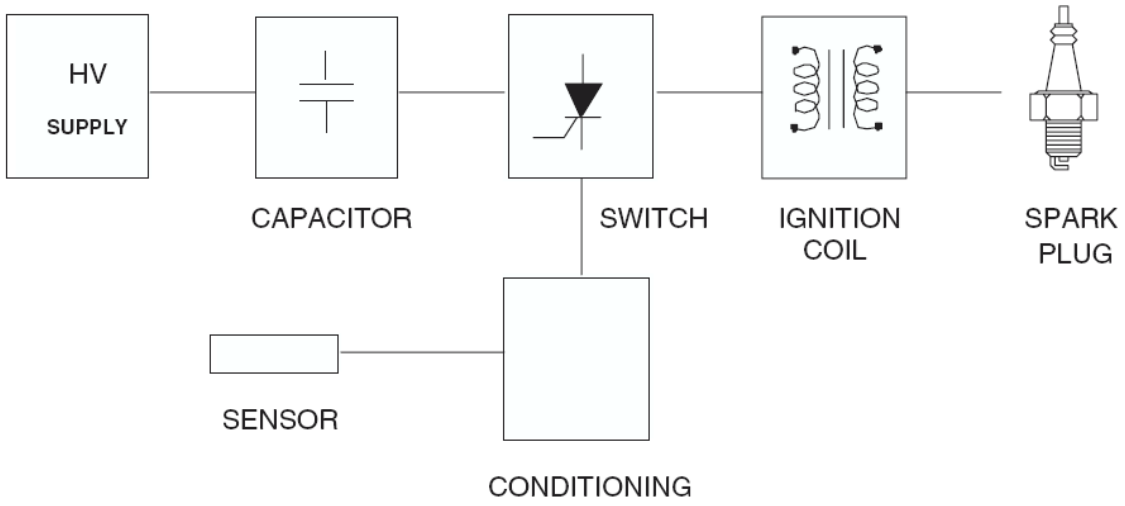
CAPACITIVE DISCHARGE IGNITION [CDI]

CDI เป็นอุปกรณ์ที่มีวิวัฒนาการตามความเจริญก้าวหน้าทางเทคโนโลยีอิเล็กทรอนิกส์ ซึ่งถูกนำมาใช้เป็นตัวจุดระเบิดในห้องเผาไหม้ของเครื่องยนต์แก๊สโซลีน ทั้งในเครื่องยนต์ สองจังหวะ และ สี่จังหวะ ทั้งแต่เครื่องยนต์ หนึ่งสูบจนไปถึงเครื่องยนต์ แปด สูบ ซึ่งการออกแบบ CDI ในยุกต์ปัจบันได้ก้าวหน้าไปมาก ซึ่ง เป็นไปตามความก้าวหน้าของเทคโนโลยีระบบควบคุมด้วยไมโครคอมพิวเตอร์ ประกอบกับ ข้อกำหนดในเทคโนโลยียานยนต์ในปัจจุบัน ที่มีการควบคุมเรื่องมลภาวะในอากาศ ดังนั้นการออกแบบ CDI ให้มีประสิทธิภาพสูง ก็มีส่วนเป็นอย่างมาก ที่จะให้การเผาไหม้เชื้อเพลิงในห้องเผาไหม้ของเครื่องยนต์เป็นไปอย่างมีประสิทธิ ซึ่งส่งผลถึง การปล่อยมลพิษ และ ประสิทธิภาพของเครื่องยนต์ ในการจัดทำเอกสารขึ้นครั้งนี้ ผู้เรียบเรียงได้จัดทำขึ้นเพื่อใช้ในการฝึกอบรม และ นักประดิษฐทั้งหลายที่สนใจ เกี่ยวกับการออกแบบและสร้างระบบ CDI ด้วยตัวเอง

ระบบการทำงานของ CDI ที่สำคัญ ๆ แบ่ง ออกได้เป็น 7 ระบบด้วยกันดังแสดงตามภาพข้างล่าง



ระบบ CDI

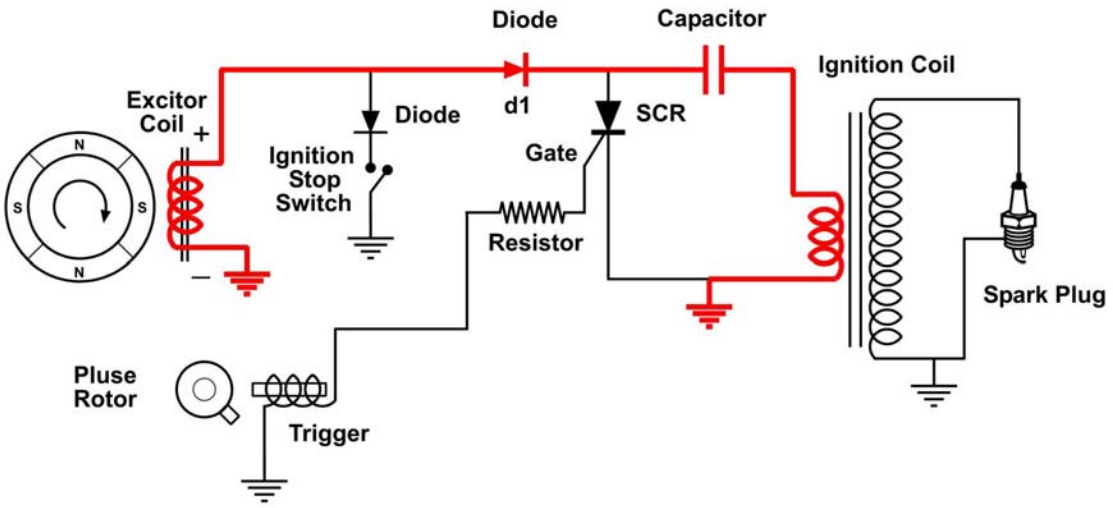
การทำงานของแต่ละระบบพอจะอธิบายคล่าว ๆ ได้ดังนี้

HV SUPPLY

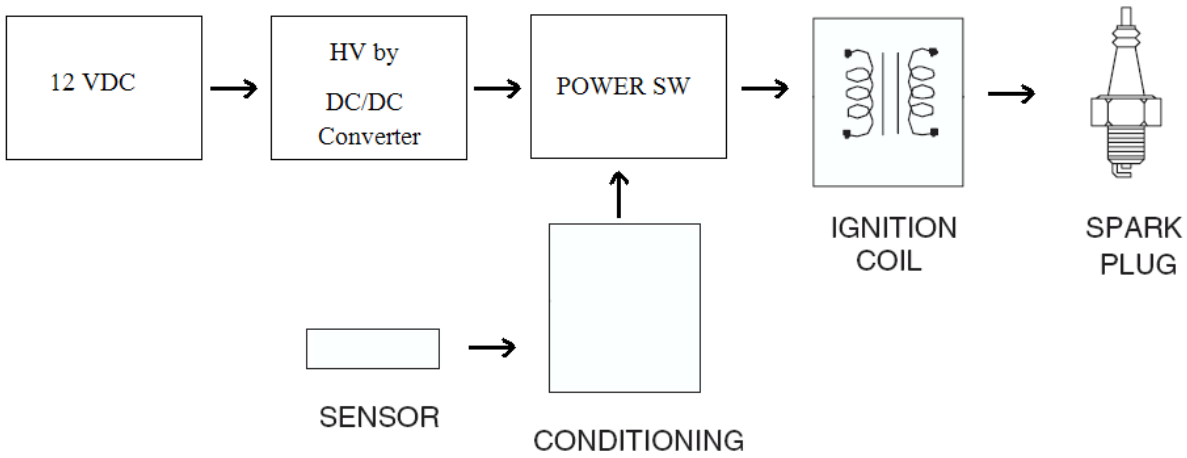
เป็นตัวสร้างแรงดันไฟฟ้าสูงเพื่อป้อนให้กับระบบ ซึ่งที่เห็นอยู่ในปัจจุบันมีสองแบบด้วยกันคือ ได้มาจากชุดกำเหนิดไฟฟ้าในตัวเครื่องยนต์ และ ได้มาจากวงจรปรุงแต่งในรูปของ DC/DC Converter ดังแสดงตามภาพข้างล่าง

POWER SW

เป็นตัวปลดปล่อยพลังงานที่สระสมอยู่ใน CAPACITOR ในรูปแรงดันไฟฟ้าสูง



ระบบ CDI ใช้แรงดันสูงจากตัวเครื่อง



ระบบ CDI ใช้แรงดันสูงจากวงจร DC/DC Converter

ซึ่งจะอธิบายรายละเอียดทั้งสองระบบนี้ในโอกาสข้างหน้าต่อไป

CAPACITOR

มีหน้าที่สระสมพลังงานในรูปแรงดันไฟฟ้าที่ได้จากแหล่งกำเนิดแรงดันไฟฟ้าสูง และพร้อมที่ จะคายประจุให้กับขดลวดปฐมภูมิของหม้อแปลงจุดระเบิด

POWER SW

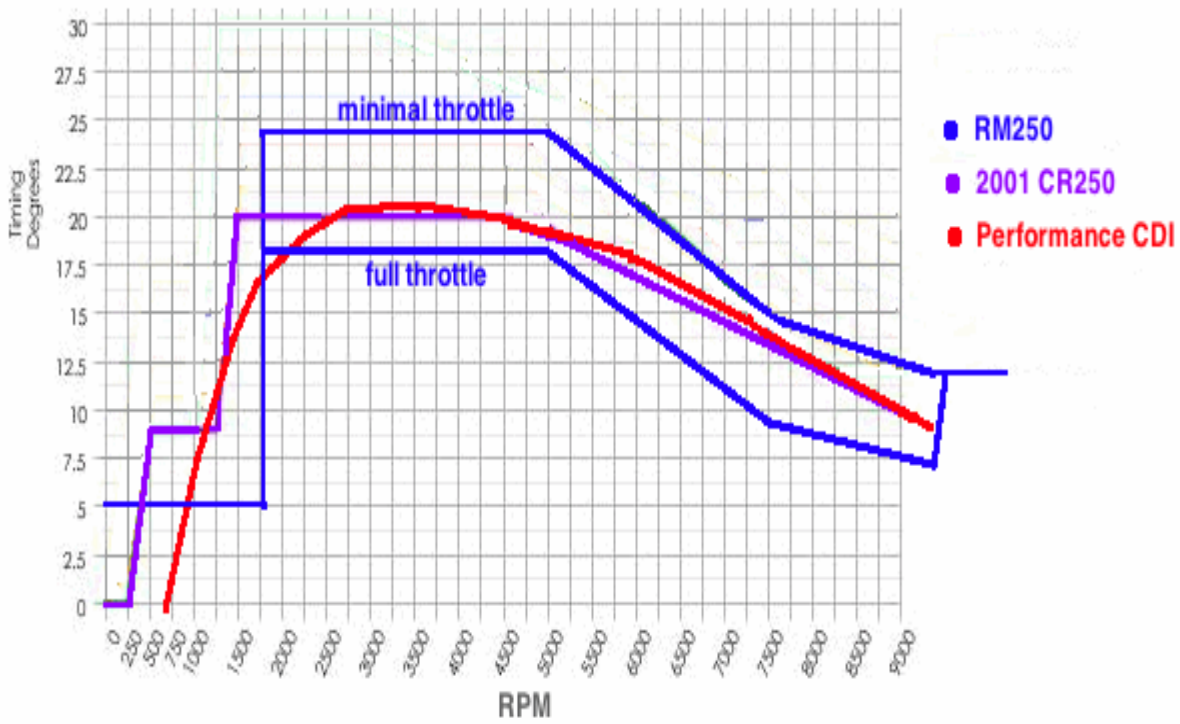
ทำหน้าที่ปลดปล่อยพลังงานที่สระสมอยู่ใน CAPACITOR ให้กับขดลวดปฐมภูมิของหม้อแปลงจุดระเบิด

SENSOR

ภารกิจหลักของ SENSOR เป็นตัวบอกแกนเวลา หรือ ตำแหน่งการหมุนของเครื่องยนต์ เพื่อ ให้การทำงานของระบบเครื่องยนต์สอดคล้องและประสานงานกันอย่างลงตัว สำหรับ บทบาทที่ชัดเจนของ SENSOR กับCDI คือ ตัวบอกตำแหน่งจุดระเบิดในแต่ละวัตจักรของยานยนต์

CONDITIONING

เป็นส่วนที่สำคัญมาก ๆ ที่จะทำให้การทำงานของเครื่องยนต์ ทำงานได้อย่างมีประสิทธิภาพภายใต้สภาวะการต่างของเครื่องยนต์ เช่น ที่ความเร็วรอบต่าง กัน ภาระโหลดต่างกัน รวมไปถึงสภาวะแวดล้อมที่แตกกัน เช่น อากาศ เย็น ร้อน หนาว เป็นต้น ส่วนสำคัญที่ CONDITIONING ต้องทำให้ดีที่สุดคือควบคุมองศาการจุดระเบิดล่างหน้าศูนย์ตายบน [BTDC] ให้เป็นไปตามภาวการณ์ข้างต้นดังกล่าว รูปข้างล่างแสดงให้เห็นมุมจุดระเบิดที่สภาวะต่าง ๆ



แสดงมุมจุดเบิดที่สอดคล้องในแต่ละความเร็วรอบและตำแหน่งลิ้นผีเสื้อ

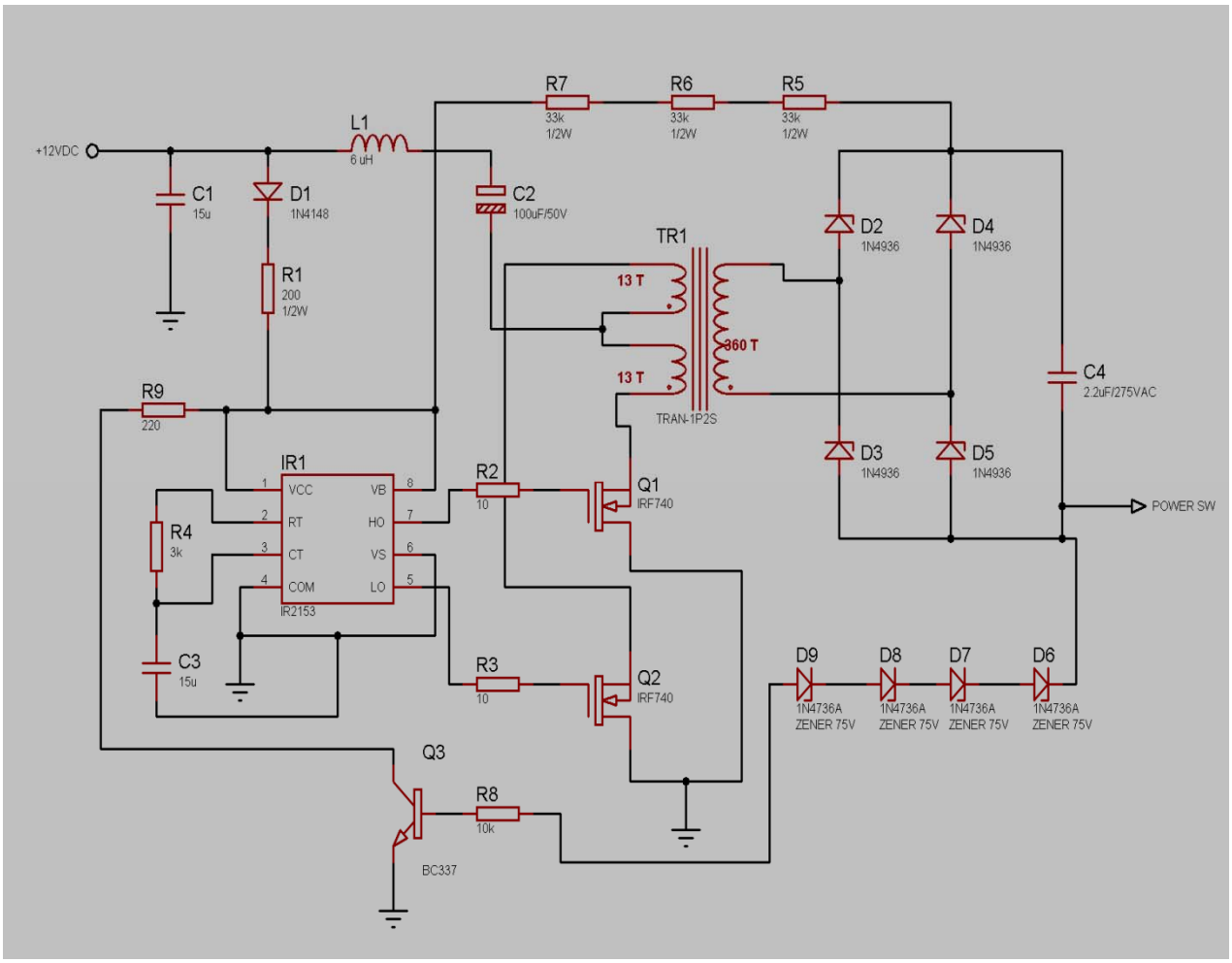
IGNITION COIL

เป็นหม้อแปลง STEP UP เพื่อยกระดับแรงดันเพื่อให้กระแสไฟฟ้าไหลผ่านช่องว่างของตัวนำที่เป็นโลหะได้ ทำให้เกิดประกายไฟในการจุดระเบิดเชื้อเพลิงได้

SPARK PLUG

เป็นส่วนสุดท้ายของระบบจุดระเบิด ที่มีส่วนสำคัญต่อกระบวนการจุดระเบิด ให้เป็นไปอย่างมีประสิทธิภาพ เนื่องจากเป็นส่วนที่ก่อให้เกิดประกายไฟ SPARK PLUG ที่ดีจะทำให้การใช้พลังงานในการสร้างประกายไฟน้อยลง

CDI ที่ใช้ระบบแรงดันสูงจาก DC/DC CONVERTER



วง HV DC TO DC CONVERTER

ส่วนของภาค HV DC TO DC CONVERTER ที่จะต้องเรียนรู้และทดลอง

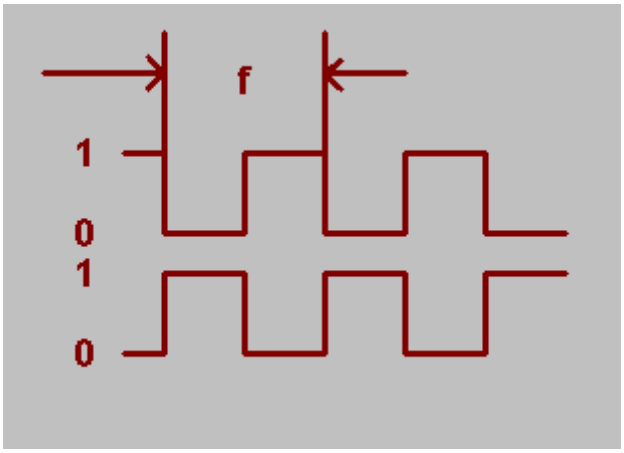
1 ภาคไฟเลี้ยง IC IR2153 จากวงจรประกอบด้วยสองส่วนด้วยกันคือ

- จาก ไฟ DC +12 V ผ่าน D1 ,R1 R1 เป็นตัวจำกัดกระแสที่จ่ายให้กับ IR2153 ที่ต้องให้ความสนใจคือ ถ้ากระแสไม่เพียงพอ IR2153 จะไม่ทำงาน

- จากไฟแรงดันสูงจากทางดัน output ของหม้อแปลง SWITCHING และถูกจำกัดกระแสด้วย R5 R6 R7 ส่วนนี้เป็นส่วนช่วยให้แหล่งจ่ายไฟเลี้ยง IR2153 เป็นไปอย่างมีประสิทธิภาพทุกขณะการทำงาน

2 ส่วนในการสร้างสัญญาณควบคุมการทำงานของ Q1 Q2 ซึ่งมาจากค่าของ R4 C3 ซึ่งการคำนวณค่าความถี่ให้กับ SWITCHING ดังสมการข้างล่าง

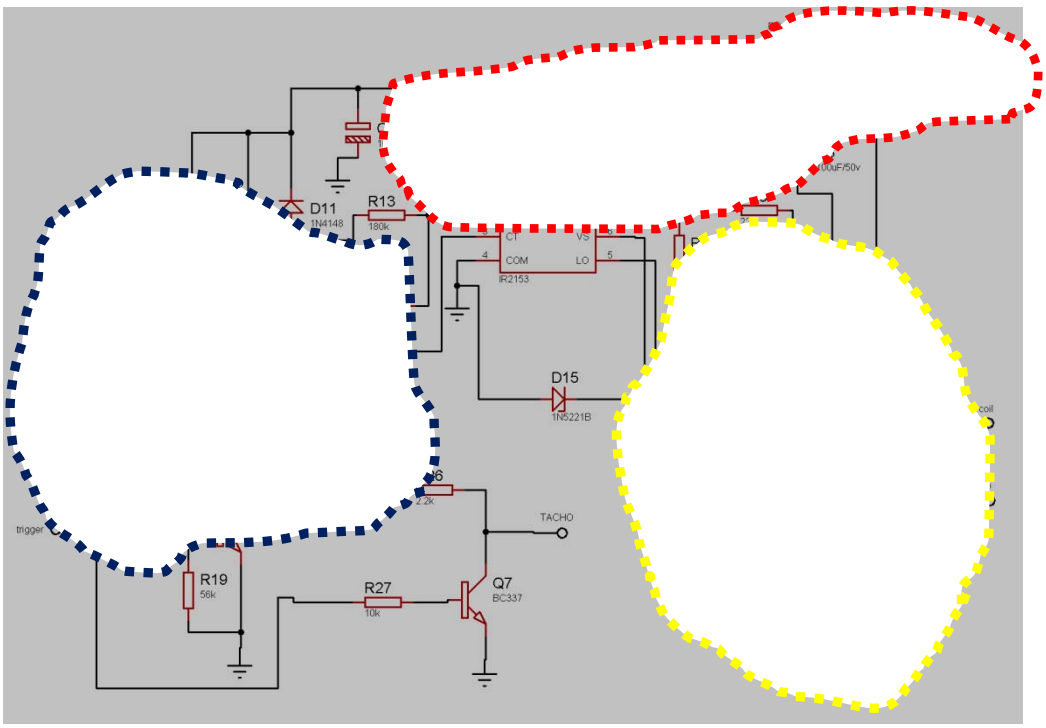
$$f = \frac{1}{1.4 \times (R_T + 75\Omega) \times C_T}$$



สัญญาณควบคุมที่ขา 5 และ 7 ของ IR2153

3 ส่วนควบคุมแรงดันไฟแรงดันสูงให้อยู่ในระดับ 300 VDC โดยใช้ แรงดัน Breakdown ของ ZENER D6 D7 D8 D9 ไปควบคุมการทำงานของ transistor Q3 เป็นลักษณะของการควบคุมป้อนกลับ แบบ ON – OFF Control

ส่วนของภาค POWER SW ที่จะต้องเรียนรู้และทดลอง

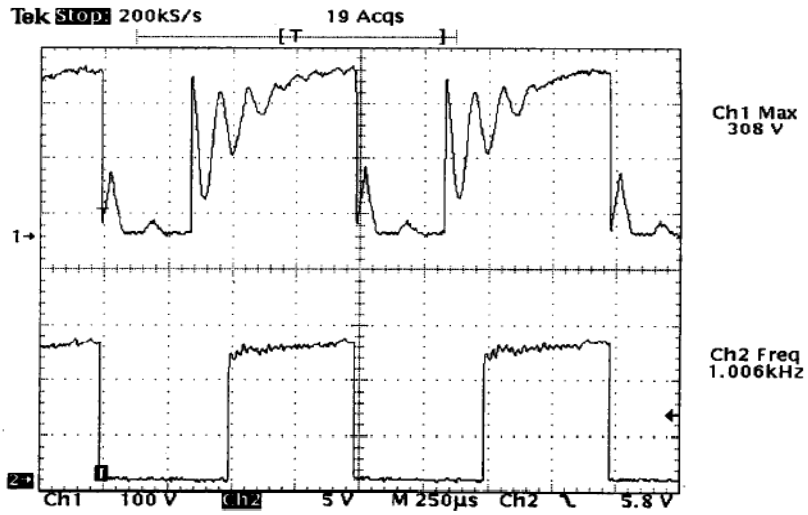


ภาค POWER SW ระบบ CDI ที่ใช้ระบบแรงดันสูงจาก DC/DC CONVERTER

1 ส่วนของไฟเลี้ยง IC IR2153 ซึ่งรับมาจากภาคแรงดันสูง HV DC/DC Converter จำกัดกระด้วย R10 R11 R12 ตามกรอบในเส้นประสีแดงดังรูป (ภาวะปกติแรงดันตกคร่อมขา 1 ของ IC อยู่ที่ 15 vdc)

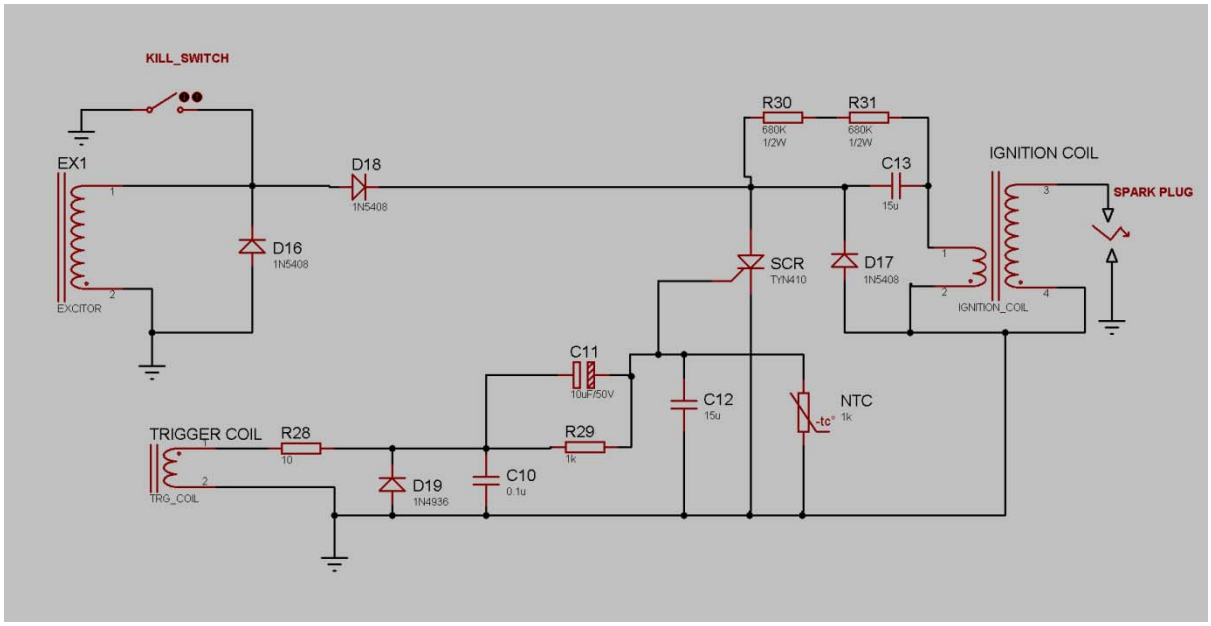
2 ส่วนของการควบคุมความกว้างของ PULS ในการ ON – OFF POWER มาจากส่วนของวงจร โมโนสเตเบิลดังแสดงในกรอบเส้นประสีน้ำเงิน(การทำงานของวงจรโดยละเอียดให้ดูจากเอกสารอ้างอิงท้ายเล่ม)

3 ส่วนของการ CHARGE และ Discharge ของ Capacitor เพื่อถ่ายพลังงานให้กับขดลวดชุด Primary ของ ignition Coil ดังแสดงตามรูปข้างล่าง



แรงตกคร่อม Capacitor ขณะ ON – OFF

ระบบ CDI ที่ใช้แรงดันสูงจากเครื่องยนต์

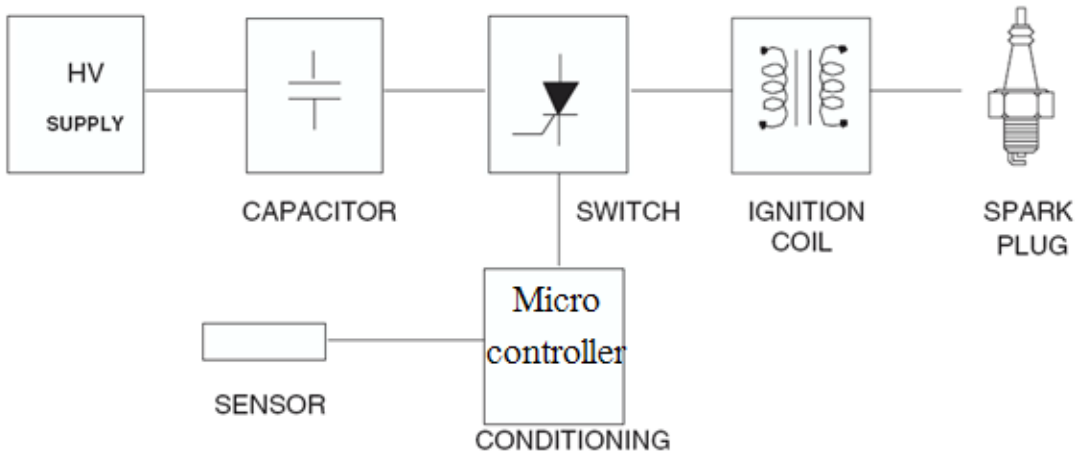


วงจร CDI แบบใช้แรงดันสูงจากเครื่องยนต์

ระบบ CDI ที่ใช้แหล่งแรงดันสูงจากเครื่องยนต์ เป็นระบบที่เป็นไปด้วยความเรียบง่าย องค์ประกอบของระบบทั้ง 7 ส่วนที่กล่าวมาข้างต้น ครบทุกระบบ จากอุปกรณ์อิเล็กทรอนิกส์เพียงไม่กี่ตัว อีกทั้งระบบดังกล่าวไม่จำเป็นต้องมีแหล่งจ่ายไฟจากข้างนอก เครื่อง ยนต์ก็สามารถ ทำงานได้ ระบบนี้จะพบได้ในเครื่องยนต์ยุกต์เก่า ๆ ในส่วนของการเรียนรู้และทดลองน่าจะมีแค่สองส่วนหลัก ๆ คือ สัญญาณควบคุมการ ON – OFF Power SW ที่ขา GATE ของ SCR แรงดันตกคร่อม CAPACITOR ขณะ CHARGE และ Discharge แรงดันให้กับขดลวด Primary ของ IGNITION Coil

การพัฒนาส่วนของ CONDITIONING โดยใช้ไมโครคอนโทรเลอร์

ดังที่ได้กล่าวมาข้างต้นแล้วว่า ส่วนของ CONDITIONING มีบทบาทสำคัญมากต่อการเพิ่มประสิทธิภาพให้กับ CDI และเป็นส่วนที่มีวิวัฒนาการมากที่สุดตั้งแต่เริ่มกำเนิด CDI มาจนถึงยุคปัจจุบัน เนื่องจากได้มีการนำเทคโนโลยีไมโครคอนโทรเลอร์ มาเป็นตัวประเมินผลหลักให้กับส่วนของ CONDITIONING ส่งผลให้การปรับปรุงแก้ไขจุดบกพร่องต่าง ๆ เพื่อให้ ระบบตอบสนองต่อการทำงานของเครื่องยนต์ในสภาวะการต่าง ๆ ที่กล่าวมาแล้วข้างต้นเป็นไปอย่างมีประสิทธิภาพ และ มีวิวัฒนาการอย่างรวดเร็ว



การใช้ไมโครคอนโทรเลอร์เป็นตัวประเมินผลหลักให้กับส่วนของ Conditioning

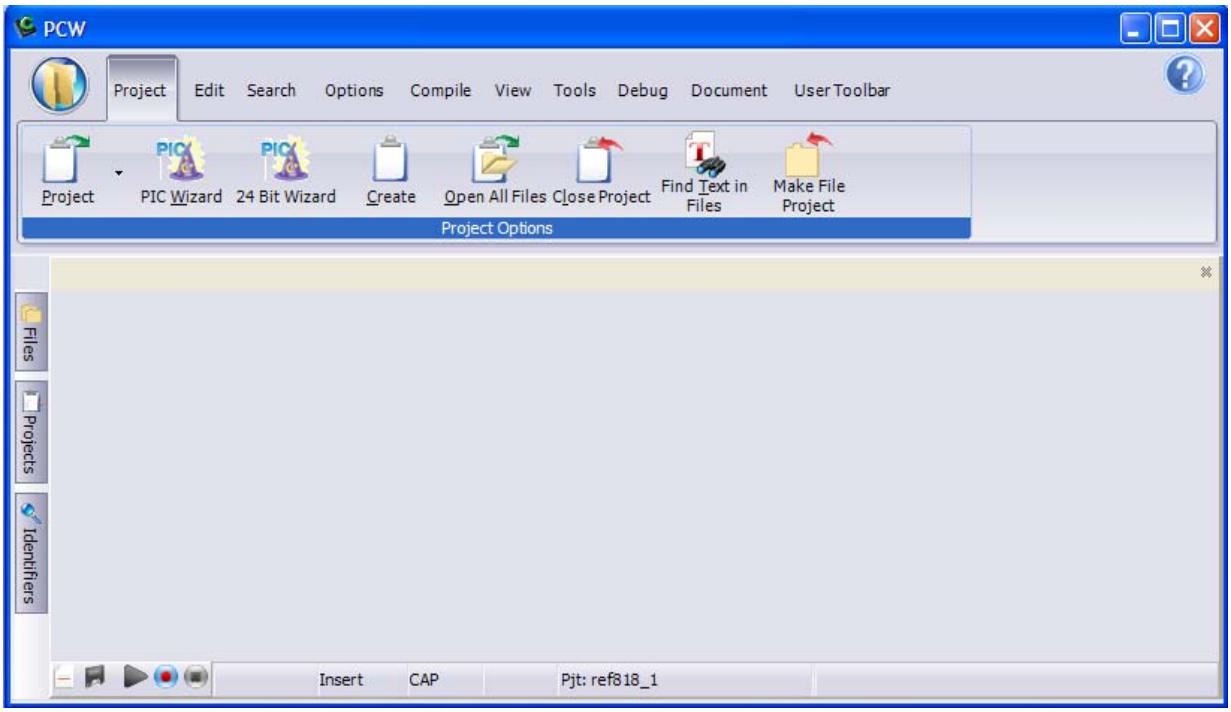
การเรียนรู้และทำความเข้าใจเพื่อนำเทคโนโลยีไมโครคอนโทรเลอร์ มาประยุกต์ใช้ในระบบควบคุมต่าง ๆ ในปัจจุบันนั้นถือเป็นสิ่งจำเป็นกับนักประดิษฐ์ หรือ วิศวกรเกือบทุกสาขา เนื่องจากเทคโนโลยีดังกล่าว ได้แซกซึมอยู่ในสินค้าต่าง ๆ มากมาย ตั้งแต่สินค้าที่ใช้ในครัวเรือนราคาไม่กี่บาท เช่น พัดลม ตู้เย็น เครื่องปรับอากาศ จนไปถึงสินค้าราคาแพง เช่น รถยนต์ เป็นต้น ต่อไปนี้จะได้กล่าวถึงเครื่องมือในการเรียนรู้และทำความเข้าใจกับการนำเทคโนโลยีไมโครคอนโทรเลอร์มาประยุกต์ใช้ในส่วนของ CONDITIONING ในกล่อง CDI ซึ่งมี สององค์ประกอบหลักด้วยกัน คือ

1 ส่วนของเครื่องมือในการพัฒนาโปรแกรม (Software) ประกอบด้วย
   - PIC C Compiler
   - Proteus ในการตรวจสอบการทำงานของโปรแกรม
2 ส่วนของการออกแบบวงจร (Hardware)
   - วงจรตรวจจับตำแหน่ง SENSOR
   - วงจรรองรับการพัฒนาโปรแกรมเพื่อปรับแต่งองศาการจุดระเบิด [BTDC]

การใช้งาน PIC C COMPILER

การใช้ PROGRAM PIC _ C COMPILER V 4.084 [11]

PIC_C COMPILER เป็นเครื่องมืออีกตัวหนึ่งที่นำมา ใช้เป็นตัวกลางในการแปลงคำสั่ง (Translators) ระหว่างภาษามนุษย์กับภาษาเครื่อง (Machine Language)ซึ่งในเวอร์ชั่น 4.084 ได้เพิ่มความสามารถและสิ่งอำนวยความสะดวกให้กับผู้ใช้งานอย่างมาก โดยสามารถใช้งานได้กับ PIC microcontroller ได้ตั้งแต่ PIC12 , PIC16 PIC18 , PIC24 ไปจนถึง DSPIC และในส่วนของ user interface ก็หน้าตาดีมีการแบ่งออกเป็นหมวด หมู่ ทำให้ง่ายต่อการใช้งานเป็นอย่างมาก หน้าต่างของโปรแกรมแสดงดังรูป



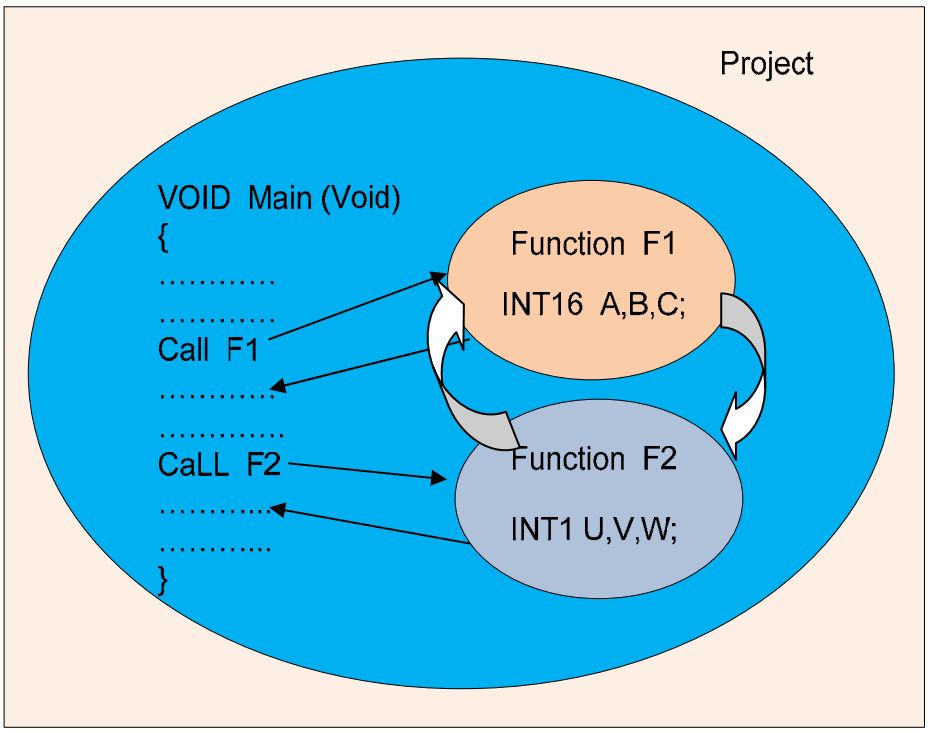
User Interface Of PIC C V4.08

จากรูป แสดงถึงการแบ่งกลุ่มเมนูเป็นกลุ่มดังนี้

1) TAB project เป็นเมนูเริ่มต้นในการสร้างงาน หน้าที่หลักของเมนู project คือนำ source file (*.c) ซึ่งในแต่ละโปรเจ็กอาจจะประกอบด้วยหลาย source file แต่ต้องมี 1 source file ที่มีฟังชั่น main() อยู่ใน project เนื่องจากงานที่สร้างขึ้นทั้งหมดจะถูกบัญชาการด้วยฟังชั่นmain()นั่นเอง
2) TAB Edit ใช้ในการแก้ไขปรับปรุงการเขียนโปรแกรมเป็นหลัก
3) TAB Search ใช้อำนวยความสะดวกในการค้นหาและ แทนที่คำใน project
4) ใช้ในการกำหนดคุณลักษณะต่าง ๆ ของโปรเจ็ก เช่น Editor Tollbar และ Project เป็นต้น
5) Tab Compile ใช้ในการตรวจสอบไวยากรณ์และแปรงโปรเจ็กเป็น HEX File

6) Tab view ใช้ในการเยี่ยมชมองค์ประกอบต่างต่าง ๆ ของโปรเจ็ก เช่น datasheet cpu , CPU Register การใช้ Preprocesser ต่าง ๆ เป็นต้น

ในการใช้งานโปรแกรม PIC _ C COMPILER จำเป็นต้องเข้าใจลักษณะโครงสร้างการทำงานของโปรแกรม รูปข้างล่าง แสดงโครงสร้างการทำงานของ PIC _ C COMPILER โดยฟังชั่น main() เป็นฟังชั่นหลักในการบริหารจัดการระบบงานใน project ทั้งหมด นั่นคือว่า sub function ต่าง ๆ ที่อยู่ใน project จะถูกเรียกใช้งานโดย function main() และระหว่าง sub function ก็สามารถเรียกใช้ซึ่งกันและกันได้ ในส่วนของตัวแปรถ้าประกาศไว้นอกฟังชั่นให้ถือว่าเป็นตัวแปรแบบ global (ยึดพื้นที่หน่วยความจำอย่างถาวร) ส่วนตัวแปรที่ประกาศในฟังชั่นจะเป็นตัวแปรแบบ private (ใช้พื้นที่หน่วยความจำเมื่อมีการเรียกใช้ฟังชั่นเท่านั้น)



Structure pic c compiler

นอกจากการทำความเข้าใจกับโครงสร้างของโปรแกรม PIC _ C COMPILER แล้วยังมีสิ่งต่าง ๆ ทีจำเป็นต้องทำความใจมีดังนี้ คือ

- Pre-Processor
- Data Definition
- Function Definition
- Operators
- ส่วนควบคุมการทำงานของโปรแกรม (Process Control)

- Built-in-function

- การสร้าง Project

- Pre-Processor

Pre-processor ใช้สำหรับเตรียมความพร้อมให้กับ compiler ให้ทำความรู้จักกับองค์ประกอบต่าง ๆ ของ Project อาธิเช่น cpu (#include<30F2010.h>) การกำหนดฟังชั่นการทำงานให้กับ cpu(#Fuse NOWDT, NOPUT, PROTECT)และกำหนด clock speed ให้กับ cpu(#use delay(clock=20000000)) เป็นต้น

- Data Definition

การพัฒนาโปรแกรมในเรื่องใด ๆ ก็ตามการประกาศตัวแปรปืนสิ่งที่หลีกเลี่ยงได้ยาก เนื่องจากงานด้านพัฒนาโปรแกรมเป็นการนำอินพุทข้อมูลที่มีการเปลี่ยนแปรงตามสภาวะต่างมาประเมินผลเพื่อให้ได้ค่าเอาท์พุทตามที่เราต้องการนั่นเอง ชนิดและวิธีการประกาศตัวแปร

ชนิดและวิธีการประใช้ตัวแปร

type	Number bit	value	signint
int	1	0 to 1	n/a
Int8	8	0 to 255	-128 to 127
Int16	16	0 to 65535	-32768 to 32767
Int32	32	0 to 4294967295	-2147483648 to 2147483647
Float32	32	-1.5×10^{45} to 3.4×10^{38}	

Constance Data

A =123; ฐานสิบ

B = O123; ฐานแปด

C = 0X0F; ฐานสิหก

D = 0B00110011; ฐานสอง

- Function Definition

รูปแบบการสร้างและการเรียกใช้ sub function มีดังนี้

- ประกาศฟังชั่น prototype ให้ compiler ทราบทุกครั้งก่อนการเรียกใช้ฟังชั่น
- กำหนดรูปแบบการรับและส่งค่าของฟังชั่นให้ถูกต้อง

ตัวอย่างโปรแกรมที่ประกอบด้วยฟังชั่นย่อยแบบต่าง ๆ

```
#include <30F2010.h>

void f1(void);

void f2(int8 data);

int16 f3(int8 a,int8 b);

void main(){

  int16 x;

   while(true)

  {

   f1();    f2(10);    f3(10,20);

  }

}

void f1(void)

{  output_b(0xff); }

 void f2(int8 data)

{ output_b(data);}

int16 f3(int8 a,int8 b)

{  return a+b;}
```

Function

Call Function

- Operators

Operator ในโปรแกรม PIC C Compiler แบ่งออกได้เป็น 4 กลุ่มด้วยกันคือ

1 กลุ่มที่ใช้ดำเนินการทางคณิตศาสตร์(Arithmetic Operator)

เป็น Operator ใช้ในการคำนวณค่าทางคณิตศาสตร์

ตัวดำเนินการทางคณิตศาสตร์

+	Addition Operator
+=	Addition assignment operator, x += y ,is the same as x = x + y
++	Increment
--	Decrement
-=	Subtraction assignment operator, x -= y ,is the same as x = x - y
/	Division Operator
/=	Division assignment operator, x /= y ,is the same as x = x / y
*	Multiplication Operator
*=	Multiplication assignment operator, x *= y ,is the same as x = x * y

2 กลุ่มที่ใช้ดำเนินการทางลอจิก(Logic Operator)

เป็น Operator ที่ใช้ดำเนินการต่าง ๆ ทางลอจิก ดังแสดงในตารางที่ 2.4

Operator ที่ใช้ดำเนินการต่าง ๆ ทางลอจิก

&	Bitwise
&=	Bitwise assignment operator X &= y, is the same as x = x & y
^	Bitwise exclusive Opertor
|	Bitwise inclusive OR Operator
|=	Bitwise inclusive OR assignment Operator
&&	Logic AND Operator
||	Logic OR Operator
!	Logic Negation Operator
~	One Complement Operator

กลุ่มที่ใช้ดำเนินการเปรียบเทียบข้อมูล(Relational Operator)

ใช้สำหรับเปรียบเทียบข้อมูลซึ่งผลที่ได้จะเป็นค่า จริง(True) หรือ เท็จ(False)

กลุ่มที่ใช้ดำเนินการเปรียบเทียบข้อมูล(Relational Operator)

<==	Left shif assignment operator, x < == y , , is the same as x = x< y
<	Less than Operator
<<	Left Shif Operator
>>	Right Shif Operator
>>=	Right Shif assignment operator,x >>= y, is the same as x = x>> y
==	Equality
!=	Inequality

กลุ่มที่ใช้ดำเนินการ การเข้าถึงข้อมูลในหน่วยความจำข้อมูล(Memory Excess Operator)

เป็น Operator ที่ใช้ดำเนินการในการเข้าถึงข้อมูลในหน่วยความจำ

กลุ่มที่ใช้ดำเนินการ การเข้าถึงข้อมูลในหน่วยความจำ

*	Indirection Operator
&	Address Operator

- ส่วนควบคุมการทำงานของโปรแกรม (Process Control)

ส่วนที่ใช้ควบคุมลำดับการทำงานของโปรแกรมประกอบด้วยกลุ่มคำสั่ง 3 กลุ่มด้วยกันคือ

กลุ่มคำสั่ง Branching , Iteration และ Condition

กลุ่มคำสั่ง Branching

เป็นกลุ่มคำสั่งที่ใช้ในการกระโดดข้ามการทำงานจากคำสั่งหนึ่งไปยังอีกคำสั่งหนึ่ง

กลุ่มคำสั่ง Branching

คำสั่ง	รูปแบบการใช้
Goto Label;	Goto loop;

Label : statement;	Loop :a++;
break	Break;
continue	Continue;

กลุ่มคำสั่ง Iteration

เป็นคำสั่งที่มีลักษณะการทำซ้ำ บ่อยครั้งที่โปรแกรมจำเป็นต้องทำงานที่ซ้ำ ๆ กันภายใต้เงื่อนไขใด ๆ

กลุ่มคำสั่ง Iteration

คำสั่ง	รูปแบบการใช้
While(Expr){statement;}	While(x<10){x++;}
Do {statement;}while(Expr)	Do{x++;}while(x<10)
For(expr1;expr2;expr3){statment}	For(x=0;x<10;x++){printf("x");}

กลุ่มคำสั่ง Condition

ใช้กำหนดให้กลุ่มคำสั่งทำงานภายใต้เงื่อนไขที่กำหนดเป็นจริง

กลุ่มคำสั่ง Condition

คำสั่ง	รูปแบบการใช้
If(expr){statment}	If(x==10){x=0;}
If(expr) {statment} Else if(expr) {statement} Else{statment}	If(x==1) {y=1;} Else if(x==2) {y=3;} Else{y=4;}
Switch(expr)	Switch(x)

Case expr1:{statement} Case expr2:{statement} Default {statement}	Case 0:{y=2;break;} Case 1:{y=4;break;} Default {y=0;break;}

- Built-in-function

Built in function นับได้ว่าเป็นสิ่งที่ช่วยอำนวยความสะดวกกับผู้ใช้โปรแกรม pic c compiler เป็นอย่างมากโดยที่ผู้ใช้แค่ทำความเข้าใจกับการส่งผ่านค่าตัวแปรและ การ return ค่าของฟังชั่น เท่านั้น ตัวอย่างที่ใช้บ่อย ๆ เช่น Delay_ms(100) Delay_us(200) เป็นฟังชั้นที่ใช้สำหรับการหน่วงเวลาซึ่งมีความจำเป็นต้องใช้เกือบทุกโปรแกรม

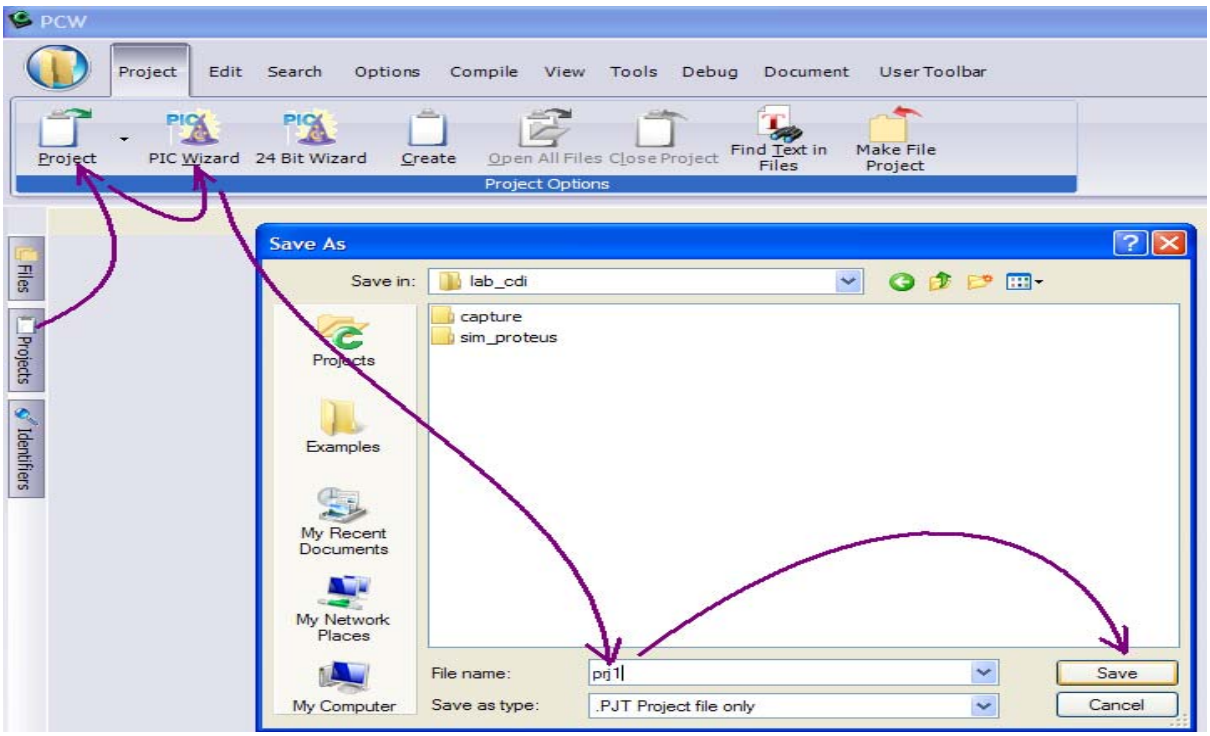
การสร้าง Project

การสร้าง Project มีทางเลือกได้ 2 ทางดังนี้คือ

1) ใช้ Project Wizard
2) ใช้ Manual create Project

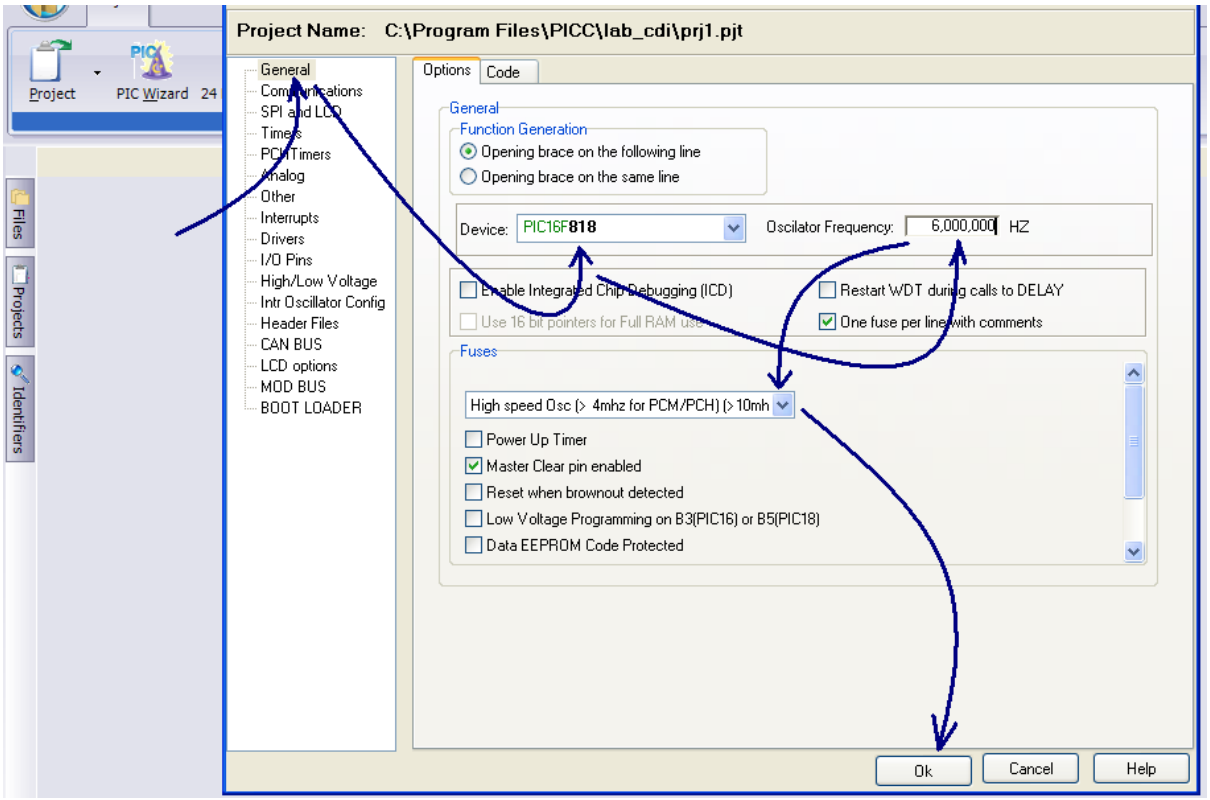
Project Wizard

นับได้ว่ามีประโยชน์ต่อผู้เริ่มต้นใช้โปรแกรมเป็นอย่างมาก เนื่องจาก pic microcontroller มี register ค่อนข้างมาก ทำให้ยากต่อการ config ในการนำไปใช้งานในแต่ละอย่าง และ project wizard จะช่วยนำทางมือใหม่สู่เป้าหมายได้ ขั้นตอนการใช้ Project wizard แสดงดังรูป



เริ่ม project wizard Step 1

- เลือก Tab Project → Pic Wizard สำหรับ(8 bit databus) or 24 bit Wizard(16 bit data bus) → project name → save



Step 2 register config

Pic c compiler จะแบ่งกลุ่มส่วนที่ผู้ใช้ต้องกำหนดค่าเริ่มต้นให้กับ compilerไว้ในlist box ทางด้านซ้ายมือ เช่น ส่วนของ general,Interrupt,Driver,I/O Pin เป็นต้นซึ่งการกำหนดค่าในแต่ละส่วนนั้นจำเป็นต้องทำความเข้าใจโครงสร้างและฟังชั่นการทำงานอย่างละเอียด หลังจากที่เราเข้าไปกำหนดส่วนต่าง ๆ เรียบร้อยแล้ว click ok โปรแกรมจะสร้าง source code ส่วนหนึ่งมาใส่ไว้ในฟังชั่นหลักเพื่อกำหนดการทำงานเริ่มต้นให้กับ CPU เพื่อพร้อมที่จะใช้งานตามความต้องการต่อไป

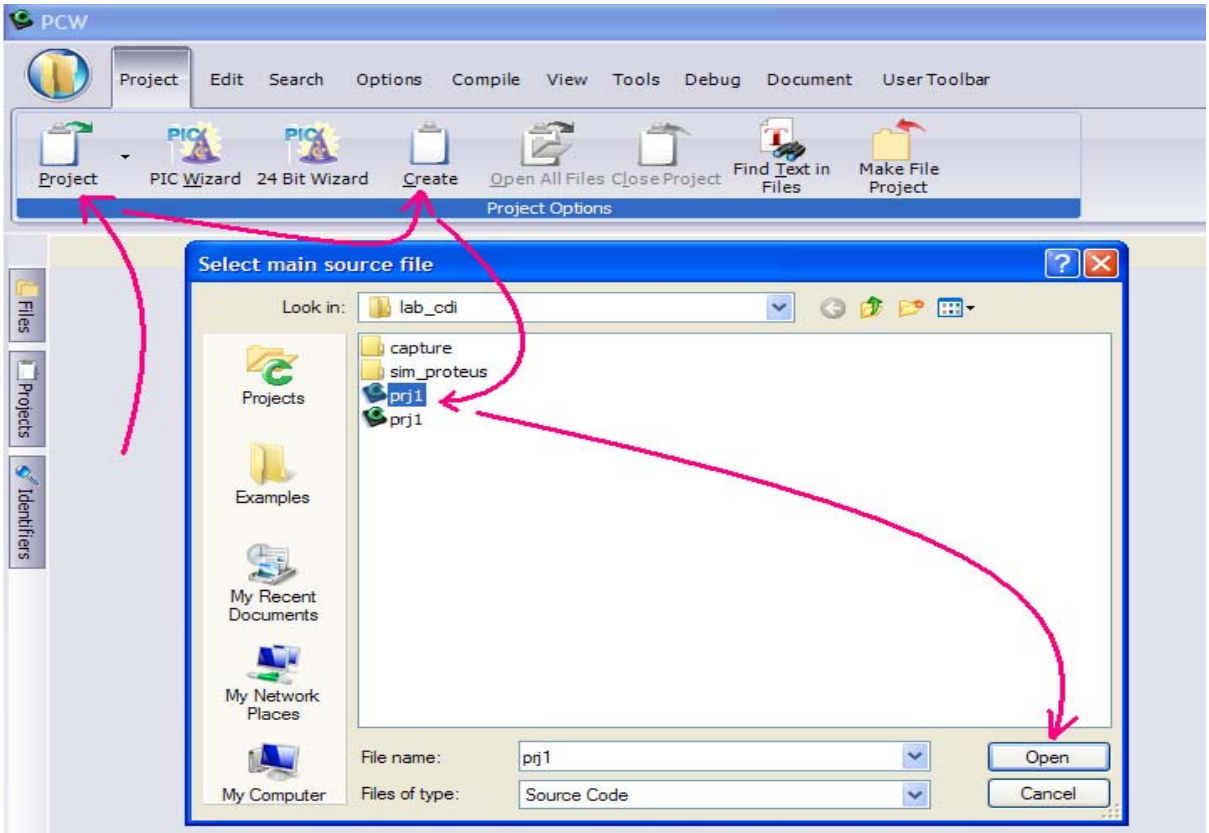
ตัวอย่าง code ได้จากการ wizard

```
#include "C:\Program Files\PICC\lab_cdi\prj1.h"

void main()

{

  setup_adc_ports(NO_ANALOGS);

  setup_adc(ADC_OFF);
```

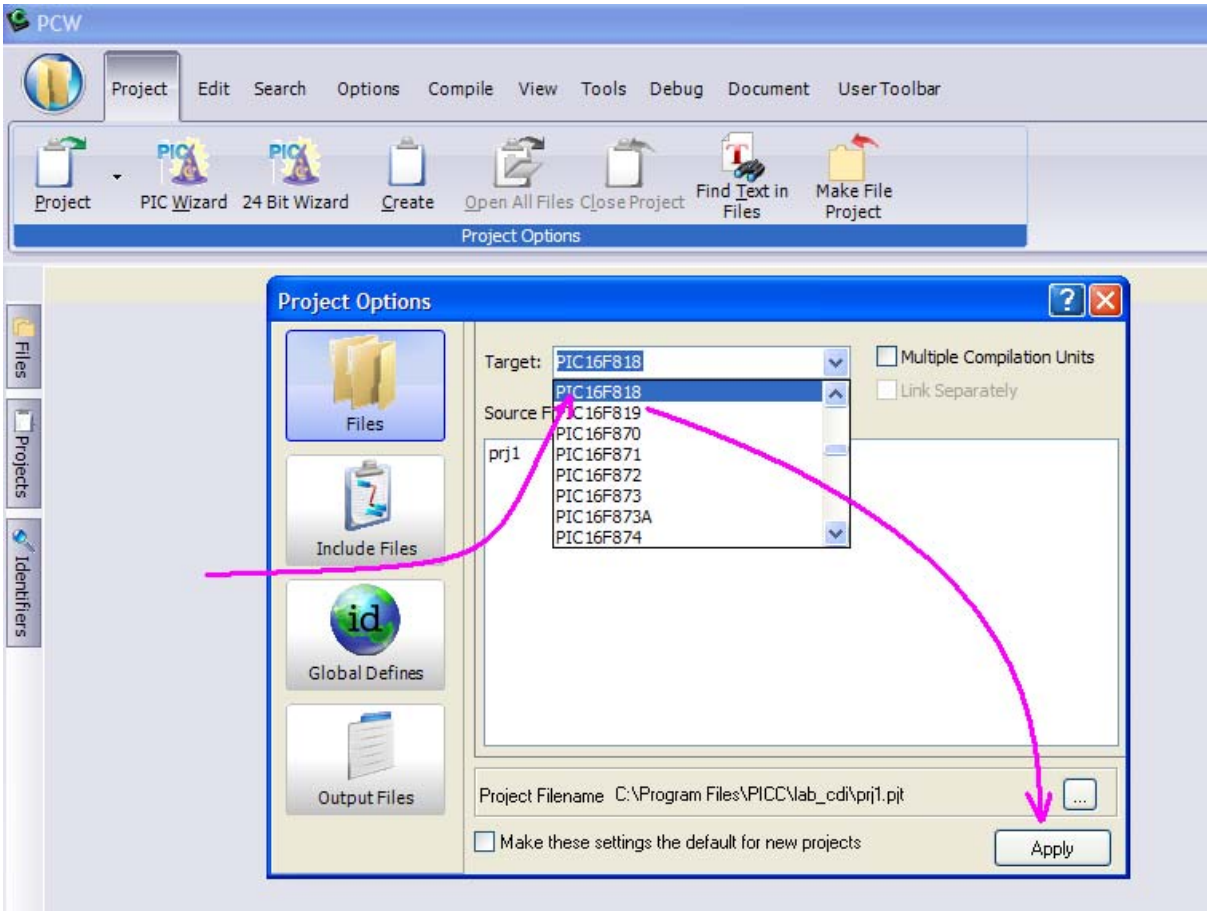
```
  setup_spi(SPI_SS_DISABLED);

  setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);

  setup_timer_1(T1_DISABLED);

  setup_timer_2(T2_DISABLED,0,1);

//Setup_Oscillator parameter not selected from Intr Oscillator Config tab

  // TODO: USER CODE!!

}
```

Manual Create Project

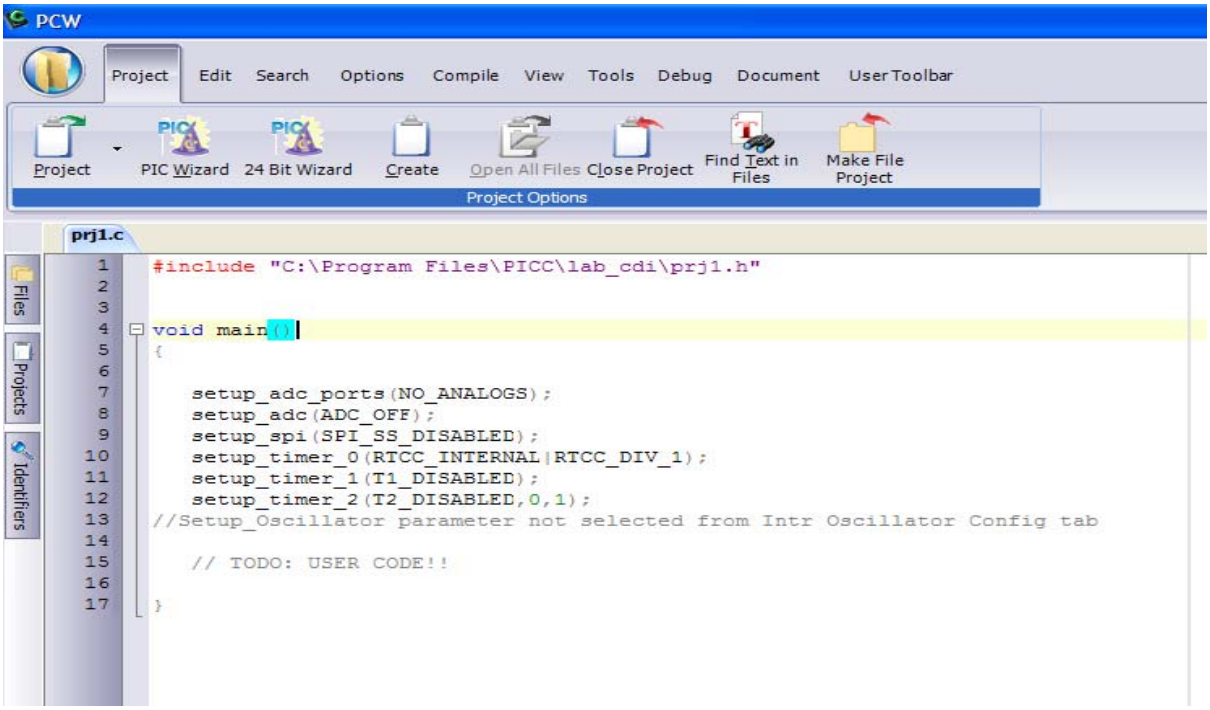
เป็นวิธีการนำ source file ที่ถูกสร้างขึ้นมาแล้ว มาประกอบรวมเป็น project อย่างไรก็ตามจะต้องมี function main() อยู่ในsource file ด้วยเนื่องจากกฎเกณฑ์ของภาษาซีที่ project ใด ๆ ก็ตามจะต้องใช้ function main()ในการบริหารจัดการกับ project ทั้งหมด ขั้นตอนการ manual create project แสดงให้เห็นดังรูป



step 1 เส้นทางสู่ การเลือก source file (* . c)



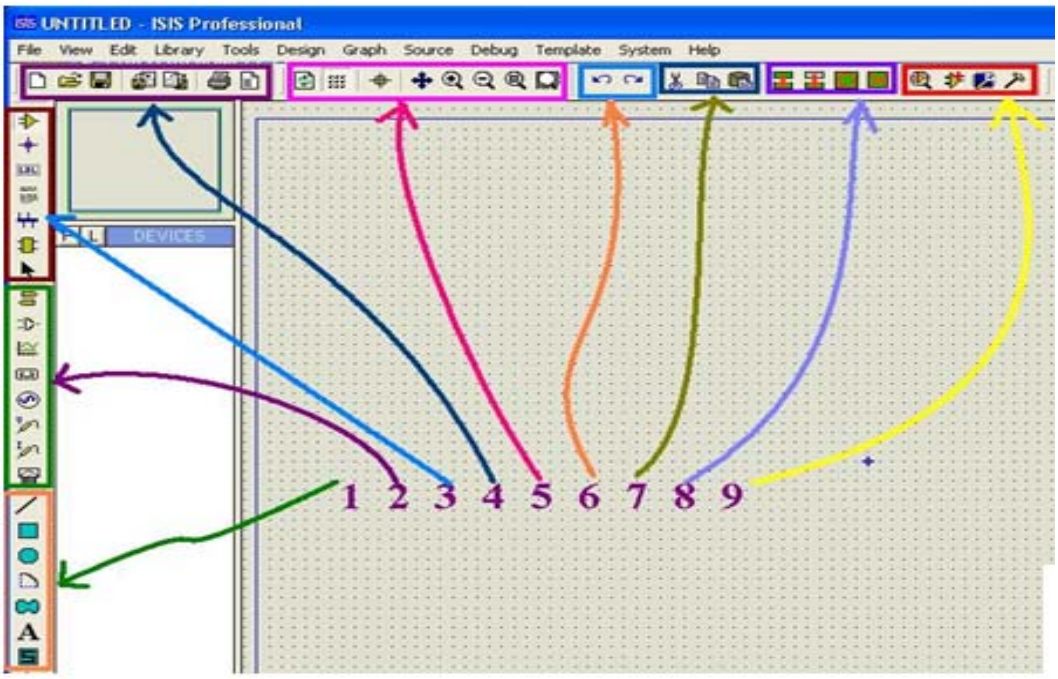
เลือก Device [PIC16F818]



การ manual create project เสร็จสมบูรณ์

พื้นฐานโดยรวมการใช้โปรแกรม **Proteus**

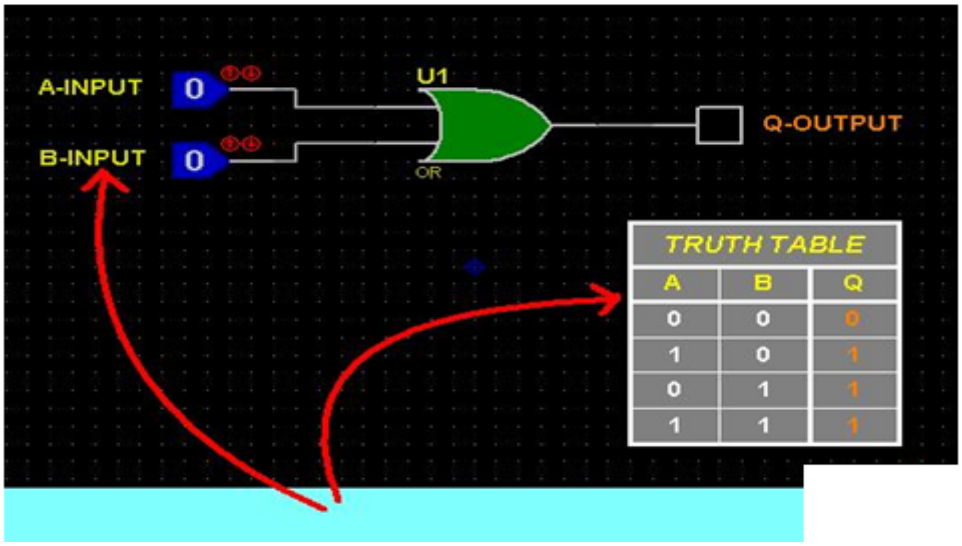
พื้นฐานการใช้งาน โปรแกรม PROTEUS สำหรับการออกแบบระบบ



รูปที่ 1 หน้าตาและสภาพแวดล้อมโดยทั่วไปของโปรแกรม PROTEUS

จากรูปที่ 1 เป็นการจัดแบ่งกลุ่มของ tool bar เพื่อความสะดวกในการใช้งาน ซึ่งพอจะสรุปการใช้งานแต่ละกลุ่มดังนี้

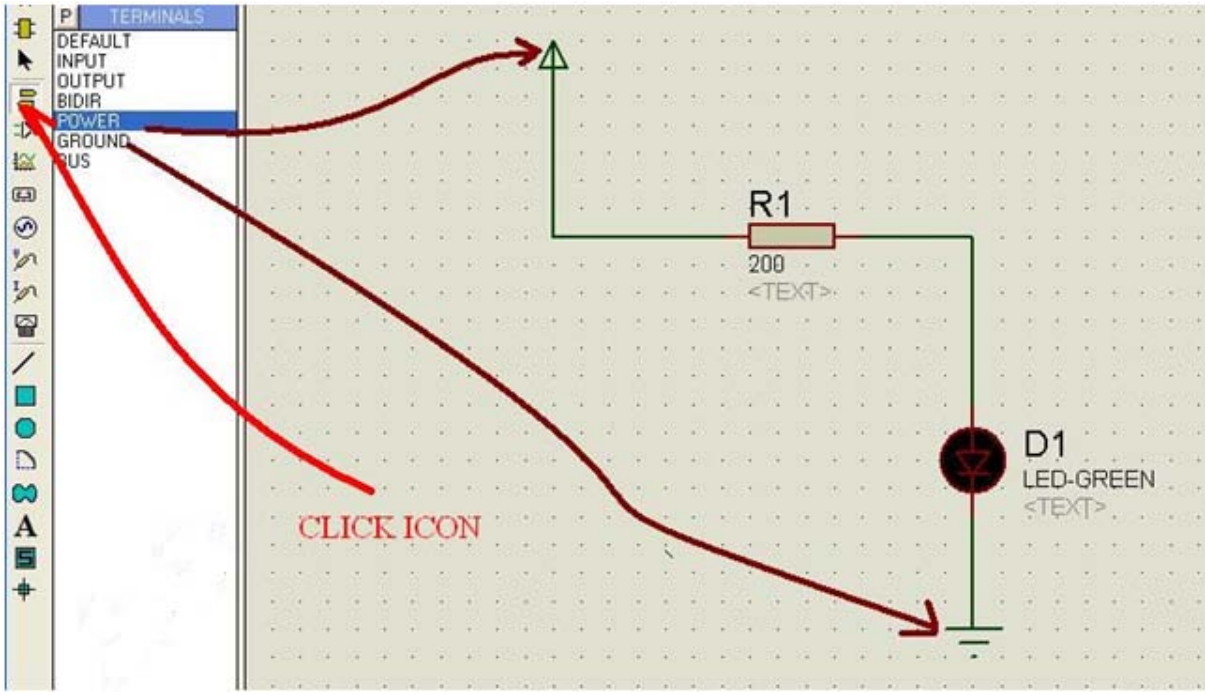
<u>กลุ่มที่ 1</u> เป็นกลุ่มเครื่องมือที่ใช้งานทางด้าน การสร้าง อุปกรณ์เพิ่มเติม หรือ งานทางด้าน ตกแต่ง หรือ อธิบายวาจร ดังตัวอย่างดังรูป



ตัวอย่างการใช้เครื่องมือกลุ่มที่ 1

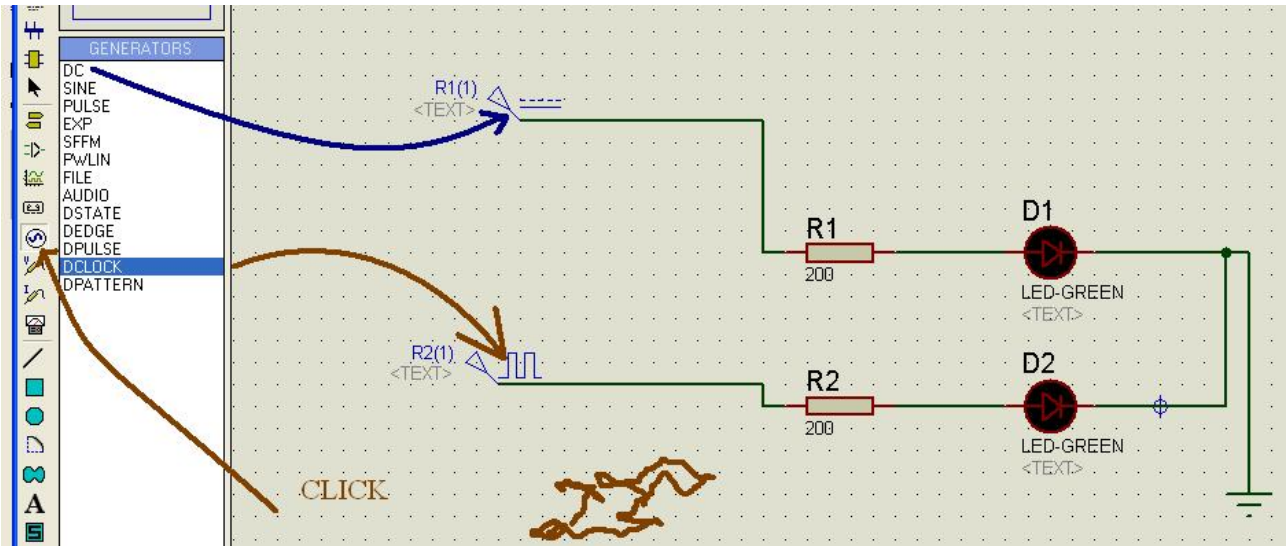
กลุ่มที่ 2 แหล่ง กำเหนิด เครื่องมือตรวจจับสัญญาน รวมไปถึง กราวด์ อินพุท และ เอาท์พุท

ตัวอย่าง 2.1 การใส่ power และ ground ให้กับวงจร



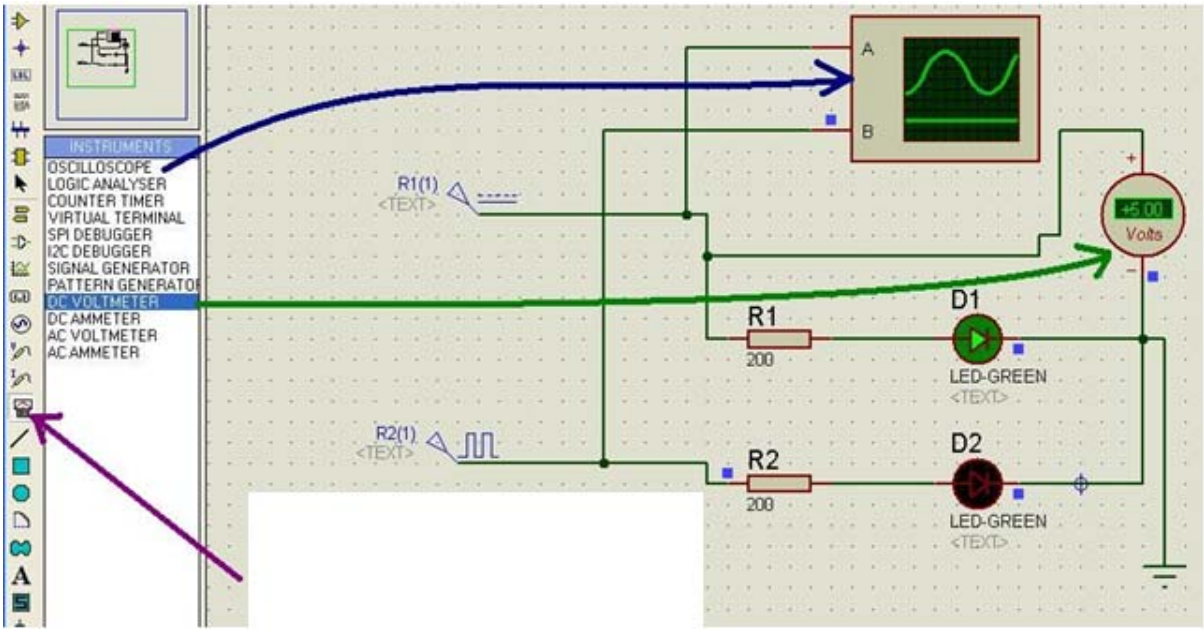
ตำแหน่ง icon ที่อยู่ของ จุดต่อ เพาเวอร์ และ กราวด์

ตัวอย่าง 2.2 การใส่ อินพุทัญญาน puls และ สัณญาน DC



แสดง tool bar แหล่งกำเหนิดสัญญาน

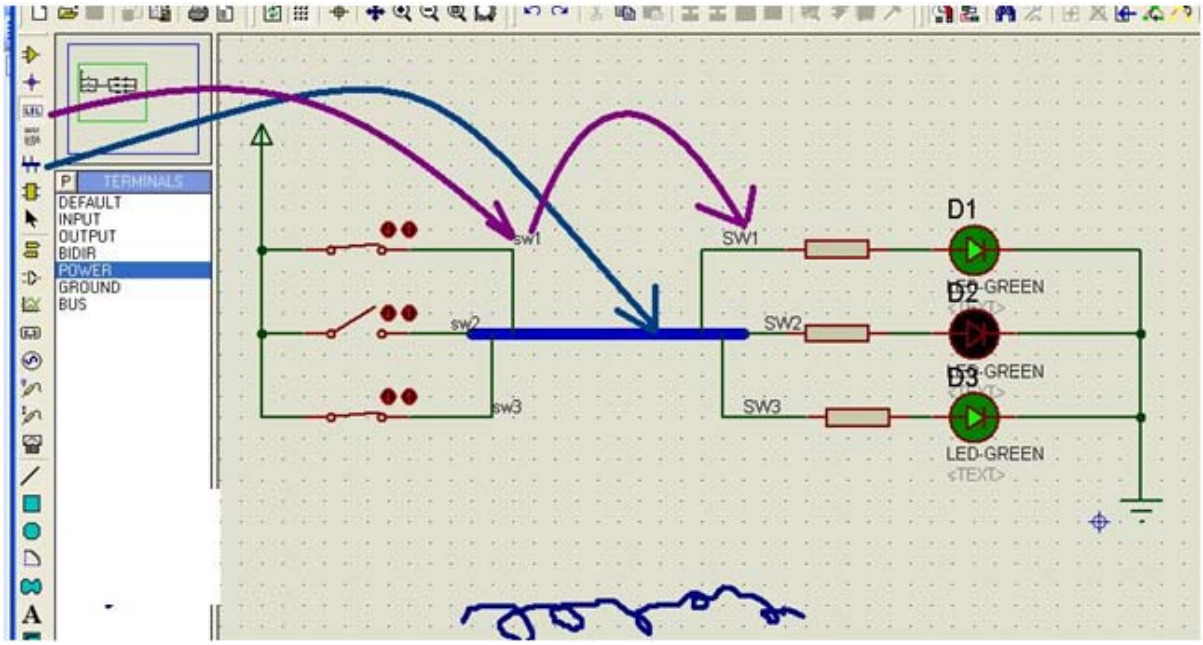
ตัวอย่าง 2.3 ตำแหน่งจัดเก็บเครื่องมือวัด



รูปที่ 5 แสดง tool bar ที่

3 **กลุ่มที่ใช้สำหรับ** จัดหาอุปกรณ์ รวมไปถึงการต่อวงจร

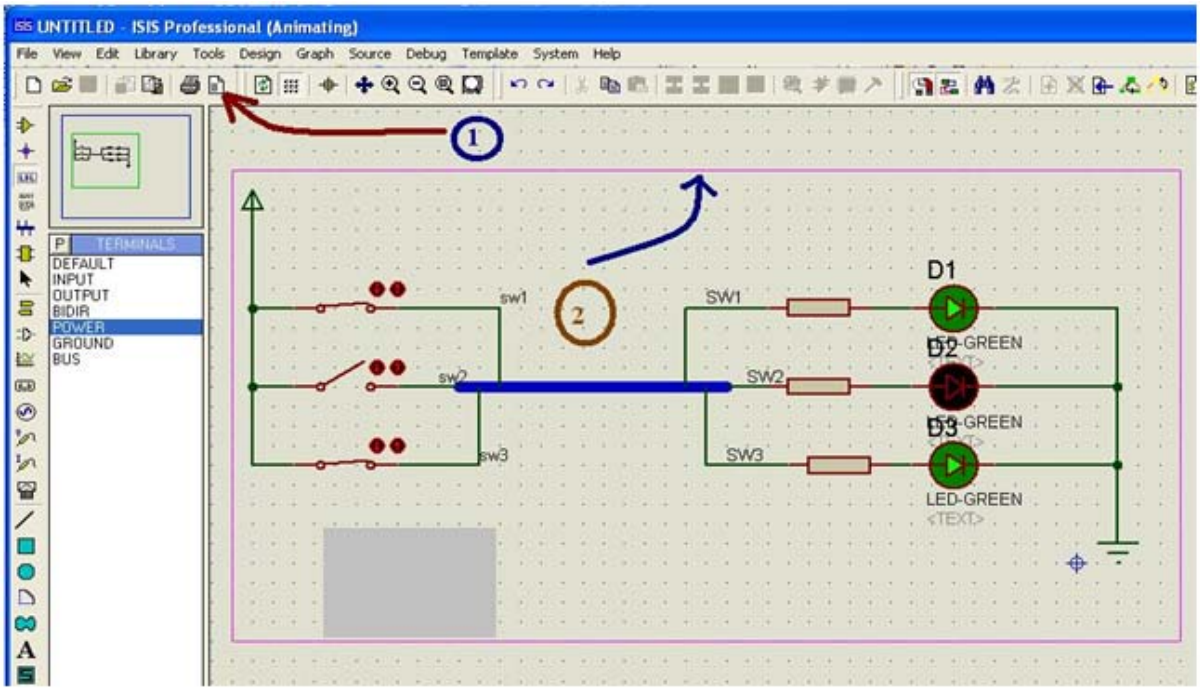
ตัวอย่าง 3.1 การต่อวงจรโดยใช้ bus bar และการใช้ label



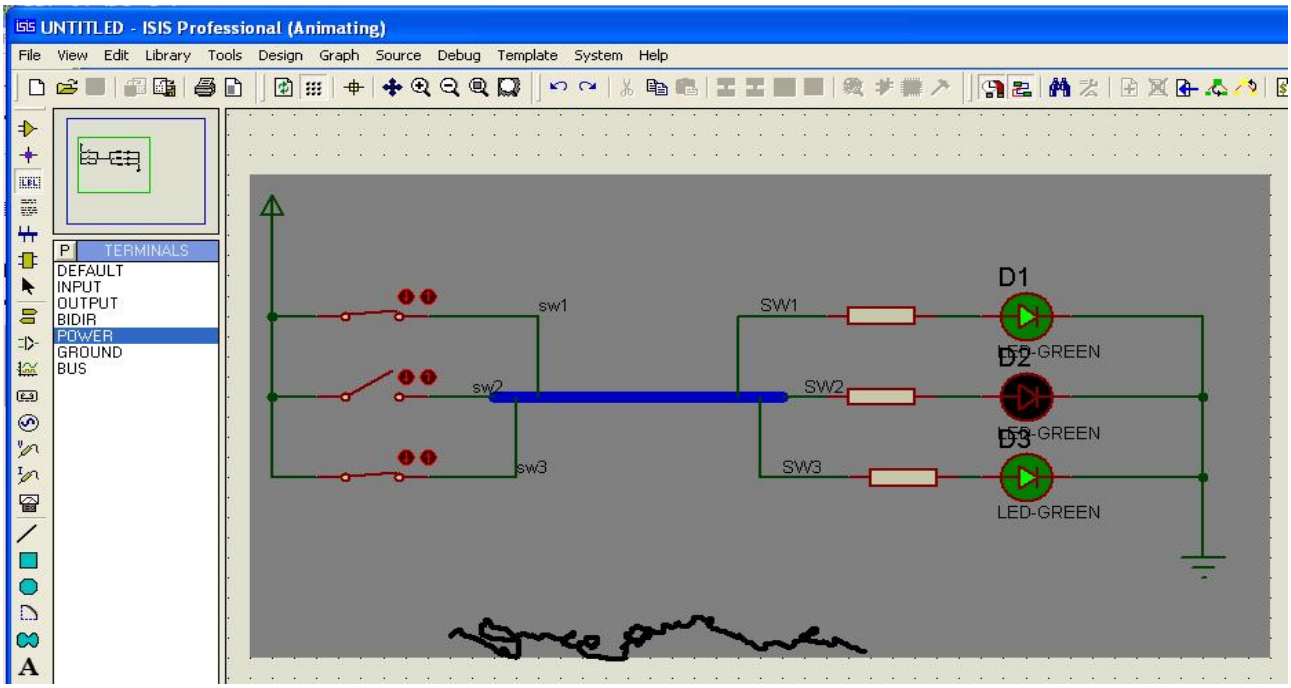
รูปที่ 6 การต่อวงจรโดยใช้ bus bar + label

4 กลุ่ม tool bar ที่ใช้สำหรับการ print และ save file

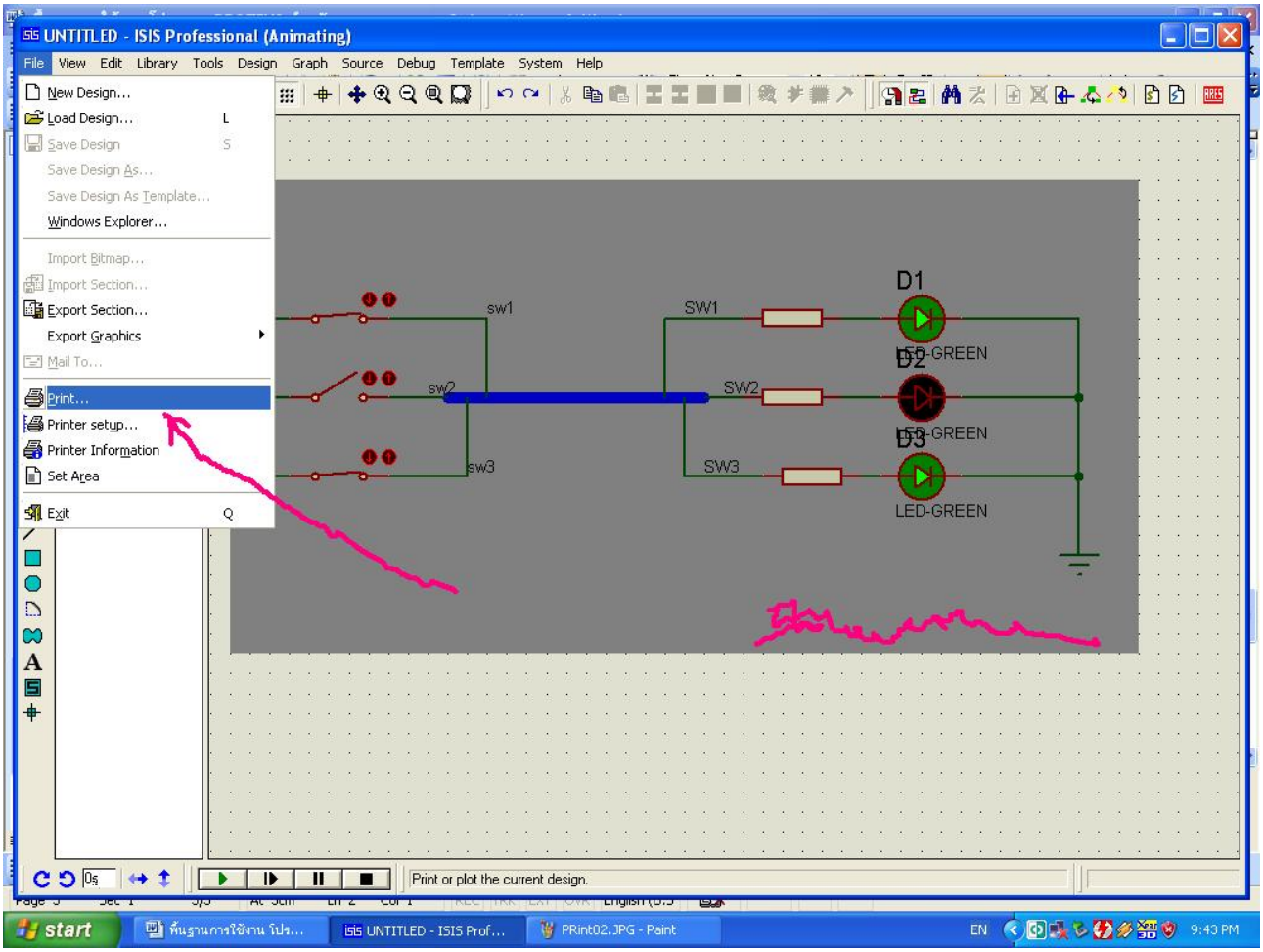
ตัวอย่างที่ 4.1 การระบุขอบเขตการ print วงจรออกทางเครื่องพิมพ์



รูปที่ 7 การระบุขอบเขตการ print 1



รูปที่ 7 การระบุขอบเขตการ print 2



รูปที่ 7 การระบุขอบเขตการ print 3

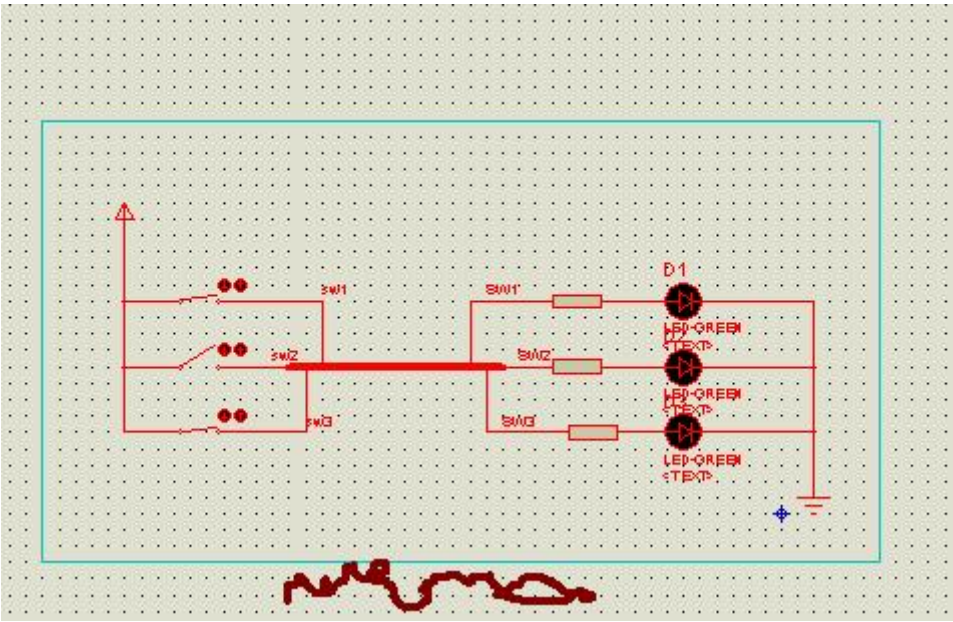
5 ตัวช่วยสำหรับการดูวงจร เช่น ย่อ ขยาย เป็นต้น

6 ตัวช่วยสำหรับการมองย้อนหลังหรือข้างหน้ากรณีเกิดข้อสงสัยหรือการทำงานผิดพลาด

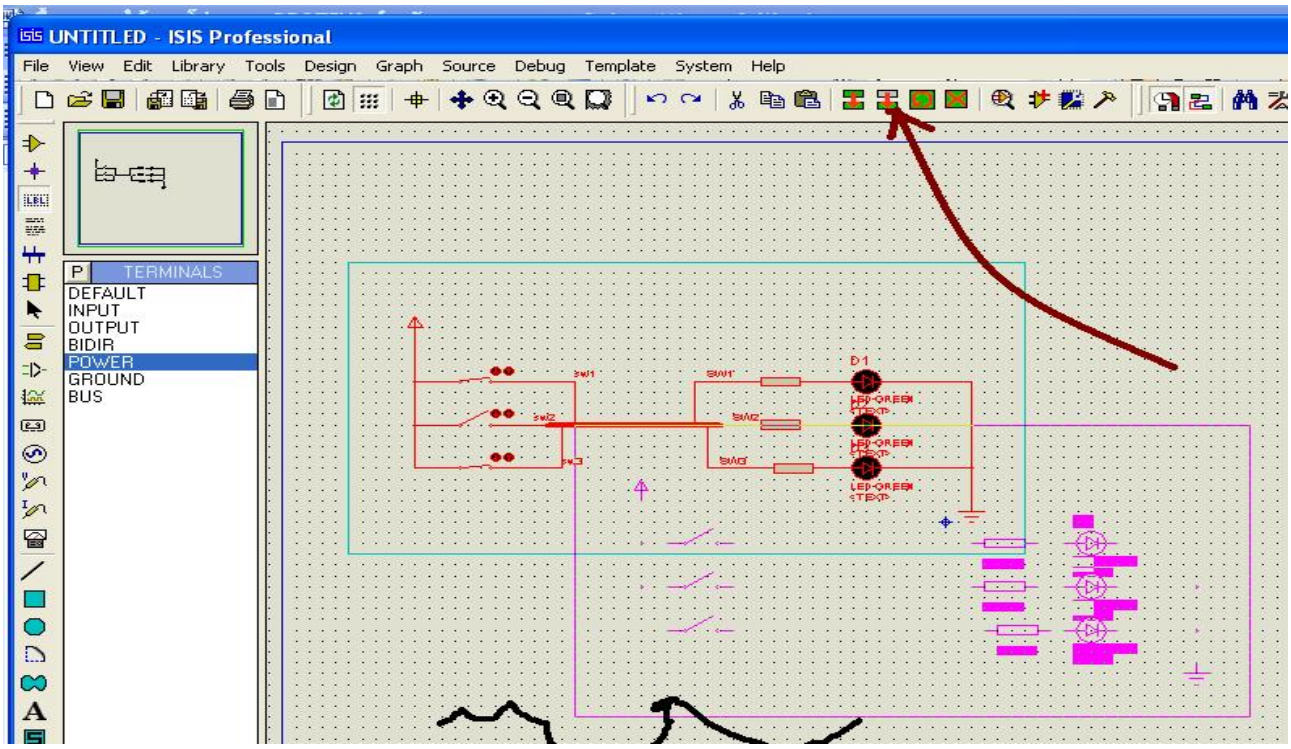
7 ตัวช่วยสำหรับการลอกแบบ วงจร ใช้เช่นเดียวกับกรณีการพิมพ์งาน

8 ตัวช่วยสำหรับการการกระทำกับจรที่มีลักษณะเป็นกลุ่ม เช่น ย้ายตำแหน่ง หรือ ทำสำเนาวงจร

ตัวอย่าง 8.1 การย้ายวงจร

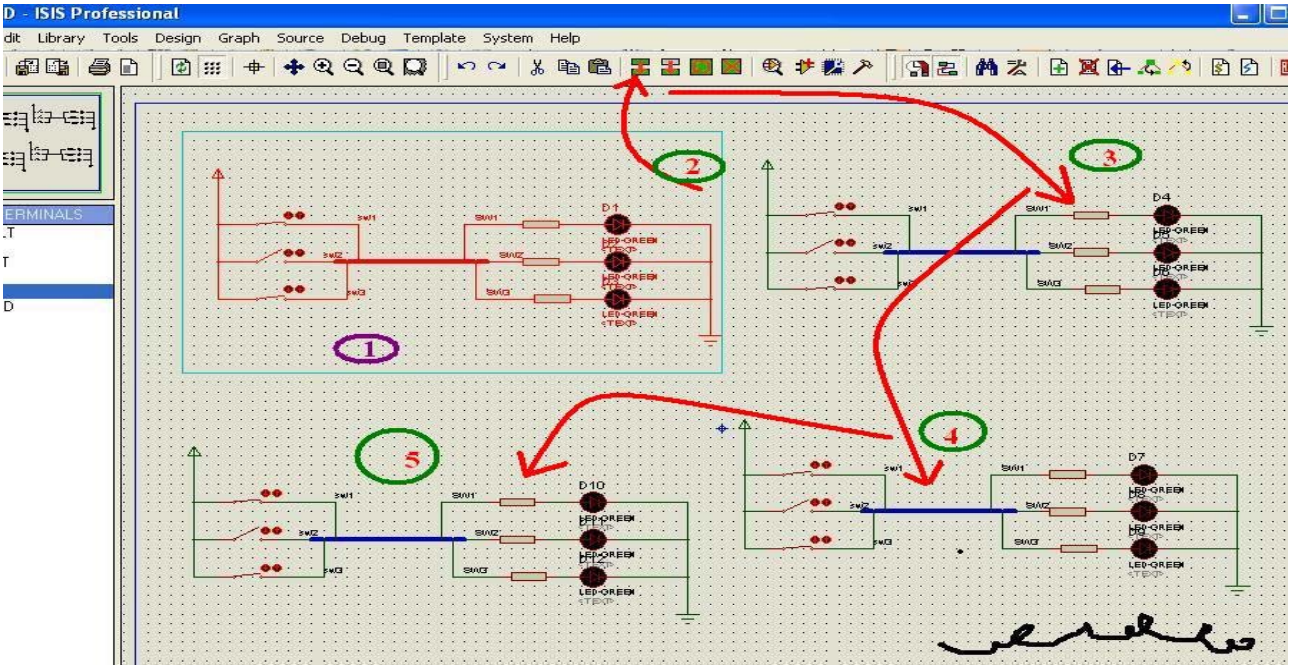


รูปที่ 8 การย้ายวงจร step 1 (กดเมาส์ปุ่มขวาค้างแล้วตีกรอบ)



รูปที่ 8 การย้ายวงจร step 2 (click tool bar move tag move object)

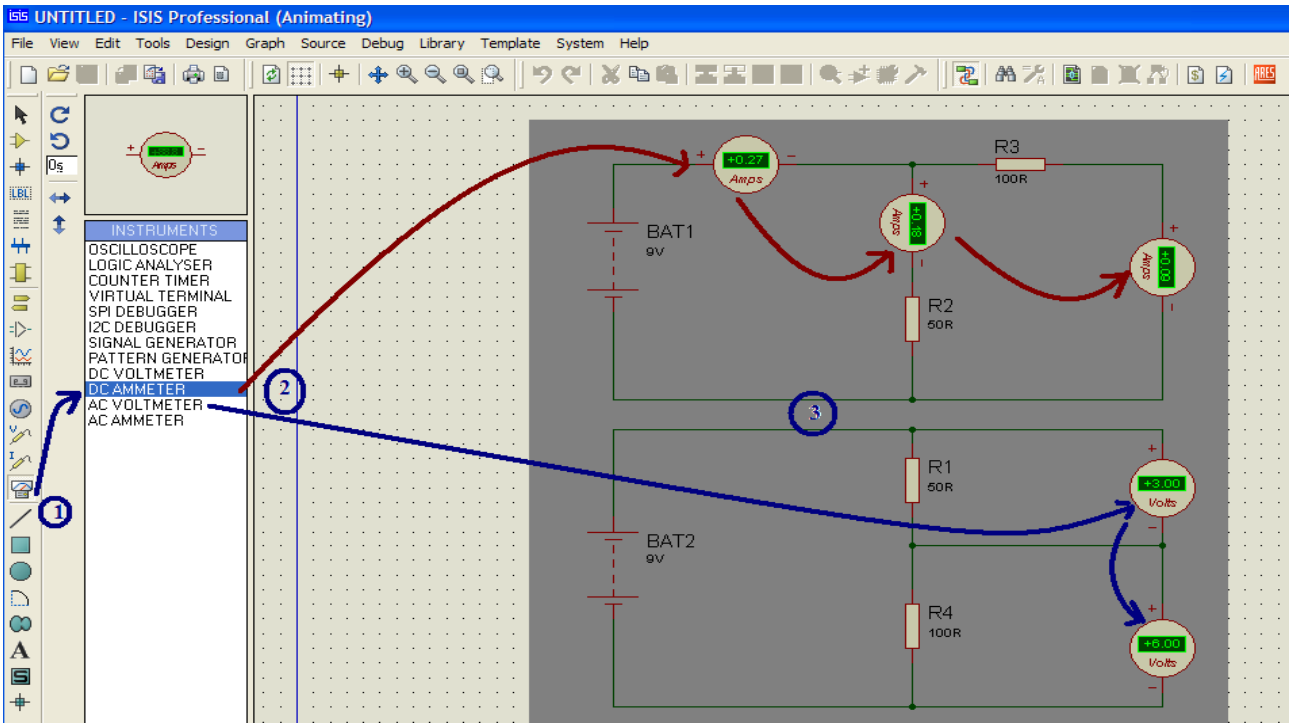
ตัวอย่าง 8.2 การทำสำเนาวงจร



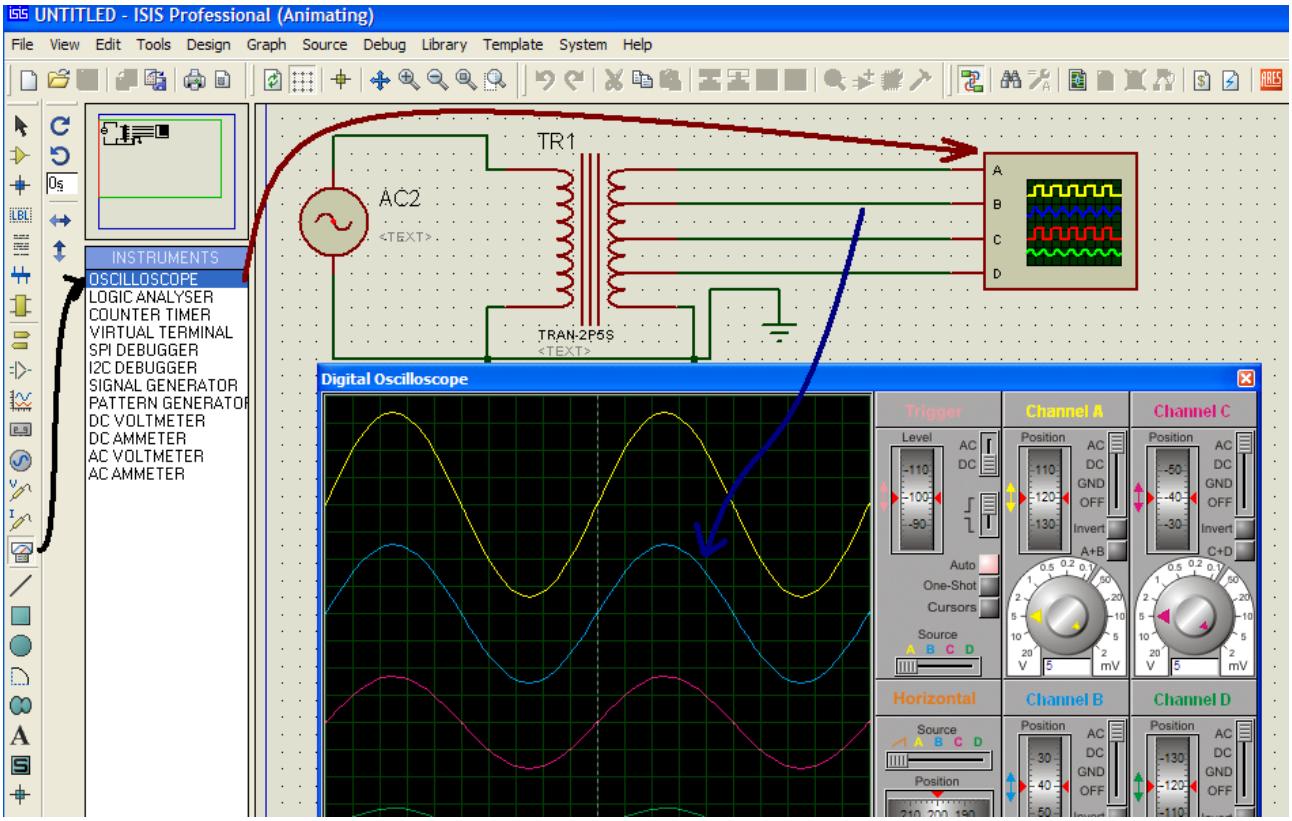
รูปที่ 9 การทำสำเนาวงจร

ตัวอย่าง 8.3 การใช้เครื่องมือตรวจสอบและวิเคราะห์วงจร

8.3.1 การวิเคราะห์วงจรโดยใช้ Amp Meter และ Volt Meter

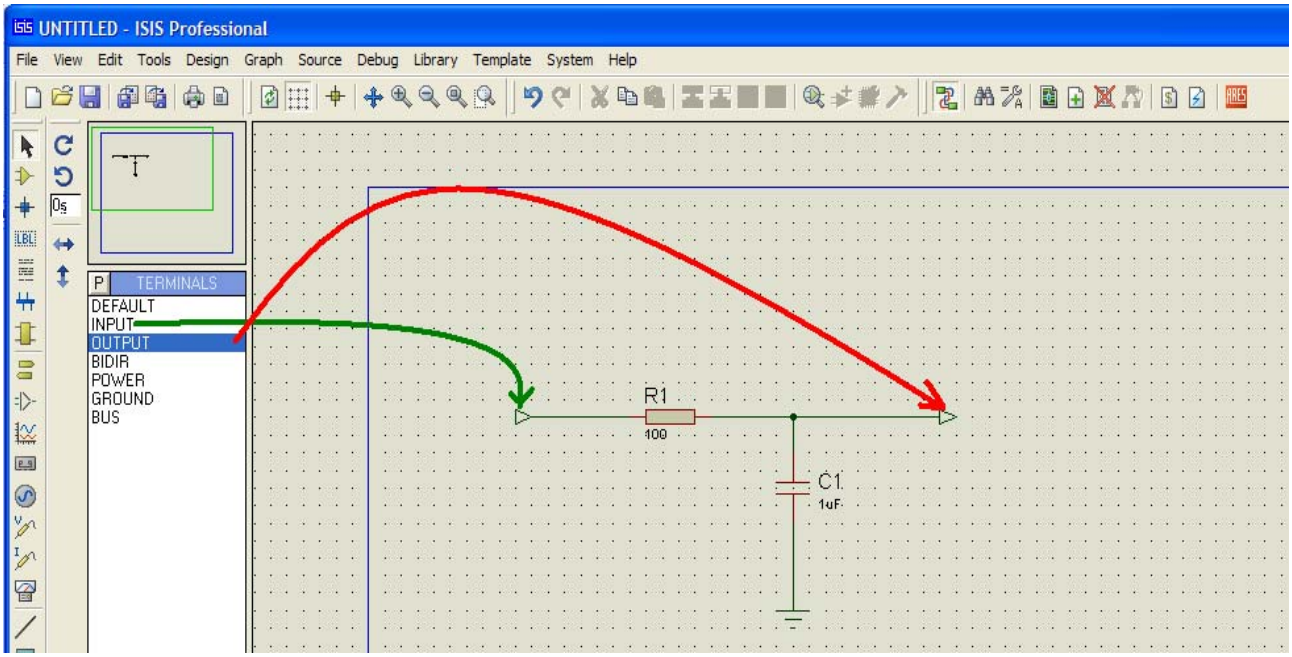


8.3.2 การวิเคราะวงจรโดยใช้ **Oscilloscope**

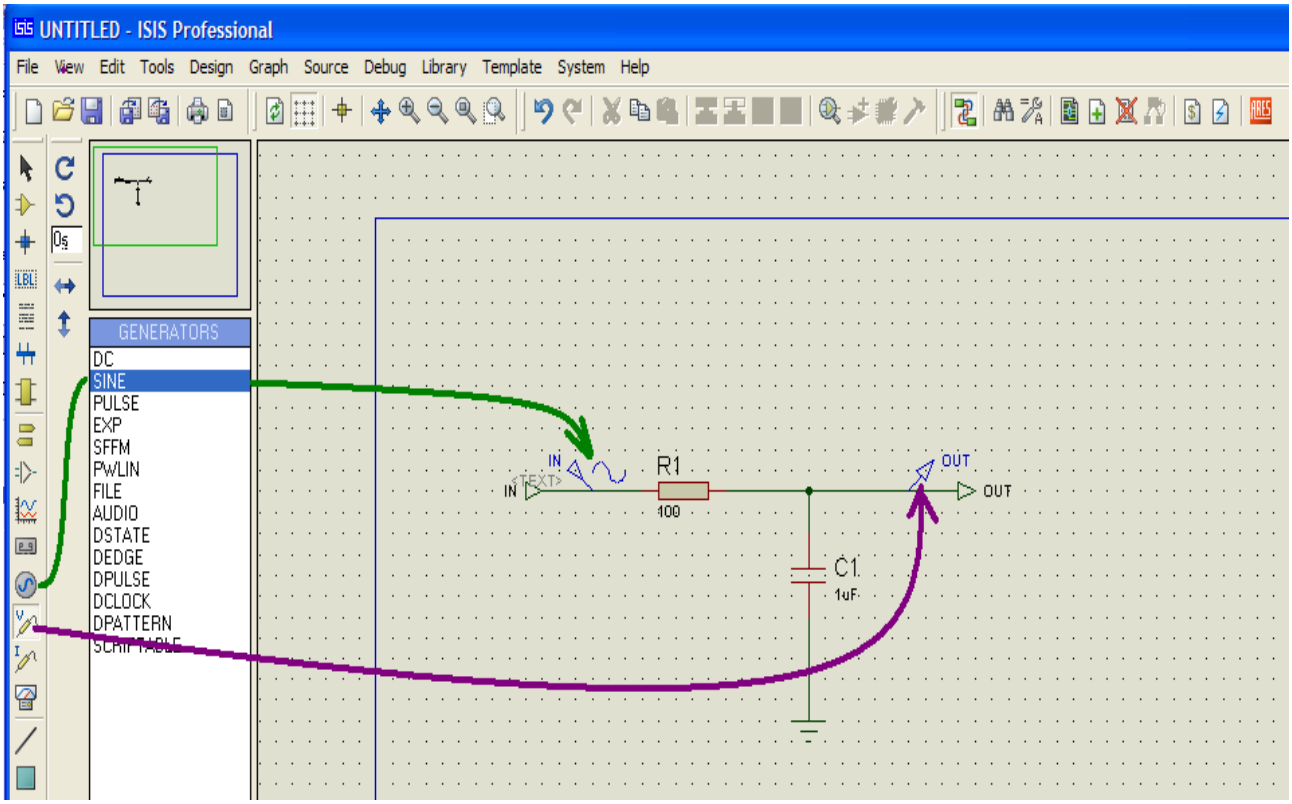


8.3.3 การวิเคราะห์วงจรโดยการใช้กราฟ

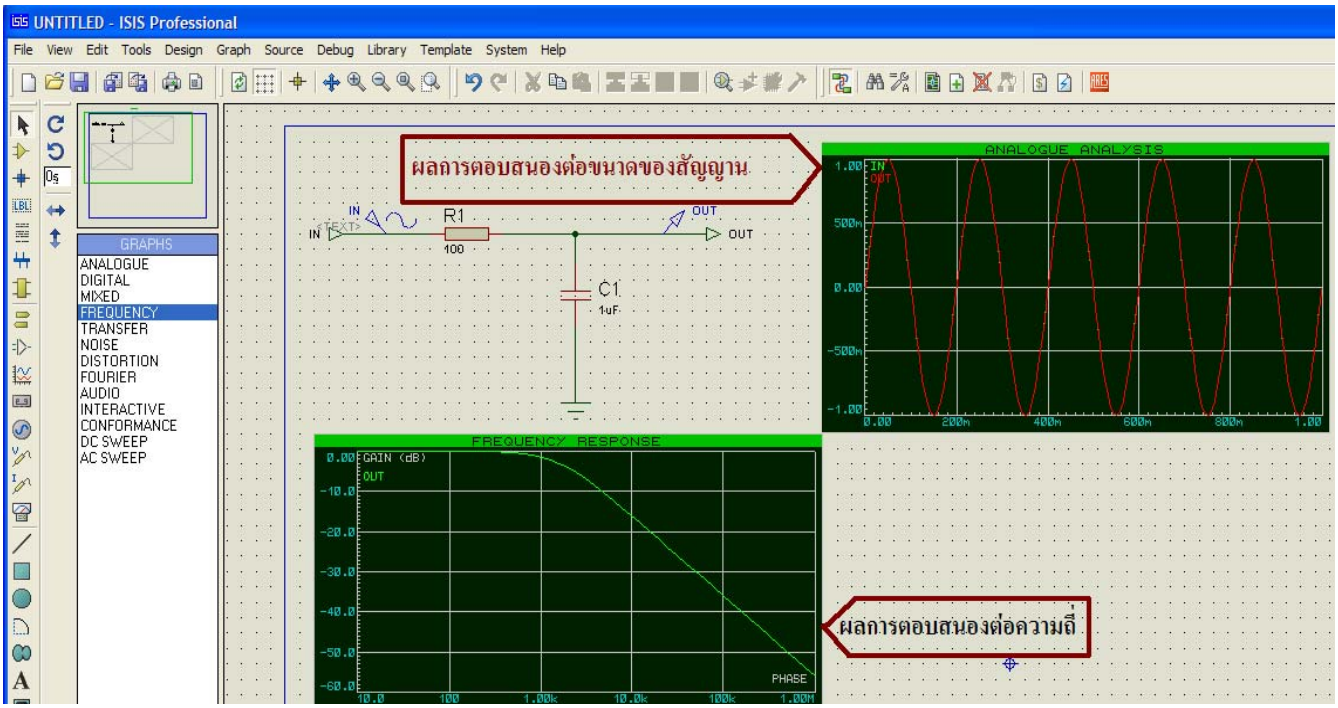
8.3.3.1 กำหนด อินพุท/เอาท์พุท



8.3.3.2 กำหนดสัญญานทางด้านอินพุท และ ตัววัดสัญญานทางด้านเอาท์พุท



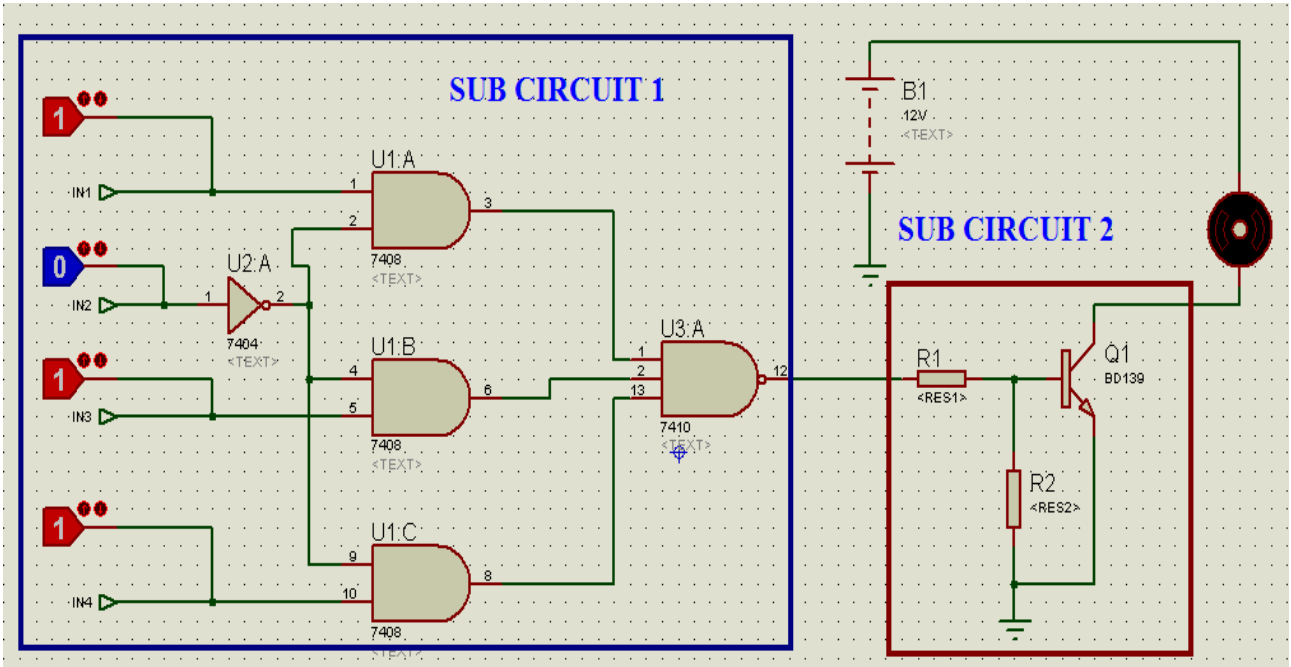
8.3.3.3 เลือกชนิดของกราฟที่จะนำมาวิเคราะห์สัญญาน



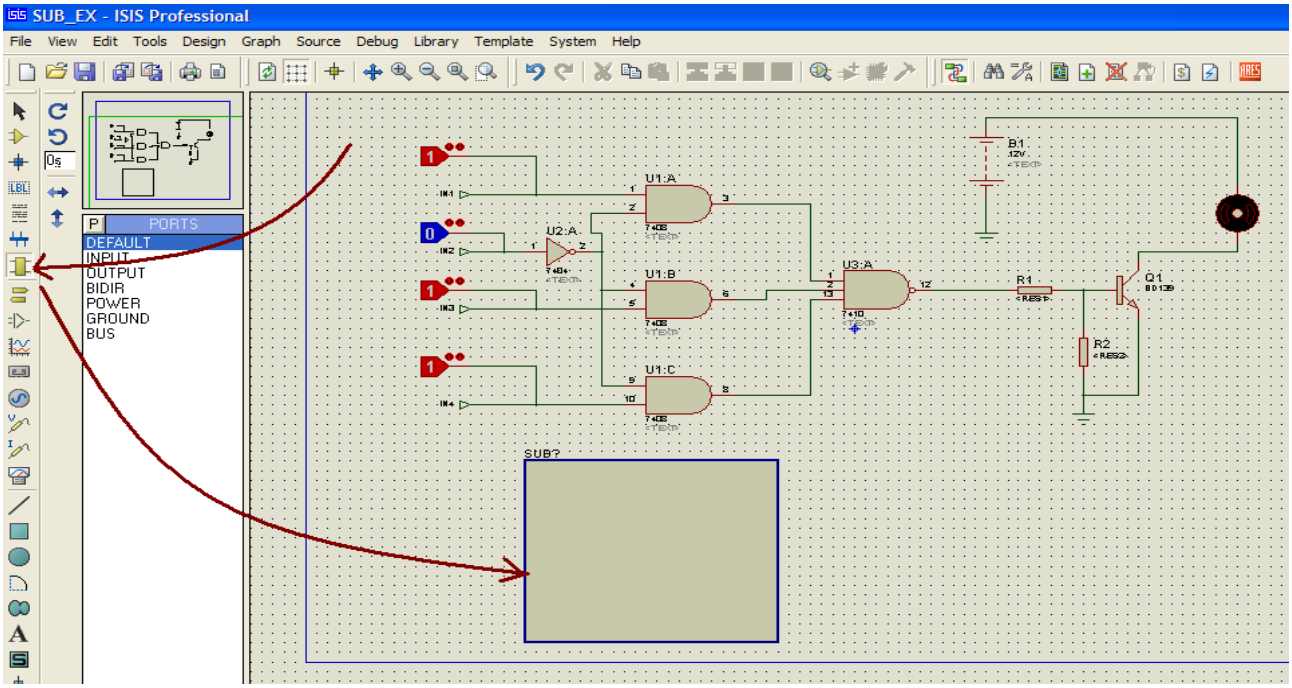
9 การสร้างวงจรย่อย (sub circuit)

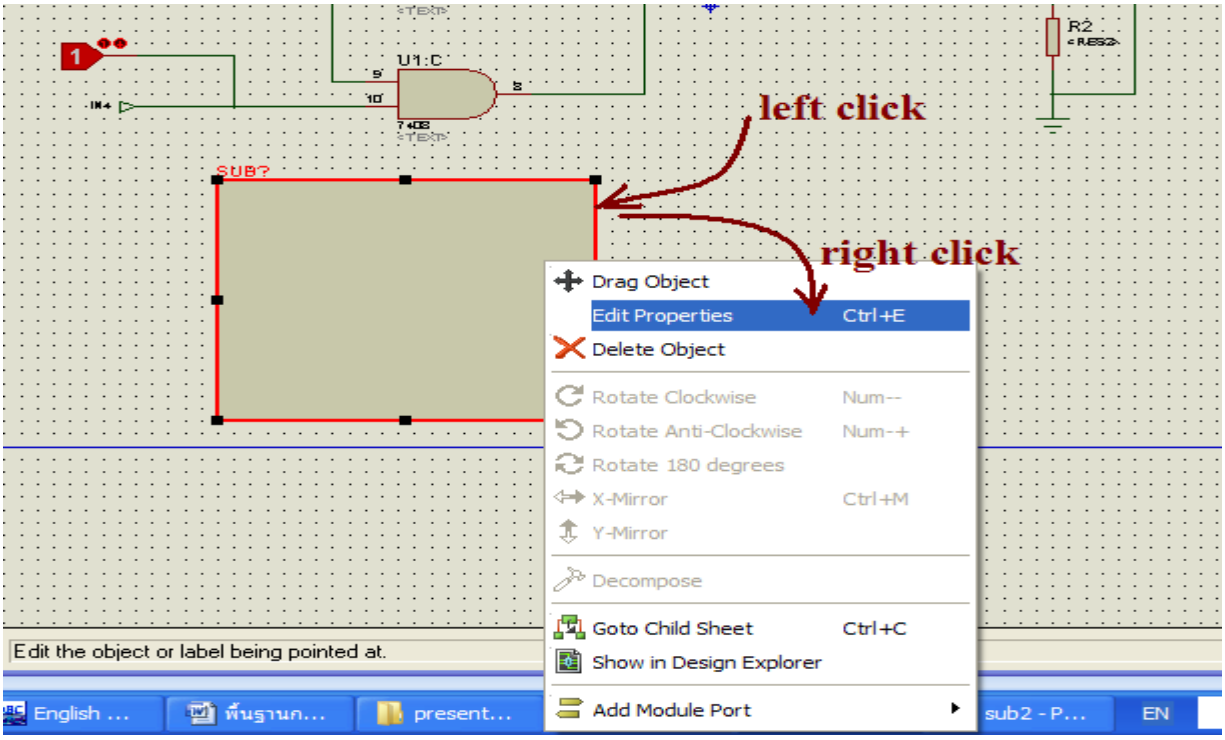
การสร้างวงจรย่อยจะช่วยให้นักออกแบบสะดวกในการตรวจสอบและวิเคราะห์วงจรโดยจะแบ่งวงจรที่มีขนาดใหญ่หรือมีความซับซ้อนออกเป็นส่วนย่อย ๆ ทำให้สามารถตรวจสอบวงจรในแต่ละส่วนได้ง่ายขึ้นนั่นเอง

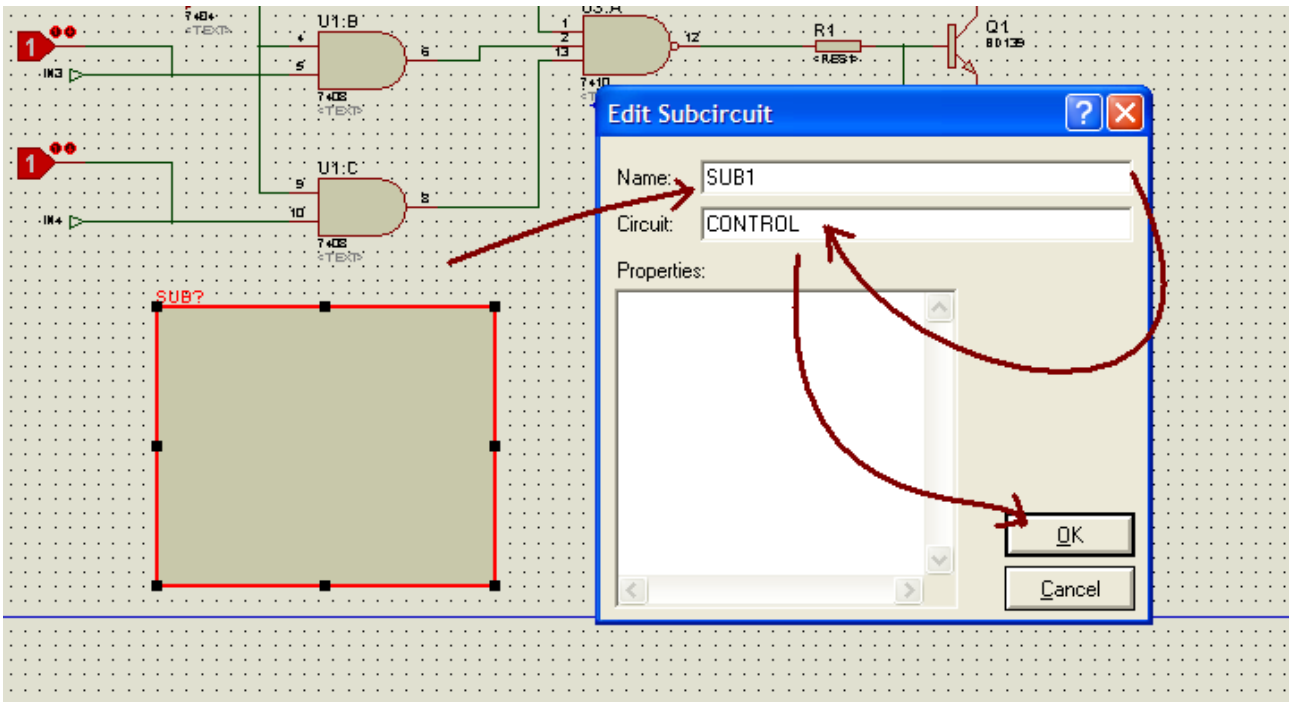
9.1 ตัวอย่างการแบ่งวงจรตามรูปข้างล่างออกเป็นสองวงจรย่อย

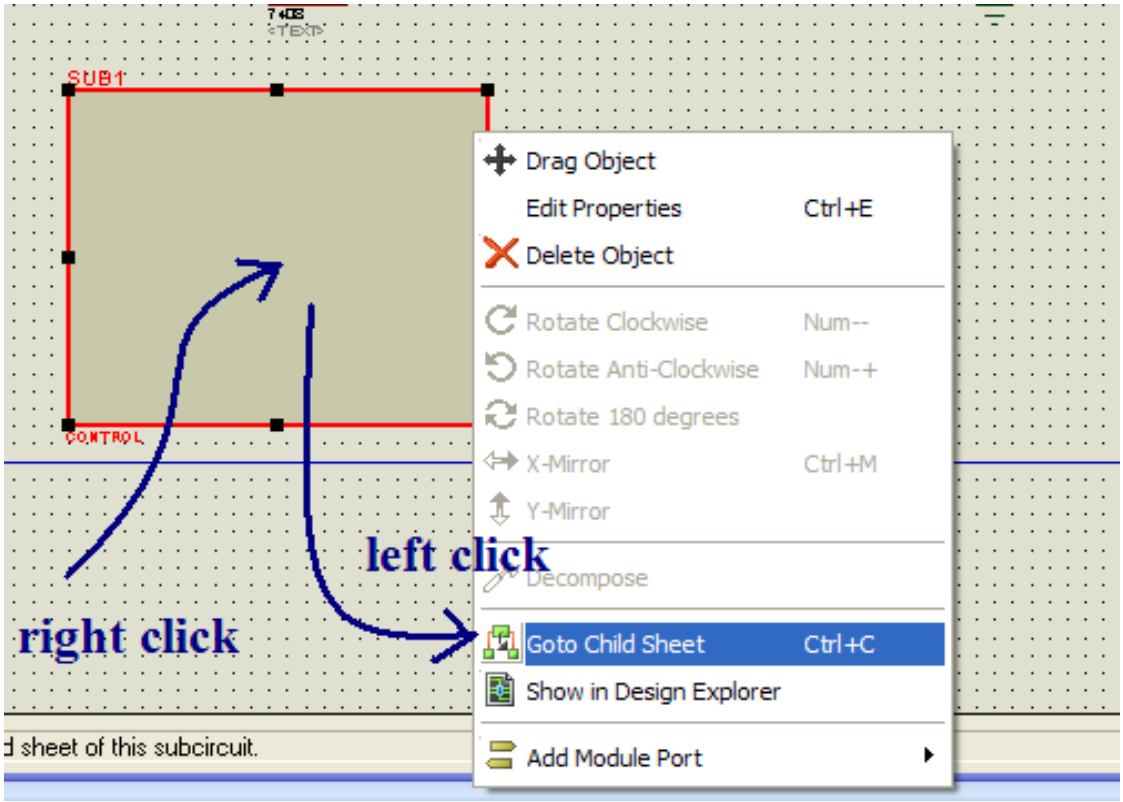


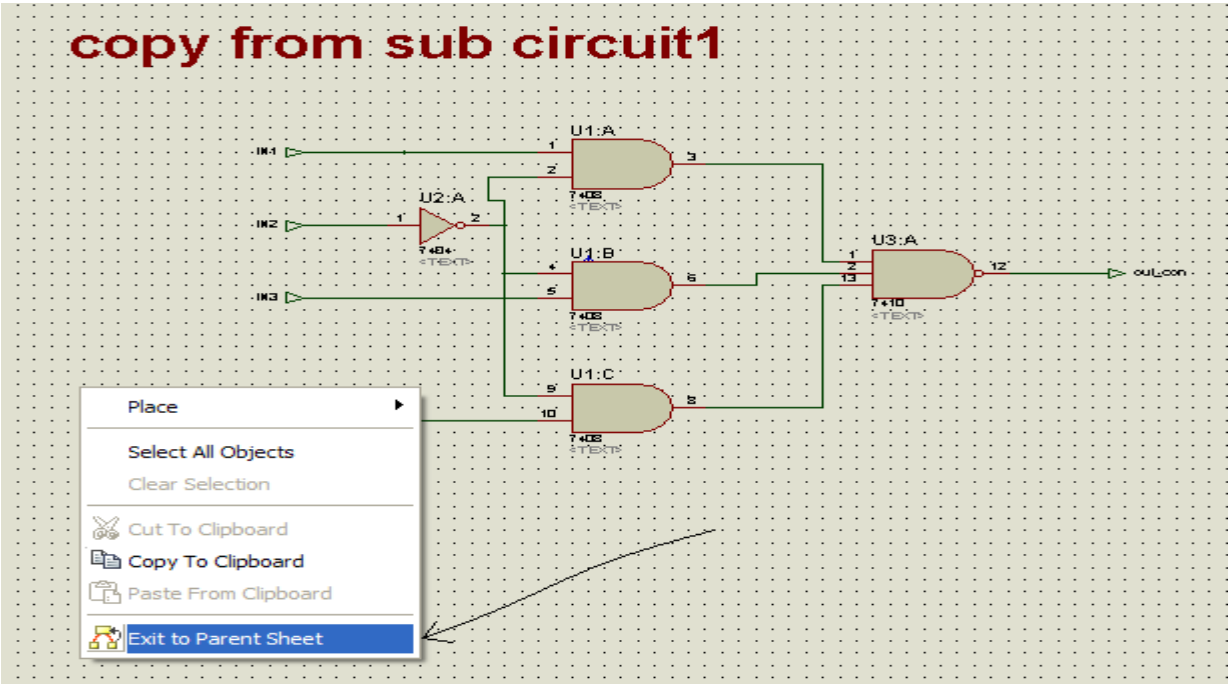
9.1.1 สร้าง sub circuit 1

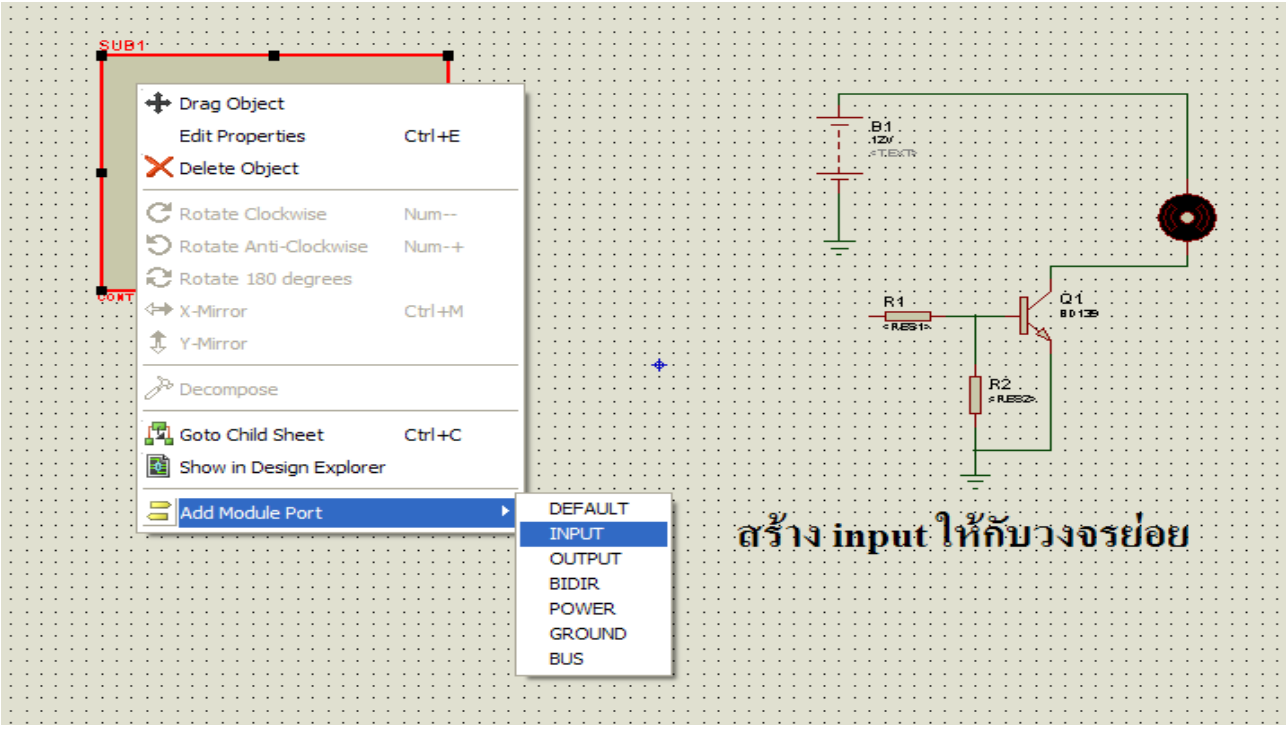



left click
right click
SUB?
U1:C
R2
Drag Object
Edit Properties Ctrl+E
Delete Object
Rotate Clockwise Num--
Rotate Anti-Clockwise Num-+
Rotate 180 degrees
X-Mirror Ctrl+M
Y-Mirror
Decompose
Goto Child Sheet Ctrl+C
Show in Design Explorer
Add Module Port
Edit the object or label being pointed at.
English ...
present...
sub2 - P...
EN

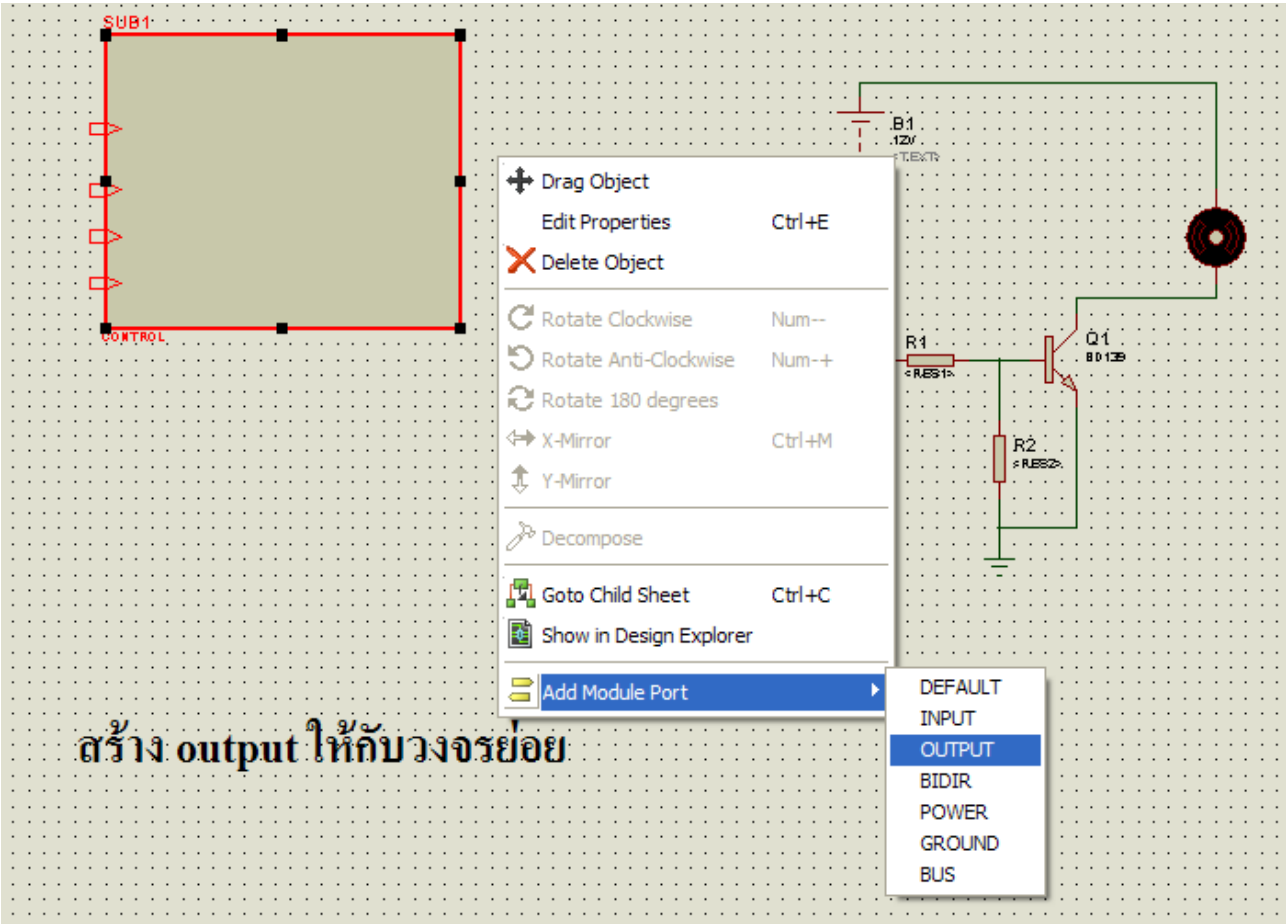

U1:B
U1:C
R1
Q1
SUB?
Edit Subcircuit
Name: SUB1
Circuit: CONTROL
Properties:
OK
Cancel


SUB1
CONTROL
Drag Object
Edit Properties Ctrl+E
Delete Object
Rotate Clockwise Num--
Rotate Anti-Clockwise Num-+
Rotate 180 degrees
X-Mirror Ctrl+M
Y-Mirror
Decompose
Goto Child Sheet Ctrl+C
Show in Design Explorer
Add Module Port
left click
right click
d sheet of this subcircuit.

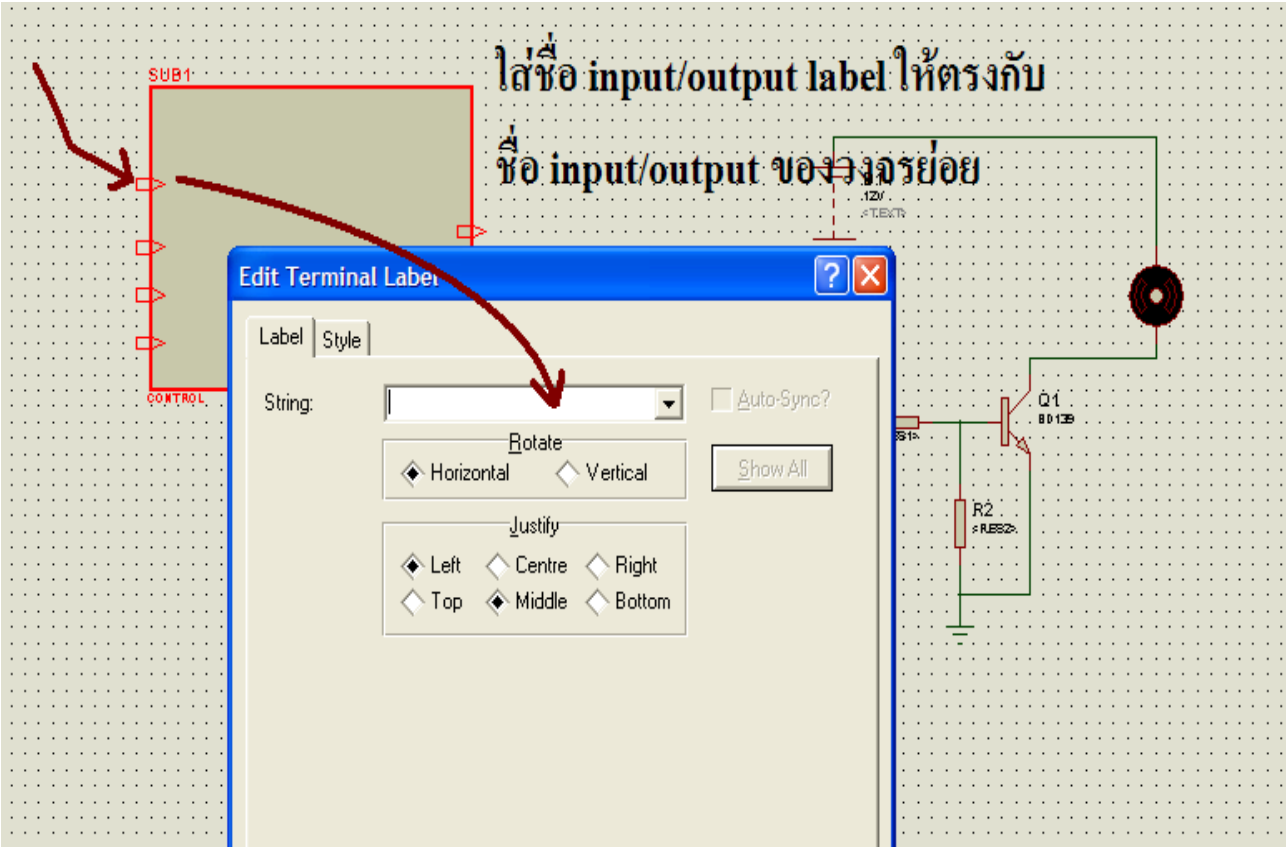

copy from sub circuit1
U1:A
U2:A
U1:B
U1:C
U3:A
IN1
IN2
IN3
out_con
Place
Select All Objects
Clear Selection
Cut To Clipboard
Copy To Clipboard
Paste From Clipboard
Exit to Parent Sheet

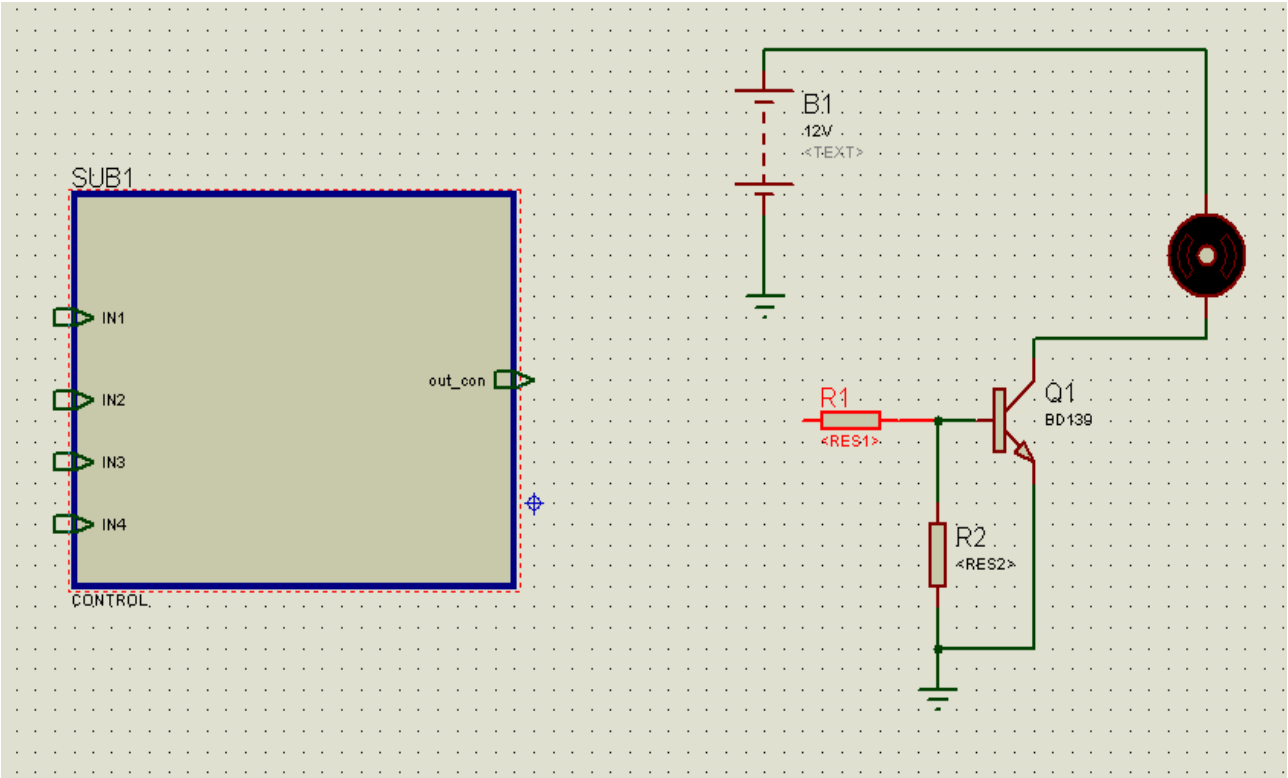


สร้าง input ให้กับวงจรย่อย



สร้าง output ให้กับวงจรย่อย

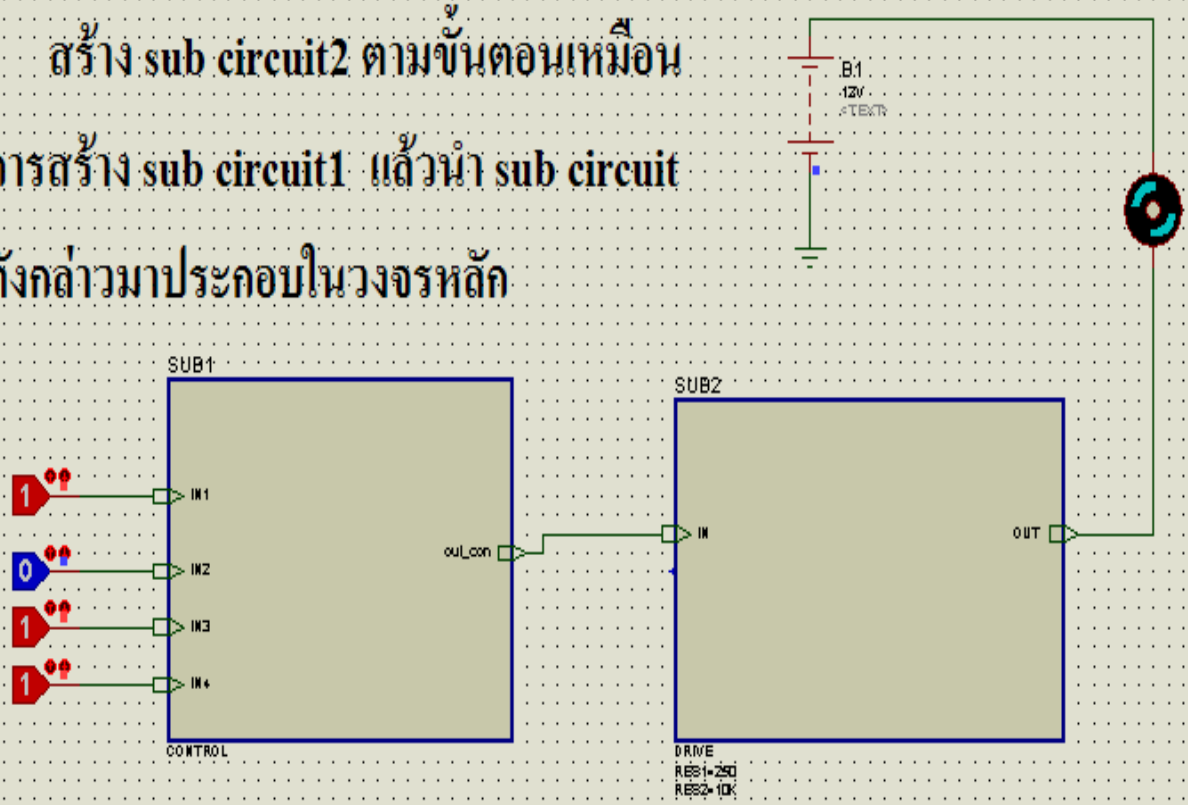

ใส่ชื่อ input/output label ให้ตรงกับ
ชื่อ input/output ของวงจรย่อย
SUB1
CONTROL
Edit Terminal Label
Label
Style
String:
Auto-Sync?
Rotate
Horizontal
Vertical
Show All
Justify
Left
Centre
Right
Top
Middle
Bottom
Q1
BD139
R2
<RES2>


SUB1
IN1
IN2
IN3
IN4
out_con
CONTROL
B1
12V
<TEXT>
R1
<RES1>
Q1
BD139
R2
<RES2>

สร้าง sub circuit2 ตามขั้นตอนเหมือน

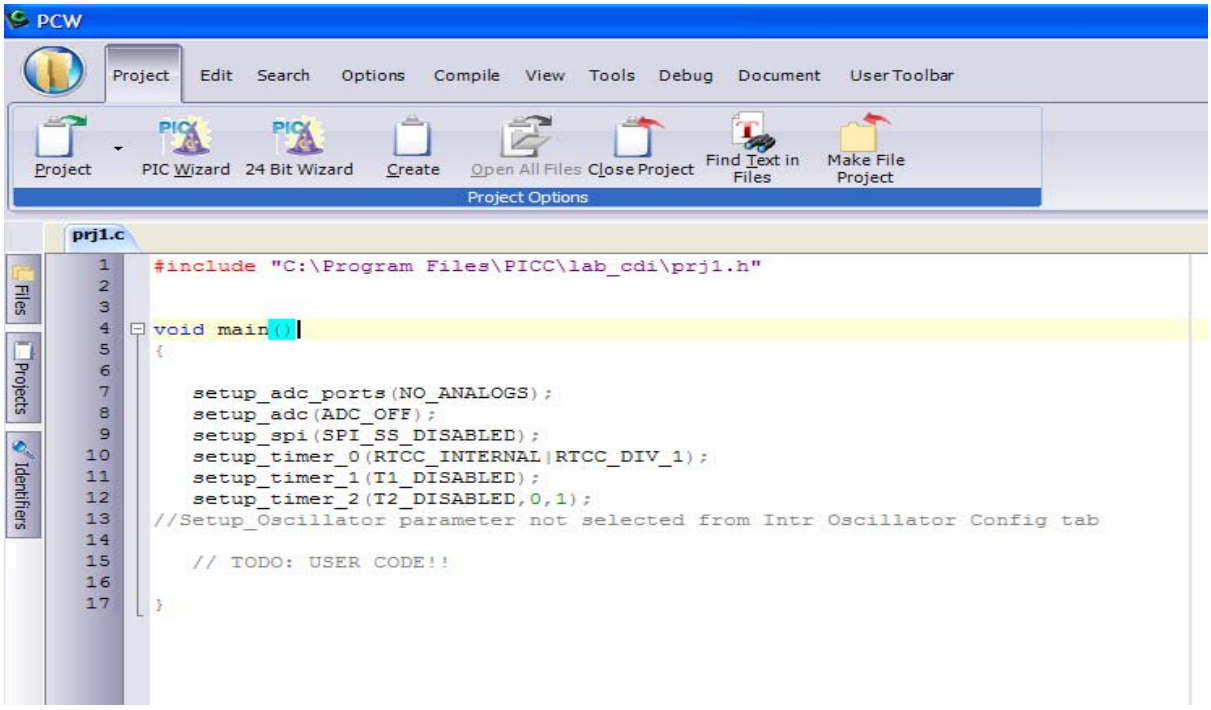
การสร้าง sub circuit1 แล้วนำ sub circuit

ดังกล่าวมาประกอบในวงจรหลัก

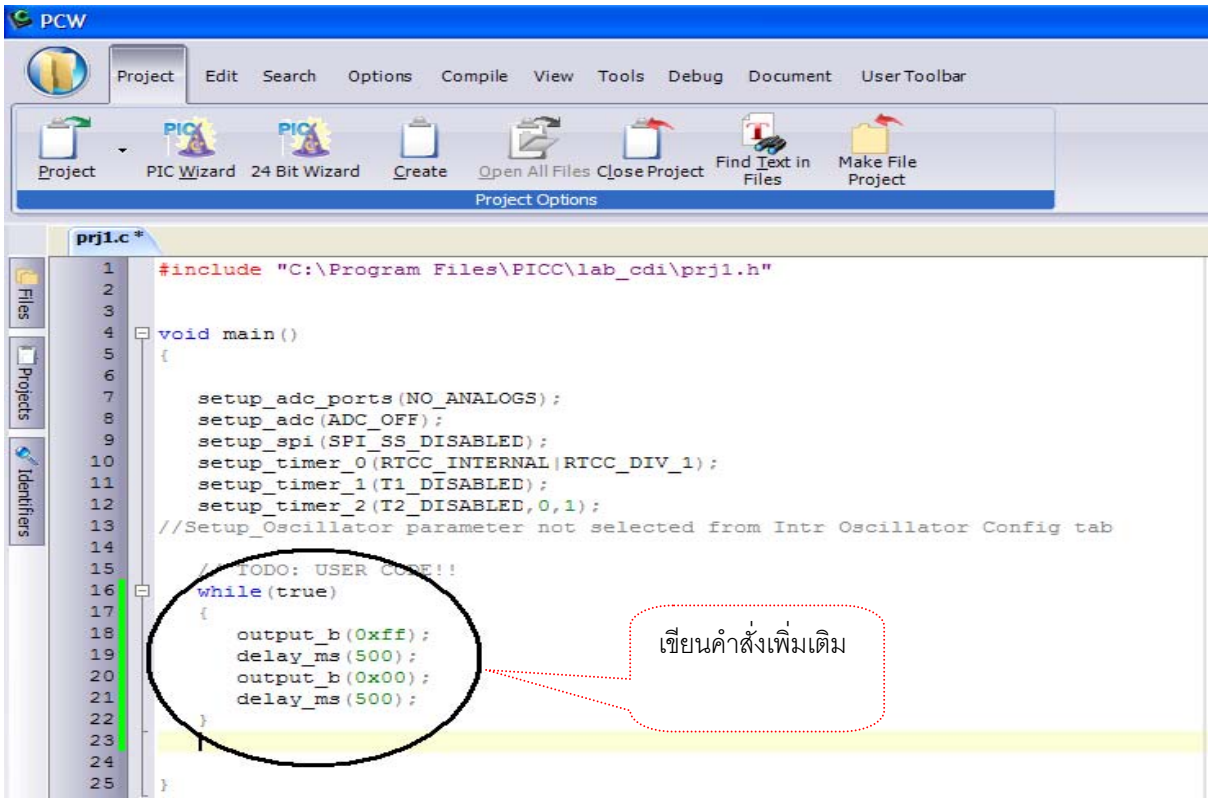


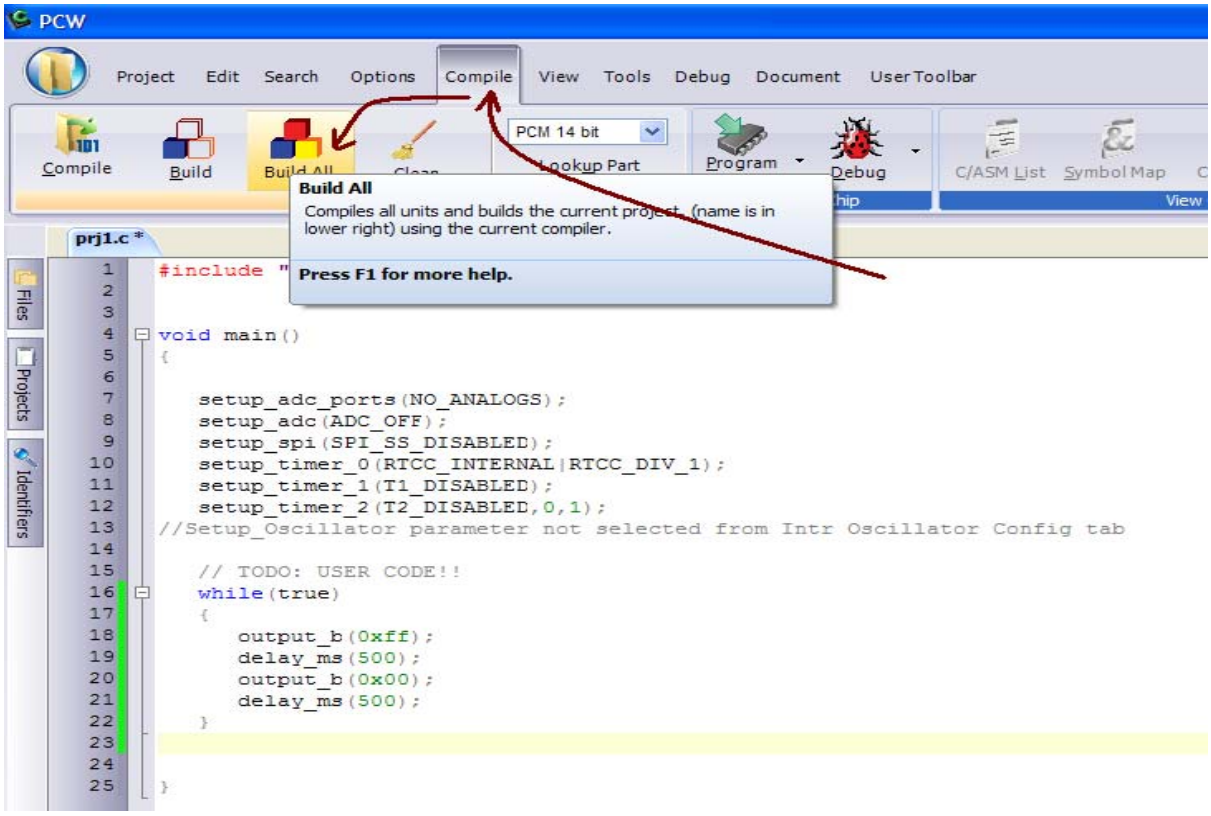
ตัวอย่างการใช้โปรแกรม PIC C Compiler ร่วมกับโปรแกรม Proteus เพื่อใช้สำหรับการเรียนรู้เรื่องของการพัฒนาโปรแกรม

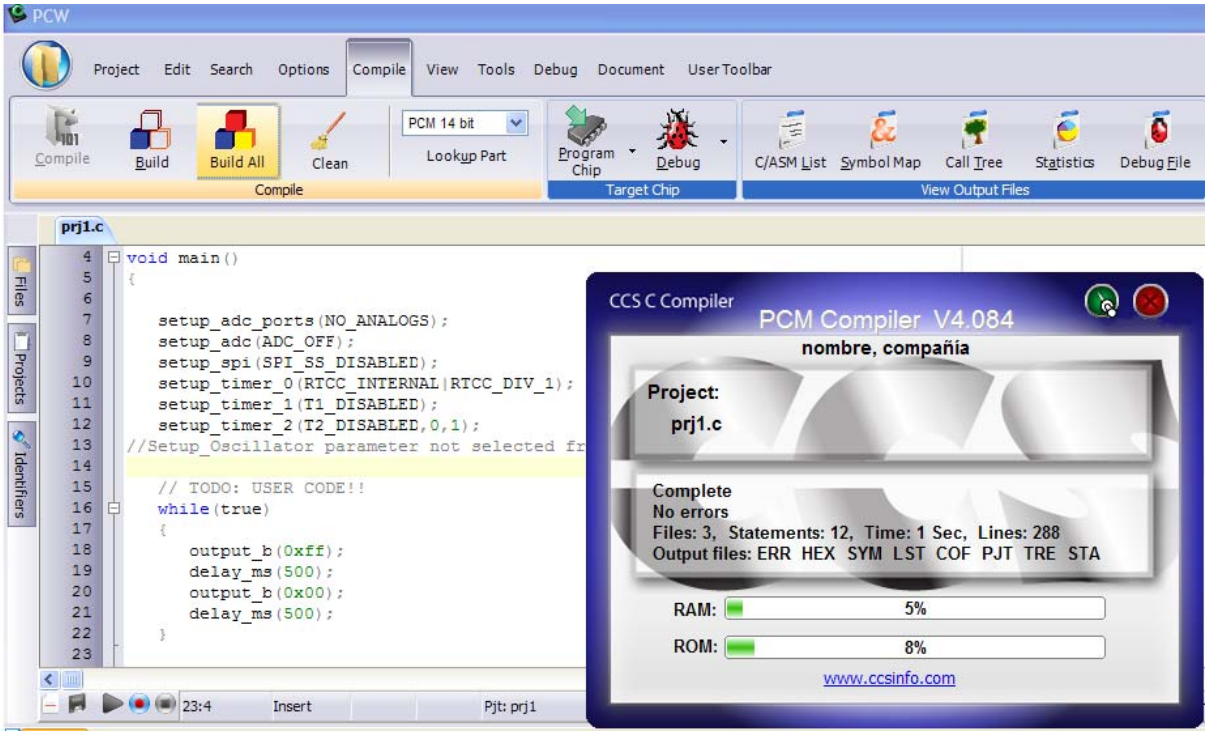
1 นำ project ที่ได้จากสร้างในข้างต้นมาเขียนคำสั่งเพิ่มเติม และ ทำตามขั้นตอนดังแสงตามรูป



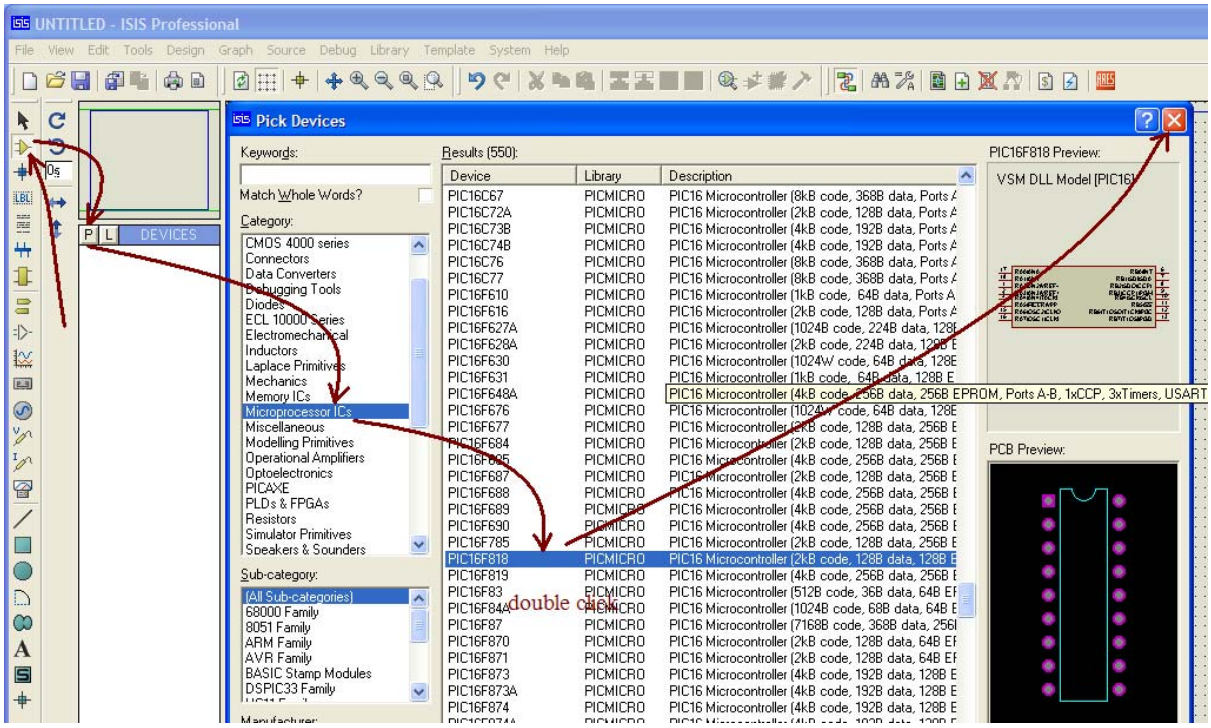
Project ที่ถูกสร้างขึ้นในข้างต้น




PCW
Project Edit Search Options Compile View Tools Debug Document User Toolbar
Compile
Build
Build All
Clean
PCM 14 bit
Lookup Part
Program
Debug
C/ASM List
Symbol Map
View
Build All
Compiles all units and builds the current project. (name is in lower right) using the current compiler.
Press F1 for more help.
prj1.c *
Files
Projects
Identifiers
#include "
void main()
{
setup_adc_ports(NO_ANALOGS);
setup_adc(ADC_OFF);
setup_spi(SPI_SS_DISABLED);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
setup_timer_1(T1_DISABLED);
setup_timer_2(T2_DISABLED,0,1);
//Setup_Oscillator parameter not selected from Intr Oscillator Config tab
// TODO: USER CODE!!
while(true)
{
output_b(0xff);
delay_ms(500);
output_b(0x00);
delay_ms(500);
}
}

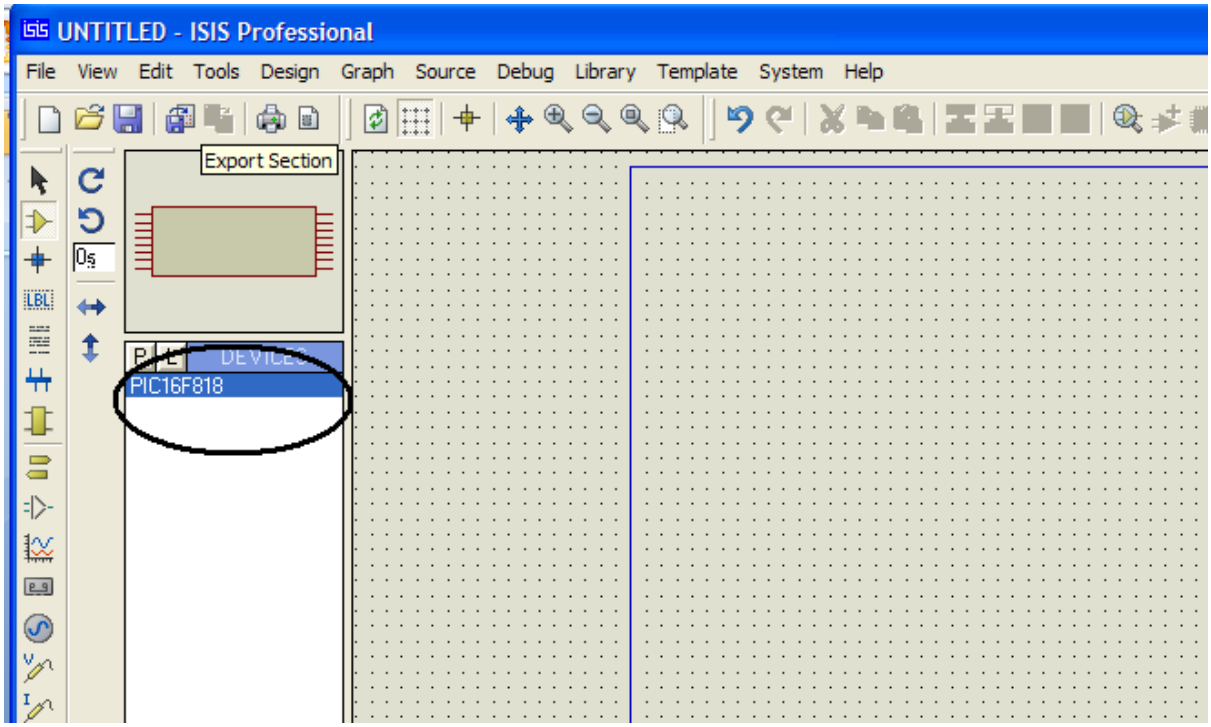

PCW
Project Edit Search Options Compile View Tools Debug Document User Toolbar
Compile
Build
Build All
Clean
PCM 14 bit
Lookup Part
Program Chip
Debug
C/ASM List
Symbol Map
Call Tree
Statistics
Debug File
Compile
Target Chip
View Output Files
prj1.c
Files
Projects
Identifiers
void main()
{
setup_adc_ports(NO_ANALOGS);
setup_adc(ADC_OFF);
setup_spi(SPI_SS_DISABLED);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
setup_timer_1(T1_DISABLED);
setup_timer_2(T2_DISABLED,0,1);
//Setup_Oscillator parameter not selected fr
// TODO: USER CODE!!
while(true)
{
output_b(0xff);
delay_ms(500);
output_b(0x00);
delay_ms(500);
}
23:4
Insert
Pjt: prj1
CCS C Compiler
PCM Compiler V4.084
nombre, compañia
Project:
prj1.c
Complete
No errors
Files: 3, Statements: 12, Time: 1 Sec, Lines: 288
Output files: ERR HEX SYM LST COF PJT TRE STA
RAM: 5%
ROM: 8%
www.ccsinfo.com

2 เปิดโปรแกรม Proteus

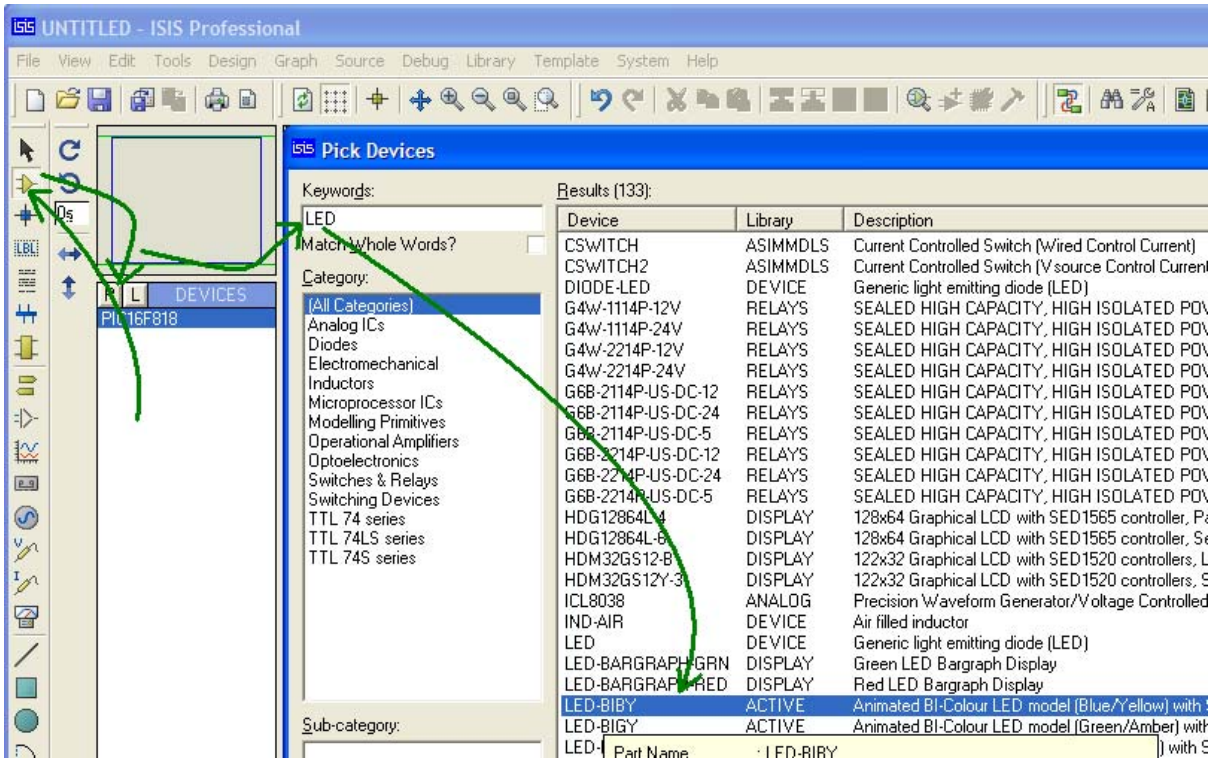


เลือกอุปกรณ์ PIC16F818

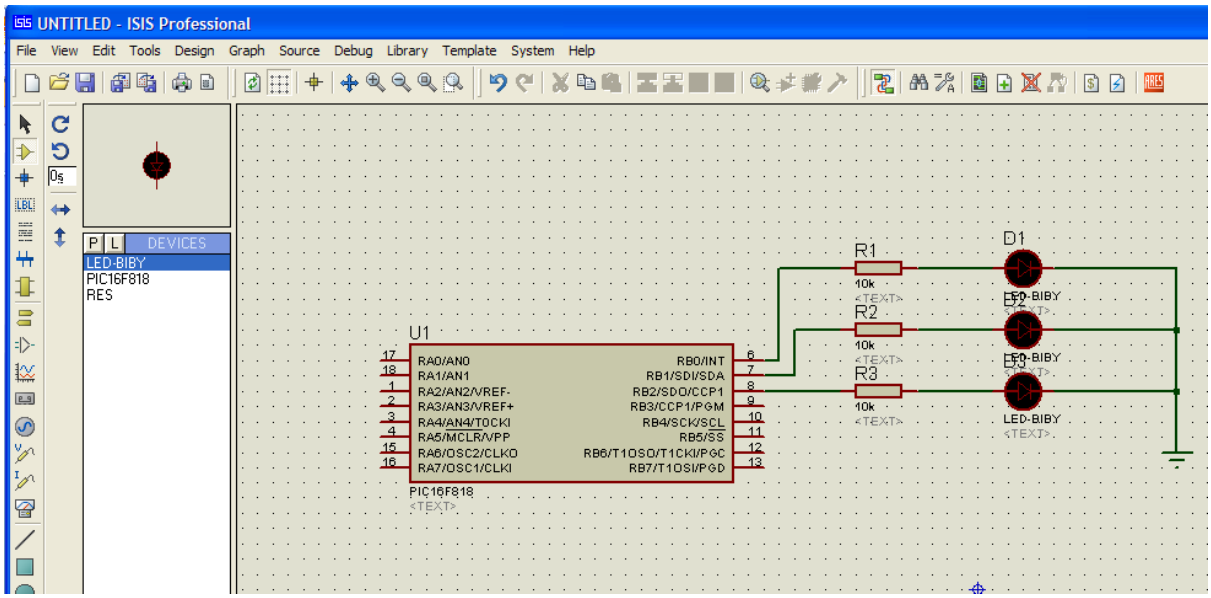
2.1 จัดหาอุปกรณ์ที่จะใช้ในการทดลอง



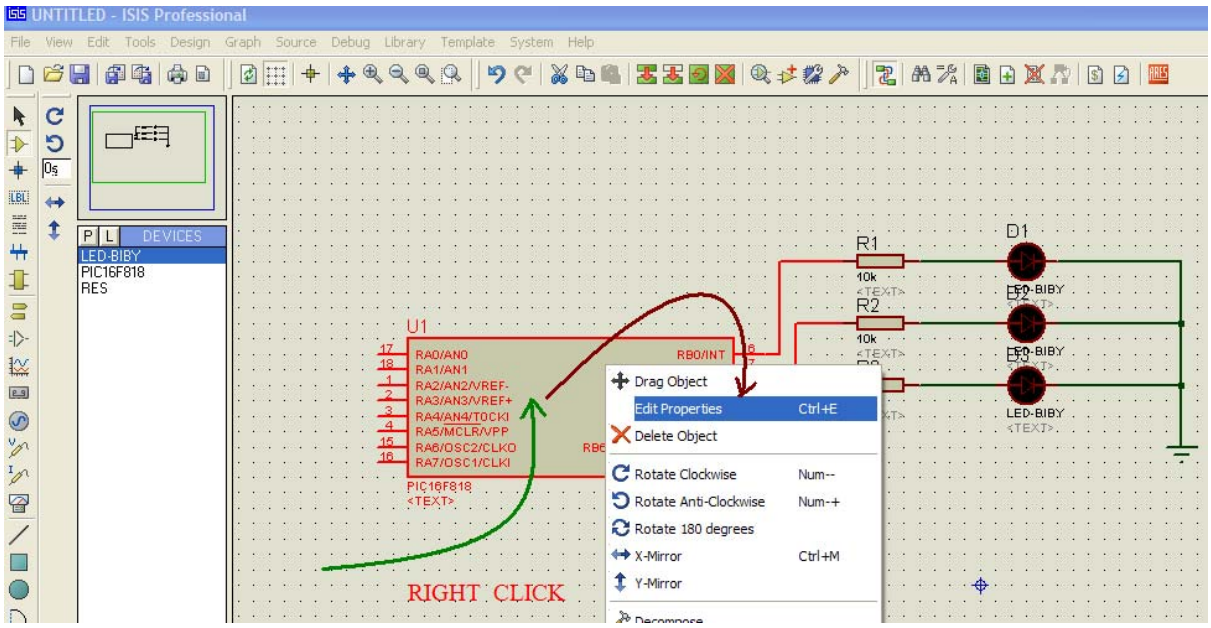
ใช้ Keywrod ช่วยในการหาอุปกรณ์

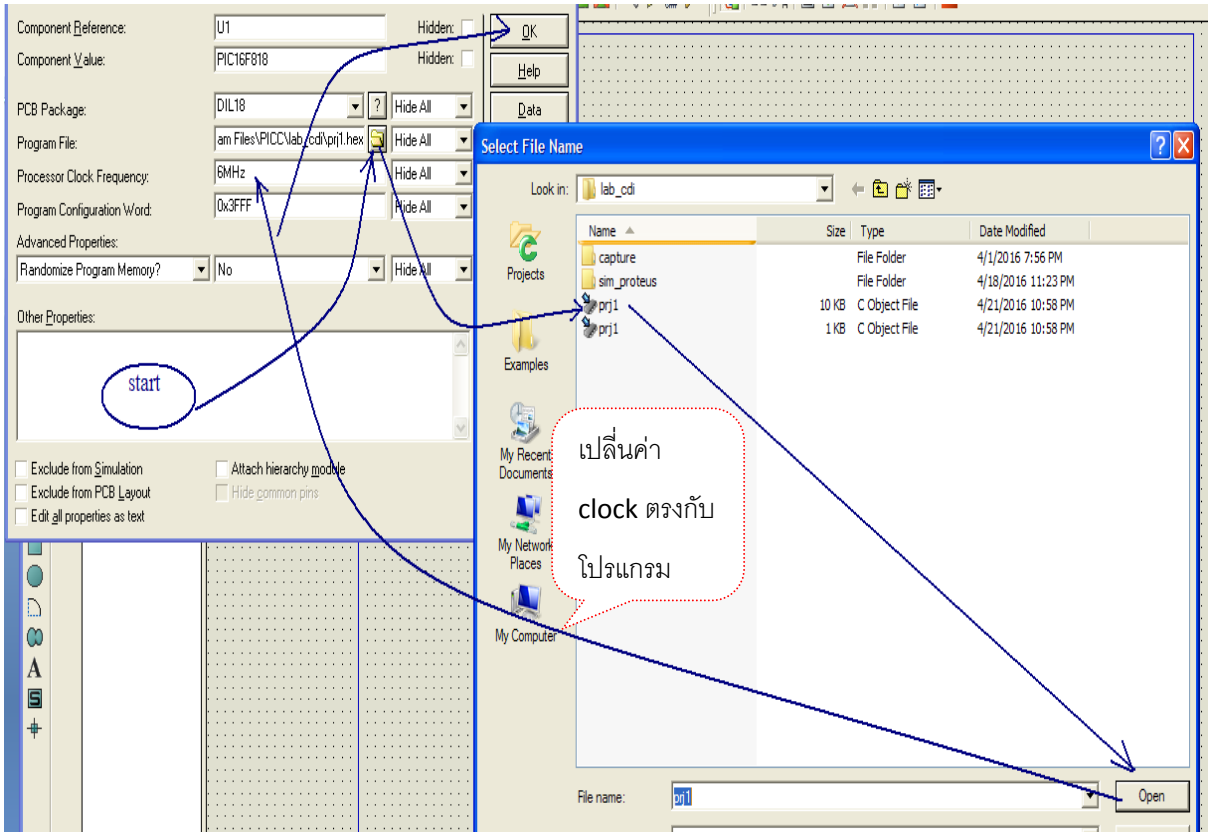


เลือกอุปกรณ์ให้ครบ และ ต่อวงจรตามรูป

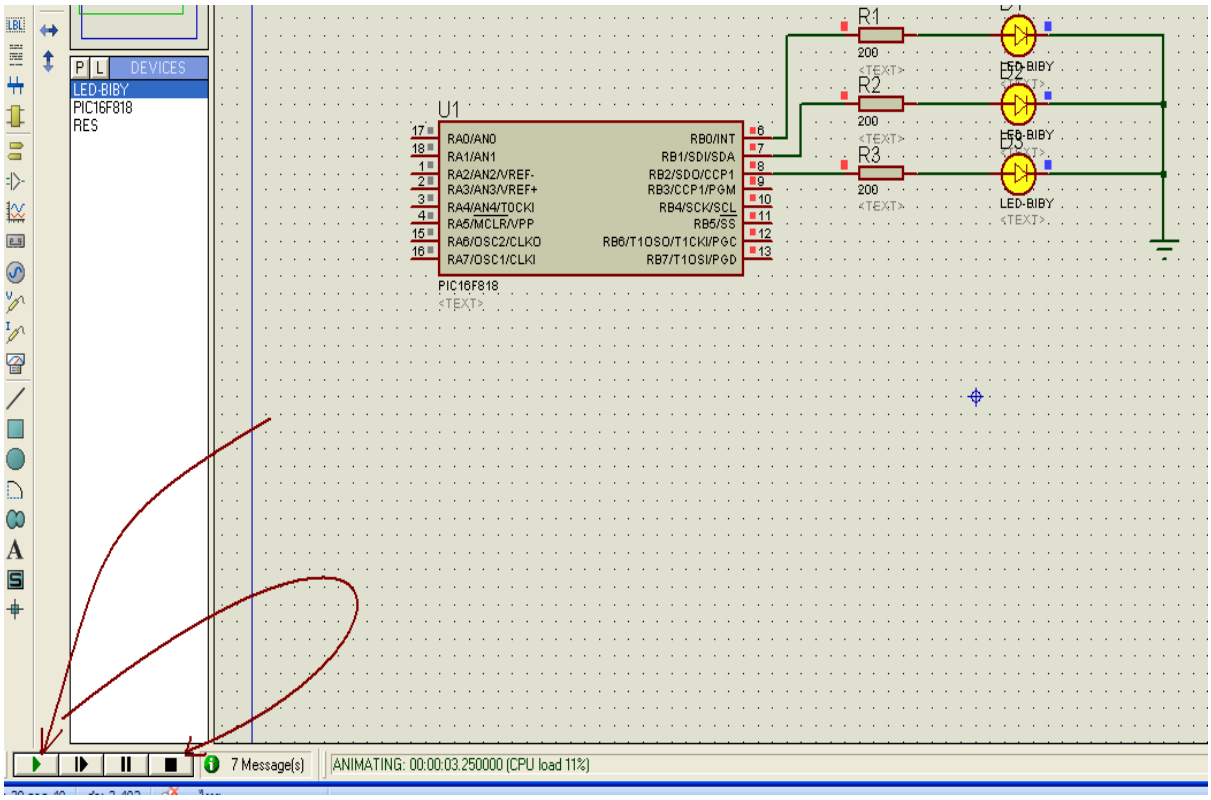


นำ HEX FILE ที่ได้รับจากการ Compile จากโปรแกรม PIC C มาบรรจุลงในตัว PIC16F818





การเริ่ม และ การหยุดตรวจสอบโปรแกรม

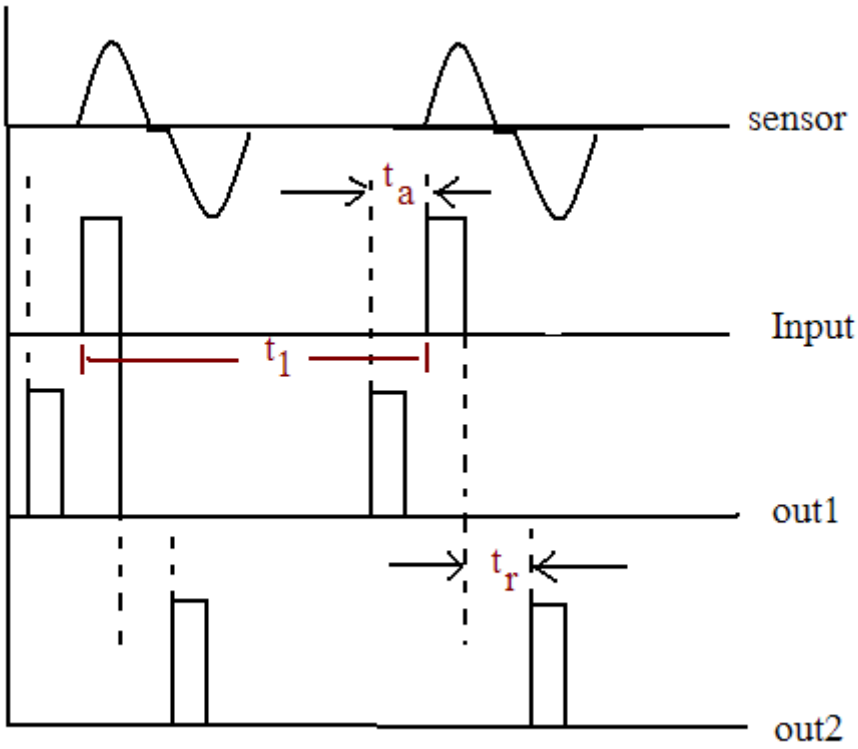


ขอให้ผู้สนใจการสร้างระบบควบคุมที่ประกอบขึ้นจาก Hardware และ Software ให้เรียนรู้ศึกษาการใช้เครื่องมือ เพิ่มเติมให้มากที่สุด จากแหล่งข้อมูลต่าง ๆ ที่มีอยู่มากมายในยุกต์ปัจจุบัน ที่มากับ Internet

การออกแบบวงจรในส่วนของ Conditioning

โดยใช้ไมโครคอนโทรลเลอร์ เพื่อใช้ปรับแต่งองศาการจุดระเบิด

ตามที่ได้กล่าวมาแล้วในเบื้องต้นว่า การออกแบบวงจรในส่วนของ Conditioning ครั้งนี้เพื่อให้ระบบสามารถปรับแต่ง องศาการจุดระเบิด [BTDC] จากสัญญาณอินพุทที่ได้รับจาก Sensor ได้ อิบายได้ตาม Timing Diagram ข้างล่าง



แสดงระบบที่สามารถปรับเฟสระหว่างสัญญาณ อินพุท กับ เอาท์พุท ให้ต่างเฟสกันได้

จาก Timing Diagram แสดงให้เห็นว่าเราสามารถควบคุมองศาการจุดระเบิดให้นำหน้า [BTDC] ล้าหลัง [ATDC] หรือตรงกันกับ [TDC] ก็ได้

$$T_a = t_1 pre - T_{advance} \text{.....(1)}$$

$$T_r = t_{1pre} + T_{restart} \text{.....(2)}$$

โดยให้

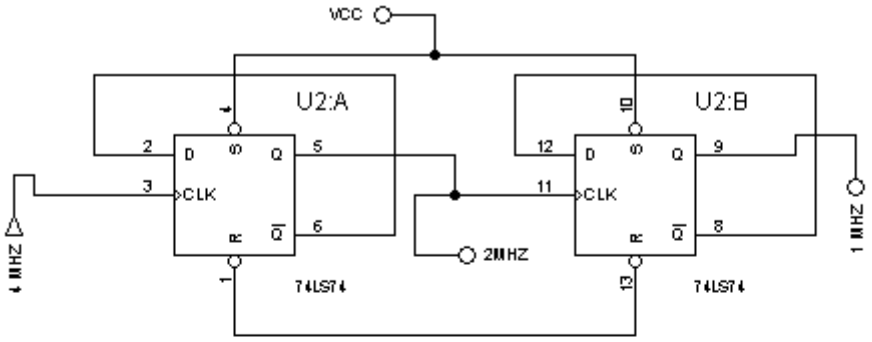
T_a = เวลา ที่ทำให้องศาการจุดระเบิดเกิดก่อน TDC

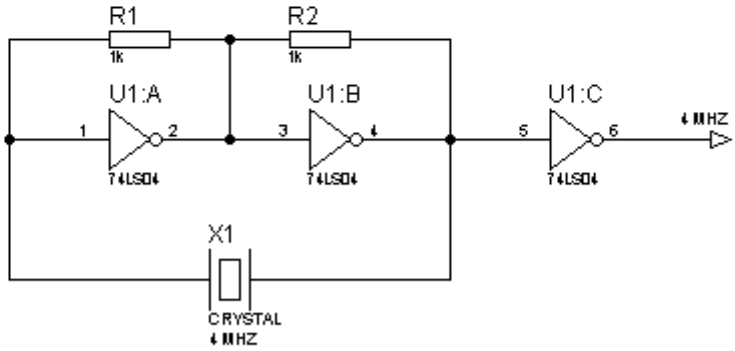
T_r = เวลา ที่ทำให้องศาการจุดระเบิดเกิดหลัง TDC

$t_1 pre$ = เวลาในแต่ละวัตจักรก่อนหน้า (TDC ถึง TDC)

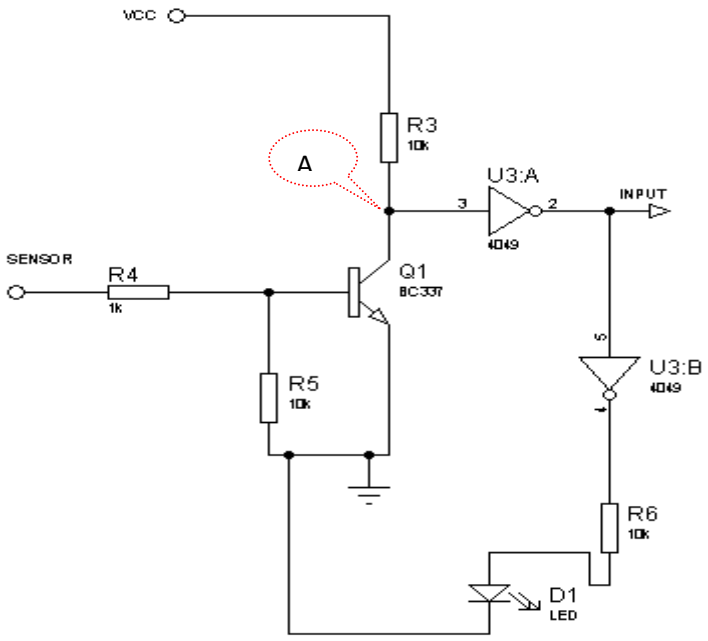
การออกแบบวงจรเพื่อใช้รองรับการทำงานของระบบ

การออกแบบวงจรเพื่อให้รองรับการทำงานของระบบให้บรรลุตามวัตถุประสงค์ผู้ออกแบบจำเป็นต้องอาศัยเวลาและการสั่งสมประสบการณ์ ในการทำความเข้าใจ การทำงานของระบบ เนื่อง จากการออกแบบวงจรที่รองรับระบบได้ดี จะทำให้ง่าย ในการพัฒนาโปรแกรม ตามมาด้วย รูปข้างล่างแสดงการออกแบบวงจร

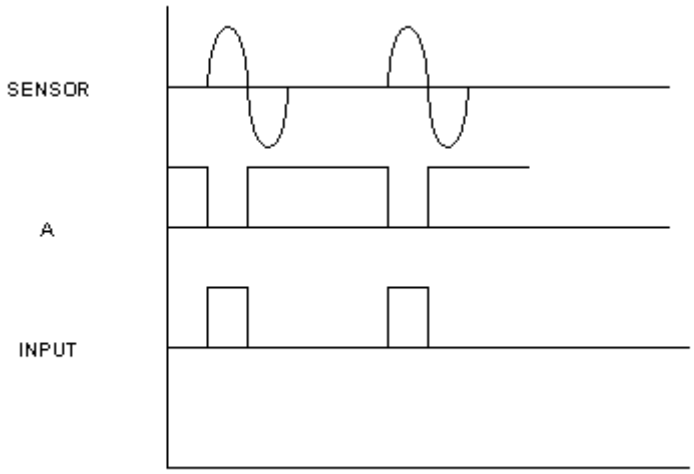




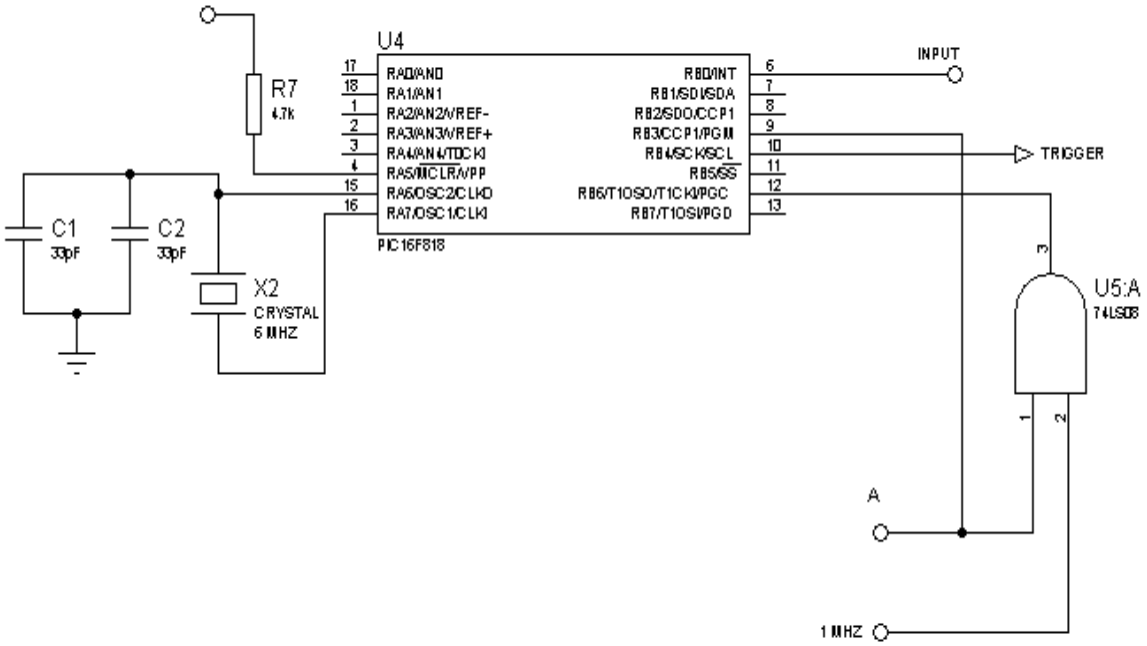
Time Base 1 Mhz



วงจรปรับแต่งสัญญาณ จาก Sensor



รูปสัญญาณแต่ละจุด



สัญญาณ อินพุท และ เอาท์พุท ของไมโครคอนโทลเลอร์