ỦY BÂN NHÂN DÂN TP THỦ ĐỨC
TRƯỜNG TRUNG CẤP NGHỀ ĐÔNG SÀI GÒN

GIÁO TRÌNH

Tên mô đun: Vi điều khiển cơ bản

NGHỀ: ĐIỆN TỬ CÔNG NGHIỆP
TRÌNH ĐỘ TRUNG CẤP

(Ban hành kèm theo Quyết định số: ...../QĐ-TCN ngày ..... tháng ... năm 20... của Hiệu trưởng Trường trung cấp nghề Đông Sài Gòn)

TP Thủ Đức, năm 2023
(Lưu hanh nội bộ)

8051 Cơ Bản

MỤC LỤC

Bài 1: Giới thiệu về vi điều khiển

Chú ý : Đây hoàn toàn là phần lí thuyết, mình đã rút gọn tối đa, do đó các bạn nên đọc hết. Chưa nên thực hành vội vì tôi sẽ hướng dân sau.

1.1 Tiêu chuẩn trong lựa chọn một bộ vi điều khiển

Là khả năng sẵn sàng đáp ứng về số lượng trong hiện tại và tương lai. Đối với một số nhà thiết kế điều này là quan trong hơn cả . Hiện nay, các bộ vi điều khiển 8 bit đứng đầu là họ 8051 có số lương lớn nhất các nhà cung cấp đa dạng (nhiều nguồn). Nhà cung cấp có nghĩa là nhà sản xuất bên cạnh nhà sáng chế của bộ vi điều khiển. Trong trường hợp 8051 thì nhà sáng chế của nó là Intel, nhưng hiện nay có rất nhiều hãng sản xuất nó (cũng như trước kia đã sản xuất).

Các hãng này bao gồm: Intel, Atmel, Philips/signe-tics, AMD, Siemens, Matra và Dallas, Semicndictior.

Bảng địa chỉ của một số hãng sản xuất các thành viên của họ 8051.

Hãng	Địa chỉ Website
Intel	www.intel.com/design/mcs51
Antel	www.atmel.com
Plips/ Signetis	www.semiconductors.philips.com
Siemens	www.sci.siemens.com
Dallas Semiconductor	www.dalsemi.com

8051 là một bộ xử lý 8 bit có nghĩa là CPU chỉ có thể làm việc với 8 bit dữ liệu tại một thời điểm. Dữ liệu lớn hơn 8 bit được chia ra thành các dữ liệu 8 bit để cho xử lý. 8051 có tất cả 4 cổng vào - ra I/O mỗi cổng rộng 8 bit. Các nhà sản xuất đã cho xuất xưởng chỉ với 4K byte ROM trên chíp.

Bảng các đặc tính của 8051 đầu tiên.

Đặc tính	Số lượng
ROM trên chíp	4K byte
RAM	128 byte
Bộ định thời	2
Các chân vào - ra	32
Cổng nối tiếp	1
Nguồn ngắt	6

1.2.Vi điều khiển và vi xử lí:

Xin nhắc đến cái máy tính của bạn, con chíp Intel hay ADM của bạn là 1 bộ vi xử lí, nó không có RAM, ROM,cổng IO và các thiết bị ngoại vi on Chip. Còn vi điều khiển chứa 1 bộ vi xử lí và RAM,ROM, cổng IO, và có thể có các thiết bị ngoại vi.

1.3. Sơ đồ chân vi điều khiển 8051:

Là IC đóng vỏ dạng DIP có 40 chân, mỗi chân có một kí hiệu tên và có các chức năng như sau:

Chân 40: nối với nguồn nuôI +5V.

Chân 20: nối với đất(Mass, GND).

Chân 29 (PSEN)(program store enable) là tín hiệu điều khiển xuất ra của 8051, nó cho phép chọn bộ nhớ ngoài và được nối chung với chân của OE (Outout Enable) của EPROM ngoài để cho phép đọc các byte của chương trình. Các xung tín hiệu PSEN hạ thấp trong suốt thời gian thi hành lệnh. Những mã nhị phân của chương trình được

đọc từ EPROM đi qua bus dữ liệu và được chốt vào thanh ghi lệnh của 8051 bởi mã lệnh.(chú ý việc đọc ở đây là đọc các lệnh (khác với đọc dữ liệu), khi đó VXL chỉ đọc các bit opcode của lệnh và đưa chúng vào hàng đợi lệnh thông qua các Bus địa chỉ và dữ liệu)

Chân 30 (ALE : Adress Latch Enable) là tín hiệu điều khiển xuất ra của 8051, nó cho phép phân kênh bus địa chỉ và bus dữ liệu của Port 0.

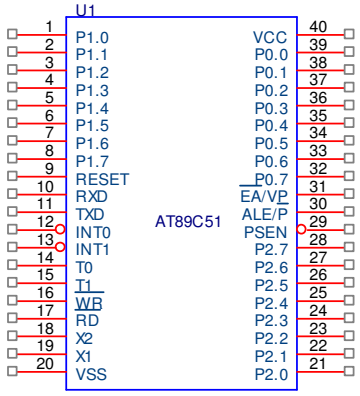
Chân 31 (EA : Eternal Acess) được đưa xuống thấp cho phép chọn bộ nhớ mã ngoàI đối với 8051.

Đối với 8051 thì : EA = 5V : Chọn ROM nội. EA = 0V : Chọn ROM ngoại.

32 chân còn lại chia làm 4 cổng vào ra:

Vào ra tức là có thể dùng chân đó để đọc mức logic (0;1 tương ứng với 0V ; 5V)vào hay xuất mức logic ra(0;1)

P0 từ chân 39 → 32 tương ứng là các chân P0_0 → P0_7
P1 từ chân 1 → 8 tương ứng là các chân P1_0 → P1_7
P2 từ chân 21 →28 tương ứng là các chân P2_0 → P2_7
P3 từ chân 10 → 17 tương ứng là các chân P3_0 → P3_7



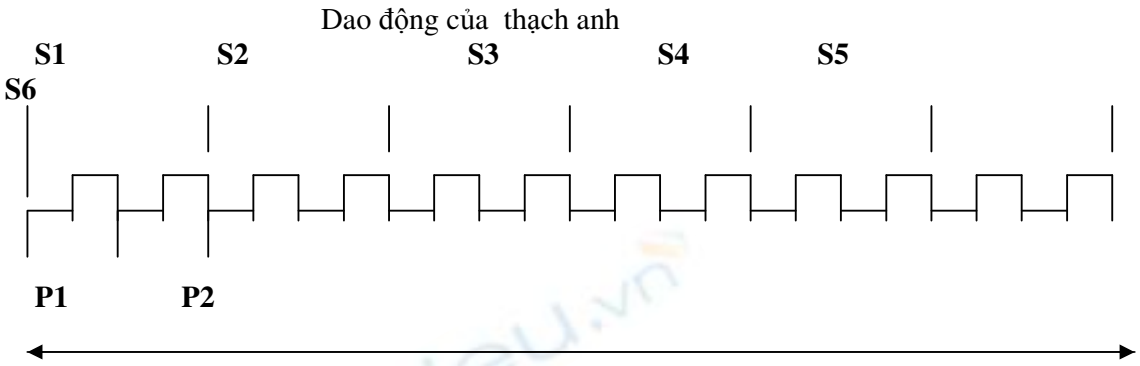
P0	P1	P2	P3	Port's Bit
P0.0	P1.0	P2.0	P3.0	D0
P0.1	P1.1	P2.1	P3.1	D1
P0.2	P1.2	P2.2	P3.2	D2
P0.3	P1.3	P2.3	P3.3	D3
P0.4	P1.4	P2.4	P3.4	D4
P0.5	P1.5	P2.5	P3.5	D5
P0.6	P1.6	P2.6	P3.6	D6
P0.7	P1.7	P2.7	P3.7	D7

Riêng cổng 3 có 2 chức năng ở mỗi chân như trên hình vẽ:

P3.0 – RxD : chân nhận dữ liệu nối tiếp khi giao tiếp RS232(Cổng COM).
P3.1 _ TxD : phân truyền dữ liệu nối tiếp khi giao tiếp RS232.
P3.2 _ INTO : interrupt 0 , ngắt ngoài 0.
P3.3 _ INT1: interrupt 1, ngắt ngoài 1.
P3.4 _T0 : Timer0 , đầu vào timer0.
P3.5_T1 : Timer1, đầu vào timer 1.
P3.6_ WR: Write, điều khiển ghi dứ liệu.
P3.7 _RD: Read , điều khiển đọc dữ liệu.

Chân 18, 19 nối với thạch anh tạo thành mạch tạo dao động cho VĐK

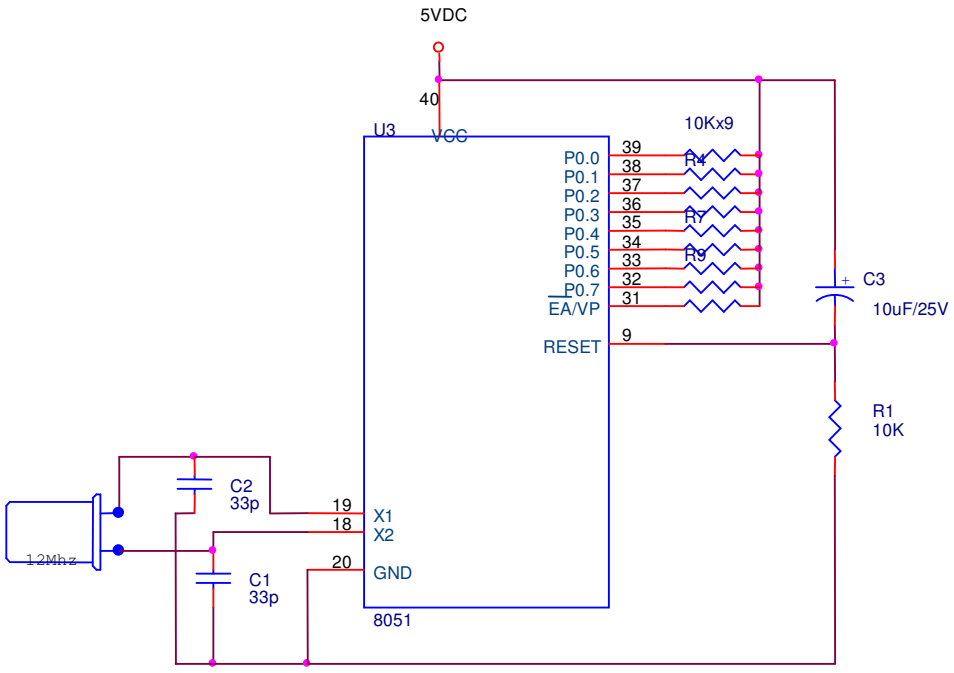
Tần số thạch anh thường dùng trong các ứng dụng là : 11.0592Mhz(giao tiếp với cổng com máy tính) và 12Mhz Tần số tối đa 24Mhz. Tần số càng lớn VĐK xử lí càng nhanh.



Riêng cổng 3 có thêm chức năng như dưới đây

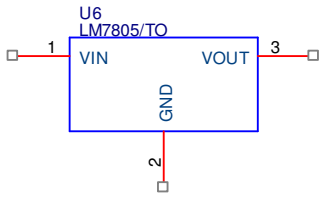
P3.0 – RxD : chân nhận dữ liệu nối tiếp khi giao tiếp RS232(Cổng COM).
P3.1 _ TxD : phần truyền dữ liệu nối tiếp khi giao tiếp RS232.
P3.2 _ INTO : interrupt 0 , ngắt ngoài 0.
P3.3 _ INT1: interrupt 1, ngắt ngoài 1.
P3.4 _T0 : Timer0 , đầu vào timer0.
P3.5_T1 : Timer1, đầu vào timer 1.
P3.6_ WR: Write, điều khiển ghi dứ liệu.
P3.7 _RD: Read , điều khiển đọc dữ liệu.

Chân 18, 19 nối với thạch anh tạo thành mạch tạo dao động cho VĐK
Tần số thạch anh thường được dùng trong các ứng dụng là : 11.0592Mhz(giao tiếp với cổng com máy tính) và 12Mhz
Tần số tối đa 24Mhz. Tần số càng lớn VĐK xử lý càng nhanh.



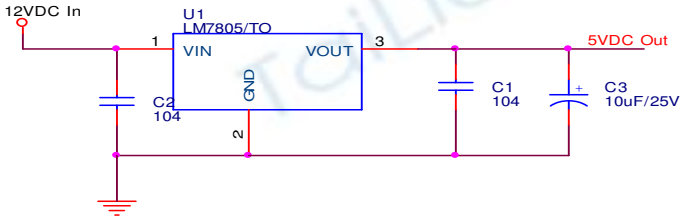
Mạch này chưa có khối nguồn để tạo nguồn 5V các bạn dùng con IC sau:

Sơ đồ chân:

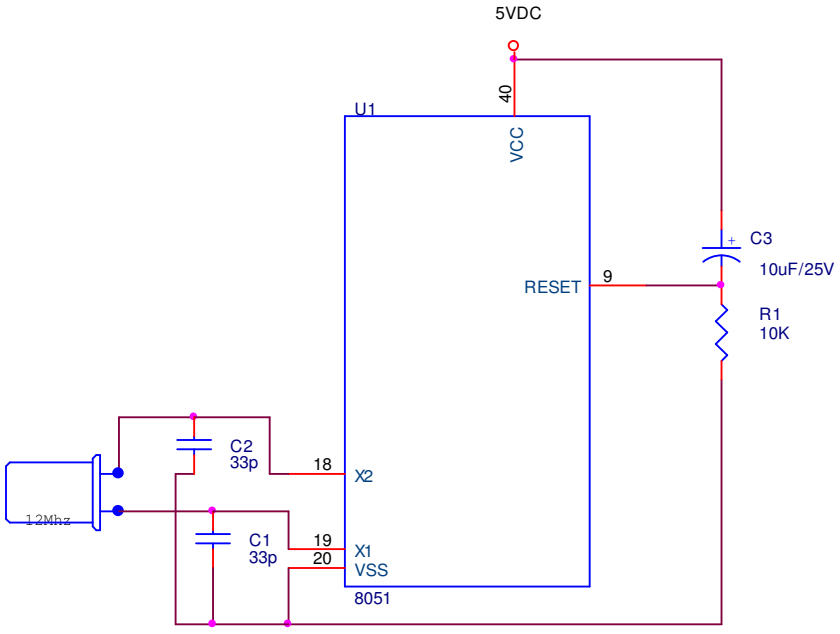


Giới thiệu IC ổn áp 7805 : Đầu vào > 7V đầu ra 5V 500mA. Mạch ổn áp: cần cho VĐK vì nếu nguồn cho VĐK không ổn định thì sẽ treo VĐK, không chạy đúng, hoặc reset liên tục, thậm chí là chết chíp.

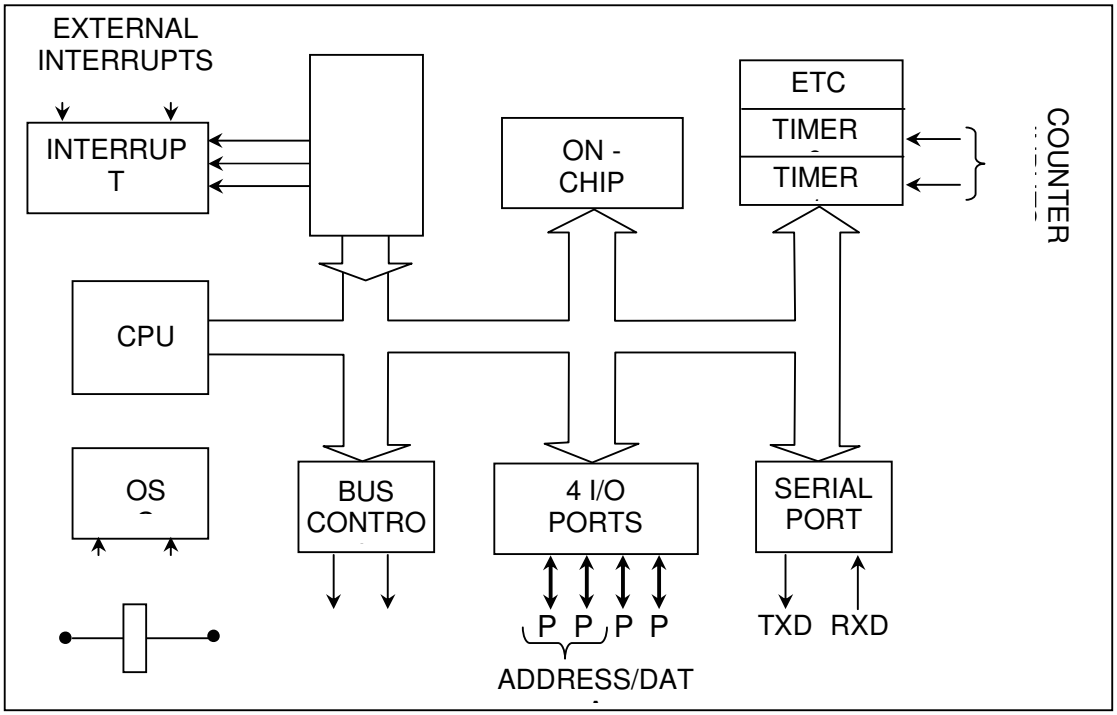
Mạch nguồn



1.4. Mạch vi điều khiển cơ bản



1.5. Cấu trúc vi điều khiển 89c51



Chú ý:Sơ đồ khối bên trong con 8051 có những tài nguyền . Interrupt, Ram, Timer, Serial prort.

1.6.Các thành viên khác của họ 8051:

Có hai bộ vi điều khiển thành viên khác của họ 8051 là 8052 và 8031.

Bộ vi điều khiển 8052: 8052 có tất cả các đặc tính chuẩn của 8051 ngoài ra nó có thêm 128 byte RAM và một bộ định thời nữa. Hay nói cách khác là 8052 có 256 byte RAM và 3 bộ định thời. Nó cũng có 8K byte ROM. Trên chíp thay vì 4K byte như 8051.

Bảng : So sánh các đặc tính của các thành viên họ 8051.

Đặc tính	**8051**	**8052**
ROM trên chip	**4K byte**	**8K byte**
RAM	**128 byte**	**256 byte**
Bộ định thời	**2**	**3**
Cổng nối tiếp	**1**	**1**
Nguồn ngắt	**6**	**8**

Do vậy tất cả mọi chương trình viết cho 8051 đều chạy trên 8052 nhưng điều ngược lại là không đúng.Đặc biệt : Một nhà sản xuất chính của họ 8051 khác nữa là Philips Corporation. Hàng này có một dải lựa chọn dộng lớn cho các bộ vi điều khiển

họ 8051. Nhiều sản phẩm của hãng đã có kèm theo các đặc tính như các bộ chuyển đổi ADC, DAC, chân PWM, cổng I/0 mở rộng .

Update sản phẩm 8051 mới tại các trang web của các nhà sản xuất địa chỉ đã có ở phần giới thiệu. Chủ yếu:www.atmel.com

1.7.Ram nội và các thanh ghi

Cỏc cú địa chỉ và FFH

F0	**F7**	**F6**	**F5**	**F4**	**F3**	**F2**	**F1**	**F0**
E0	**E7**	**E6**	**E5**	**E4**	**E3**	**E2**	**E1**	**E0**
D0	**D7**	**D6**	**6D**	**6C**	**6B**	**6A**	**69**	**68**
B8	**-**	**-**	**-**	**BC**	**BB**	**BA**	**B9**	**B8**
B0	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
A8	**AF**	**AE**	**AD**	**AC**	**AB**	**AA**	**A9**	**A8**
A0	**A7**	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**
99	**Không định địa chỉ từng bít**							
98	**9F**	**9E**	**9D**	**9C**	**9B**	**9A**	**99**	**98**
90	**97**	**96**	**95**	**94**	**93**	**92**	**91**	**90**
8D	**Không định địa chỉ từng bít**							
8C	**Không định địa chỉ từng bít**							
8B	**Không định địa chỉ từng bít**							
8A	**Không định địa chỉ từng bít**							
89	**Không định địa chỉ từng bít**							
88	**8F**	**8E**	**8D**	**8C**	**8B**	**8A**	**89**	**88**
87	**Không định địa chỉ từng bít**							
83	**Không định địa chỉ từng bít**							
82	**Không định địa chỉ từng bít**							
81	**Không định địa chỉ từng bít**							
80	**87**	**86**	**85**	**84**	**83**	**82**	**81**	**80**
THANH GHI CHỨC NĂNG ĐẶC BIỆT								

thanh ghi SFR nằm giữa 80H các địa chỉ này

ở trên 80H, vì các địa chỉ từ 00 đến 7FH là địa chỉ của bộ nhớ RAM bên trong 8051. Không phải tất cả mọi địa chỉ từ 80H đến FFH đều do SFR sử dụng, nhưng vị trí ngăn nhớ từ 80H đến FFH chưa dùnglà để dữ trữ và lập trình viên 8051 cũng không được sử dụng.

Bảng : chức năng của thanh ghi chức năng đặc biệt SFR

SFR định địa chỉ từng bit(những thanh ghi cần nhớ đối khi lập trình cơ bản C)

Thanh ghi / Bit	Ký hiệu	Chức năng					
TMOD		Chọn model cho bộ định thời 1					
7	GATE	Bít điều khiển cổng. Khi được set lên 1, bộ định thời chỉ hoạt động trong khi INT1 ở mức cao					
6	C/T	Bít chọn chức năng đếm hoặc định thời:					
		1= đếm sự kiện					
		0= định thời trong một khoảng thời gian					
5	M1	Bit chọn chế độ thứ nhất					
4	M0	Bit chọn chế độ thứ 2					
		M1	M0	Chõ đé	Chức năng		
		0	0	0	Chế độ định thời 13 bit		
		0	1	1	Chế độ định thời 16 bit		
		1	0	2	Chế độ tự động nạp lại 8 bit		
		1	1	3	Chế độ định thời chia xẻ		
3	GATE	Bit điều khiển cổng cho bộ định thời 0					
2	C/T	Bit chọn chức năng đếm / định thời cho bộ định thời 0					
1	M1	Bit chọn chế độ thứ nhất cho bộ định thời 0					
0	M0	Bit chọn chế độ thứ 2 cho bộ định thời 0					
TF1 TR1	TF1	TR0	IE1	IT1	IE0	IT0	
Thanh ghi / Bit	Ký hiệu	Chức năng					
TCON		Điều khiển bộ đinh thời					
TCON.7	TF1	Cờ tràn của bộ định thời 1. Cờ này được set bởi phần cứng khi có tràn, được xoá bởi phần mềm, hoặc bởi phần cứng khi bộ vi xử lý trỏ đến trình phục vụ ngắt					
TCON.6	TR1	Bit điều khiển hoạt động của bộ định thời 1. Bit này được set hoặc xoá bởi phần mềm để điều khiển bộ định thời hoạt động hay ngưng					
TCON.5	TF0	Cừ tràn của bộ định thời 0					
TCON.4	TR0	Bit điều khiển hoạt động của bộ định thời 0					
TCON.3	IE1	Cừ ngắt bên ngoài 1 (kích khởi cạnh). Cờ này được set bởi phần cứng khi có cạnh âm (cuống) xuất hiện trên chân INT1, được xoá bởi phần mềm, hoặc phần cứng khi CPU trỏ đến trình phục vụ ngắt					

TCON.2	IT1	Cờ ngắt bên ngoài 1 (kích khởi cạnh hoặc mức). Cờ này được set hoặc xoá bởi phần mềm khi xảy ra cạnh âm hoặc mức thấp tại chân ngắt ngoài
TCON.1	IE0	Cờ ngắt bên ngoài 0 (kích khởi cạnh)
TCON.0	IT0	Cờ ngắt bên ngoài 0 (kích khởi cạnh hoặc mức)

EA	ET2	ES	ET1	EX1	EX0	ET0

IE	Điều khiển các nguồn ngắt (0: không cho phép; 1: cho phép)
IE.7	EA	Cho phép/ không cho phép toàn cục
IE.6	---	Không sử dụng
IE.5	ET2	Cho phép ngắt do bộ định thời 2
IE.4	ES	Cho phép ngắt do port nối tiếp
IE.3	ET1	Cho phép ngắt cho bộ định thời 1
IE.2	EX1	Cho phép ngắt từ bên ngoài (ngắt ngoài 1)
IE.1	EX0	Cho phép ngắt từ bên ngoài (ngắt ngoài 0)
IE.0	ET0	Cho phép ngắt do bộ định thời 0

CHÚ Ý

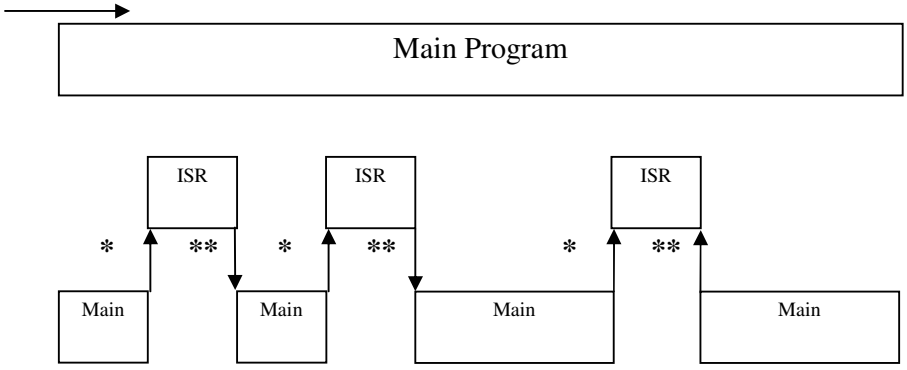
3 thanh ghi này cũng rất cơ bản, nhớ tên thanh ghi, tên các bít trong thanh ghi, chức năng từng thanh ghi và từng bít trong thanh ghi.

1.8.Giới thiệu sơ qua các nguồn ngắt:

Ngắt do	Cờ	Địa chỉ vector
Reset hệ thống	RST	0000H
Ngắt ngoài 0	IE0	0003H
Bộ định thời 0	TF0	000BH
Ngắt ngoài 1	IE1	0013H
Bộ định thời 1	TF1	001BH
Port nối tiếp	RI hoặc TI	0023H
Bộ định thời 2	TF2 hoặc EXF2	002BH

Timer

Program excution without interrut



Program excution with interrupt.

Một chương trình chính không có ngắt thì chạy liên tục, còn chương trình có ngắt thì cứ khi nào điều kiện ngắt được đảm bảo thì con trỏ sẽ nhảy sang hàm ngắt thực hiện xong hàm ngắt lại quay về đúng chỗ cũ và thực hiện tiếp chương trình. Ta có 1 ví dụ như sau: Bạn đang ăn cơm , có tiếng điện thoại , bạn đạt bát cơm ra nghe điện thoại , nghe xong lại quay về bưng bát cơm lên ăn tiếp. Thì quá trình ăn cơm của bạn là chương trình chính,có điện thoại gọi đến là điều kiện ngắt, bạn ra nghe điện thoại là thực hiện chương trình ngắt(Interrupt Service Rountine),quay về ăn cơm tiếp là tiếp tục thực hiện chương trình chính.

Ngắt đối với người mới học vi điều khiển là rất khó hiểu, vì đa số các tài liệu đều không giải thích ngắt để làm gì. Có nhiều loại ngắt khác nhau nhưng tất cả đều có chung 1 đặc điểm, ngắt dùng cho mục đích đa nhiệm. Đa tức là nhiều, nhiệm tức là nhiệm vụ. Thực hiện nhiều nhiệm vụ .Các bạn nhìn vào tiền trình của hàm main với chương trình có ngắt :

Chương trình chính đang chạy, ngắt xảy ra, thực hiện hàm ngắt rồi quay lại chương trình chính. Chương trình trong vi điều khiển khác với ví dụ ăn cơm nghe điện thoại của tôi ở chỗ , thời gian thực hiện hàm chính là rất lớn,thời gian thực hiện hàm ngắt là rất nhỏ, cho nên thời gian thực thi hàm ngắt không ảnh hưởng nhiều lắm đên chức năng hàm chính. Như vậy trong hàm ngắt các bạn làm 1 việc, trong hàm chính các bạn làm 1 việc

như vậy coi như các bạn làm được 2 việc(đa nhiệm) trong 1 quang thời gian tương đối ngắn cỡ mS, chứ thực ra tại 1 thời điểm vi điều khiển chỉ thực thi 1 lệnh.

Ví dụ : Bạn thử nghĩ xem làm thế nào để vừa điều chế xung PWM để điều chỉnh tốc độ động cơ , vừa đọc các cảm biến đầu vào mà tốc độ động cơ phụ thuộc đầu vào cảm biến.

BÀI 2: NGÔN NGỮ LẬP TRÌNH C

2.1.Cấu trúc một chương trình:

```
//Đính kèm các file
#include <file.h>
#include <file.c>
//Khai báo biến toàn cục
unsigned char x,y;
int z;
long n=0;
//Khai báo và định nghĩa các hàm
void Hàm1(void)
{
  …//Các câu lệnh
}

void Hàm2(unsigned char x)
{
  …//Các câu lệnh
}
```

```
//Hàm chính bắt buộc chương trình nào cũng phảI có
void main(void)
{
  …//Các câu lệnh
}
```

Các câu lệnh trong hàm chính có thể có lời gọi các hàm đã khai báo ở trên hoặc không

Khi có lời gọi hàm nào thì chương trình nhảy đến hàm đó thực hiện hàm đó xong con trỏ lại quay về chương trình chính(hàm main) thực hiện tiếp các hàm hoặc câu lệnh.

Các câu lệnh trong C kết thúc bằng dấu ";"

Các lời giảI thích được đặt trong dấu: Mở đầu bằng "/*" kết thúc bằng "*/"

Nếu lời giảI thích trên 1 dòng thì có thể dùng dấu: "//"

Khi lập trình nên giảI thích các câu lệnh khối lệnh làm gì để về sau khi chương trình lớn dễ sửa lỗi.

2.2.Các loại biến trong C:

Dạng biến	Số Bit	Số Byte	Miền giá trị
char	**8**	**1**	**-128 đến +127**
unsigned char	**8**	**1**	**0 đến 255**
short	**16**	**2**	**-32,768 đến +32,767**
unsigned short	**16**	**2**	**0 đến 65,535**
int	**16**	**2**	**-32,768 đến +32,767**
unsigned int	**16**	**2**	**0 đến 65,535**
long	**32**	**4**	**-2,147,483,648 đến+2,147,483,647**
unsigned long	**32**	**4**	**0 đến 4,294,697,295**

Khai báo biến

Cấu trúc : Kiểu biến Tên biến

VD: unsigned char x;

Khi khai báo biến có thể gán luôn cho biến giá trị ban đầu.

VD :

Thay vì: unsigned char x;

x=0;

Ta chỉ cần : unsigned char x=0;

Có thể khai báo nhiều biến cùng một kiểu một lúc

VD: unsigned int x,y,z;

Ngoài ra để dùng cho vi điều khiển trình dịch chuyên dụng còn hỗ trợ các loại biến sau:

Dạng biến	Số Bit	Số Byte	Miền giá trị
bit	**1**	**0**	**0 ; 1**
sbit	**1**	**0**	**0 ; 1**

sfr	**8**	**1**	**0 đến 255**
sf16	**16**	**2**	**0 đến 65,535**

VD:
Ngoài ra ,chúng ta có thể định nghĩa biến kiểu bít hay kiểu SFR (special function register)
Bit Kiemtra;
Sfr P10=0x90;
VD:
Bit Kiemtra;
Sfr P10=0x90;
Các SFR không cần phải học thuộc chỉ cần biết, và chúng được khai báo trong thư viện
AT89X51.H và AT89X52.H

2.3.Hàm trong C:

Hàm trong C có cấu trúc như sau
Có 2 loại hàm
Hàm trả lại giá trị:
Cấu trúc: Kiểu giá trị hàm trả lạii Tên hàm (Biến truyền vào hàm)

```
{
// Các lệnh xử lý ở đây
}
```

VD :

```
unsigned char Cong(unsigned char x, unsigned char y)
{
// Các lệnh xử lý ở đây
}
```

Hàm không trả lại giá trị
Cấu trúc: void Tên hàm (Biến truyền vào hàm)

```
{
// Các câu lệnh xử lý ở đây
}
```

VD dụ :

```
void Cong(unsigned char x, unsigned char y)
{
// Các câu lệnh xử lý ở đây
}
```

Hàm có thể truyền vào biến hoặc không
VD
Hàm không có biến truyền vào:

```
unsigned char  Tênhàm(void)
{
    // Các câu lệnh xử lí ở đây
}
```

Hàm có biến truyền vào:

```
void  Tênhàm(unsigned char x)
```

```
{
    // Các câu lệnh xử lí ở đây
}
```

Số biến truyền vào tùy ý(miễn đủ bộ nhớ), ngăn cách bởi dấu ","
Ví dụ:

```
Void TênHàm(unsigned char x, unsigned char y, unsigned char z)
{
// Các câu lệnh xử lí ở đây
}
```

Ngoài ra riêng cho vi điều khiển phần phềm Keil C còn có một loại hàm đó là hàm ngắt:
Cấu trúc:

```
Void  Tênhàm(void) interrupt nguồnngắt using băngthanhghi
{
}
```

Hàm ngắt không được phép trả lại giá trị hay truyền tham biến vào hàm.
Tên hàm bất kì
Interrupt là từ khóa chỉ hàm ngắt

Nguồn ngắt từ 0 tới 5 theo bảng vector ngắt

Ngắt do	Cờ	Địa chỉ vector
Reset hệ thống	**RST**	**0000H**
Ngắt ngoài 0	**IE0**	**0003H**
Bộ định thời 0	**TF0**	**000BH**
Ngắt ngoài 1	**IE1**	**0013H**
Bộ định thời 1	**TF1**	**001BH**
Port nối tiếp	**RI hoặc TI**	**0023H**
Bộ định thời 2	**TF2 hoặc EXF2**	**002BH**

Không tính ngắt reset hệ thống bắt đầu đếm từ ngắt ngoài 0
Băng thanh ghi trên ram chọn từ 0 đến 3

2.4.Các toán tử cơ bản:

Phép gán: =
VD: x=y; // x phải là biến y có thể là biến hoặc giá trị nhưng phải phù hợp kiểu
Phép cộng: +
Phép trừ: -
Phép nhân: *
Phép chia: /
Các toán tử logic:
  Bằng : ==
  And: &&
  Or: ||
  Not: !
  Dịch trái: <<
Dịch phải: >>

2.5.Các cấu trúc lệnh rẽ nhánh, kiểm tra thường dùng:

Câu lệnh rẽ nhánh if:

Cấu trúc: if (Điềukiện) { // Các câu lệnh xử lí }
Giải thích: Nếu Điềukiện đúng thì xử lí các câu lệnh bên trong còn sai thì nhảy qua

Câu lệnh lựa chọn switch:

```
Cấu trúc:   switch(Biến)
              {
              case giatrị1:  { // Các câu lệnh   break; }
              case giatrị2:  { // Các câu lệnh   break; }
              case giatrị3:  { // Các câu lệnh   break; }
              …
              case giatrịn:  { // Các câu lệnh   break; }
               }
```

Giải thích : Tùy vào Biến có giatrị1 thì thực hiện các câu lệnh sau đó tương ứng rồi thoát khỏi cấu trúc nhờ câu lệnh break;
Biến có giatrị2 thì thực hiện các câu lệnh sau đó tương ứng rồi thoát
….
Biến có giatrịn thì thực hiện các câu lệnh sau đó tương ứng rồi thoát

Câu lệnh vòng lặp xác định for:
Cấu trúc: for(n=m; n<l; n++) {// Các câu lệnh xử lí }
GiảI thích:
Trong đó m,l là giá trị (m>l), còn n là biến
Thực hiện lặp các câu lệnh (l-m) lần

Câu lệnh vòng lặp không xác định while:

```
Cấu trúc:
    While( Điều kiện)
  {
     //Các câu lệnh
   }
```

GiảI thích:
Thực hiện lặp các câu lệnh khi điều kiện đúng, nếu câu lệnh sai thi thoát khỏi vòng lặp

2.6. Bộ tiền xử lý:

#define : Dùng để định nghĩa. Ví dụ:

```
#define dung 1
#define sai 0
```

có nghĩa là dung có giá trị bằng 1. Trong chương trình có thể có đoạn code như sau:

```
 bit kiemtra
if (bit==dung)  { // Các câu lệnh}
if (bit==sai)  { // Các câu lệnh}
```

Việc này giúp lập trình dễ sửa lỗi hơn.

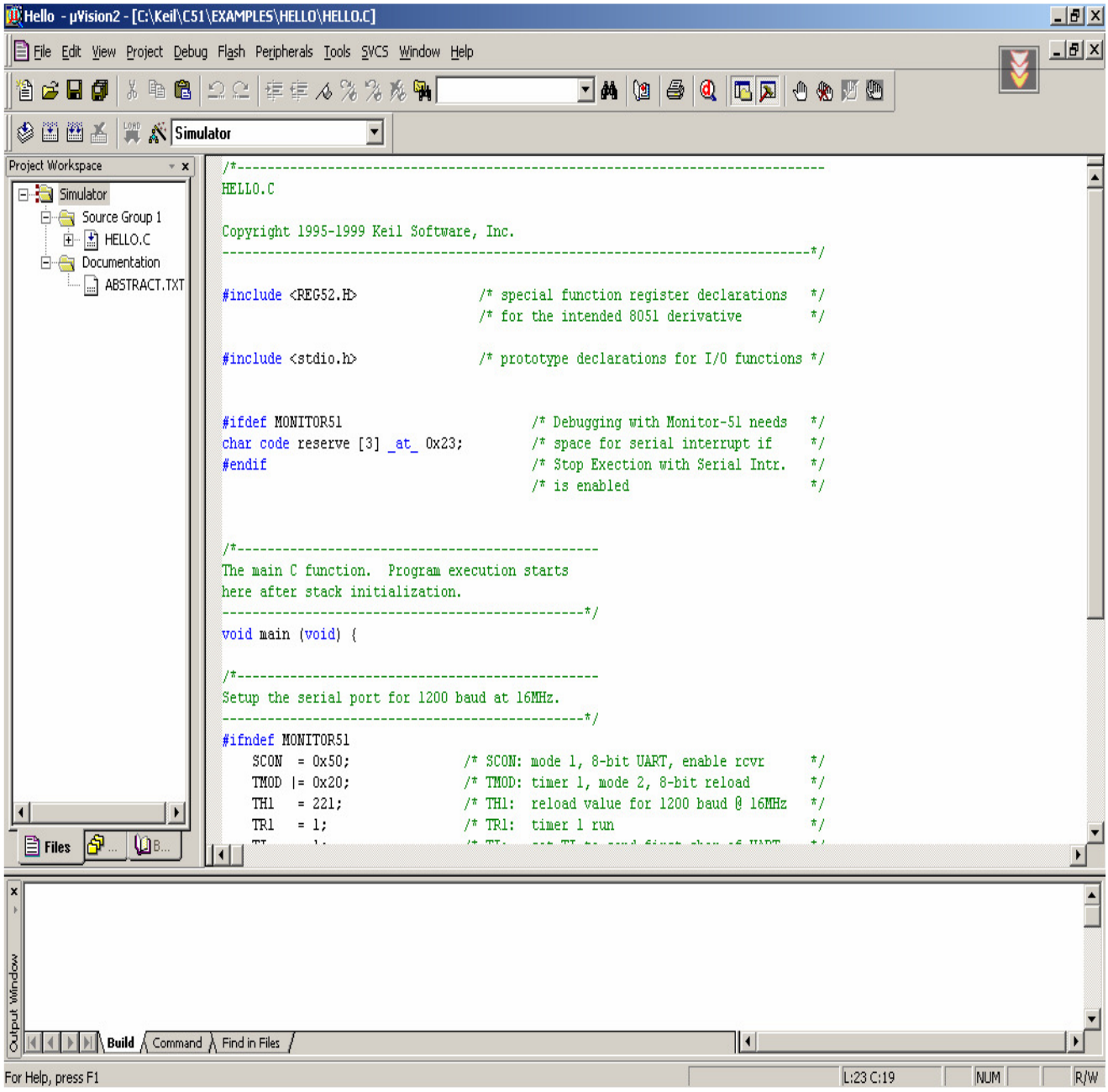
Một số web hay :

www.dientuvietnam.net	www.atmel.com
www.svbkol.org	www.keil.com
www.diendandientu.com	www.iguanalabs.com

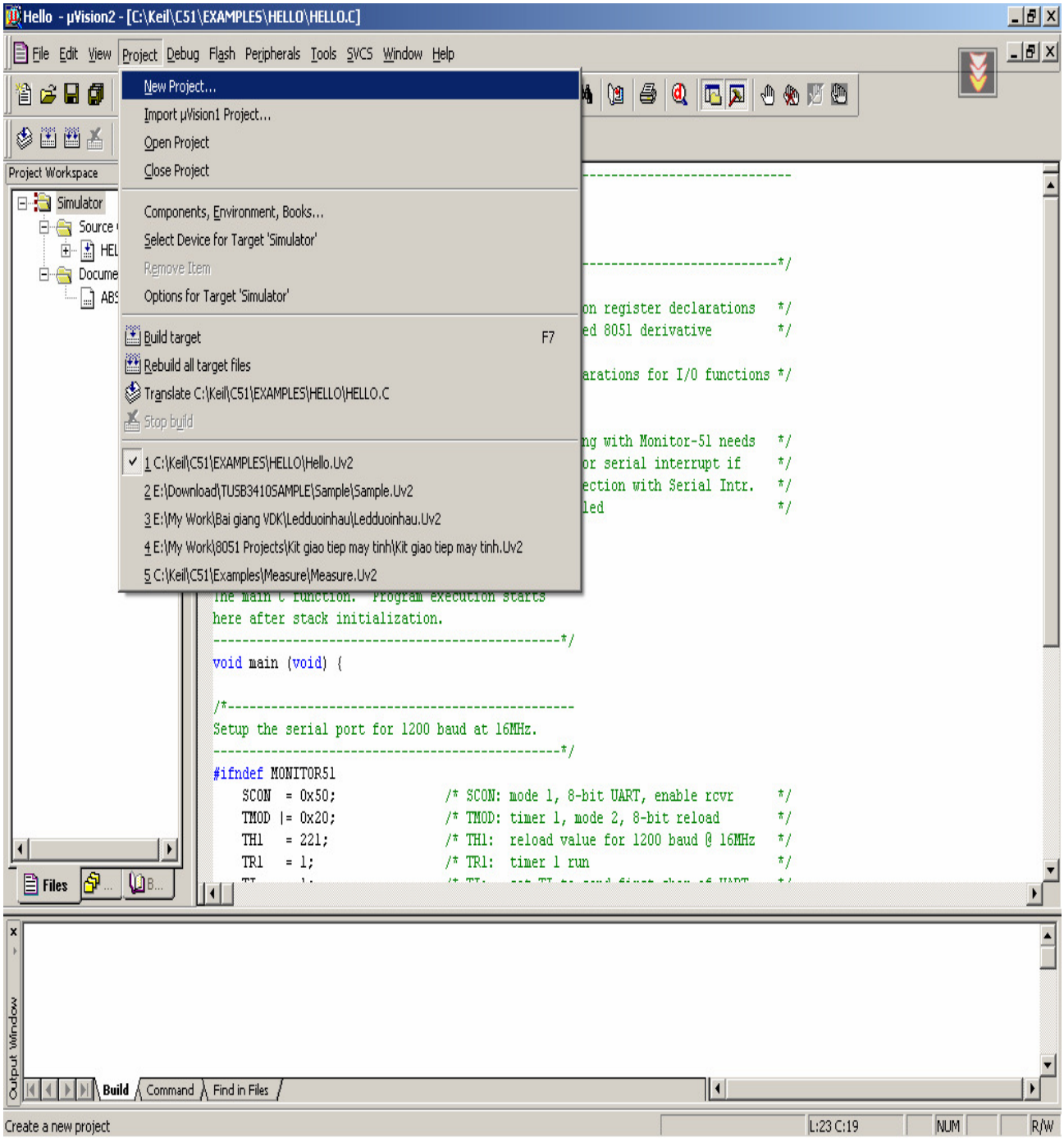
www.microchip.com
www.elechtro-tech-online.com
www.diendansv.hutech.edu.vn
www.ttvnol.com
www.8052.com
www.kmitl.ac.th
www.ftdichip.com

Bài 2 (Tiếp)
Phần 2: Sử dụng Keil C.

Sau khi cài đặt
1> Khởi tạo cho Project:



Để tạo 1 project mới chọn Project → New project như sau:



Được hình sau: