

第一章 流水灯实验

1.1 . 实验所需要的开发工具/软件

- 1) 安装 WinAVR20070525 编译开发软件
- 2) 安装 Source Insight 软件 (可选)
- 3) 简易并行 ISP 下载线
- 4) 安装 PonyProg 单片机 ISP 下载软件
- 5) USB 电源线一根
- 6) 实验板一块

1.2. MCU 相关硬件基础知识介绍

请参考《ATMEGA16 内部资源使用教程》的第一章。

1.3. 实验硬件原理图

请参考《AVR8 开发板原理图》。

1.4 实验功能

LED0~LED7 全亮全灭闪烁，亮灭间隔为 1 秒。

1.5 实验源代码及注释分析

Program 1.1 (文件名为 LED.C)

```
#include <avr/io.h> //包含单片机的定义文件
#define uchar unsigned char //定义用 uchar 代替 unsigned char
#define uint unsigned int //定义用 uint 代替 unsigned int
/*-----
port_init : IO 口初始化，所有的 IO 口都为输出口，且输出高电平
输入参数：无
输出参数：无
-----*/
void port_init(void)
{
    PORTA = 0xFF;
    DDRA = 0xFF;
```

```

PORTB = 0xFF;
DDRB = 0xFF;
PORTC = 0xFF;
DDRC = 0xFF;
PORTD = 0xFF;
DDRD = 0xFF;
}
/*-----*/
delay : 延时函数，由输入参数 t 决定延时长短
输入参数：t—无符号字符型，取值范围 0~255
输出参数：无
-----*/

void delay(uchar t)
{
    uchar i;
    uint j;
    for(i=0;i<t;i++){
        for(j=0;j<60000;j++){ //延时大约 5ms
            asm("nop");
        }
    }
}
/*-----*/
main : 主函数
输入参数：无
输出参数：无
-----*/

void main(void)//主函数
{
    port_init();
    while(1)
    {
        delay(200); //延时大约 1s
        PORTC = 0xff; //8 个 LED 全灭
        delay(200); //延时大约 1s
        PORTC = 0x0; //8 个 LED 全亮
    }
}

```

1.6 对 C 源文件的编辑、编译及 HEX 文件的烧写

具体请参考《AVR 单片机的开发环境建立及使用教程》。

第二章 查询式按键实验

2.1 实验所需要的开发工具/软件

- 1) 安装 WinAVR20070525 编译开发软件
- 2) 安装 Source Insight 软件 (可选)
- 3) 简易并行 ISP 下载线
- 4) 安装 PonyProg 单片机 ISP 下载软件
- 5) USB 电源线一根
- 6) 实验板一块

2.2 MCU 相关硬件基础知识介绍

请参考《ATMEGA16 内部资源使用教程》的第 2 章。

2.3 实验硬件原理图

请参考《AVR8 开发板原理图》。

2.4 实验功能

按 KEY1~KEY4 中的任何一个键，会将对应得一个 LED 灯点亮。

- 1) 若按下 KEY1 键，则 LED0 被点亮，其余 7 个 LED 灯全灭；
- 2) 若按下 KEY2 键，则 LED1 被点亮，其余 7 个 LED 灯全灭；
- 3) 若按下 KEY3 键，则 LED2 被点亮，其余 7 个 LED 灯全灭；
- 4) 若按下 KEY4 键，则 LED3 被点亮，其余 7 个 LED 灯全灭；
- 5) 若同时按下两个以上的键，则 8 个 LED 灯全灭。

2.5 实验源代码及注释分析

Program 2.1 (文件名为 POLL_BUZZER.C)

```
#include <avr/io.h> //包含单片机的定义文件
#define uchar unsigned char //定义用 uchar 代替 unsigned char
#define uint unsigned int //定义用 uint 代替 unsigned int
#define LED0_On (PORTC&=~(1<<PC0)) //使 PC^0=0，点亮 LED0
#define LED1_On (PORTC&=~(1<<PC1)) //使 PC^1=0，点亮 LED1
#define LED2_On (PORTC&=~(1<<PC2)) //使 PC^2=0，点亮 LED2
```

更多资料请登陆 www.mcudriver.cn 本站主要探讨 AVR、51、及 Freescale8 位单片机。提供常用电子元器件的零售、邮购业务。专注于 1-wire、I2C、SPI 等串行器件的驱动开发等。 版权所有 转载需注明出处

```
#define LED3_On (PORTC&=~(1<<PC3)) //使 PC^3=0，点亮 LED3
```

```
/*-----*/
```

port_init：IO 口初始化，所有的 IO 口都为输出口，且输出高电平

输入参数：无

输出参数：无

```
-----*/
```

```
void port_init(void)
```

```
{  
    PORTA = 0xFF;  
    DDRA = 0xFF;  
    PORTB = 0xFF;  
    DDRB = 0xFF;  
    PORTC = 0xFF;  
    DDRC = 0xFF;  
    PORTD = 0xFF;  
    DDRD = 0xFF;
```

```
}
```

```
/*-----*/
```

delay：延时函数，由输入参数 t 决定延时长短

输入参数：t—无符号字符型，取值范围 0~255

输出参数：无

```
-----*/
```

```
void delay(uchar t)
```

```
{  
    uchar i;  
    uint j;  
    for(i=0;i<t;i++){  
        for(j=0;j<60000;j++){ //延时大约 5ms  
            asm("nop");  
        }  
    }  
}
```

```
/*-----*/
```

key_exe：按键响应函数

输入参数：keynum—按键值

输出参数：无

```
-----*/
```

```
void key_exe(uchar keynum)
```

```
{  
    if(keynum == 0xE0) LED0_On; //key1 按下，LED0 点亮  
    else if(keynum == 0xD0) LED1_On; //key2 按下，LED1 点亮  
    else if(keynum == 0xB0) LED2_On; //key3 按下，LED2 点亮  
    else if(keynum == 0x70) LED3_On; //key4 按下，LED3 点亮
```

```
else PORTC=0xff; //错误键值，LED 全灭
}
/*-----*/
main：主函数
输入参数：无
输出参数：无
-----*/
void main(void)//主函数
{
    uchar temp_pd,temp0,temp1;
    uchar n; //定义无符号字符型局部变量 n
    port_init();
    //将与按键相连的四个 IO 口设为输入口：
    DDRD=~((1<<PD7)|(1<<PD6)|(1<<PD5)|(1<<PD4));
    while(1) //进入循环，检测是否有键按下
    {
        temp_pd=PIND; //读取 IO 口的引脚值
        temp0=temp_pd&0xf0; //屏蔽低四位无效值

        if(temp0!=0xf0){ //如果不为 0xf0，说明可能有键按下
            //延时去抖动，并再次检测键值
            delay(6); //大约 30ms
            temp_pd=PIND;
            temp1=temp_pd&0xf0; //屏蔽低四位无效值
            if(temp0==temp1) { //若前后两次值相同，说明有键按下
                key_exe(temp1); //执行按键响应函数 key_exe
            }
        }
    }
}
```

2.6 对 C 源文件的编辑、编译及 HEX 文件的烧写

具体请参考《AVR 单片机的开发环境建立及使用教程》。

第三章 蜂鸣器实验

3.1 实验所需要的开发工具/软件

- 1) 安装 WinAVR20070525 编译开发软件
- 2) 安装 Source Insight 软件（可选）
- 3) 简易并行 ISP 下载线

- 4) 安装 PonyProg 单片机 ISP 下载软件
- 5) USB 电源线一根
- 6) 实验板一块

3.2 MCU 相关硬件基础知识介绍

请参考《ATMEGA16 内部资源使用教程》的第 1 章。

3.3 实验硬件原理图

请参考《AVR8 开发板原理图》。

3.4 实验功能

蜂鸣器间隔鸣叫 3 次，鸣叫一次时间长为 1 秒。

3.5 实验源代码及注释分析

Program 3.1 (文件名为 BUZZER.C)

```
#include <avr/io.h> //包含单片机的定义文件
#define uchar unsigned char //定义用 uchar 代替 unsigned char
#define uint unsigned int //定义用 uint 代替 unsigned int
#define Buzzer_On (PORTD|=(1<<PD7)) //使 PD^7=0，开启蜂鸣器
#define Buzzer_Off (PORTD&=~(1<<PD7)) //使 PD^7=1，关闭蜂鸣器

/*-----
port_init : IO 口初始化，所有的 IO 口都为输出口，且输出高电平
输入参数：无
输出参数：无
-----*/
void port_init(void)
{
    PORTA = 0xFF;
    DDRA = 0xFF;
    PORTB = 0xFF;
    DDRB = 0xFF;
    PORTC = 0xFF;
    DDRC = 0xFF;
    PORTD = 0xFF;
    DDRD = 0xFF;
}
```

```

}

/*-----*/
delay：延时函数，由输入参数 t 决定延时长短
输入参数：t—无符号字符型，取值范围 0~255
输出参数：无
-----*/
void delay(uchar t)
{
    uchar i;
    uint j;
    for(i=0;i<t;i++){
        for(j=0;j<60000;j++){ //延时大约 5ms
            asm("nop");
        }
    }
}

/*-----*/
main：主函数
输入参数：无
输出参数：无
-----*/
void main(void)//主函数
{
    uchar n; //定义无符号字符型局部变量 n
    port_init();
    Buzzer_Off;//先关

    for(n=0;n<3;n++){ //循环 3 次
        Buzzer_On;
        delay(200); //延时大约 1s
        Buzzer_Off;
        delay(200); //延时大约 1s
    }
    while(1);//进入死循环
}

```

3.6 对 C 源文件的编辑、编译及 HEX 文件的烧写

具体请参考《AVR 单片机的开发环境建立及使用教程》。

第四章 中断式按键实验

4.1 实验所需要的开发工具/软件

- 1) 安装 WinAVR20070525 编译开发软件
- 2) 安装 Source Insight 软件（可选）
- 3) 简易并行 ISP 下载线
- 4) 安装 PonyProg 单片机 ISP 下载软件
- 5) USB 电源线一根
- 6) 实验板一块

4.2 MCU 相关硬件基础知识介绍

请参考《ATMEGA16 内部资源使用教程》的第 1 章和第 2 章。

4.3 实验硬件原理图

请参考《AVR8 开发板原理图》。

4.4 实验功能

按 KEY1~KEY4 中的任何一个键，会将对应得一个 LED 灯点亮。

- 1) 若按下 KEY1 键，则 LED0 被点亮，其余 7 个 LED 灯全灭；
- 2) 若按下 KEY2 键，则 LED1 被点亮，其余 7 个 LED 灯全灭；
- 3) 若按下 KEY3 键，则 LED2 被点亮，其余 7 个 LED 灯全灭；
- 4) 若按下 KEY4 键，则 LED3 被点亮，其余 7 个 LED 灯全灭；
- 5) 若同时按下两个以上的键，则 8 个 LED 灯全灭。

4.5 实验源代码及注释分析

Program 4.1（文件名为 INT_BUZZER.C）

```
#include <avr/io.h> //包含单片机的定义文件
#include <avr/signal.h>
#include <avr/interrupt.h>
#define uchar unsigned char //定义用 uchar 代替 unsigned char
#define uint unsigned int //定义用 uint 代替 unsigned int
```

更多资料请登陆 www.mcudriver.cn 本站主要探讨 AVR、51、及 Freescale8 位单片机。提供常用电子元器件的零售、邮购业务。专注于 1-wire、I2C、SPI 等串行器件的驱动开发等。 版权所有 转载需注明出处


```
#define LED0_On (PORTC&=~(1<<PC0)) //使 PC^0=0，点亮 LED0
#define LED1_On (PORTC&=~(1<<PC1)) //使 PC^1=0，点亮 LED1
#define LED2_On (PORTC&=~(1<<PC2)) //使 PC^2=0，点亮 LED2
#define LED3_On (PORTC&=~(1<<PC3)) //使 PC^3=0，点亮 LED3
```

```
/*-----
port_init：IO 口初始化，所有的 IO 口都为输出口，且输出高电平
输入参数：无
输出参数：无
-----*/
```

```
void port_init(void)
{
    PORTA = 0xFF;
    DDRA = 0xFF;
    PORTB = 0xFF;
    DDRB = 0xFF;
    PORTC = 0xFF;
    DDRC = 0xFF;
    PORTD = 0xFF;
    DDRD = 0xFF;
}
```

```
/*-----
INT_Init：外部中断初始化
输入参数：无
输出参数：无
-----*/
```

```
void INT_Init()
{
    cli(); //关总中断
    MCUCR = 0x02; //INT0 下降沿触发中断
    GICR = 0x40; //使能外部中断 0
    sei(); //开总中断
}
```

```
/*-----
delay：延时函数，由输入参数 t 决定延时长短
输入参数：t—无符号字符型，取值范围 0~255
输出参数：无
-----*/
```

```
void delay(uchar t)
{
    uchar i;
    uint j;
    for(i=0;i<t;i++){
        for(j=0;j<60000;j++){ //延时大约 5ms
            asm("nop");
        }
    }
}
```

```

    }
}
}
/*-----*/
key_exe : 按键响应函数
输入参数 : keynum—按键值
输出参数 : 无
-----*/
void key_exe(uchar keynum)
{
    if(keynum ==0xE0) LED0_On; //key1 按下 , LED0 点亮
    else if(keynum ==0xD0) LED1_On; //key2 按下 , LED1 点亮
    else if(keynum ==0xB0) LED2_On; //key3 按下 , LED2 点亮
    else if(keynum ==0x70) LED3_On; //key4 按下 , LED3 点亮
    else PORTC=0xff; //错误键值 , LED 全灭
}
/*-----*/
SIGNAL(SIG_INTERRUPT0) : INT0 中断服务程序
输入参数 : 无
输出参数 : 无
-----*/
SIGNAL(SIG_INTERRUPT0)
{
    uchar temp_pd,temp0,temp1;
    temp_pd=PIND; //读取 IO 口的引脚值
    temp0=temp_pd&0xf0; //屏蔽低四位无效值
    if(temp0!=0xf0){ //如果不为 0xf0 , 说明可能有键按下
        //延时去抖动 , 并再次检测键值

        delay(6); //大约 30ms
        temp_pd=PIND;
        temp1=temp_pd&0xf0; //屏蔽低四位无效值
        if(temp0==temp1) { //若前后两次值相同 , 说明有键按下
            key_exe(temp1); //执行按键响应函数 key_exe
        }
    }
}
/*-----*/
main : 主函数
输入参数 : 无
输出参数 : 无
-----*/
void main(void)//主函数
{

```

```
port_init();
//将与按钮相连的四个 IO 口设为输入口：
DDRD=~((1<<PD7)|(1<<PD6)|(1<<PD5)|(1<<PD4));
INT_Init();
while(1) //进入循环，等待 INTO 中断，检测是否有键按下
{
}
}
```

4.6 对 C 源文件的编辑、编译及 HEX 文件的烧写

具体请参考《AVR 单片机的开发环境建立及使用教程》。

第五章 定时器控制流水灯

5.1 实验所需要的开发工具/软件

- 1) 安装 WinAVR20070525 编译开发软件
- 2) 安装 Source Insight 软件（可选）
- 3) 简易并行 ISP 下载线
- 4) 安装 PonyProg 单片机 ISP 下载软件
- 5) USB 电源线一根
- 6) 实验板一块

5.2 MCU 相关硬件基础知识介绍

请参考《ATMEGA16 内部资源使用教程》的第 1 章和第 3 章。

5.3 实验硬件原理图

请参考《AVR8 开发板原理图》。

5.4 实验功能

LED 1、3、5、7 和 2、4、6、8 交叉点亮。

5.5 实验源代码及注释分析

Program 5.1（文件名为 TIMER0.C）

```

#include <avr/io.h> //包含单片机的定义文件

#include <avr/signal.h>

#include <avr/interrupt.h>

#define uchar unsigned char //定义用 uchar 代替 unsigned char
#define uint unsigned int   //定义用 uint 代替 unsigned int
#define LED0_On (PORTC&=~(1<<PC0)) //使 PC^0=0，点亮 LED0
#define LED1_On (PORTC&=~(1<<PC1)) //使 PC^1=0，点亮 LED1
#define LED2_On (PORTC&=~(1<<PC2)) //使 PC^2=0，点亮 LED2
#define LED3_On (PORTC&=~(1<<PC3)) //使 PC^3=0，点亮 LED3
/*-----*/
port_init：IO 口初始化，所有的 IO 口都为输出口，且输出高电平
输入参数：无
输出参数：无
-----*/

void port_init(void)
{
    PORTA = 0xFF;
    DDRA = 0xFF;
    PORTB = 0xFF;
    DDRB = 0xFF;
    PORTC = 0xFF;
    DDRC = 0xFF;
    PORTD = 0xFF;
    DDRD = 0xFF;
}
/*-----*/

delay：延时函数，由输入参数 t 决定延时长短
输入参数：t—无符号字符型，取值范围 0~255
输出参数：无
-----*/

void delay(uchar t)
{
    uchar i;
    uint j;
    for(i=0;i<t;i++){
        for(j=0;j<60000;j++){ //延时大约 5ms
            asm("nop");
        }
    }
}
/*-----*/

SIGNAL(SIG_INTERRUPT0)：INT0 中断服务程序
输入参数：无

```

输出参数：无

```

-----*/
SIGNAL(SIG_INTERRUPT0)
{
    uchar temp_pd,temp0,temp1;
    temp_pd=PIND; //读取 IO 口的引脚值
    temp0=temp_pd&0xf0; //屏蔽低四位无效值
    if(temp0!=0xf0){ //如果不为 0xf0，说明可能有键按下
        //延时去抖动，并再次检测键值

        delay(6); //大约 30ms
        temp_pd=PIND;
        temp1=temp_pd&0xf0; //屏蔽低四位无效值
        if(temp0==temp1) { //若前后两次值相同，说明有键按下
            key_exe(temp1); //执行按键响应函数 key_exe
        }
    }
}
/*-----*/
main：主函数
输入参数：无
输出参数：无
-----*/
void main(void) //主函数
{
    port_init();
    //将与按键相连的四个 IO 口设为输入口：
    DDRD=~((1<<PD7)|(1<<PD6)|(1<<PD5)|(1<<PD4));

    while(1) //进入循环，等待 INTO 中断，检测是否有键按下
    {
    }
}

```

5.6 对 C 源文件的编辑、编译及 HEX 文件的烧写

具体请参考《AVR 单片机的开发环境建立及使用教程》。

第六章 RS232 串口通信实验

6.1 实验所需要的开发工具/软件

- 1) 安装 WinAVR20070525 编译开发软件
- 2) 安装 Source Insight 软件 (可选)
- 3) 简易并行 ISP 下载线
- 4) 安装 PonyProg 单片机 ISP 下载软件
- 5) USB 电源线一根
- 6) 串口线一根
- 7) 实验板一块

6.2 MCU 相关硬件基础知识介绍

请参考《ATMEGA16 内部资源使用教程》的第 4 章。

6.3 实验硬件原理图

请参考《AVR8 开发板原理图》。

6.4 实验功能

先向 PC 端发送一串字符“This is a RS232 Communication Program!”, 再进入无限循环, 等待 PC 端发送字符至下位机, 下位机每接收到一个字符之后, 都将该字符发送回 PC 端回显。

6.5 实验源代码及注释分析

Program 6.1 (文件名为 RS232.C)

```
#include <avr/io.h> //包含单片机的定义文件
#define uchar unsigned char //定义用 uchar 代替 unsigned char
#define uint unsigned int //定义用 uint 代替 unsigned int
#define fosc 12000000 //定义晶振大小：12MHZ
#define baud 9600 //设置 RS232 通信波特率
/*-----
port_init : IO 口初始化, 所有的 IO 口都为输出口, 且输出高电平
输入参数 : 无
输出参数 : 无
```

```

-----*/
void port_init(void)
{
    PORTA = 0xFF;
    DDRA = 0xFF;
    PORTB = 0xFF;
    DDRB = 0xFF;
    PORTC = 0xFF;
    DDRC = 0xFF;
    PORTD = 0xFF;
    DDRD = 0xFF;
}
/*-----*/

uart_init : UART 初始化
输入参数：无
输出参数：无
-----*/

void uart_init(void)
{
    UCSRB=(1<<RXEN)|(1<<TXEN);//允许发送和接收
    UBRRL=(fosc/16/(baud+1))%256;
    UBRRH=(fosc/16/(baud+1))/256; //设置波特率寄存器，以获得波特率为 9600
    UCSRC=(1<<URSEL)|(1<<UCSZ1)|(1<<UCSZ0);//8 位数据+1 位 STOP 位(无校验)
}
/*-----*/

getchar : 获取字符
输入参数：无
输出参数：无
-----*/

uchar getchar(void)
{
    while(!(UCSRA& (1<<RXC)));
    return UDR;
}
/*-----*/

putchar : 发送一个字符
输入参数：无
输出参数：无
-----*/

void putchar(uchar c)
{
    while (!(UCSRA&(1<<UDRE)));
    UDR=c;
}

```

```
/*-----  
puts：发送一个字符串，并带回车换行  
输入参数：无  
输出参数：无  
-----*/  
int puts(char *s)  
{  
    while (*s)  
    {  
        putchar(*s);  
        s++;  
    }  
    putchar(0x0a);  
    putchar(0x0d); //回车换行  
    return 1;  
}  
/*-----  
main：主函数  
输入参数：无  
输出参数：无  
-----*/  
void main(void)  
{  
    uchar temp;  
    port_init();  
    uart_init();  
    puts("This is a RS232 Communication Program!"); //显示一串字符  
    while(1) //进入循环  
    {  
        temp= getchar(); //等待接收从 PC 端发过来的字符，赋给 temp  
        putchar(temp); //将接收到的字符，再发送回 PC 端显示  
    }  
}
```

6.6 对 C 源文件的编辑、编译及 HEX 文件的烧写

具体请参考《AVR 单片机的开发环境建立及使用教程》及《串口使用教程》。

6.7 实验结果验证

在开发板上电前，先打开配套安装软件文件夹中的串口调试助手软件：comdebug.exe，进入下图界面：



将串口线一端接 PC 一端接上开发板，再连好 USB 线将开发板上电，在串口调试助手接收区，首先会接收到一串字符“This is a RS232 Communication Program!”，再进入无限循环，等待 PC 端发送字符至下位机。

用户可将发送区清空，在发送区输入任何字符，点击“发送(S)”控件，将字符发送给开发板，此时接收区也会接收到该字符(由于开发板将接收到的字符又发送回 PC)。

第七章 A/D 转换实验

7.1 实验所需要的开发工具/软件

- 1) 安装 WinAVR20070525 编译开发软件
- 2) 安装 Source Insight 软件 (可选)
- 3) 简易并行 ISP 下载线
- 4) 安装 PonyProg 单片机 ISP 下载软件
- 5) USB 电源线一根
- 6) 串口线一根
- 7) 实验板一块

7.2 MCU 相关硬件基础知识介绍

请参考《ATMEGA16 内部资源使用教程》的第 4 章和第 5 章。

7.3 实验硬件原理图

请参考《AVR8 开发板原理图》。

7.4 实验功能

先向 PC 端发送一串字符“This is a A/D Conversion Program!”,再进入无限循环,单片机连续转换 ADC 通道 0 的输入电压,并将转换结果发送到 PC 端。

7.5 实验源代码及注释分析

Program 7.1 (文件名为 ADC0.C)

```
#include <avr/io.h> //包含单片机的定义文件
#define uchar unsigned char //定义用 uchar 代替 unsigned char
#define uint unsigned int //定义用 uint 代替 unsigned int
#define fosc 12000000 //定义晶振大小：12MHZ
#define baud 9600 //设置 RS232 通信波特率
uint adc_rel; //AD 转换结果 adc_rel
uchar adc_mux; //AD 转换通道
/*-----
port_init : IO 口初始化,所有的 IO 口都为输出口,且输出高电平
输入参数：无
输出参数：无
-----*/
void port_init(void)
{
    PORTA = 0xFF;
    DDRA = 0xFF;
    PORTB = 0xFF;
    DDRB = 0xFF;
    PORTC = 0xFF;
    DDRC = 0xFF;
    PORTD = 0xFF;
    DDRD = 0xFF;
}
/*-----
uart_init : UART 初始化
```

更多资料请登陆 www.mcudriver.cn 本站主要探讨 AVR、51、及 Freescale8 位单片机。提供常用电子元器件的零售、邮购业务。专注于 1-wire、I2C、SPI 等串行器件的驱动开发等。 版权所有 转载需注明出处

输入参数：无

输出参数：无

```
-----*/
void uart_init(void)
{
    UCSRB=(1<<RXEN)|(1<<TXEN);//允许发送和接收
    UBRRL=(fosc/16/(baud+1))%256;
    UBRRH=(fosc/16/(baud+1))/256; //设置波特率寄存器，以获得波特率为 9600
    UCSRC=(1<<URSEL)|(1<<UCSZ1)|(1<<UCSZ0);//8 位数据+1 位 STOP 位(无校验)
}
/*-----*/
```

getchar：获取字符

输入参数：无

输出参数：无

```
-----*/
uchar getchar(void)
{
    while(!(UCSRA&(1<<RXC)));
    return UDR;
}
/*-----*/
```

putchar：发送一个字符

输入参数：无

输出参数：无

```
-----*/
void putchar(uchar c)
{
    while(!(UCSRA&(1<<UDRE)));
    UDR=c;
}
/*-----*/
```

puts：发送一个字符串，并带回车换行

输入参数：无

输出参数：无

```
-----*/
int puts(char *s)
{
    while (*s)
    {
        putchar(*s);
        s++;
    }
    putchar(0x0a);
    putchar(0x0d); //回车换行
}
/*-----*/
```

```

    return 1;
}
/*-----*/
adc_init：对单片机内部 ADC 功能的初始化
输入参数：无
输出参数：无
-----*/
void adc_init(void)
{
    adc_mux=0;//通道 0
    DDRC=0x00;
    PORTC=0x00; //设置 PC 口为输入口（用作 ADC 输入）
    ADCSRA = 0x00;
    //选择内部 AVCC 为基准,输入通道为 0 通道：
    ADMUX =(1<<REFS0)|(adc_mux&0x0f);
    ACSR  =(1<<ACD); //关闭模拟比较器
    //分频设置：64 分频
    ADCSRA=(1<<ADEN)|(1<<ADSC)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1); }
}
/*-----*/
SIGNAL：ADC 转换完成，中断服务子程序
输入参数：无
输出参数：无
-----*/
SIGNAL(SIG_ADC)
{
    adc_rel=ADC&0x3ff;//读取 AD 转换结果
    //选择内部 AVCC 为基准,输入通道为 0 通道，此处可以切换到其他通道
    ADMUX =(1<<REFS0)|(adc_mux&0x0f); //若更改了通道，需要加一定的延时
    ADCSRA|=(1<<ADSC); //启动 AD 转换
}
/*-----*/
ADCtoBCD：将 ADC 结果转换成电压值
输入参数：temp：ADC 的转换值
输出参数：无
-----*/
void ADCtoBCD(uint temp)
{
    unsigned char result;
    unsigned char num_bit[3];
    temp=(unsigned int)(((unsigned long)((unsigned long)temp*Vref))/0x3ff);
    temp2=temp;
    /******将转换结果发送到串行口******/
    for(i=0;i<3;i++)

```

```

{
    num_bit[2-i]=temp%10;
    temp=temp/10;
}
putchar(48+num_bit[0]);
putchar('.');
putchar(48+num_bit[1]);
putchar(48+num_bit[2]);
putchar(' ');
putchar(0x0d);
putchar(0x0a);//回车换行
}
/*-----*/
delay : 短延时
输入参数 : n : 延时长短参数
输出参数 : 无
-----*/
void delay(uchar n)
{
    uint i;
    for(i=0;i<n;i++)
        for(j=0;j<60000;j++)  asm("nop");//5ms
}
/*-----*/
main : 主函数
输入参数 : 无
输出参数 : 无
-----*/
void main(void)
{
    cli();//close interrupt
    port_init();
    uart_init();
    adc_init();
    sei();//enable interrupt
    puts("This is a A/D Conversion Program!");//显示一串字符
    while(1) //进入循环
    {
        ADCtoBCD(adc_rel);//发送转换结果至 PC
        delay(200);//delay 1s
    }
}

```

7.6 对 C 源文件的编辑、编译及 HEX 文件的烧写

更多资料请登陆 www.mcudriver.cn 本站主要探讨 AVR、51、及 Freescale8 位单片机。提供常用电子元器件的零售、邮购业务。专注于 I-wire、I2C、SPI 等串行器件的驱动开发等。 版权所有 转载需注明出处

具体请参考《AVR 单片机的开发环境建立及使用教程》及《串口使用教程》。

7.7 实验结果验证

参考 6.7 节，打开串口调试助手软件，在接收区会接收到从开发板发过来的 AD 转换电压值。用户可用万用表测量开发板上的 ADC 的输入电压值与转换值是否相符。也可旋转可调电阻的旋钮来改变输入电压值，来测试转换值是否随着它改变。

第八章 PS/2 键盘实验

8.1 实验所需要的开发工具/软件

- 1) 安装 WinAVR20070525 编译开发软件
- 2) 安装 Source Insight 软件（可选）
- 3) 简易并行 ISP 下载线
- 4) 安装 PonyProg 单片机 ISP 下载软件
- 5) USB 电源线一根
- 6) 串口线一根
- 7) PC 及 PS/2 接口键盘各一个
- 8) 实验板一块

8.2 MCU 相关硬件基础知识介绍

请参考《ATMEGA16 内部资源使用教程》的第 1、2、4 章。

8.3 实验硬件原理图

请参考《AVR8 开发板原理图》。

8.4 实验功能

先向 PC 端发送一串字符“**This is a PS/2 Comunication Program!**”，再进入无限循环；当 PS/2 键盘有按键被按下时，下位机立刻将对应的按键码发送到 PC 端显示。

8.5 实验源代码及注释分析

Program 8.1 (文件名为 PS2.C)

```
#include <avr/io.h> //包含单片机的定义文件
#define uchar unsigned char //定义用 uchar 代替 unsigned char
#define uint unsigned int //定义用 uint 代替 unsigned int
#define fosc 12000000 //定义晶振大小：12MHZ
#define baud 9600 //设置 RS232 通信波特率
static unsigned char IntNum=0; //中断次数计数
static unsigned char KeyV=0; //键值
static unsigned char DisNum=0; //显示用指针
static unsigned char Key_UP=0; //Key_UP 是键松开标识
static unsigned char BF=0; //标识是否有字符被收到
unsigned char UnShifted[59][2] = { //PS/2 键盘的键码 (未按 shift 键的对应码)
0x1C, 'a',
0x32, 'b',
0x21, 'c',
0x23, 'd',
0x24, 'e',
0x2B, 'f',
0x34, 'g',
0x33, 'h',
0x43, 'i',
0x3B, 'j',
0x42, 'k',
0x4B, 'l',
0x3A, 'm',
0x31, 'n',
0x44, 'o',
0x4D, 'p',
0x15, 'q',
0x2D, 'r',
0x1B, 's',
0x2C, 't',
0x3C, 'u',
0x2A, 'v',
0x1D, 'w',
0x22, 'x',
0x35, 'y',
0x1A, 'z',
0x45, '0',
0x16, '1',
0x1E, '2',
0x26, '3',
0x25, '4',
0x2E, '5',
```

```

0x36, '6',
0x3D, '7',
0x3E, '8',
0x46, '9',
0x0E, '^',
0x4E, '|',
0x55, '=' ,
0x5D, '\\',
0x29, '|',
0x54, '[',
0x5B, ']',
0x4C, ';',
0x52, '\',
0x41, ',',
0x49, ':',
0x4A, '/',
0x71, ':',
0x70, '0',
0x69, '1',
0x72, '2',
0x7A, '3',
0x6B, '4',
0x73, '5',
0x74, '6',
0x6C, '7',
0x75, '8',
0x7D, '9',
};
/*-----*/
port_init : IO 口初始化，所有的 IO 口都为输出口，且输出高电平
输入参数：无
输出参数：无
-----*/

void port_init(void)
{
    PORTA = 0xFF;
    DDRA = 0xFF;
    PORTB = 0xFF;
    DDRB = 0xFF;
    PORTC = 0xFF;
    DDRC = 0xFF;
    PORTD = 0xFF;
    DDRD = 0xFF;
}

```



```

/*-----
uart_init : UART 初始化
输入参数 : 无
输出参数 : 无
-----*/
void uart_init(void)
{
    UCSRB=(1<<RXEN)|(1<<TXEN);//允许发送和接收
    UBRRL=(fosc/16/(baud+1))%256;
    UBRRH=(fosc/16/(baud+1))/256; //设置波特率寄存器 , 以获得波特率为 9600
    UCSRC=(1<<URSEL)|(1<<UCSZ1)|(1<<UCSZ0);//8 位数据+1 位 STOP 位(无校验)
}

/*-----
getchar : 获取字符
输入参数 : 无
输出参数 : 无
-----*/
uchar getchar(void)
{
    while(!(UCSRA& (1<<RXC)));
    return UDR;
}

/*-----
putchar : 发送一个字符
输入参数 : 无
输出参数 : 无
-----*/
void putchar(uchar c)
{
    while (!(UCSRA&(1<<UDRE)));
    UDR=c;
}

/*-----
puts : 发送一个字符串 , 并带回车换行
输入参数 : 无
输出参数 : 无
-----*/
int puts(char *s)
{
    while (*s)
    {
        putchar(*s);
        s++;
    }
}

```

```

    }
    putchar(0x0a);
    putchar(0x0d); //回车换行
    return 1;
}

void Decode(unsigned char ScanCode)
{
    uchar No;
    if (!Key_UP) { //当键盘松开时
        switch (ScanCode){
            case 0xF0 : // 当收到 0xF0 , Key_UP 置 1 表示断码开始
                Key_UP = 1; break;
            case 0x12 : // 左 SHIFT
                Shift = 1; break;
            case 0x59 : // 右 SHIFT
                Shift = 1; break;
            default:
                if (DisNum > 15) { //每接收超过 15 个字符时 , 换行输出
                    putchar(0x0d);
                    putchar(0x0a);
                    DisNum = 0;
                }
                for (No = 0; (UnShifted[No][0] != ScanCode)
                    && (No < 59); No++); //查表显示
                if (UnShifted[No][0] == ScanCode)
                {
                    putchar(UnShifted[No][1]); //将按下的字符发送到 PC
                }
                DisNum++;
                break;
            }
        }
    }
    else{
        Key_UP = 0;
    }
    BF = 0; //标识字符处理完了
}

```

```

SIGNAL(INTERRUPT1)

```

```

{
    if ((IntNum > 0) && (IntNum < 9)){
        KeyV = KeyV >> 1; //因键盘数据是低>>高 , 结合上一句所以右移一位
        if (PIND & 0x10) KeyV = KeyV | 0x80; //当键盘数据线为 1 时为 1 到最高位
    }
}

```

```

}
IntNum++;
while (!(PIND&0x08)); //等待 PS/2CLK 拉高
if (IntNum > 10){
    IntNum = 0; //当中断 11 次后表示一帧数据收完，清变量准备下一次接收
    BF = 1; //标识有字符输入完了
    cli(); //关中断等显示完后再开中断（注：如这里不用 BF 和关中断直接
        //调 Decode()则所 Decode 中所调用的所有函数要声明为再入函数)
    }
}
}
/*-----*/
main：主函数
输入参数：无
输出参数：无
-----*/
void main(void)
{
    uchar temp;
    cli(); //关中断
    port_init();
    uart_init();
    MCUCR = 0x0;
    GICR = 0x80; //使能 INT1，低电平触发中断
    sei(); //开中断

    puts("This is a PS/2 Program!"); //显示一串字符
    while(1)
    {
        if(BF) Decode(KeyV);
        else sei(); //开中断
    }
}

```

8.6 对 C 源文件的编辑、编译及 HEX 文件的烧写

具体请参考《AVR 单片机的开发环境建立及使用教程》及《串口使用教程》。