

HT MCU 的 C 编程说明 (I)

文件编码：HA0046s

介绍：

本文针对 HT-IDE3000 编写。所有 HT-IDE2000 的功能，HT-IDE3000 版本均包含在内。但请注意，一些在 HT-IDE2000 上的函数，如 peekPX 等已经被其他形式代替，如需要移植程序，则必须进行部分修改。

本文主要目的是告诉读者，在汇编程序所实现的某些功能，如何用 HOLTEK C 来实现。以及两者在使用中的一些区别和限制，希望能对使用 HOLTEK 芯片的用户有所帮助。更多的 HOLTEK C 说明，请见《HT-IDE 3000 USER'S GUIDE》。

目录：

1、位变量定义

- i. 定义位变量
- ii. 定义位变量在指定位置

2、I/O 口读写操作

3、表格操作

- i. 建立表格
- ii. 读取表格内容

4、ROMBANK 和 RAMBANK

- i. RAMBANK 的设置和访问
- ii. ROMBANK 的设置和访问

5、范例：流水灯

位变量定义

. 定义位变量

bit name

这是在 C 语言中定义变量的格式。“*bit*”定义变量类型是“位”，“*name*”是变量名。当“*name*”是一个全局变量时，系统会自动分配一个位空间。

Example:

```
bit flag1,flag2;    //系统自动分配 bit 空间给位变量 flag1、flag2
main(){
    flag1=flag2=1;
}
```

.定义位变量在指定位置

#define name _ramaddr_bitaddr

前置处理指令**#define**可以定义位变量的具体 RAM 地址，这里仅局限于 RAMBANK0。

“*name*”是位变量名；“*ramaddr*”是 RAM 中具体地址；“*bitaddr*”是位变量的位址。

Example:

```
#define ab _40_3 //ab 位于 RAM bank0 的[40h].3
main(){
    ab=1;
}
```

I/O 口读写操作

I/O 口可以在 C 程序中，直接对其访问和赋值（以 BYTE 为单位）。在程序开头，必须包含 include 文件“htXXX.h”，举例如下：

```
#include "ht48r70a-1.h"
...
main(){
    ...
    _pa=_pb=_pc=0;           //设置 PA,PB,PC 值为 0
    _pac=_pbc=_pcc=0;       //设置 PA,PB,PC 为输出口
    ...
}
```

表格操作

i. 建立表格

const var-type name[num]={date0,data1,data2,...}

在程序中，表格是不可或缺的一部分，即是在 ROM 空间写入一连串指定内容，在程序中可以取出需要的表格数据。C 程序中，const 用来定义常数，同时常数存放在 ROM 空间。所以，我们定义一个常数数组，就是向 ROM 空间写入一连串指定值。

“var-type”是常数类型，可以是整数或者字符；“name”是常数数组名；“num”是数组大小；“data0,data1,data2”是常数数组内容，即表格内容。“data0,data1,data2”可以是 byte，或者 word。

Example1:

```
const long array[3]={0x1234,0x5678,0xabcd};
long temp;
main(){
temp=array[0];          //temp=0x1234
}
```

Example2:

```
const char array[2]={'a','b'};
long temp;
main(){
temp=array[1];          //temp='a',a 的字符代码是 61h
}
```

这样就建立了一张表格。值得注意的是，每一张表格的大小限制在 256byte 之内，一旦表格大小超出范围，则会发生查表错误。

ii. 读取表格内容

读取表格内容，即读取常数数组的第 n 个数据。因此我们读取 array[n]的内容，就可以了。数组的第一个数据从 array[0]开始。

Example:

```
const long array[3]={0x1234,0x5678,0xabcd};
long t1,t2,t3;
main(){
t1=array[0];            //t1=0x1234
t2=array[1];            //t2=0x5678
t3=array[2];            //t3=0xabcd
}
```

ROMBANK 和 RAMBANK

i. ROMBANK 的设置和访问

在 HOLTEK 很多 MCU 的硬件结构中，ROM 如果大于 8K 的空间的话，就被分为若干个 BANK，8K=1BANK。例如 HT48RA3 有三个 ROMBANK (BANK0,BANK1,BANK2)。

在对 ROMBANK 的操作上，汇编程序和 C 程序有着概念上的不同，即汇编需要明确程序具体在哪个 ROMBANK，如何去访问等等；C 程序完全不必考虑这点，C 编译器会实现自动 ROMBANK 的设置和访问动作。

因此，我们在 C 程序中看不见任何有关 ROMBANK 的语句。

ii. RAMBANK 的设置和访问

HOLTEK 一些 MCU 如 HT49 系列 (带 LCD 显示) HTG21 系列 (GAME) 有多个 RAMBANK。变量可以指定在各个 RAMBANK，如下的定义：

```
int v1 @ 0x5B      ;定义 v1 在 RAM bank0 的地址 5BH
int v2 @ 0x2F0     ;定义 v2 在 RAM bank2 的地址 F0H
```

范例：流水灯程序

硬件说明：采用 HT48R50A-1 芯片，PA、PB 口直接 LED 输出，系统频率 4MHz。0.5S 流水灯转动一次。

软件程序：

```
#include <ht48r50a-1.h>

#pragma vector extern_i @ 0x4 //定义中断子程序位置
#pragma vector timer_i0 @ 0x8
#pragma vector timer_i1 @ 0xc
#define _te _0e_4

unsigned int count;
unsigned int light0, light1;

//ISR
void extern_i(){
}

//Timer0 用于 2ms 中断时间
void timer_i0(){
    if (count==250) //0.5sec
    {_c=0;
        if (light1==0x1)
        {_c=1;
            _rrc(&light0); //流水灯循环输出
            _rrc(&light1);
            count=0;
        }
        else count++;
    }
}

void timer_i1(){
}

//MAIN 主程序
void main(){
    _clrwdt();
    _intc=00;
```

```
_tmr0c=0x87;
_tmr0=0xe1;
_pac=_pbc=00;           //设置 pa , pb 口为输出口
light0=0x80;
light1=0x00;
_et0i=1;                 //开中断
_emi=1;
_te=1;
while (1)
{_clrwdt();
_pa=light0;              //pa,pb 口输出
_pb=light1;
}
}
```