ภาษาซีสำหรับ MCS-51

รศ.ฉรงค์ บวบทอง
ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
มหาวิทยาลัยธรรมศาสตร์

ทึ่ม

ทึ่มเข้าสอบทีไรใช้ลอกเพื่อน
ครูขู่เตือนจะไล่ออกถ้าจับได้
สอบปลายปีทึ่มจึงไม่ลอกใคร
ทำไม่ได้กระดาษเปล่าเอาส่งคืน

พวกเพื่อนเพื่อนที่ให้ทึ่มได้ลอก
สอบเสร็จบอกกับทึ่มว่าอย่าขมขื่น

ขนาดกูดูหนังสือมาทั้งคืน
กูยังยื่นกระดาษเปล่าเอาส่งครู

ทึ่มโมโหตัดพ้อแล้วต่อว่า
กูอุตสาห์ไม่ลอกมึงตามคำขู่
มึงดันส่งกระดาษเปล่าเอาเหมือนกู
เดี๋ยวคุณครูเขารู้หรอกว่าลอกกัน

วัตถุประสงค์

- เพื่อให้มีความเข้าใจเกี่ยวกับภาษา ซี
- เพื่อให้เขียนโปรแกรมภาษา ซี สำหรับไมโครคอนโทรลเลอร์ได้

ความรู้เบื้องต้นของภาษา C

- พ.ศ.2515 เดนนิส รีตชี่ (Dennis Ritchie) ได้พัฒนาภาษา C ขึ้น
- พ.ศ. 2531 ได้รูปแบบมาตราฐานของภาษา C ที่เรียกว่า ANSI-C

ตัวแปลภาษา C สำหรับ MCS-51

ผลิตภัณฑ์	Google
Keil C51 µVision2 µVision3	**36,700**
Mikro C51	**33,200**
http://www.mikroe.com/en/download/	
Rkit-51 เป็นของบริษัท Raisonance	**5,420**

รูปแบบโครงสร้างภาษา C

```
#include

#define

function

main()
{

}
```

1) การประกาศ Header File
ในส่วนนี้จะเป็นการแจ้งให้คอมไพล์เลอร์ทราบถึงไฟล์ ต่างๆ ที่เก็บชุดคำสั่งซึ่งได้มีการเรียกใช้ภายในโปรแกรม
2) ส่วนประกาศค่าคงที่และตัวแปรชนิดโกลบอล
3) ส่วนของการสร้างฟังก์ชัน
เป็นส่วนที่ผู้ใช้งานสามารถสร้างฟังก์ชัน หรือโปรแกรมย่อยเพื่อการใช้งานต่างๆ ได้เองตามที่ต้องการ โดยปกติจะเขียนไว้ก่อนฟังก์ชัน main ( ) เพื่อให้ฟังก์ชัน main ( ) สามารถเรียกใช้งานฟังก์ชันต่างๆ ที่เราสร้างขึ้นมาได้
4) ฟังก์ชัน main ( )
สำหรับฟังก์ชันนี้ถือเป็นฟังก์ชันหลักของโปรแกรมภาษา C ที่จำเป็นต้องมีเสมอและจะมีได้เพียงฟังก์ชันเดียวเท่านั้น หน้าที่สำคัญของฟังก์ชันนี้คือ เป็นจุดเริ่มต้นการทำงานของโปรแกรมนั่นเอง ดังนั้นในการเขียนโปรแกรมจึงเริ่มต้นที่ฟังก์ชันนี้เสมอ
ข้อระบุอื่นๆ
คำสั่งต่างๆ ของภาษา C จะต้องใช้ตัวอักษรพิมพ์เล็ก และทุกคำสั่งจะต้องปิดท้ายด้วยเครื่องหมาย " ; "
// หรือ /* */ ใช้ทำหมายเหตุ

ตัวอย่าง

```
/*-----------------------------------------*/
#include <AT89X051.h>
#include <stdio.h>
unsigned int i;
/*-------Set serial port --------------------*/
void ini_serial(void)
{
...
return;
}
/*------MAIN C function ------------------*/
void main (void)
{
    char c;
   ini_serial();
      while (1)
       {
       i = toascii(c);
       printf ("\nYou typed character %c \n", c);
       }
 }
```

การประกาศ Preprocessor ของ ไมโครคอนโทรลเลอร์ ตระกูล AT89c1051 และ AT89c2051

การประกาศไลบรารี่ stdio สำหรับฟังก์ชั่นการใช้งานพอร์ทอนุกรม และ ไลบรารี่ ctype

การเขียนฟังก์ชั่นเพื่อกำหนดรูปแบบของพอร์ทอนุกรม

ส่วนหลักของโปรแกรม ไม่มีการส่งข้อมูลเข้าและไม่มีข้อมูลออก

เรียกใช้ฟังก์ชั่น toascii

ฟังก์ชั่น printf เป็นการส่งข้อมูลออกทางพอร์ทอนุกรม

ส่วน Preprocessor

ส่วนนี้เป็นส่วนบอกให้ตัวแปลรู้ว่าจะใช้ไมโครคอนโทรลเลอร์เบอร์ใด เพื่อให้ไมโครคอนโทรลเลอร์รู้จักกับรีจิสเตอร์และบิตควบคุมต่างๆ ของคอนโทรลเลอร์นั้น การประกาศใช้เครื่องหมายชาร์ป (#) และไดเร็กทีฟ include ร่วมกับชื่อไฟล์ที่บอกรายละเอียดของคอนโทรลเลอร์

- #include <AT89X051.h>
  เป็นไฟล์ Preprocessor ของตระกูล AT89C1051 และ AT89C2051
- #include <89C51AC2.h>
  เป็นไฟล์ Preprocessor ของตระกูล T89C51AC2
- #include <C51rd2.h>
  เป็นไฟล์ Preprocessor ของตระกูล p89C51RD2 ของบริษัทฟิล
- #include <reg52.h>
  เป็นไฟล์ Preprocessor ของตระกูล AT89C52 และ AT89S52/53

ส่วนไลบรารี (Library)

ไลบรารี่เป็นไฟล์ที่มีฟังก์ชั่นบรรจุอยู่ภายใน ไฟล์นี้มีนามสกุลเป็น .h มาพร้อมกับตัวแปล สามารถตรวจสอบได้จากไฟล์ Help

การประกาศใช้ไดเร็กทีฟ #include แล้วต่อด้วยชื่อไฟล์

#include <stdio.h>

ประกอบด้วยฟังก์ชั่นที่ใช้ติดต่อกับพอร์ทอนุกรม ตัวอย่างฟังก์ชั่นได้แก่ printf _getkey getchar gets putchar puts scanf sprintf sscanf ungetchar vprintf และ vsprintf

#include <math.h>

ประกอบด้วยฟังก์ชั่นที่ใช้คำนวณทางคณิตศาสตร์

#include <string.h>

ประกอบด้วยฟังก์ชั่นที่ใช้จัดการเกี่ยวกับสตริง (String) และบัฟเฟอร์ (Buffer)

#include < intrins.h >

ประกอบด้วยฟังก์ชั่นเกี่ยวกับการหมุนบิต และการตรวจสอบสถานะของเลขแบบ Floating-point

ส่วนค่าคงที่ของโปรแกรม

การประกาศค่าคงที่ในภาษา C ใช้ไดเร็กทีฟ #defined แล้วตามด้วยค่าคงที่ดังนี้

#define setValue 99 กำหนดค่า setValue ให้เป็น 99

หรือใช้แทนการเขียนคำสั่งเพื่อให้สื่อความหมายได้ชัดเจน

#define Motor_ON (P2_7 = 1)

#define Motor_OFF (P2_7 = 0)

ส่วนการใช้งานร่วม (Global Declarations)

ตัวแปรแบบใช้งานร่วมหรือ ตัวแปรแบบโกลบอล

หมายถึงตัวแปรที่สามารถเรียกใช้ได้ทุกที่ภายในโปรแกรม

การประกาศให้ตัวแปรเป็นแบบโกลบอล ให้ประกาศไว้นอกขอบเขตของส่วนโปรแกรมใดๆ

ตัวแปรแบบโลคัล (Local)

ตัวแปรใช้ได้เฉพาะภายในส่วนของโปรแกรมใดโปรแกรมหนึ่ง

การประกาศให้ประกาศไว้ภายในโปรแกรมที่ต้องการใช้งาน

```
/*-------------------------------------------*/
#include <AT89X051.h>
#include <stdio.h>
unsigned int i;
/*-------Set serial port ---------------------*/
void ini_serial(void)
{
...
return;
}
/*------MAIN C function  --------------------*/
void main (void)
{
  char c;
  ini_serial();
    while (1)
    {
   i = toascii(c);
   printf ("\nYou typed character %c \n", c);
    }
}
```


ฟังก์ชั่น (Function)

- เป็นส่วนที่ใช้ประกาศถึงฟังก์ชั่นของโปรแกรม
- โปรแกรมหนึ่งๆ จะมีกี่ฟังก์ชั่นก็ได้
- ต้องมีฟังก์ชั่นหลักอยู่หนึ่งฟังก์ชั่น ใช้เป็นฟังก์ชั่นสำหรับเริ่มการทำงานของโปรแกรม เรียกฟังก์ชั่นนี้ว่า main
- ส่วนของคำสั่งของฟังก์ชั่นจะอยู่ภายในวงเล็บปีกกา { }และสามารถกำหนดให้มีการส่งผ่านตัวแปรได้
- ถ้าไม่ต้องการให้ส่งตัวแปรเข้าจะใช้คำว่า void ไว้ภายในวงเล็บและถ้าไม่มีการส่งค่าคืน จะใส่คำว่า void ไว้หน้าชื่อฟังก์ชั่น

```
void   func_abcd(void)
{
  ..............................
}
```

ส่วนหมายเหตุ (Comment)

- เป็นส่วนที่ให้ผู้เขียนโปรแกรมใช้อธิบายความหมายของโปรแกรมหรือคำสั่งที่เขียน เพื่อให้ง่ายต่อการทำความเข้าใจ ส่วนนี้ต้องอยู่ในเครื่องหมาย /* ..... */ หรืออยู่หลังเครื่องหมาย //

```
/*-------------------------------------------------------------------------
Set serial port for 9600 baud at 11.0592 MHz.  Note that we use Timer 1
for the baud rate generator.
-------------------------------------------------------------------------*/

/* ------------------------  Main Function ------------------------------*/


// PS2 I/P Program
```


ข้อกำหนดอื่นๆ

- คำสั่งส่วนใหญ่ต้องเขียนด้วยอักษรตัวพิมพ์เล็ก
- ตัวแปรสามารถตั้งชื่อเป็นตัวพิมพ์เล็กหรือใหญ่ก็ได้แต่ต้องเรียกใช้ให้ถูกตามตัวพิมพ์ และต้องไม่ตรงกับคำสงวนของตัวคอมไพเลอร์ที่ใช้
- ทุกคำสั่งต้องมีเครื่องหมาย ; แสดงการจบคำสั่ง และสามารถเขียน 1 คำสั่งใน 1 บรรทัดหรือจะเขียนต่อกันยาวก็ได้ แต่ก็จะทำให้แก้ไขโปรแกรมได้ยาก
- คำสั่งใดๆในโปรแกรมสามารถมีเลเบลได้ถ้าต้องการ และต้องมีเครื่องหมาย : ตามท้ายด้วย

ค่าคงที่และตัวแปร

ชนิดของตัวแปร

ชนิด (Type)	ขนาดความกว้าง(Width)		ช่วงของค่าที่เก็บ(Range)
	Bits	Bytes	
bit †	1		0 to 1
signed char	8	1	-128 to +127
unsigned char	8	1	0 to 255
enum	8 / 16	1 or 2	-128 to +127 or -32768 to +32767
signed short	16	2	-32768 to +32767
unsigned short	16	2	0 to 65535
signed int	16	2	-32768 to +32767
unsigned int	16	2	0 to 65535
signed long	32	4	-2147483648 to 2147483647
unsigned long	32	4	0 to 4294967295
float	32	4	±1.175494E-38 to ±3.402823E+38
sbit †	1		0 to 1
sfr †	8	1	0 to 255
sfr16 †	16	2	0 to 65535

ตัวดำเนินการ

ตัวดำเนินการทางคณิตศาสตร์

เครื่องหมาย	ความหมาย
+	บวก
-	ลบ
*	คูณ
/	หาร
%	หารแบบใช้เศษของการหาร(Mod)

ตัวดำเนินการ

ตัวดำเนินการที่ใช้ในการแปลงค่า

เครื่องหมาย	ความหมาย
=	นำค่าทางขวามาให้ทางซ้าย
++	เพิ่มค่าขึ้น 1
--	ลดค่าลง 1
+=	เพิ่มค่าเท่ากับค่าทางขวา
-=	ลดค่าเท่ากับค่าทางขวา
*=	นำตัวเองคูณกับค่าทางขวา
/=	นำตัวเองหารกับค่าทางขวา
%=	นำตัวเองหารกับค่าทางขวาเอาเศษ
&=	นำตัวเองมาAND กับค่าทางขวา
|=	นำตัวเองมา OR กับค่าทางขวา
^=	นำตัวเองมา XORกับค่าทางขวา
<<=	นำตัวเองมาเลื่อนบิตไปทางซ้ายจำนวนครั้งเท่ากับค่าทางขวา
>>=	นำตัวเองมาเลื่อนบิตไปทางขวาจำนวนครั้งเท่ากับค่าทางขวา

ตัวดำเนินการ

ตัวดำเนินการที่ใช้ในการเปรียบเทียบและตัวดำเนินการทางบิต

เครื่องหมาย	ความหมาย
==	เท่ากับ
!=	ไม่เท่ากับ
!	ตรงกันข้าม(NOT)
&&	และ (AND)
>=	มากกว่าหรือเท่ากับ
<=	น้อยกว่าหรือเท่ากับ
||	หรือ(OR)
<	น้อยกว่า
>	มากกว่า

เครื่องหมาย	ความหมาย
&	AND บิต
|	OR บิต
^	XOR บิต
<<	Left shift บิต
>>	Right shift บิต
~	One' Complement (Inverse)

อะเรย์และพอยน์เตอร์ (Array and Pointer)

ตัวแปร์อะเรย์

ตัวแปรอะเรย์ คือกลุ่มของตัวแปร เช่นกลุ่มของตัวแปร char กลุ่มของตัวแปร int สมาชิกของกลุ่มตัวแปรนี้มีจำนวนจำกัด เฉพาะเจาะจงเมื่อกำหนดจำนวนแล้วไม่สามารถลดหรือเพิ่มได้ ดังนั้นการกำหนดอะเรย์สำหรับไมโครคอนโทรลเลอร์ต้องคำนึงถึงหน่วยความจำที่มีใช้งานด้วย อะเรย์นี้สามารถกำหนดให้เป็นแบบหลายๆมิติได้ ขึ้นอยู่กับว่าตัวแปรภาษาย่อมให้มีได้เท่าไร

รูปแบบการกำหนดอะเรย์

แบบ 1 มิติ ชนิดของตัวแปร ชื่อตัวแปร[n]
แบบ 2 มิติ ชนิดของตัวแปร ชื่อตัวแปร[n][m]
โดย n และ m หมายถึงจำนวนสมาชิก

ตัวอย่างตัวแปรและการกำหนดค่าให้กับตัวแปร

```
void main( void )
{
        unsigned char x, y[3], z[2][3];
        x = 4;
        y[0] = 1;
        y[1] = 2;
        y[2] = 3;

        z[0][0] = 10;
        z[0][1] = 11;
        z[0][2] = 12;
        z[1][0] = 13;
        z[1][1] = 14;
        z[1][2] = 15;
}
```

- x เป็นตัวแปร
- y เป็นตัวแปร 1 มิติ
- z เป็นตัวแปร 2 มิติ

ตัวแปรแบบพอยเตอร์

ตัวแปรพอยเตอร์หรือตัวแปรตัวชี้ เป็นตัวแปรที่มีความสำคัญมากในภาษาซี เพราะว่าให้ความยืดหยุ่นในการใช้งานได้ดี คล้ายกับภาษาแอสเซมบลี้ ตัวแปรแบบพอยเตอร์มีหน้าที่เก็บตำแหน่งของตัวแปรอื่นๆ ว่าอยู่ที่ใดในหน่วยความจำ

รูปแบบการกำหนดตัวแปรพอยเตอร์

ชนิดของตัวแปร ***ชื่อตัวแปร**

ตัวอย่างการใช้ตัวแปรแบบพอยเตอร์

```
void main( void )
{
        unsigned char *ptr1;
        unsigned char *ptr2;
        unsigned char a;
        ptr1 = 0x40;
        ptr2 = 0x50;
        a = *ptr1;
        a = a+5;
        *ptr2 = a;
}
```

เป็นการกำหนดพอยเตอร์ขึ้น 2 ตัว เพื่อใช้ชี้ตำแหน่งของหน่วยความจำข้อมูลภายใน

ตำแหน่งที่ 40H

และตำแหน่งที่ 50H

และระบุตัวแปร a เป็นแบบ Unsigned char

การอ่านข้อมูลจากหน่วยความจำมาไว้ที่ a ก็จะอ่านที่ละ 8 บิต

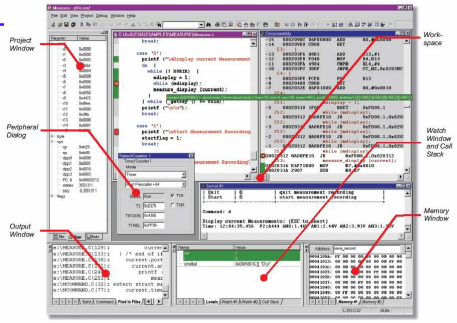
เท่านั้น แต่ถ้าระบุ a เป็นตัวแปรแบบอื่นเช่น char ก่านอ่านก็จะทำที่ละ 16 บิต ดังนั้นการกำหนดชนิดตัวแปรนี้ต้องระมัดระวังให้ดี

คำสั่งควบคุมต่างๆในภาษา C

- If
- if-else
- Switch
- For
- While
- do-while

µVision2

Project Manager Editor & Debugger



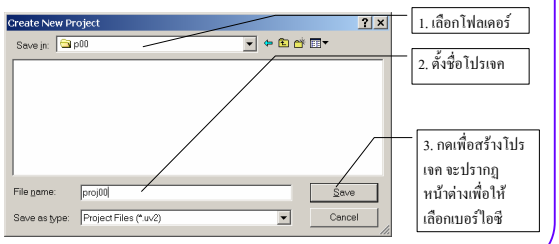
การใช้งาน uVision2 พัฒนาโปรแกรมสำหรับ MCS-51

1. สร้างโปรเจคใหม่
2. เลือก Option
3. สร้างไฟล์โปรแกรมภาษาแอสเซมบลี้หรือภาษาซี
4. เพิ่มไฟล์ (add file) เข้ากับโปรเจค
5. แปล
6. ทดสอบโปรแกรม (Debug)

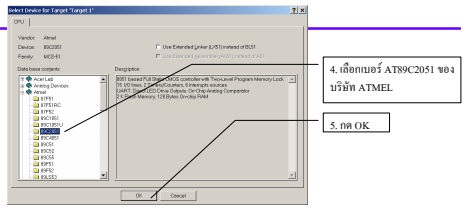
1. สร้างโปรเจคใหม่

1.1 project new

ใช้คำสั่ง New Project ในเมนู Project ตั้งโปรเจคลงในโฟลเดอร์ที่ต้องการ

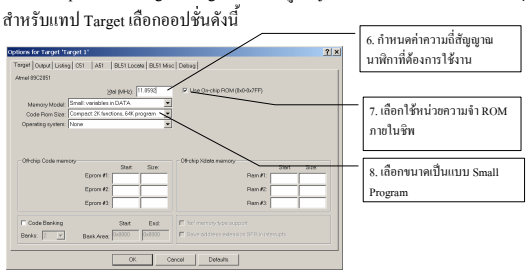


1.2 เลือก ซีพียู 89c2051

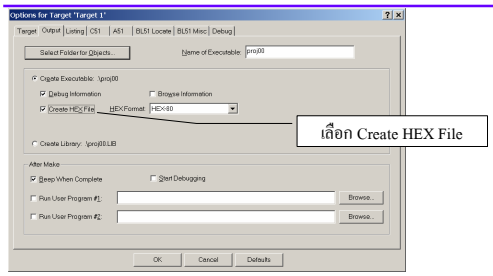


2. กำหนดค่า Optiont

ใช้คำสั่ง Options for Target ‘Target 1’ ในเมนู Project เพื่อกำหนดพารามิเตอร์ต่างๆ สำหรับแทป Target เลือกออปชั่นดังนี้



สำหรับแทป Output เลือกออปชั่นดังนี้



3.สร้างซอร์สไฟล์ภาษาแอสเซมบลี้หรือซี

ใช้เมนู File คำสั่ง New แล้วพิมพ์โปรแกรมดังนี้

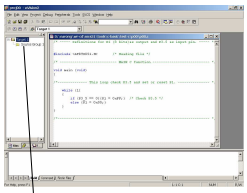
ตัวอย่างภาษาซี

```
/* ----- Definitions for P1 (8 bits)as output and P3.5 as input pin. ----- */

#include <AT89X051.H>                          /* Heading file*/

/* -------------------------- MAIN C function -------------------------- */

void main (void)
{

/*-------------- This loop check P3.5 and set or reset P1. -------------- */

            while (1)
            {
               if (P3_7 == 0){P1 = 0xFF;}          /* Check P3.5 */
                           else {P1 = 0x00;}
            }
}
```

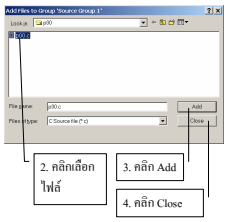
เมื่อเขียนเสร็จแล้วใช้คำสั่ง Save As ในเมนู File เซฟโปรแกรมไว้ในโฟลเดอร์ที่กำหนดไว้

4.เพิ่มไฟล์ (add file) เข้ากับโปรเจค

ที่หน้าต่าง File ให้เลื่อนเมาส์ไปชี้ที่ Source Group1 แล้วคลิกขวา เพื่อเลือกคำสั่ง Add File to Group 'Source Group 1'



1.คลิกขวาเลือกคำสั่ง Add File to Group 'Source Group 1'



2. คลิกเลือกไฟล์

3. คลิก Add

4. คลิก Close

5. แปลภาษาซีให้เป็นไฟล์เฮกซ์

ใช้คำสั่ง Build Target ในเมนู Project เพื่อแปลไฟล์

6. ทดสอบโปรแกรม (Debug)

1. ใช้คำสั่ง Start/Stop Debug Session ในเมนู Debug
2. ใช้คำสั่ง I/O Ports ในเมนู Peripherals เลือกแสดง Port_1 และ Port_3
3. ใช้คำสั่ง Go ในเมนู Debug เพื่อให้โปรแกรมทดสอบทำงาน ในขณะนี้ทดลองใช้เมาส์คลิกตำแหน่ง Pins บิตที่ 5 ของพอร์ท 3 จะปรากฏว่าค่าพอร์ท 1 มีการเปลี่ยนแปลง
4. การใช้ Toolbox
5. การใช้ Debug Function editor

4. คำสั่งเกี่ยวกับหน่วยความจำ

B Bit—addressable RAM memory (BIT).

C Code memory (CODE).

D Internal directly-addressable RAM memory of the 8051 (DATA).

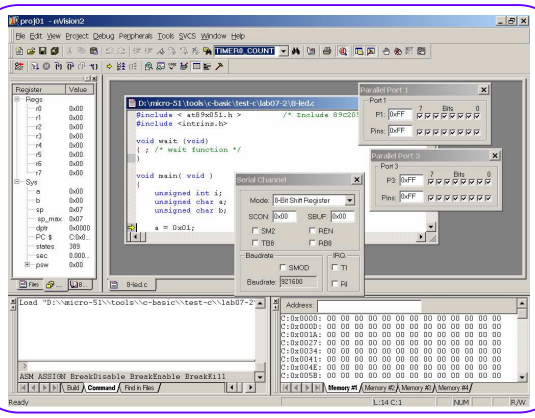
I Internal indirectly-addressable RAM memory of the 8051 (IDATA).

X External RAM memory (XDATA).

เช่น

D:0x00	แสดงค่าในหน่วยความจำภายใน
C:0X0000	แสดงค่า Code Memory
X:0x001000	แสดงค่าในหน่วยความจำข้อมูลภายนอกตำแหน่งที่ 1000 เป็นต้นไป

I:0x00



จบ