

-----部分内容来自网络-----

//config.h 文件

#include<reg52.h>

typedef unsigned long ulong;

sbit DDSCLK = P2^4;       // P2\_4, DDS 时钟信号

sbit DDSEN = P2^5;       // P2\_5, DDS 片选或者写数据 有效信号

sbit DDSData =P2^5;       // P2\_6, DDS 数据信号

extern void DDSIni();

extern void LDelay(unsigned char DelayNum);

extern void Fre\_To\_DDSDData(unsigned char Fre);

extern void SDelay(unsigned char DelayNum);

extern void AD9833\_reset();

extern void AD9833\_writedata(unsigned int DDSdata);

extern void Load\_wave(unsigned int Contr\_Reg\_data,unsigned int Fre\_MSBdata,unsigned int Fre\_LSBdata,unsigned int Phs\_data);

//C Files

#include "config.h"

////////////////////////////////////

/\*函数描述: DDS 初始化

/\*写入参数: 无

/\*返回参数: 无

/\*作者: Mikewang

/\*日期: 2009-07-15

/\*描述:

/\*       根据硬件连接选通 MCU 与 AD9833 的数据线路; 将 AD9833 的内部数据进行清零操作

\*/////////////////////////////////////

void DDSIni()

{

    DDSEN=1;           //DDSEN disable

    DDSCLK=1;         //Clock high

    DDSData=0;

    AD9833\_reset();

}

////////////////////////////////////

/\*函数描述: 长延迟

/\*写入参数: Delaynum, 延迟时间: Delaynum\*1.6mS

/\*返回参数: 无

/\*作者: Mikewang

/\*日期: 2008.\*\*.\*\*

/\*描述:

```

/*
*////////////////////////////////////
void LDelay(unsigned char DelayNum)
{
    unsigned char Delay1p6mS=0XFF;
    while(DelayNum--)
    {
        while(Delay1p6mS--);
    }
}
////////////////////////////////////
/*函数描述：短延迟
/*写入参数：Delaynum，延迟时间：不详
/*返回参数：无
/*作者：Mikewang
/*日期：2008.**.**
/*描述：
/*
*////////////////////////////////////

void SDelay(unsigned char DelayNum)
{
    while(DelayNum--);
}

////////////////////////////////////
/*函数描述：AD9833 写数据操作
/*写入参数：AD9833 控制字，16bit
/*返回参数：无
/*作者：Mikewang
/*日期：2009-07-15
/*描述：
        标准时序
*////////////////////////////////////

void AD9833_writedata(unsigned int DDSdata)
{
    unsigned char data_num=0;
    unsigned int DDSdata_temp;

    DDSdata_temp=DDSdata;
    DDSCLK=1;
    SDelay(4);
    DDSEN=0;    //DDS able

```

```

SDelay(4);
for(data_num=0;data_num<16;data_num++)
{
    if(DDSdata_temp & 0X8000 )
    {
        DDSData=1;
    }
    else{
        DDSData=0;
    }
    SDelay(5);
    DDSCLK=0;
    SDelay(5);
    DDSData=0;
    SDelay(5);
    DDSCLK=1;
    DDSdata_temp=DDSdata_temp<<1;
    if(data_num < 15)
    {
        SDelay(50);
    }
}
DDSEN=1;
}
////////////////////////////////////
/*函数描述：AD9833 清零操作
/*写入参数： 无
/*返回参数： 无
/*作者： Mikewang
/*日期： 2009-07-15
/*描述：
        清零内容：控制寄存器复位、清零频率寄存器 1 和 2、清零相位寄存器 1 和 2
*////////////////////////////////////
void AD9833_reset()
{
    //add code here
    AD9833_writedata(0x0100);    //寄存器复位
    AD9833_writedata(0x2000);    //准备清空频率寄存器 0

    AD9833_writedata(0x4000);    //Filled with 0
    AD9833_writedata(0x4000);    //Filled with 0
    AD9833_writedata(0x2000);    //Prepare for clear Fre1
    AD9833_writedata(0x8000);    //Filled with 0
    AD9833_writedata(0x8000);    //Filled with 0

```

```

AD9833_writedata(0xD000);      //clear PHS 0
AD9833_writedata(0xF000);      //clear PHS 1
}
////////////////////////////////////
/*函数描述： AD9833 加载波新
/*写入参数： 控制字、32bit（28bit 有效）的频率数据，16bit（12bit 有效）的相位数据
/*返回参数： 无
/*作者： Mikewang
/*日期： 2009-07-15
/*描述：
        根据不同的控制字完成不同的功能，建议使用 28bit 连续写模式
*////////////////////////////////////
void Load_wave(unsigned int Contr_Reg_data,unsigned int Fre_MSBdata,unsigned int
Fre_LSBdata,unsigned int Phs_data)
{
    //add
    unsigned int Fre_MSBdata_temp=Fre_MSBdata;
    unsigned int Fre_LSBdata_temp=Fre_LSBdata;
    unsigned int Contr_Reg_data_temp=Contr_Reg_data;
    unsigned int Phs_data_temp=Phs_data;
    AD9833_writedata(Contr_Reg_data_temp);

    AD9833_writedata(Fre_LSBdata_temp);
    AD9833_writedata(Fre_MSBdata_temp);

    AD9833_writedata(Phs_data);
}

//main files
#include "config.h"
main()
{
    ulong fout;
    volatile unsigned int freql,freqh;
    volatile ulong temp0,temp1;
    ulong Mclk;
    LDelay(0XFF);
    DDSIni();

    fout = 50000;      //你想输出的频率 60000hz-----修改输出频率
    Mclk = 12000000;    //你的晶振频率 12000000hz----不改晶振就不用改了

    temp0 = Mclk/16384 ;

```

```
fout  = fout*16384;
temp1 = fout/temp0;

freq1 = temp1&0x3fff;
freq1 = (freq1|0x4000);

freqh = temp1>>14;
freqh = (freqh&0x3fff);
freqh =  freqh|0x4000;

Load_wave(0x2002,freqh,freq1,0xD000);
do{
}while(1);
}
```

农民也玩嵌入式

-----[zhangzhenliang@msn.cn](mailto:zhangzhenliang@msn.cn)