

การทดลองที่ 4 การใช้งาน IC ขยายพอร์ต 8255 และ การควบคุม DC Motor

วัตถุประสงค์

1. เพื่อให้นิสิตสามารถเขียนโปรแกรมภาษา C อย่างง่ายในการควบคุมไมโครคอนโทรลเลอร์ได้
2. เพื่อให้นิสิตเข้าใจการการใช้งาน IC ขยายพอร์ต 8255 และการควบคุม DC Motors

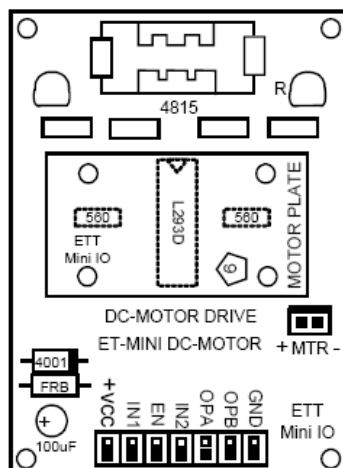
อุปกรณ์ในการทดลอง

1. ชุดทดลองไมโครคอนโทรลเลอร์
2. เครื่องคอมพิวเตอร์ PC พร้อมโปรแกรมสำหรับการเขียนและคอมไพล์ภาษา C
 - โปรแกรม Keil51 v.xx
3. สายต่อพอร์ตอนุกรม
4. วงจรควบคุม Brushed DC motor และ Stepping Motor

ทฤษฎี

วงจรควบคุม Brushed DC Motor

Module ET-MINI DC-MOTOR เป็นชุดขับ DC Motor 5V สามารถควบคุมทิศทางการหมุนได้ทั้ง ช้ายและขวา และอ่านรอบการหมุนได้โดยใช้ Opto เป็น Sensor ลักษณะของขั้วต่อใช้งาน และการส่งลอจิกควบคุมการหมุนจะแสดงดังรูปและตารางด้านล่าง โดย ขั้วต่อ IN1,IN2 จะเป็นตัวกำหนดทิศทางการหมุน ส่วนขั้วต่อ EN จะทำหน้าที่ Enable ชุดขับกระแส โดยจะทำงานที่ลอจิก “1” ขั้วต่อ OPA และ OPB จะทำหน้าที่ส่งสัญญาณลอจิก “1” ไปให้ MCU เมื่อใบพัดของ DC Motor ไปตัดที่ตัว Opto Sensor โดยถ้าไม่มีอะไรมาตัดที่ตัว Sensor ที่ขั้วต่อ OPA และ OPB จะมีสถานะเป็น “0” ซึ่งเหมาะสำหรับใช้อ่านรอบการหมุนของ Motor โดยเลือกอ่านสัญญาณจากขั้วต่อ OPA หรือ OPB ก็ได้



ตารางแสดงการ Control DC-MOTOR

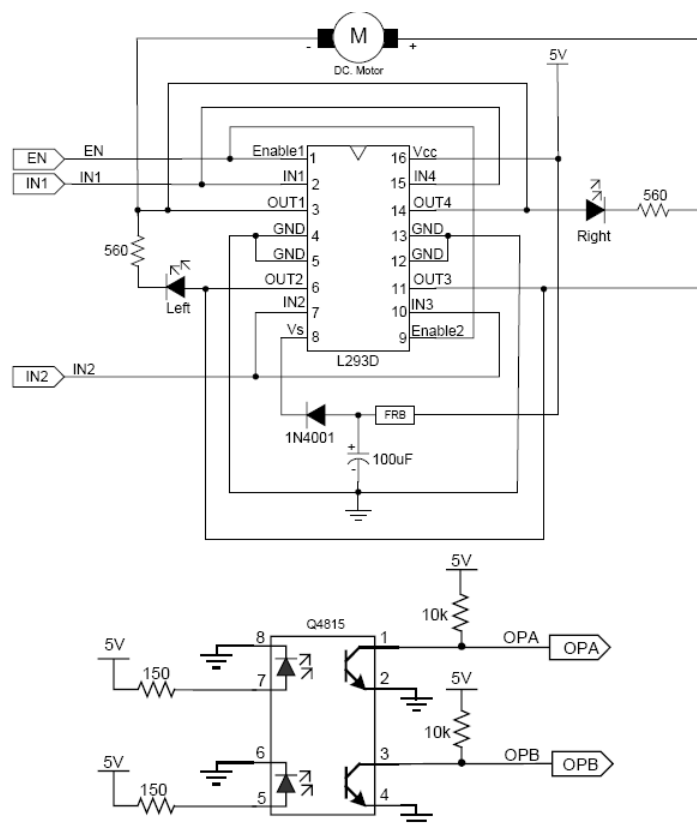
Control DC MOTOR								
RIGHT(R)			LEFT(L)			STOP		
EN	IN1	IN2	EN	IN1	IN2	EN	IN1	IN2
1	1	0	1	0	1	0	x	x



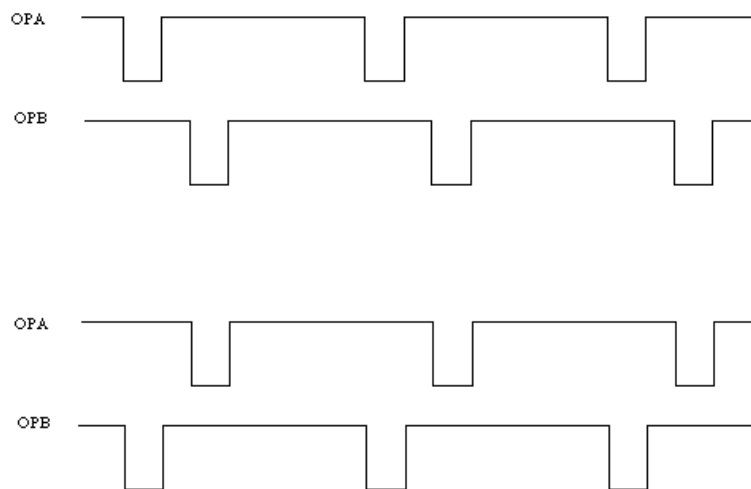
Con. DC-Motor



รูปโครงสร้าง ET-MINI DC-MOTOR และตำแหน่งขา Port ใช้งาน

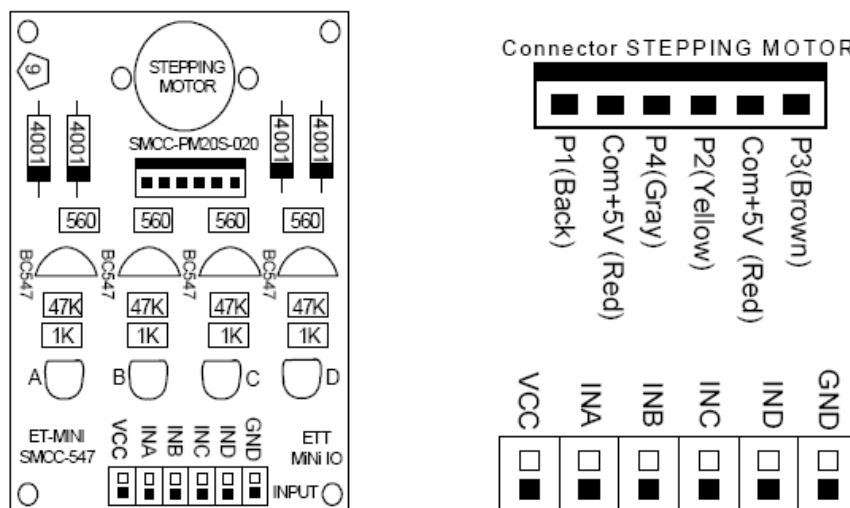


รูป วงจร ET-MINI DC-MOTOR

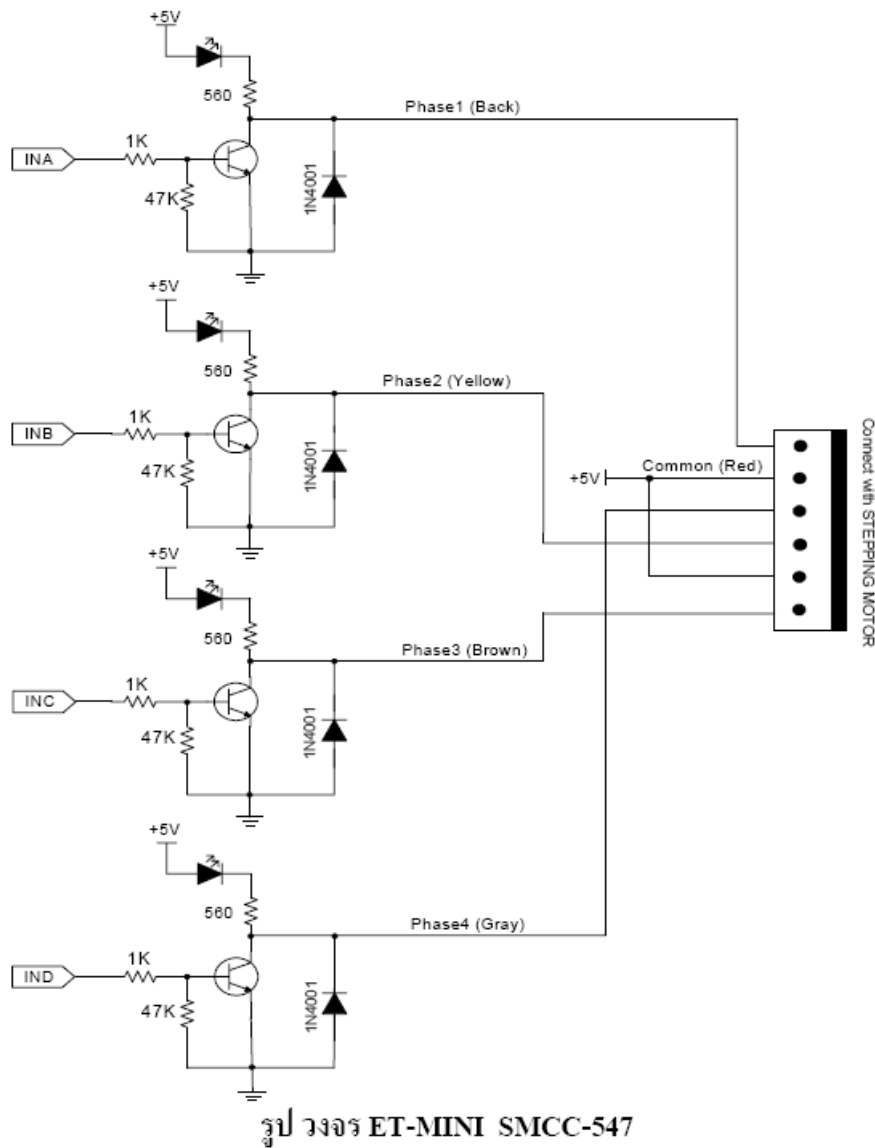


วงจรควบคุม Stepping Motor

Module ET-MINI SMCC-547 เป็นชุดขับ Stepping Motor ที่ใช้มีความละเอียดอยู่ที่ Step ละ 18 องศา ใช้ทรานซิสเตอร์ BC547 เป็นตัวขับ และใช้ LED แสดงผลการทำงานเมื่อมีการส่งลอจิก “1” มาคอนโทรลในแต่ละเฟส LED ก็จะได้เห็นว่าเฟสไหนทำงานอยู่ โดยในการส่งลอจิกนั้นจะต้องส่งเรียงเฟสกันไปเรื่อยๆซึ่งจะเริ่มจากเฟสไหนก่อนก็ได้ ใน Module นี้จะมี Pin INA, INB, INC และ IND ไว้ให้สำหรับต่อเข้ากับ ขา I/O ของ MCU เพื่อให้ MCU ส่งลอจิก “1” มาทำการควบคุม Step การหมุนโดยเรียงเฟสไปตามลำดับ การจัดเรียงขาสัญญาณต่างๆจะแสดงดังรูปด้านล่าง ส่วน Vcc ที่ใช้ จะเป็น 5 V ถ้าใช้ MCU ที่ใช้แรงดัน 3.3V จะต้องใช้ชุด Opto แปลงระดับลอจิกจาก 3V เป็น 5 V ด้วย

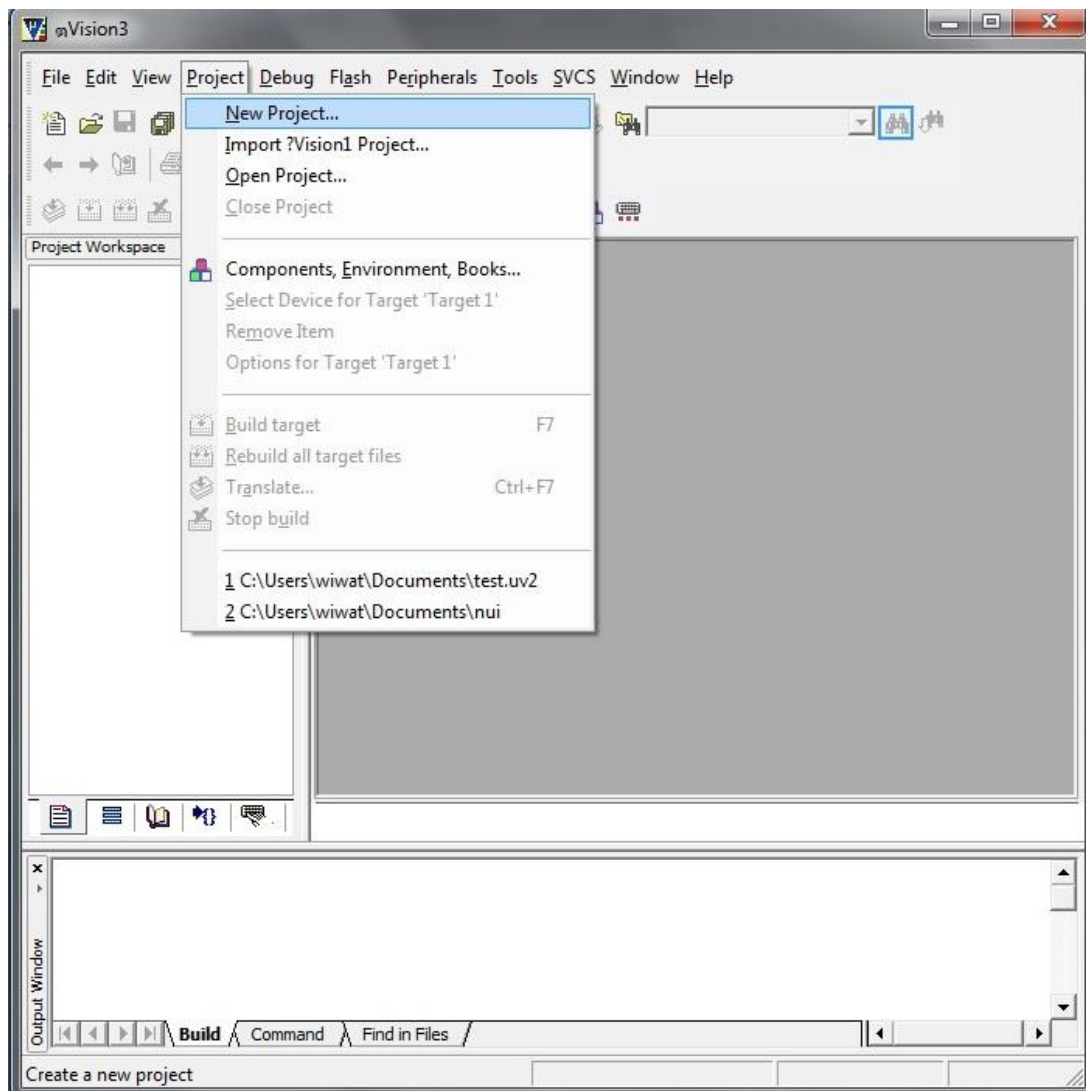


รูป โครงสร้าง ET-MINI SMCC-547 และตำแหน่งขา Port ใช้งาน

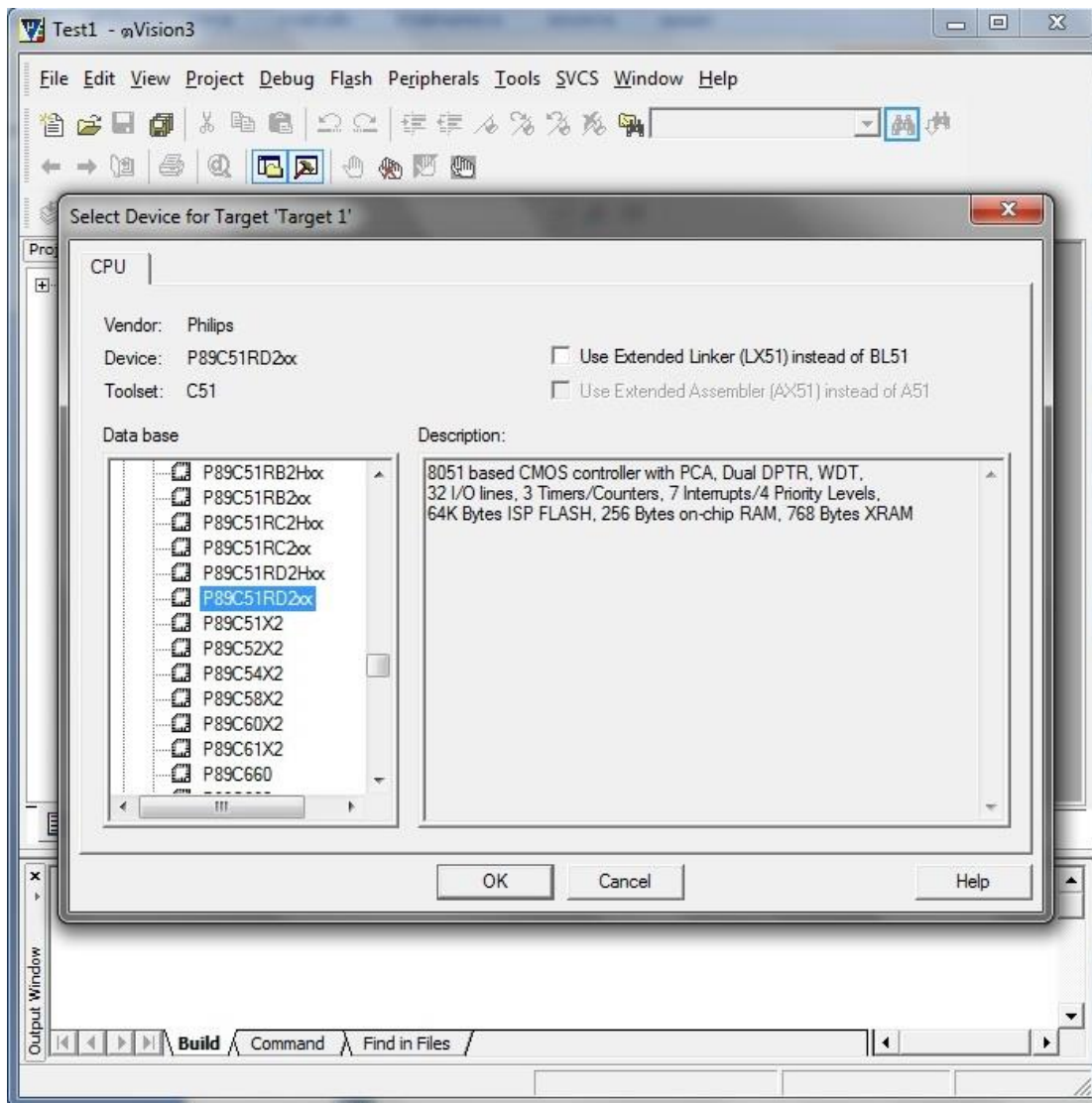


วิธีการทดลอง

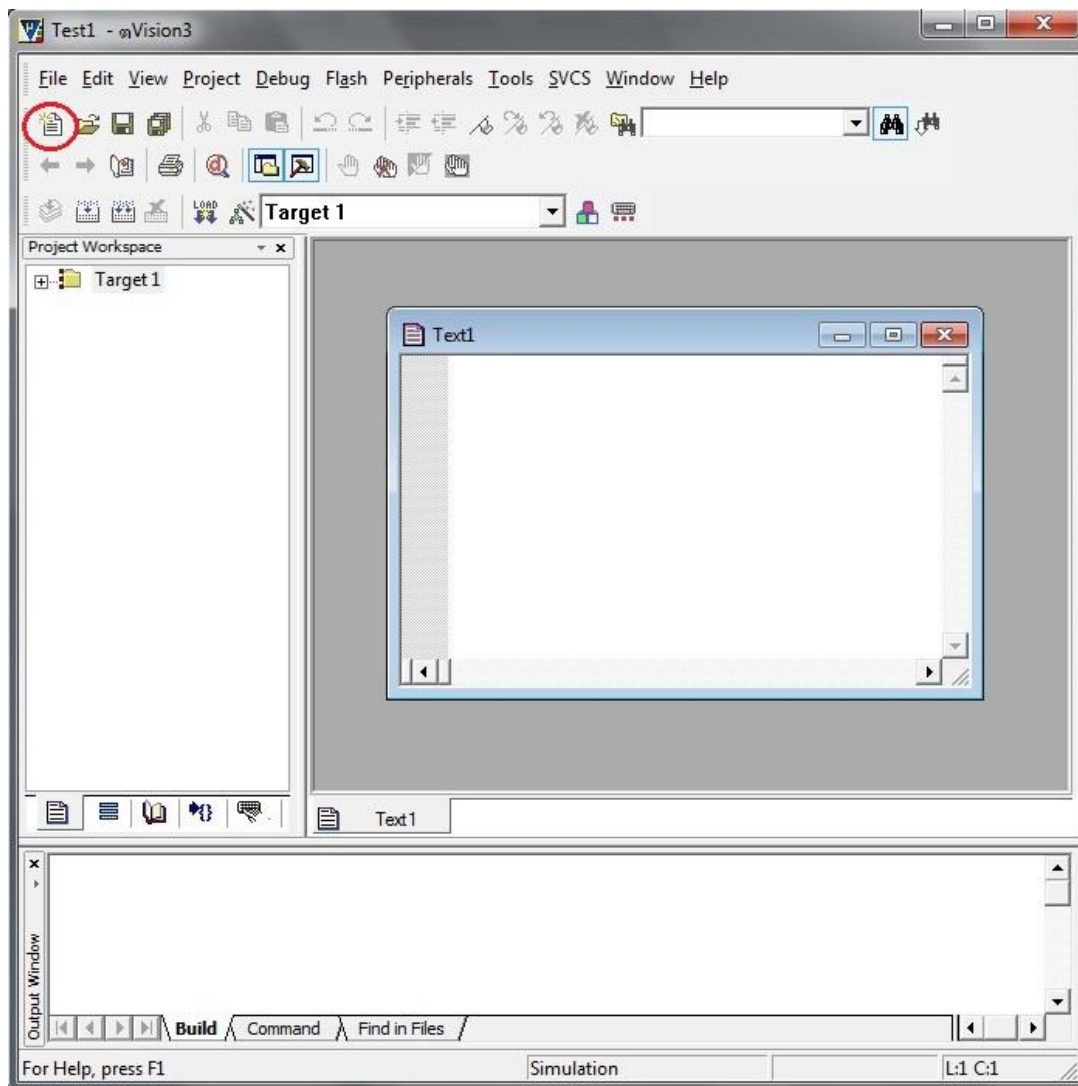
1. ใช้ IDE สำหรับพัฒนาชุดคำสั่งของ MCS-51 ด้วยภาษา C โดยใช้ Keil51 เพื่อเขียนโปรแกรม Lab04_x.c
2. สร้างโปรเจกต์ใหม่โดยเลือก New project ตั้งชื่อเป็น Lab04 แล้วกด ok



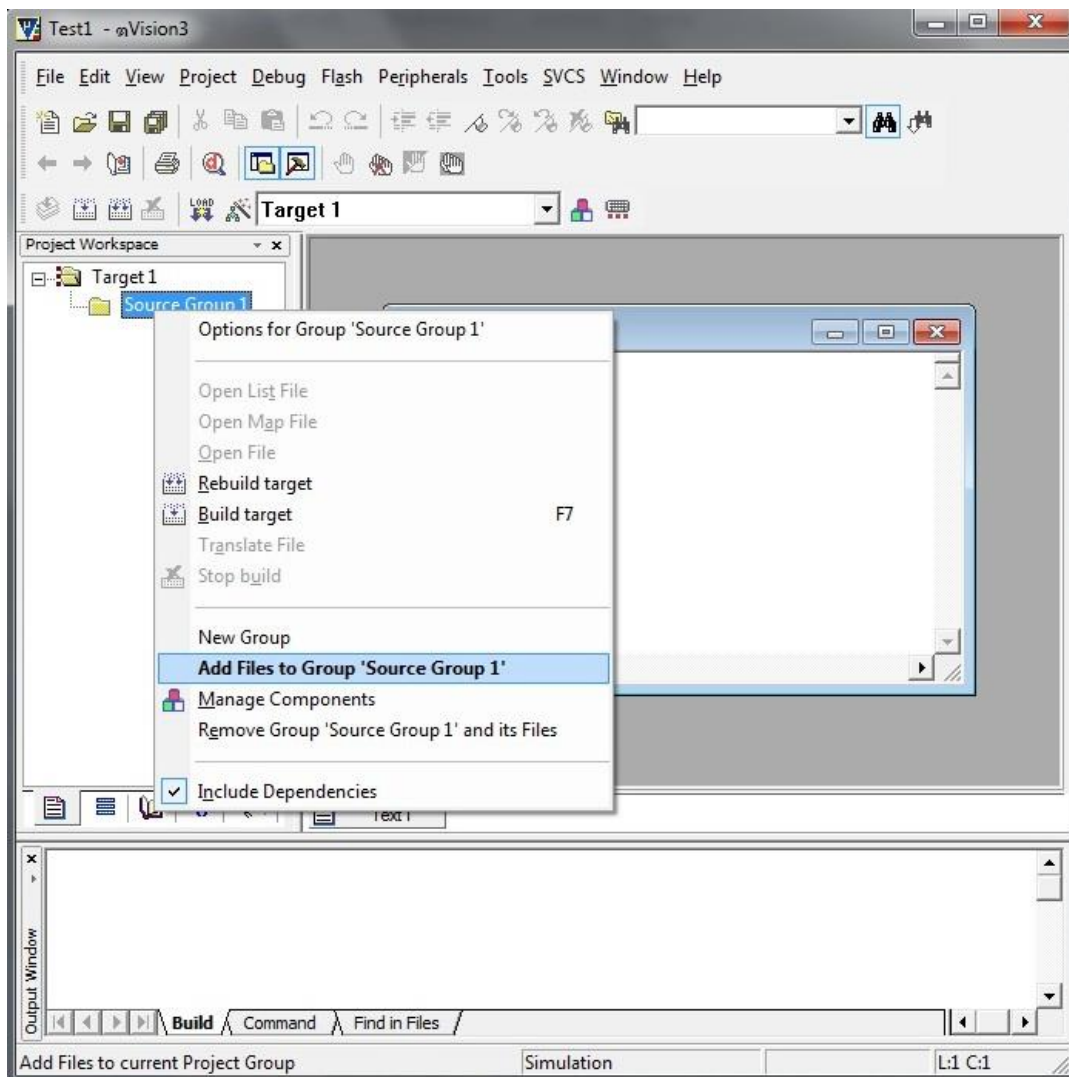
3. เลือก CPU : Philips P89C51RD2xx



4. จากนั้นคลิกที่ปุ่มวงกลมสีแดง เพื่อสร้างหน้าต่างสำหรับเขียนโปรแกรม(หน้าต่าง Text1)
สร้างไฟล์ใหม่ เขียนโปรแกรมและ save โดยใช้ชื่อ Lab04_x.c (x แทนหมายเลขการทดลอง)



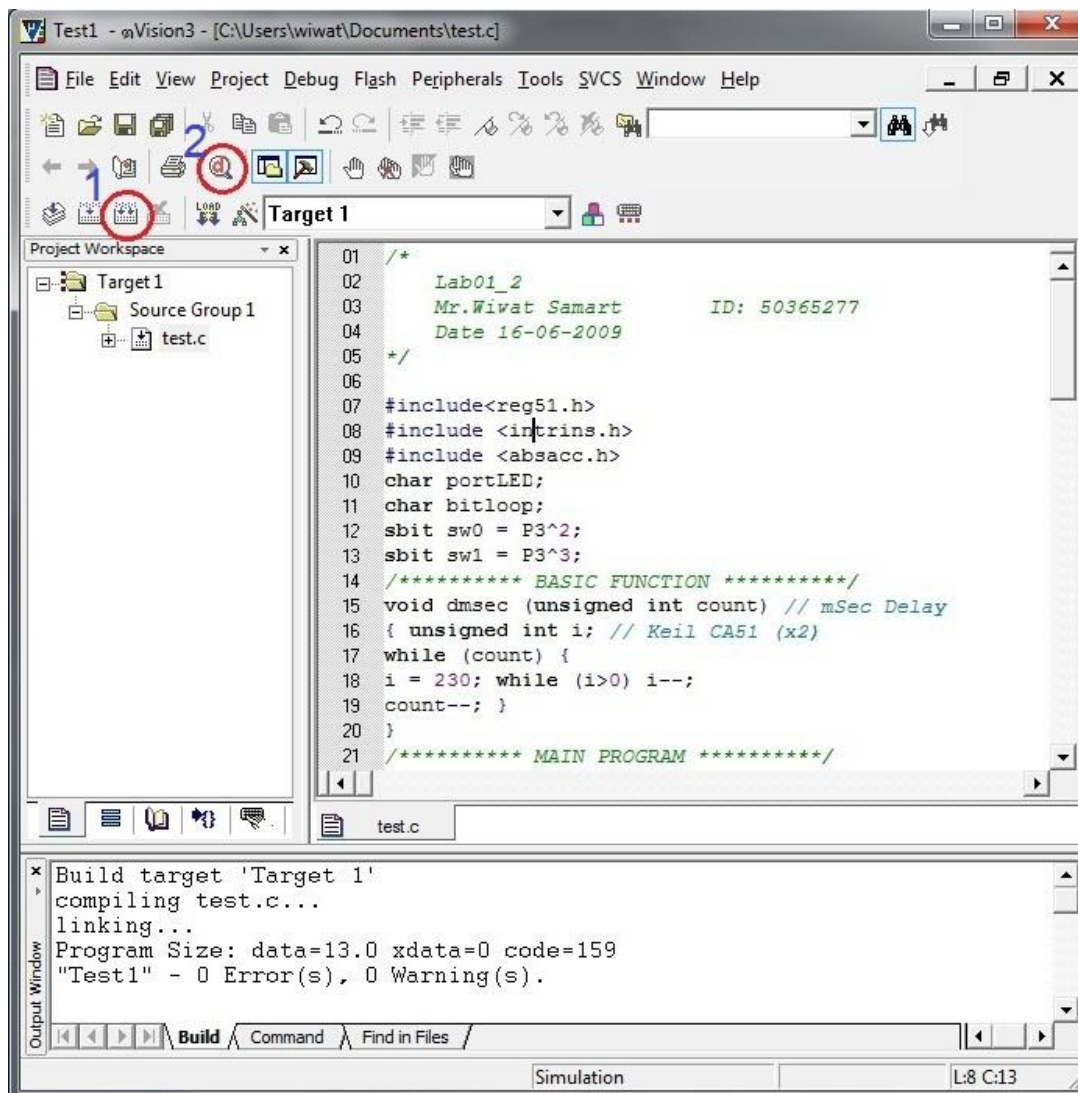
5. คลิกขวาที่ Source Group 1 แล้วเลือก Add file to group 'Source Group 1' แล้วเลือก File Lab04_x.c



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 Lab04.hex สามารถเปลี่ยนแปลงชื่อได้โดยแก้ไขชื่อ output file ลงในช่อง 'Name of Executable'



8. ให้นักศึกษาทำการทดลองที่เหลือ โดยเมื่อทำการทดลองใหม่ให้สร้างไฟล์ใหม่ เขียนโปรแกรมและ save เป็นชื่อใหม่ เช่น สร้างไฟล์ใหม่ชื่อ Lab04_2.c (2 แทนหมายเลขการทดลองที่ 2)

9. จากนั้น Click ที่ Source group แล้ว click ขวาที่ Lab04_1.c ซึ่งอยู่ใน Source group เลือก Remove file ' Lab04_1.c' เพื่อเอา file Lab04_1.c ออกจาก project

10. Click ขวาที่ Source group เลือก Add file to group แล้วเลือก File Lab04_2.c

11. ทำการทดลองที่เหลือทั้งหมด โดยการต่อวงจรอธิบายไว้ตอนต้นของแต่ละโปรแกรม โดยทดลองปรับเปลี่ยนทิศทางและความเร็วของ motor

12. การทดลอง 4.2 ใช้ Hyper Terminal แสดงผล

การทดลองที่ 4.1

<pre> /*File : LAB4_1.c Description : DC motor drive Clock : = 29.4912 MHz Hardware: DC motor module Compiler: Keil C Compiler*/ /*Connect IN1 to Port1.0, IN2 to Port1.1 and EN to Port1.2*/ #include<reg52.h> sbit IN1 = P1^0; sbit IN2 = P1^1; sbit EN = P1^2; /***** BASIC FUNCTION *****/ void delay1 (unsigned int count) // mSec Delay { unsigned int i; // Keil CA51 (x2) while (count) { i = 100; while (i>0) i--; count--; } </pre>	<pre> } /***** MAIN PROGRAM *****/ void main() { P1 = 0xff; while(1) { EN = 1; IN1 = 0; IN2 = 0; delay1(40); IN2 = 1; IN1 = 0; delay1(60); } } // end of main() </pre>
--	--

การทดลองที่ 4.2

<pre> /*File : LAB4_2.c Description : DC motor drive and counter Clock : 29.4912 MHz Hardware: DC motor module </pre>	<pre> Software: Hyper terminal Compiler: Keil C Compiler */ /*Connect OPA to P3.4 and use Counter0 to count*/ </pre>
---	---

<pre> /*Connect IN1 to Port1.0, IN2 to Port1.1 and EN to Port1.2*/ #include<reg51.h> #include <absacc.h> #include <stdio.h> /***** 8255 I/O PORT *****/ sbit IN1 = P1^0; sbit IN2 = P1^1; sbit EN = P1^2; int a,b; void start232(void); /***** DELAY FUNCTION *****/ void delay1 (unsigned int count) // Delay { unsigned int i; // Keil CA51 (x2) while (count) { i = 100; while (i>0) i--; count--; } } /***** MAIN PROGRAM *****/ void main() { start232(); </pre>	<pre> TR0=1; while(1) { EN = 1; IN1 = 0; IN2 = 0; delay1(50); IN2 = 1; IN1 = 0; delay1(50); a=TH0; b=TL0; printf("Counter0 = %d %d \n\r", a, b); } } // end of main() /***** Function void start232(void) { // speed x 2 SCON = 0x52; // set RS232 parameter TMOD = 0x25; // Timer0 as counter // TH1 = -6; PCON = 0x80; //19200 TH1 = -12; PCON = 0x80; //9600 </pre>
--	--

```

TR1 =1;
}

RI = 0;

```

การทดลองที่ 4.3

<pre> /*File : LAB4_3.c Description : Stepping motor, 1-phase drive Clock : 29.4912 MHz Hardware: Stepping motor module, Step angle 18 Compiler: Keil C Compiler */ /*Connect INA to Port1.0, INB to Port1.1, INC to Port1.2 and IND to Port1.3*/ /* 1 phase drive*/ /*Step angle 18*/ #include<reg52.h> int step; /***** DELAY FUNCTION *****/ void delay1 (unsigned int count) // Delay { unsigned int i; // Keil CA51 (x2) </pre>	<pre> while (count) { i = 100; while (i>0) i--; count--; } } /***** MAIN PROGRAM *****/ void main() { step = 0x11; while(1) { P1=step; step = _crol_(step,1); delay1(400); } } // end of main() </pre>
---	--

Reference

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

การบ้าน

1. จากการทดลอง 4.1 เขียนโปรแกรมเพื่อให้ motor หมุนเร็วด้วยความเร็ว 3 ระดับ คือ ช้า ปานกลาง และเร็วและปรับทิศทางการหมุนของมอเตอร์

2. จากการทดลอง 4.3 เขียนโปรแกรมเพื่อให้ motor หมุนเร็วด้วยความเร็ว 3 ระดับ คือ ช้า ปานกลาง และเร็ว โดยเปลี่ยนเป็นการขับแบบ 2-phase และ half step

3. จากการทดลอง 4.2 เขียนโปรแกรมเพื่อตรวจสอบและแสดงทิศทางการหมุนของมอเตอร์