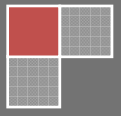
2007

Lập trình C cho VXL - Cơ bản

Vagam - giotdang

ntuan
BIA
8/15/2007



Lập trình C cho VXL - Cơ bản

I.Giới thiệu

C là một ngôn ngữ khá mạnh và rất nhiều người dùng .Nếu nói số lệnh cơ bản của C thì không nhiều . Nhưng đối với lập trình cho vxl , chúng ta chi cần biết số lượng lệnh không nhiều .
Đầu tiên bạn phải làm quen với

Các kiểu toán tử ở C
Các kiểu dữ liệu
Cẩu trúc cơ bản của một chương trình
Các cấu trúc điều khiển (chính các tập lệnh)
Cấu trúc điều kiện : if và else
Các cấu trúc lặp
Vòng lặp while
Vòng lặp do while
Vòng lặp for
Lệnh break.
Cấu trúc lựa chọn: switch. case

Biết sử dụng các hàm và chương trình con .

II.Cơ bản C

1. Các chỉ thị trước xử lý của Keil C

// chu thich

/********** chu thich**********

****

********** */

Đây là dòng chú thích. Tất cả các dòng bắt đầu bằng hai dấu sổ (//) được coi là chút thích mà chúng không có bất kì một ảnh hưởng nào đến hoạt động của chương trình. Chúng có thể được các lập trình viên dùng để giải thích hay bình phẩm bên trong mã nguồn của chương trình. Trong trường hợp này, dòng chú thích là một giải thích ngắn gọn những gì mà chương trình chúng ta làm. Còn trong dấu (/* */) bạn có thể chú thích bao nhiêu dòng tuỳ thích ,

#include <AT89X52.H>
hoặc
#include "AT89X52.H"

trình biên dịch sẽ gọi file thư viện của 89 ra (cơ bản là 51 cũng như 52)

#define bien_thay_the bien

Vd

#define Congtac P0_6

port0.6 được đặt tên là congtac , khi ta gọi tên này trình biên dịch Keil sẽ tự chuyển tới bit quản lý P0_6

Note :cách viết P0_6 phụ thuộc vào từng trình biên dịch , có chương trình thì lại viết là P0.6 , còn keil C viết như cách đầu

#define m_left_tien P1_5
#define m_left_lui P1_4

#define m_left_forward m_left_tien=0;m_left_lui=1;

các bạn chú ý đây là một cách sử dụng marco trong C
khi mình gọi m_left_forward thì chân P1_5 = 0 và P1_4=1

Các viết này gần như cho chúng ta một chương trình con , tuy nhiên không nên quá lạm dụng nó

Một ưu điểm nổi bật của C là các bạn có thể tạo ra các bộ thư viện .
Ví dụ sau là tạo thư viện thuvien.h (đuôi .h bạn có thể tạo bằng cách save as .. *.h ở Keil C).
#ifndef _thuvien_H
#define _thuvien_H

....//mã chương trình
#endif

2. Các toán tử :

->Toán tử gán (=).
Ex:

```
b = 5;
a = 2 + b;
a = 2 + (b = 5);
a = b = c = 5;
```

*->Các toán tử số học (+, -, *, /, %)*

+ cộng
- trừ
* nhân
/ chia
% lấy phần dư (trong phép chia)

->Các toán tử gán phức hợp (+=, -=, *=, /=, %=, >>=, <<=, &=, ^=, |=)

value += increase; tương đương với value = value + increase;
a -= 5; tương đương với a = a - 5;
a /= b; tương đương với a = a / b;
price *= units + 1; tương đương với price = price * (units + 1);

Tăng và giảm ++ - -

a++; <=> a+=1; <=> a=a+1;

tính chất tiền tố hoặc hậu tố (++a) # (a++) Ex

```
B=3;
B=3;A=++B;
// A is 4, B is 4

B=3;
A=B++;
// A is 3, B is 4
```

-> Các toán tử quan hệ (==, !=, >, <, >=, <=)

== Bằng
!= Khác
> Lớn hơn
< Nhỏ hơn
> = Lớn hơn hoặc bằng
< = Nhỏ hơn hoặc bằng

EX

(7 == 5) sẽ trả giá trị false
(6 >= 6) sẽ trả giá trị true

tất nhiên thay vì sử dụng các số, chúng ta có thể sử dụng bất cứ biểu thức nào. Cho a=2, b=3 và c=6

(a*b >= c) sẽ trả giá trị true.
(b+4 < a*c) sẽ trả giá trị false

Chú ý rằng = (một dấu bằng) lf hoàn toàn khác với ==
(hai dấu bằng). (==) nhằm so sánh còn (=)gán giá trị của biểu thức bên phải cho biến ở bên trái .

-> *Các toán tử logic* (!, &&, ||).

! NOT
&& AND
|| OR

EX:

!(5 == 5) trả về false vì biểu thức bên phải (5 == 5) có giá trị true.
!(6 <= 4) trả về true vì (6 <= 4)có giá trị false.
!true trả về false.
!false trả về true.
((5 == 5) && (3 > 6)) trả về false (true && false).
((5 == 5)

-> *Các toán tử thao tác bit* (&, |, ^, ~, <<, >>).

& AND Logical AND
| OR Logical OR
^ XOR Logical exclusive OR
~ NOT Đảo ngược bit
<< SHL Dịch bit sang trái
>> SHR Dịch bit sang phải

->*Thứ tự ưu tiên của các toán tử*

Thứ tự	Toán tử	Mô tả	Associativity
1	: :	scope	Trái

2	`() [ ] -> . sizeof`		Trái
3	`++ --`	tăng/giảm	Phải
	`~`	Đảo ngược bit	
	`!`	NOT	
	`& *`	Toán tử con trỏ	
	`(type)`	Chuyển đổi kiểu	
	`+ -`	Dương hoặc âm	
4	`* / %`	Toán tử số học	Trái
5	`+ -`	Toán tử số học	Trái
6	`<< >>`	Dịch bit	Trái
7	`< <= > >=`	Toán tử quan hệ	Trái
8	`== !=`	Toán tử quan hệ	Trái
9	`& ^ |`	Toán tử thao tác bit	Trái
10	`&& ||`	Toán tử logic	Trái
11	`?:`	Toán tử điều kiện	Phải
12	`= += -= *= /= %= >>= <<= &= ^= |=`	Toán tử gán	Phải
13	`,`	Dấu phẩy	Trái

3. Các kiểu dữ liệu

Các kiểu biến.chuẩn

Type	**Bits**	**Bytes**	**Range**
char	8	1	-128 to +127
unsigned char	8	1	0 to 255
enum	16	2	-32,768 to +32,767
short	16	2	-32,768 to +32,767
unsigned short	16	2	0 to 65,535

int	16	2	-32,768 to +32,767
unsigned int	16	2	0 to 65,535
long	32	4	-2,147,483,648 to +2,147,483,647
unsigned long	32	4	0 to 4,294,697,295

Kiểu dữ liệu **trong Keil C**

Type	Bits	Bytes	Range
bit	1	0	0 to 1
sbit	1	0	0 to 1
sfr	8	1	0 to 255
sf16	16	2	0 to 65,535

4. Cấu trúc cơ bản của 1 chương trình C

```
//Các chỉ thị tiền định
#include <lcd.h> //Gọi thư viện có sẵn cách viết khác "*.h"
#define led1 PORTA.0 //dùng định nghĩa các biến

char bien1,bien2; //cac bien can dung
int a,b;

void chuongtrinhcon(unsigned int  b)   // chuong trinh con
{
…
}

int ham(void)                    // chuong trinh con dang ham
{
….
Return(a);
}
```

```
void main(void) //chương trình chính
{
int  a; // khai bao bien dang so nguyen

chuongtrinhcon();

a = ham();

}
```

Chương trình con là nơi các bạn viết các chương trình nhỏ , rất tiện cho các đoạn lệnh gặp lại nhiều lần . Chương trình con có thể có thể gọi ở trong chương trình chính bất kì đâu .
Hàm là chương trình con trả về cho mình một giá trị.
Cách sử dụng hàm và chương trình con các bạn nên tham khảo thêm quyển kĩ thuật lập trình C để hiểu rõ hơn .

5. Các lệnh cơ bản của C

Cấu trúc điều kiện: if và else

->if (condition) statement

```
  if  (x  ==  100)  x++;
```

nếu x=100 thì tăng x thêm 1

->if (condition) statement1 else statement2

```
if  (x  ==  100)    x++;
else    x- -;
```

Các cấu trúc lặp

Vòng lặp while .
Dạng của nó như sau:
while (expression) statement

```
while(1) {};
```

Tạo vòng lặp mãi mãi , rất hay đùng trong lập trình VXL .Chương trình chính sẽ được viết trong dấu ngoặc.

Vòng lặp do-while
Dạng thức:
do statement while (condition);

```
do
{
```

```
x++; // cho nay cac ban co the viet nhieu cau lenh ,
}
while(x>10)
```

tăng giá trị của x cho đến khi x > 10

Chức năng của nó là hoàn toàn giống vòng lặp while chỉ trừ có một điều là điều kiện điều khiển vòng lặp được tính toán sau khi statement được thực hiện, vì vậy statement sẽ được thực hiện ít nhất một lần ngay cả khi condition không bao giờ được thoả mãn .Như vd trên kể cả x >10 thì nơ vẫn tăng giá trị 1 lần trước khi thoát

nếu x=100 thì tăng x thêm 1 còn không thì giảm x.

Vòng lặp for .

Dạng thức:

for (initialization; condition; increase) statement;
và chức năng chính của nó là lặp lại statement chừng nào condition còn mang giá trị đúng, như trong vòng lặp while. Nhưng thêm vào đó, for cung cấp chỗ dành cho lệnh khởi tạo và lệnh tăng. Vì vậy vòng lặp này được thiết kế đặc biệt lặp lại một hành động với một số lần xác định. Cách thức hoạt động của nó như sau:

1, initialization được thực hiện. Nói chung nó đặt một giá khí ban đầu cho biến điều khiển. Lệnh này được thực hiện chỉ một lần.
2, condition được kiểm tra, nếu nó là đúng vòng lặp tiếp tục còn nếu không vòng lặp kết thúc và statement được bỏ qua.
3, statement được thực hiện. Nó có thể là một lệnh đơn hoặc là một khối lệnh được bao trong một cặp ngoặc nhọn.
4, Cuối cùng, increase được thực hiện để tăng biến điều khiển và vòng lặp quay trở lại bước 2.

Phần khởi tạo và lệnh tăng không bắt buộc phải có. Chúng có thể được bỏ qua nhưng vẫn phải có dấu chấm phẩy ngăn cách giữa các phần. Vì vậy, chúng ta có thể viết for (;n<10;) hoặc for (;n<10;n++).
Bằng cách sử dụng dấu phẩy, chúng ta có thể dùng nhiều lệnh trong bất kì trường nào trong vòng for, như là trong phần khởi tạo. Ví dụ chúng ta có thể khởi tạo một lúc nhiều biến trong vòng lặp:

```
for ( n=0, i=100 ; n!=i ; n++, i-- )
{
// cái gì ở đây cũng được...
}
```

Vi dụ điển hình nhất trong lập trình VXL

```
void  delayms(int  n)
{

int  i,j;                          // khai  bao  bien  chi  trong  chuong  trinh  con
for  (i=0;i<n;i++)
for  (j=0;j<1500;j++)  {  }       // tham  so  j  tuy  thach  anh  toc  do  vxl  ma  cac
```

```
                                    //ban  thay  doi  cho  phu  hop
}
```

Các lệnh rẽ nhánh và lệnh nhảy

Lệnh break.

Sử dụng break chúng ta có thể thoát khỏi vòng lặp ngay cả khi điều kiện để nó kết thúc chưa được thoả mãn. Lệnh này có thể được dùng để kết thúc một vòng lặp không xác định hay buộc nó phải kết thúc giữa chừng thay vì kết thúc một cách bình thường. Ví dụ, chúng ta sẽ dừng việc đếm ngược trước khi nó kết thúc:

Lệnh continue.
Lệnh continue làm cho chương trình bỏ qua phần còn lại của vòng lặp và nhảy sang lần lặp tiếp theo. Ví dụ chúng ta sẽ bỏ qua số 5 trong phần đếm ngược:

Lệnh goto.
Lệnh này cho phép nhảy vô điều kiện tới bất kì điểm nào trong chương trình. Nói chung bạn nên tránh dùng nó trong chương trình C++. Tuy nhiên chúng ta vẫn có một ví dụ dùng lệnh goto để đếm ngược:

Hàm exit.
Mục đích của exit là kết thúc chương trình và trả về một mã xác định. Dạng thức của nó như sau
void exit (int exit code);
exit code được dùng bởi một số hệ điều hành hoặc có thể được dùng bởi các chương trình gọi. Theo quy ước, mã trả về 0 có nghĩa là chương trình kết thúc bình thường còn các giá trị khác 0 có nghĩa là có lỗi.

các lệnh trên mình chủ yếu chỉ dùng lệnh break để thoát khỏi vòng lặp . Các lệnh khác thường rất ít dược sử dụng

Đây là 1 đoạn code nhỏ mình trích ra từ chương trình của mình

```
 while(1)
{
lcd_gotoxy(5,0);lcd_putsf("  Run  Thuan    ");
thuan();
if(!enter)  { lcd_clear();
lcd_putsf("DA  DUNG    ");
stop();break;}
}
```

***Cấu trúc lựa chọn*: switch.**

Cú pháp của lệnh switch hơi đặc biệt một chút. Mục đích của nó là kiểm tra một vài giá trị hằng cho một biểu thức, tương tự với những gì chúng ta làm ở đầu bài này khi liên kết một vài lệnh if và else if với nhau. Dạng thức của nó như sau:

Code:

```
 switch  (expression)  {
```

```
case constant1:
   block of instructions 1
   break;
case constant2:
   block of instructions 2
   break;
.
.
.
default:
   default block of instructions
}
```

Nó hoạt động theo cách sau: switch tính biểu thức và kiểm tra xem nó có bằng constant1 hay không, nếu đúng thì nó thực hiện block of instructions 1 cho đến khi tìm thấy từ khoá break, sau đó nhảy đến phần cuối của cấu trúc lựa chọn switch.
Còn nếu không, switch sẽ kiểm tra xem biểu thức có bằng constant2 hay không. Nếu đúng nó sẽ thực hiện block of instructions 2 cho đến khi tìm thấy từ khoá break.
Cuối cùng, nếu giá trị biểu thức không bằng bất kì hằng nào được chỉ định ở trên (bạn có thể chỉ định bao nhiêu câu lệnh case tuỳ thích), chương trình sẽ thực hiện các lệnh trong phần default: nếu nó tồn tại vì phần này không bắt buộc phải có.

nếu nút enter được bấm thì chương trình sẽ thoát ra khỏi vòng lặp while .

III. C cho 8051

1. Keil Variable Extensions

```
data địa chỉ trực tiếp MOV A, 07Fh
idata địa chỉ gián tiếp MOV R0, #080h
                                    MOV A, R0
xdata bộ nhớ ngoài              MOVX @DPTR
code bộ nhớ chương trình   MOVC @A+DPTR
```

VD
unsigned int data bien = 0; // them data vao khai bao kieu bien

Chú ý rằng , bạn có thể không cần khai báo cụ thể , chỉ cần
unsigned int checksum = 0;

2. Địa chỉ ngắt

Interrupt	Vector address	Interrupt number
External 0	0003h	0

Timer 0	000Bh	1
External 1	0013h	2
Timer 1	001Bh	3
Serial	0023h	4

```
org             00h
ljmp main
org 0003h
ljmp ngat0
org 30h

main:                   // chuong trinh chinh
...
Here: sjmp Here     // vong lap vo tan

ngat0:
...
reti
```

code C

Code:

```
void  main                          // chuong trinh chinh
{
...
while(1)                            //vong lap vo tan sau khi thuc hien xong cong viec
}
void  ngat0(void)  interrupt  0   // chuong trinh ngat
{
...
}
```

3. Một ví dụ hoàn thiện về lập trình C cho 8051

```
#include<AT89X52.h>
#include<stdio.h>
#define    strai3      P0_7
#define    strai2      P0_6
#define    strai1      P0_5
#define    strai0      P0_4
#define    sphai0      P0_3
#define    sphai1      P0_2
#define    sphai2      P0_1
#define    sphai3      P0_0
//////////////////////////
#define    mtraif      P1_0
#define    mtraib      P1_1
#define    mphaif      P1_2
#define    mphaib      P1_3
#define    dc1f                P1_6
#define    dc1b                P1_7
```

```
#define    dc2f                    P1_4
#define    dc2b                    P1_5
//////////////////////////
#define    f_an0                   P2_0
#define    f_an1                   P2_1
#define    f_an2                   P2_2
#define    f_an3      P2_3
//////////////////////////
#define    dc3f                    P2_4
#define    dc3b                    P2_5
//////////////////////////
#define  start                                 P3_6
#define    ctht1_batdau                        P3_1
#define    ctht2_ketthuc                       P3_2
#define    ctht3                   P3_3
#define    ctht4                   P3_4
#define    ctht5                   P3_5
#define    ctht6                   P3_6
//////////////////////////
#define    tien                    1
#define    lui                     0
//////////////////////////
#define    v_cham_trai                         50
#define    v_cham_phai                         50
#define    delta_v_cham_trai                  20
#define    delta_v_cham_phai                   20
#define    v_nhanh_trai                        100
#define    v_nhanh_phai                        100

#define    v_quay_trai                         40
#define    v_quay_phai                         60

//////////////////////////
//vach trang sensor=1 vach xanh sensor=0
int          t=0,i=0,j=0,k=0;
int          vtrai=100,vphai=100;
int          PWMC=0;
int          dem=0;
int          dirtrai=tien,dirphai=tien;
int          phuongan=0;
//////////////////////////
void dithang(int v_left,int v_right);
void dithangcham(int v_left,int v_right);
void stop(void);
void quayphai(void);
void quaytrai(void);
void ragach(void);
void pwm(void) interrupt 1 ;
void khoitao(void);
void hanhtrinh(void);

void ham(int time,int trai,int phai);
void ham2(void);
//=======================================
void main(void)
{
          khoitao();
          khoidongthang();
          hanhtrinh();
          ragach();
          stop();

}
//=========================================
void khoitao(void)
{
          TMOD=0x02;
          TH0=0xE1; TR0=1; IE=0x82;
          P1=0x00;P2=0x00;P3=0x00;
          while(!start) stop();
}
//=========================================
void khoidongthang(void)
```

```
{
        for(i=0;i<300;i++)
        {
                dirtrai=tien;dirphai=tien;
                vtrai=40; vphai=40;
        }
}
//=========================================
void khoidongquay(void)
{
                dirtrai=tien;dirphai=tien;
                vtrai=90; vphai=80;
                for(i=1;i<=50;i++);
                vtrai=52; vphai=40;
                for(i=1;i<=580;i++);
                ham(90,20,20);
                do
                {
                        dirtrai=tien;dirphai=lui;
                        vtrai=33;vphai=10;
                }
                while(!strai3);
                for(i=1;i<=40;i++)
                {
                        dirtrai=tien;dirphai=lui;
                        vtrai=33; vphai=9;
                }
                do
                {
                        dirtrai=lui;dirphai=tien;
                        vtrai=15;vphai=32;
                }
                while(!strai0);
                ham(100,20,20);
                dirtrai=tien;dirphai=tien;
}
//=========================================

void quayphai(void)
{
        int ktra=0;
                 dirtrai=tien;dirphai=lui;
        do
        {
                 vtrai=     v_quay_trai;vphai=  v_quay_phai;
                 if (sphai3) ktra++;
        }
        while(ktra<20);
        stop();
        for(i=1;i<1000;i++){};
        dirtrai=tien; dirphai=tien;
}
////////////////////////////////////

void quaytrai(void)
{
        int ktra=0;
           dirtrai=lui;dirphai=tien;
        do
        {
                 vtrai=     60; vphai=              40;
                 if (strai3)ktra++;
        }
        while(ktra<20);
        stop();
        for(i=1;i<1000;i++){};
        dirtrai=tien; dirphai=tien;
}
//++++++++++++++++++++++++++++++++++++++
//vach trang sensor=1 vach xanh sensor=0
void stop(void)
{
        vtrai=0; vphai=0;
```

```
}
//=============================================================================================

void dithang(int v_left,int v_right)
{
        if(!(strai0||strai1||strai2||strai3||sphai0||sphai1||sphai2||sphai3)) // di thang
                {
                        dirphai=tien;dirtrai=tien;
                        vtrai=v_left;vphai=v_right;
                }
                        else
                                if((strai0&&strai1&&sphai0&&sphai1))
                                {       j=0;
                                        for(i=1;i<=15;i++)
                                        if((strai0&&strai1&&sphai0&&sphai1)) j++;
                                        if(j>=10) { dem=dem+1;t=1 ;}
                                        else t=0;
                                        while(t)//cho qua vach trang
                                        {
                                                {       //di thang
                                                        dirphai=tien;dirtrai=tien;
                                                        vtrai=40;vphai=40;
                                                }
                                                k=0;
                                                for(i=1;i<=20;i++)
                                                if(!(strai3||strai2||sphai3||sphai2))  k++;
                                                if(k>15) t=0;
                                        }
                                }
                                else  if ((sphai2||sphai3)) //re phai lon
                                                                        {
                                                                                dirtrai=tien;dirphai=tien;
                                                                                vtrai=60;vphai=0;}
                                else  if ((strai0||strai1))

        //retrainho
                                                                        {
                                                                                dirtrai=lui;dirphai=tien;
                                                                                vtrai=1;vphai=60;

                }
                                else  if ((sphai0||sphai1))

        //rephainho
                                                                        {
                                                                                dirtrai=tien;dirphai=lui;
                                                                                vtrai=60;vphai=1;

                }
                                else  if ((strai2||strai3))

         //re trai lon
                                                                        {
                                                                                dirtrai=tien;dirphai=tien;
                                                                                vtrai=0;vphai=60;}
}
////////////////////////////
void ragach(void)
{
while(!ctht2_ketthuc)
        {
                dirtrai=tien;dirphai=tien;
                vtrai=25;vphai=25;
                dc1f=1;dc1b=0;
        }
        dc1f=0;dc1b=0;
        //---------------------------------------

  for(t=1;t<=5;t++)
  {
        for(i=1;i<=10000;i++){
        dirtrai=tien;dirphai=tien;
        vtrai=25;vphai=25;}
```

```
                for(i=1;i<=10000;i++){
        dirtrai=lui;dirphai=lui;
        vtrai=0;vphai=0;}
        }
  //-------------------------
        dirtrai=lui;dirphai=lui;
        vtrai=25;vphai=25;
        for(i=1;i<=2000;i++);
}

//////////////////////////
void pwm(void) interrupt 1
{
        PWMC++;
        if(PWMC==100) PWMC=0;
        if(PWMC<vtrai)
                {
                        mtraif=dirtrai;
                        mtraib=!dirtrai;
                }
                else
                {
                        mtraif=0;
                        mtraib=0;
                }
        if(PWMC<vphai)
                {
                        mphaif=dirphai;
                        mphaib=!dirphai;
                }
                else
                {
                        mphaif=0;
                        mphaib=0;
                }
}
void ham(int time,int trai,int phai)
{
        int g;

        for(g=0;g<=time;g++)
        {
                vphai=phai;vtrai=trai;
                dirtrai=tien;dirphai=tien;
                dirtrai=lui;dirphai=lui;

        }
        dirtrai=tien;dirphai=tien;
        vphai=0;vtrai=0;

}
```

Chúc các bạn học lập trình C cho vi xử lý thật nhanh nhé . Đọc phần lý thuyết cơ bản sau đó đọc bài ví dụ cuối cùng .
Nếu bạn còn gì chưa hiểu , hãy post lên để nhóm vagam giúp các bạn .

Thân

Lê Ngọc Tuấn – giotdang1985@yahoo.com