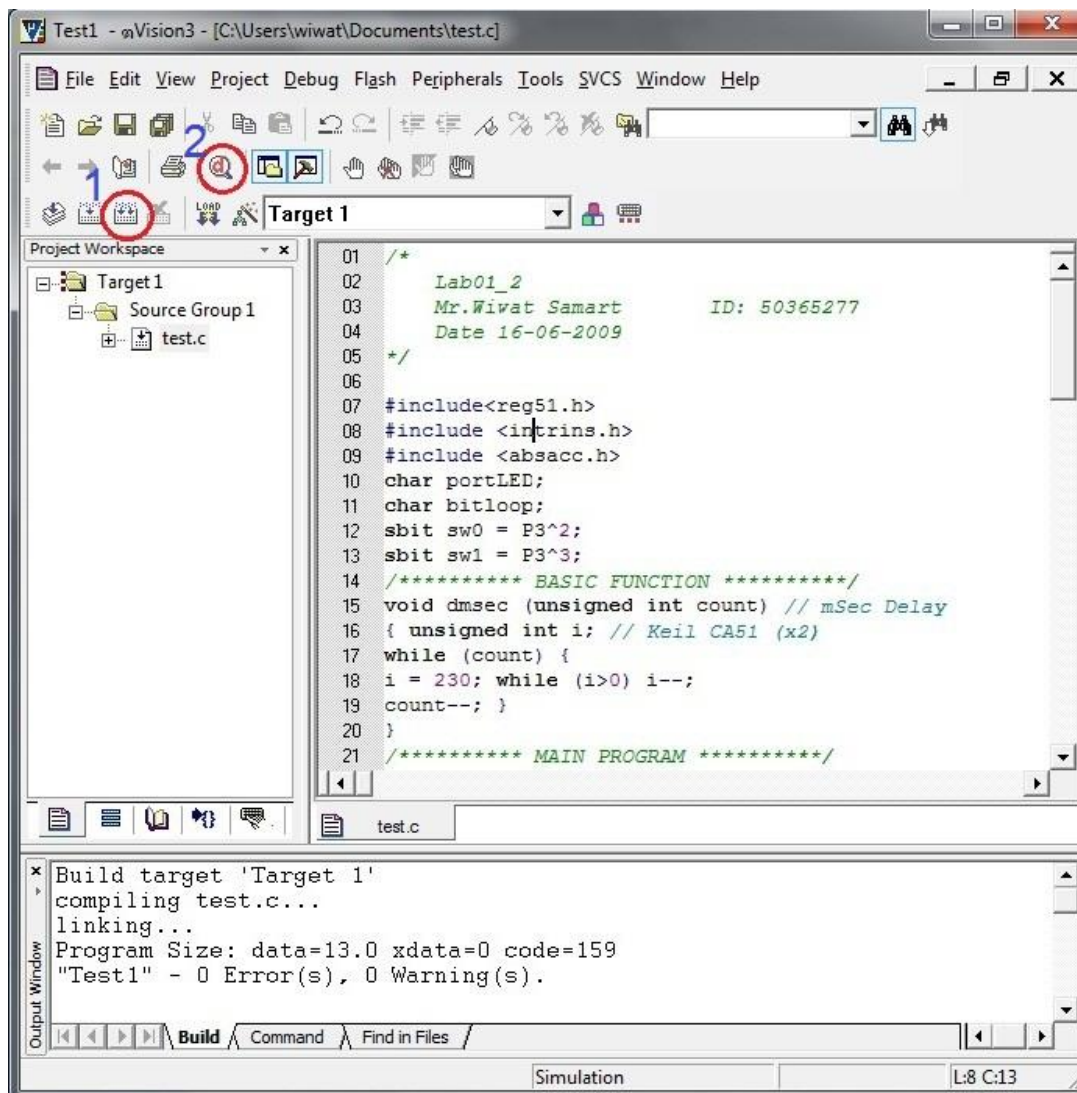


5. คลิกขวาที่ Source Group 1 แล้วเลือก Add file to group 'Source Group 1' แล้วเลือก File Lab06_x.c



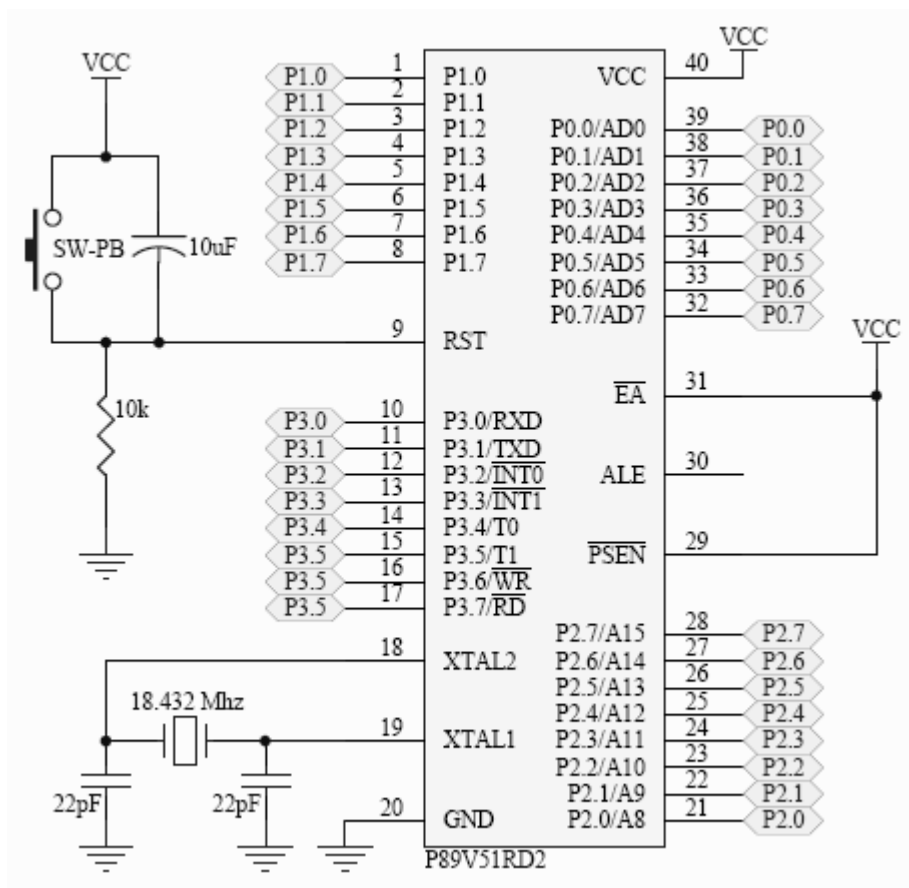
6. เขียนโปรแกรมให้เสร็จแล้วคลิกที่ วงกลมที่ 1 เพื่อทำการ compile ถ้าหากว่าไม่พบ Error โดยที่ถ้าเกิด error ขึ้น โปรแกรมจะแสดงว่า error ก็ที่ (ดู windows output ด้านล่างของโปรแกรม) คลิกที่ วงกลมที่ 2 เพื่อให้ keil ตรวจสอบ และทำการ debug พร้อมทั้งตรวจสอบขนาดของโปรแกรมที่เราสร้างขึ้นและพร้อมสำหรับการทดสอบการทำงานของโปรแกรม

7. Build โปรแกรมที่เขียน และให้เลือก option → create hex file โดยเลือกที่ check box Hex file ที่ ได้จะมีชื่อเหมือนกับ ชื่อโปรเจ็ค คือ Lab06.hex

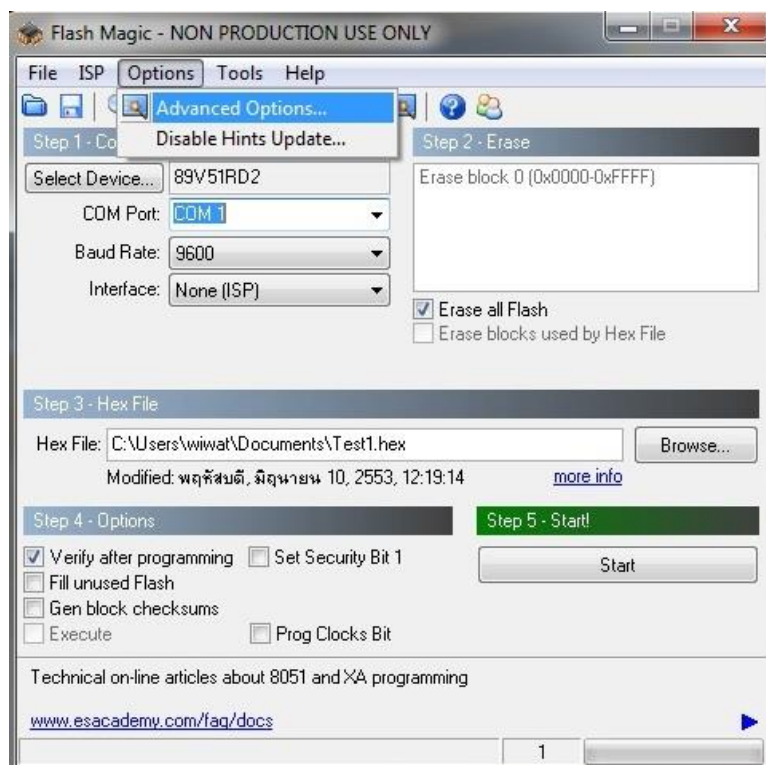


การใช้งานบอร์ด MCS-51และโปรแกรม Flash Magic

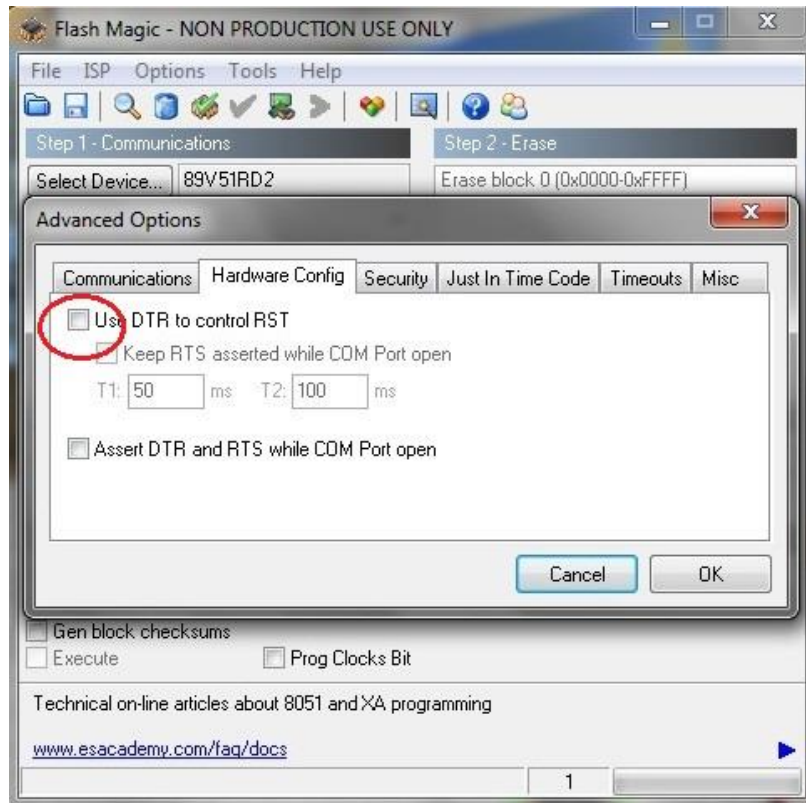
1. โหลดโปรแกรมลงบนบอร์ด MCS-51 โดยใช้บอร์ดไมโครคอนโทรลเลอร์ซึ่งมีลักษณะวงจรวัดดังนี้



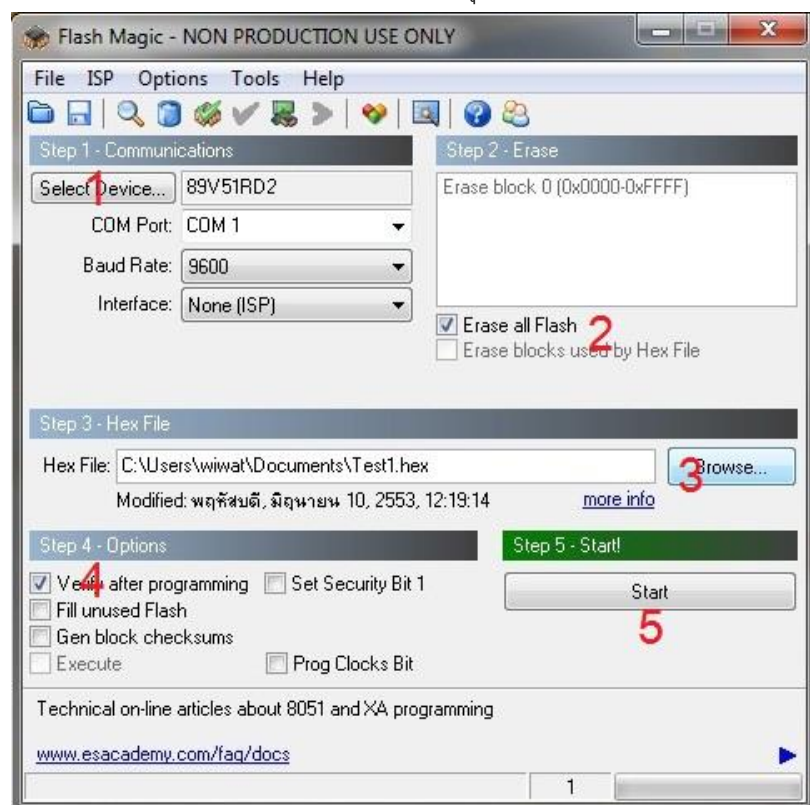
- ต่อสายจาก com1 ของ computer เข้ากับบอร์ดโดยระวังให้ connector ด้านที่เป็น ground ตรงกับ เครื่องหมาย - ทำการโหลดผ่านโปรแกรม Flash magic ดังนี้
- เปิดโปรแกรม Flash magic โดยทำการตั้งค่าต่างๆ ดังนี้ เลือกที่ Options----> Advanced Options



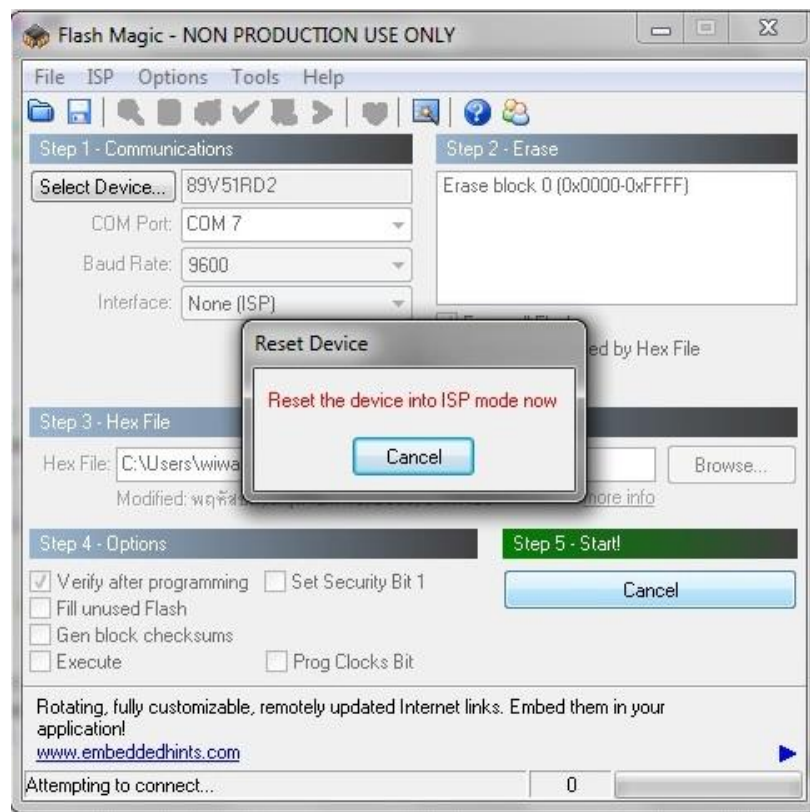
- c. คลิกเลือก Use DTR to control RST ออก



- 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



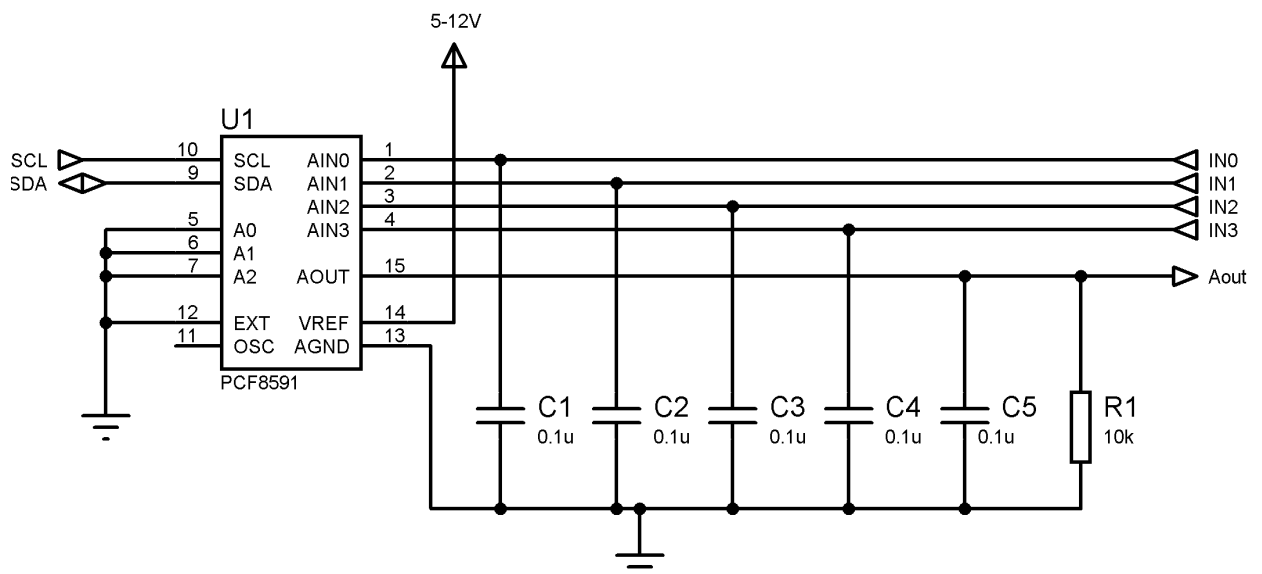
f. เชื่อมสาย serial ออก

วิธีการทดลอง

1. ต่อดวงจรมตามรูปแล้วทดลองรันโปรแกรม

V_{SS} ขา 8 ground

V_{DD} ขา 16 supply voltage



2. วัด V_{Aout} ของ ADC ที่ขา 15 พร้อมทดลองแก้ไขโปรแกรมให้ V_{out} มีค่า 4.5 โวลต์ โดยคำนวณจากสมการ

$$V_{AOUT} = V_{AGND} + \frac{V_{VREF} - V_{AGND}}{256} \sum_{i=0}^7 D_i \times 2^i$$

3. ต่อ channel 1 ของ ADC ด้วย sensor วัดระยะทาง ทดลองเปลี่ยนระยะทาง จาก sensor กับวัตถุเป็น 4, 6, 8, 10 และ 12 cm. แล้วคำนวณค่าที่ ADC อ่านได้เทียบกับค่า output ของ sensor ที่ระบุมาใน คู่มือของ sensor จากสมการเดิมแต่เปลี่ยน V_{Aout} เป็น V_{in1}

ระยะทาง*2D120	10 cm	15 cm	20 cm	25 cm	30 cm	35 cm
ค่า ADC (ตัวเลข)						
ระดับแรงดัน (คำนวณจากตัวเลข ADC)						
ระดับแรงดัน (วัดโดย multimeter)						
ระดับแรงดัน (จากคู่มือ sensor)						

ระยะทาง*2Y0A21	10 cm	20 cm	30 cm	40 cm	50 cm	60 cm
----------------	-------	-------	-------	-------	-------	-------

การทดลองที่ 6.1

```
#include <reg51.h>

#include <intrins.h>

#include <stdio.h>

#define CH0 0x40
#define CH1 0x41
#define CH2 0x42
#define CH3 0x43

/***** I2C I/O Bit *****/

sbit SCL = P1^5;          //pin10
sbit SDA = P1^6;          //pin9

/*****Basic function*****/

void dmsec (unsigned int count) {      // mSec Delay
    unsigned int i;                    // for Keil CA51 (Speed
    x1)
    while (count) {
        i = 115; while (i>0) i--;
        count--;
    }
}
```

```
/******I2C******/
```

```
void I2C_delay(void)
```

```
{
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
}
```

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

```
void SCL_high(void) // I2C clock high
```

```
{
```

```
SCL = 1;
```

```
I2C_delay();
```

```
}
```

```
void SCL_low(void) // I2C clock low
```

```
{
```

```
SCL = 0;
```

```
I2C_delay();
```

```
}
```

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

```
void I2C_stop(void) // Stop
```

```
{
```

```
SDA = 0;
```

```
SCL = 1;
```

```
I2C_delay();
```

```
SDA = 1;
```

```
}
```

```
void I2C_start(void) // Start
```

```
{
```

```
SDA = 1;
```

```
SCL = 1;
```

```
SDA = 0;
```

```
I2C_delay();
```

```
SCL = 0;
```

```
SDA = 1;
```

```
}
```

```
bit I2C_wrbyte (unsigned dat)
```

```
// write data , return 0=ok , return 1=error
```

```
{
```

```
unsigned char i;
```

```
bit outbit;
```

```
    for (i=1;i<=8;i++) // send 8 bits
```

```
    {
```

```
        outbit = dat & 0x80;
```

```
        SDA = outbit; // send 1 bit
```

```
        dat = dat << 1; // shift 1 bit
```

```
        SCL_high();
```

```
        SCL_low();
```

```
    }
```

```
SDA = 1;
```

```
SCL_high();
```

```
outbit = SDA;
```

```
SCL_low();
```

```
return(outbit);
```

```
}
```

```
unsigned char I2C_rdbyte() // read data byte
```

```
{
```

```
unsigned char i, dat;    // return 0xff = error
```

```
bit inbit;
```

```
dat = 0;
```

```
    for (i=1;i<=8;i++) // read 8 bits
```

```
    {
```

```
        SCL_high();
```

```
        inbit = SDA; // read 1 bit
```

```
        dat = dat << 1;
```

```
        dat = dat | inbit;
```

```
        SCL_low();
```

```
    }
```

```
SDA = 1;
```

```
SCL_high();
```

```
inbit = SDA;
```

```
SCL_low();
```

```
if (~inbit) dat = 0xff;
```

```
return(dat);
```

```
}
```

```
/******ADC******/
```

```
unsigned ADC(unsigned char channel)
```

```
{
```

```
unsigned char temp; //ADC input buffer
```

I2C_start();	//
I2C_wrbyte(0x90); //Write pcf8591 address(write)	speed x 2
I2C_wrbyte(0x40 channel); //Write control byte	SCON = 0x52; // set RS232 parameter
I2C_stop();	TMOD = 0x20; // Crystal = 11059200
	TH1 = -6; PCON = 0x80;
	//9600
I2C_start();	//
I2C_wrbyte(0x91); //Write address(read)	TH1 = -12; PCON = 0x80; //4800
temp = I2C_rdbyte(); //Read data	
I2C_stop();	TR1 = 1;
return(temp);	RI = 0;
}	}
	/***** MAIN *****/
	void main(void){
/***** D/A *****/	unsigned char a;
void DAC(unsigned char dat){	unsigned int w,g;
I2C_start();	start232();
I2C_wrbyte(0x90);	
I2C_wrbyte(0x40);	while(1){
I2C_wrbyte(dat);	for (g=0;g<=255;g++) {
I2C_stop();	a = ADC(CH1);
}	
/***** *****/	w = a; //Vref 5 V. Actual value = a*5/256
void start232(void)	= a*0.0195
{	

<pre>// w = a*0.0195; printf("ADC CH1 = %d \n",w); printf("ADC value = %d \n",a);</pre>	<pre> DAC(a); dmsec(1000);} }</pre>
--	--

Reference

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