

DDS 模块使用说明书

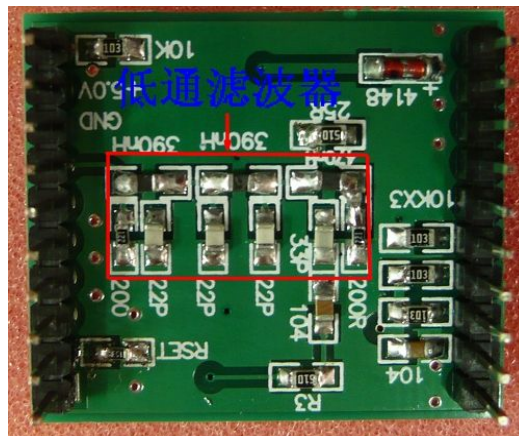
目录

1、DDS 模块实物图·····	2
2、DDS 模块管脚定义·····	2
3、与单片机的硬件连接及程序（并行送控制字方式）·····	3
4、与单片机的硬件连接及程序（串行送控制字方式）·····	4
5、DDS 模块方波的产生·····	7
6、输出波形及频率特性·····	8
7、电路参考原理图·····	14
8、电路参考 PCB 图·····	15

1、DDS 模块实物图（正面）



DDS 模块实物图（反面）



该模块系统时钟频率为 125MHZ，支持串行和并行送控制字方式。最大不失真输出频率可达 40MHZ（该模块测试的实际值）

2、各管脚定义（只针对此模块）

CLK: 系统时钟频率输出 (125MHZ)

RESET: 控制 DDS 内部 DAC 的输出电流（当需要控制输出信号的幅度时，可以控制该脚的电压值从而控制 DDS 信号输出的幅度）

00A: 内部高速比较器的正相输出端 (对应 AD9850 的 14 管脚)

00B: 内部高速比较器的反相输出端（对应 AD9850 的 13 管脚）

VIP: 内部高速比较器的同相输入端（对应 AD9850 的 16 管脚）

VIN: 内部高速比较器的反相输入端（对应 AD9850 的 15 管脚）

F0: 频率输出端 (已经经过了典型低通滤波器后的波形)

GND: 输入电源地

VDD: 输入电源正极 (+5V)

RST: AD9850 复位端（高电平，对芯片进行操作前需将该脚置为高电平，复位完成后将其置为低电平 **RST AD9850**）

FOUP: 数据更新位（串行/并行数据输入时的输入位 FO OD AD9850）

WCLK: 时钟输入端（串行/并行数据输入时的输入位 CLK AD9850）

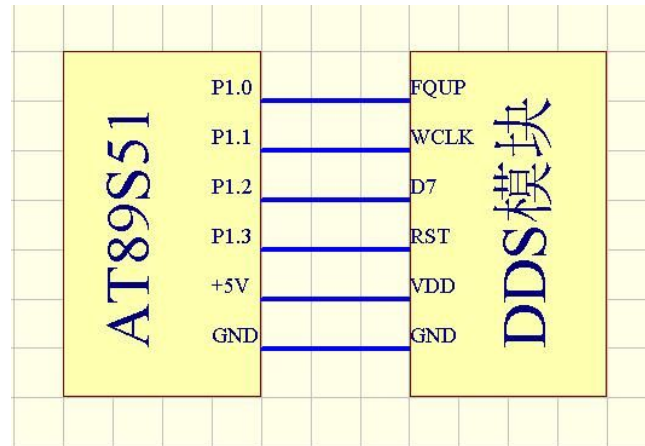
D0: 数据输入端（并行输入数据时的低位）

D1: 数据输入端 D2: 数据输入端 D3: 数据输入端

D4: 数据输入端 D5: 数据输入端 D6: 数据输入端

D7: 数据输入端（并行输入数据时的高位。当进行串行送数据时，该位是串行的数据输入位 DataIn AD9850）

3、与单片机的硬件连接及程序（串行送控制字方式）



串行读写程序：该程序实现的是固定频点的输出 1KHZ

```
#include<regx51.h>
#include <intrins.h>
sbit FQ_QD_AD9850    = P1^0;
sbit CLK_AD9850      = P1^1;
sbit DataIn_AD9850   = P1^2;
sbit RST_AD9850      = P1^3;
unsigned long int freq = 0;
unsigned char Control_AD9850 = 0x00;
// Phase0 ,power on    mode and 6 REFCLK Multiplier disable
void SentFreq()
{
    unsigned char i;
    unsigned int temp;
    FQ_QD_AD9850=0;

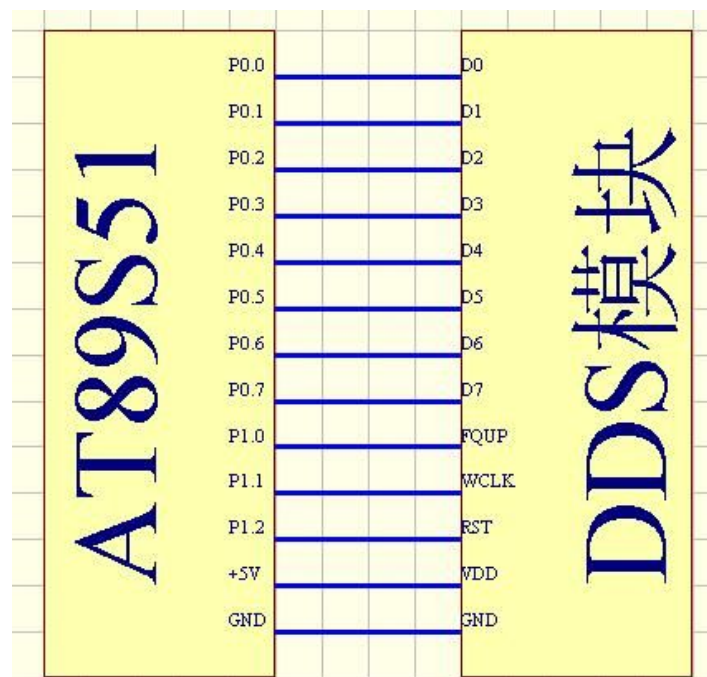
    for(i=0;i<32;i++)    //串口数据  输入频率控制字
    {
        CLK_AD9850= 0;
        temp=( ( freq  >> i ) & 1 );
        DataIn_AD9850 = temp;
        CLK_AD9850 = 1;
    }
    for(i=0;i<8;i++)
    //phase-b4 ph-b3 ph-b2 ph-b1 ph-b0
    {
        CLK_AD9850 = 0;
        temp=( ( Control_AD9850 >> i ) & 1 );
        DataIn_AD9850 = temp;
        CLK_AD9850 = 1;
    }
    CLK_AD9850 = 0;
    FQ_QD_AD9850= 1;
}
```

```

    FQ_QD_AD9850 = 0;
}
void Set_Freq(unsigned long int Frequency)
{
    freq= (unsigned long int)(34.36*Frequency);
    // SYSCLK = 125 MHz  2^32/125000000=34.36
    SentFreq();
}
void main()
{RST_AD9850=1;//复位 AD9850
  RST_AD9850=1;
  RST_AD9850=0;
  Set_Freq(1000); (如果单独执行，最好送两次频率控制字较为稳定)
  Set_Freq(1000);
    while(1)
    {
        ;
    }
}

```

4、与单片机的硬件连接图（并行送控制字方式）



并行读写程序：该程序实现的是固定频点的输出 1KHZ

```

#include<regx51.h>
#include <intrins.h>
sbit FQ_UD_AD9850=P1^0;//定义 I/O 口
sbit CLK_AD9850=P1^1;
sbit RST_AD9850= P1^2;

```

```
unsigned long int freq = 0;
unsigned char Control_AD9850 = 0x00; // Phase0 ,power on mode and 6
REFCLK Multiplier disable
unsigned char W1=0X0e;
unsigned char W2=0X38;
unsigned char W3=0Xe3;
unsigned char W4=0X8e;
void Parallel2Serial_AD9850(void)//并行写入控制字
{
    FQ_UD_AD9850=0;
    CLK_AD9850=0;
    P0=Control_AD9850;
    CLK_AD9850=1;
    CLK_AD9850=0;

    P0=W1;
    CLK_AD9850=1;
    CLK_AD9850=0;

    P0=W2;
    CLK_AD9850=1;
    CLK_AD9850=0;

    P0=W3;
    CLK_AD9850=1;
    CLK_AD9850=0;

    P0=W4;
    CLK_AD9850=1;
    CLK_AD9850=0;

    FQ_UD_AD9850=1;
    FQ_UD_AD9850=0;
}
```

```
void Set_Freq(float Frequency)//计算输出频率
{
    freq= (unsigned long int)( 34.36*Frequency); // SYSCLK = 125 MHz
    W4=(unsigned char)freq&0xff;
    freq=freq>>8;

    W3=(unsigned char)freq&0xff;
    freq=freq>>8;

    W2=(unsigned char)freq&0xff;
```

```
freq=freq>>8;

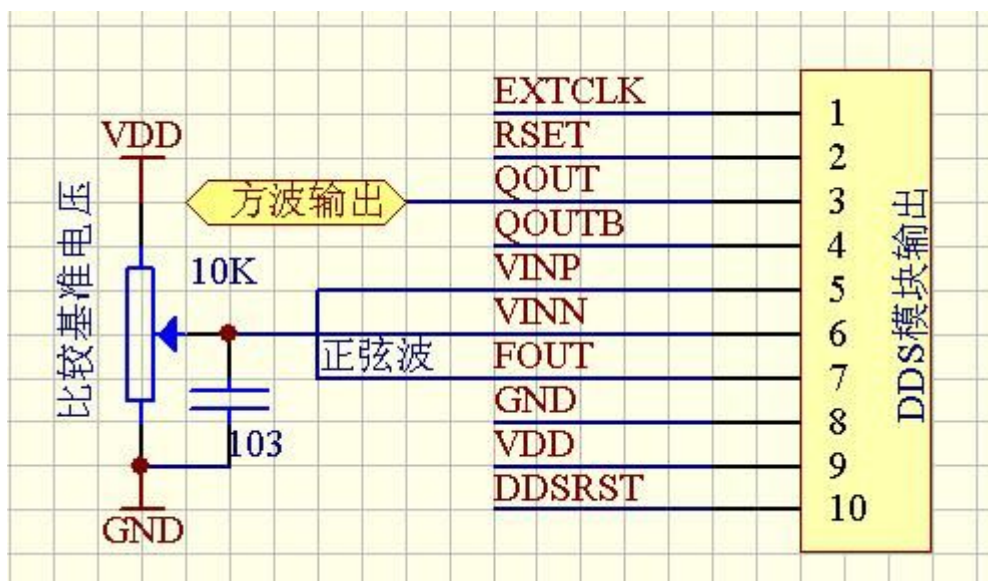
W1=(unsigned char)freq&0xff;

Parallel2Serial_AD9850();
}
void main(void)
{RST_AD9850=1;//复位模块电路
RST_AD9850=1;
RST_AD9850=0;

while(1)
{
Set_Freq(1000);//输出 1KHZ 频率
}
}
```

5、DDS 模块方波的产生

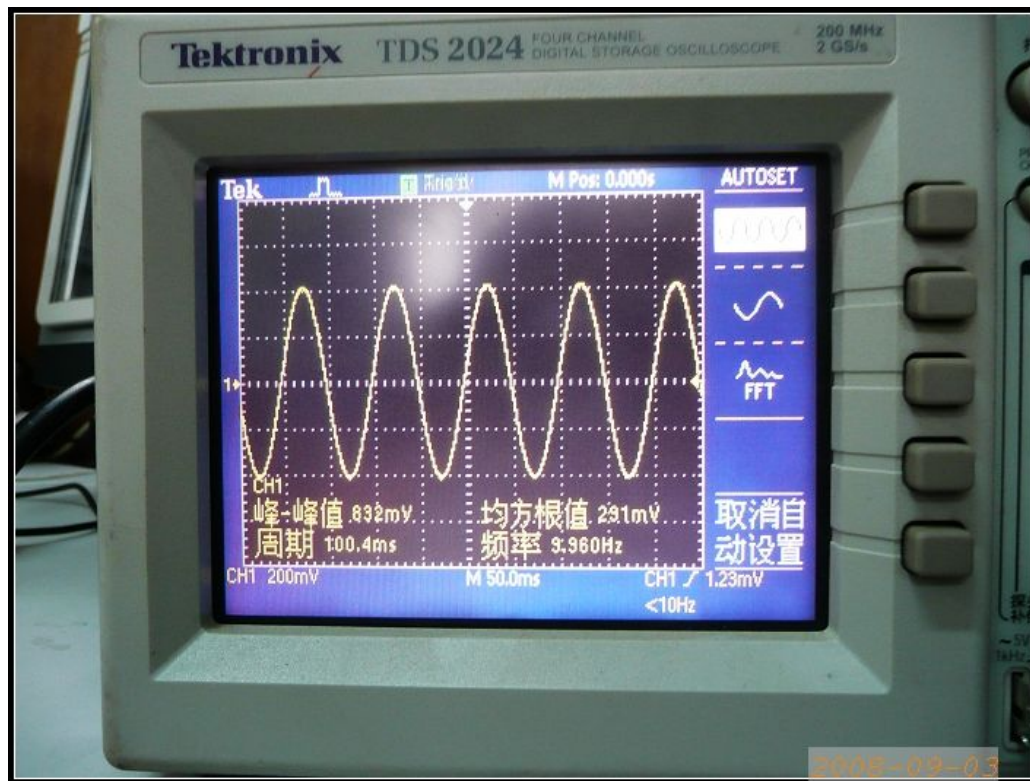
利用 AD9850 内部高速比较器产生方波，实现电路如下：



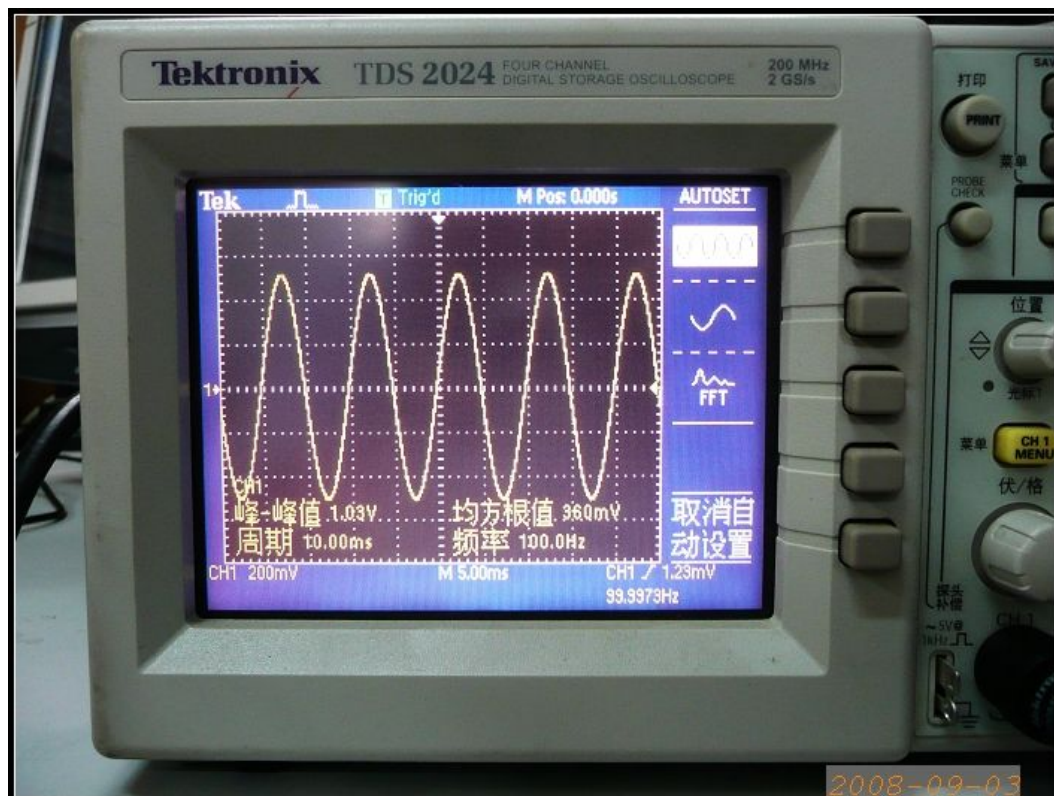
说明：该模块上面本身没有接 100K（为灵活使用预留），学习者如果要输出方波则需要外接电位器和电容，按上面的图所示连接。其原理是利用芯片 AD9850 内部的高速比较器将从模块 7 脚（正弦波 FOUT）输入到 5 脚（VINP 内部比较器的输入），从而实现在 QOUT 点输出方波。

6、输出波形及频率特性（频谱仪测量）

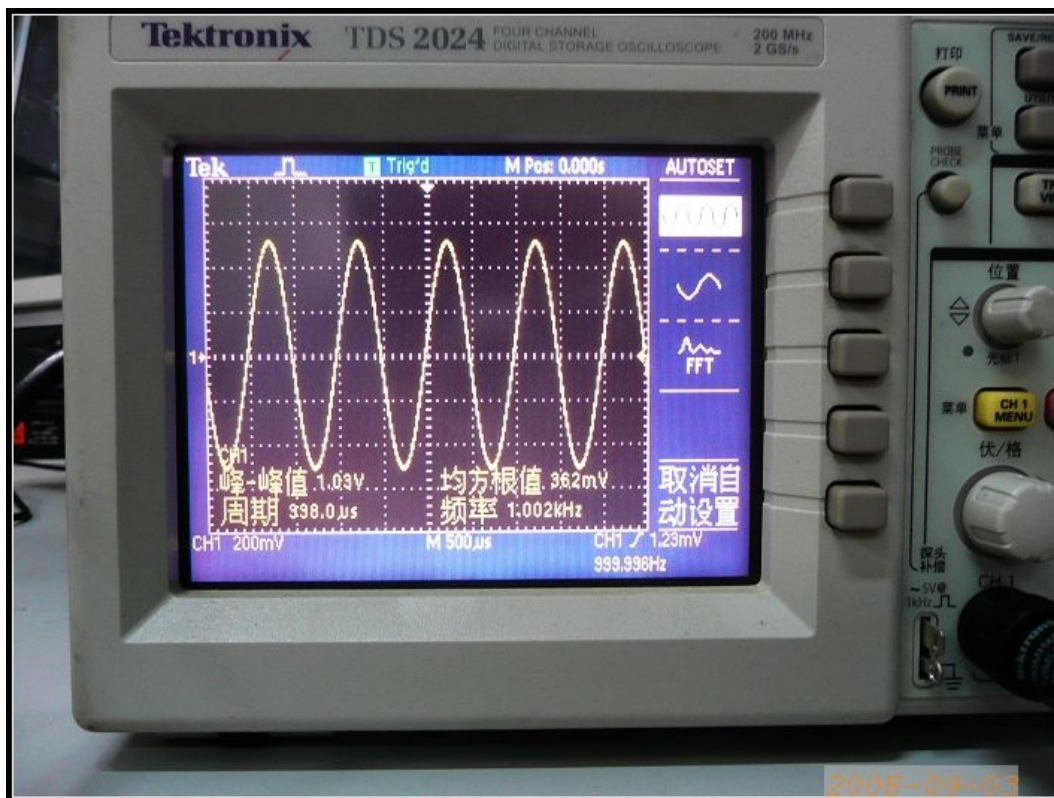
10HZ 波形输出



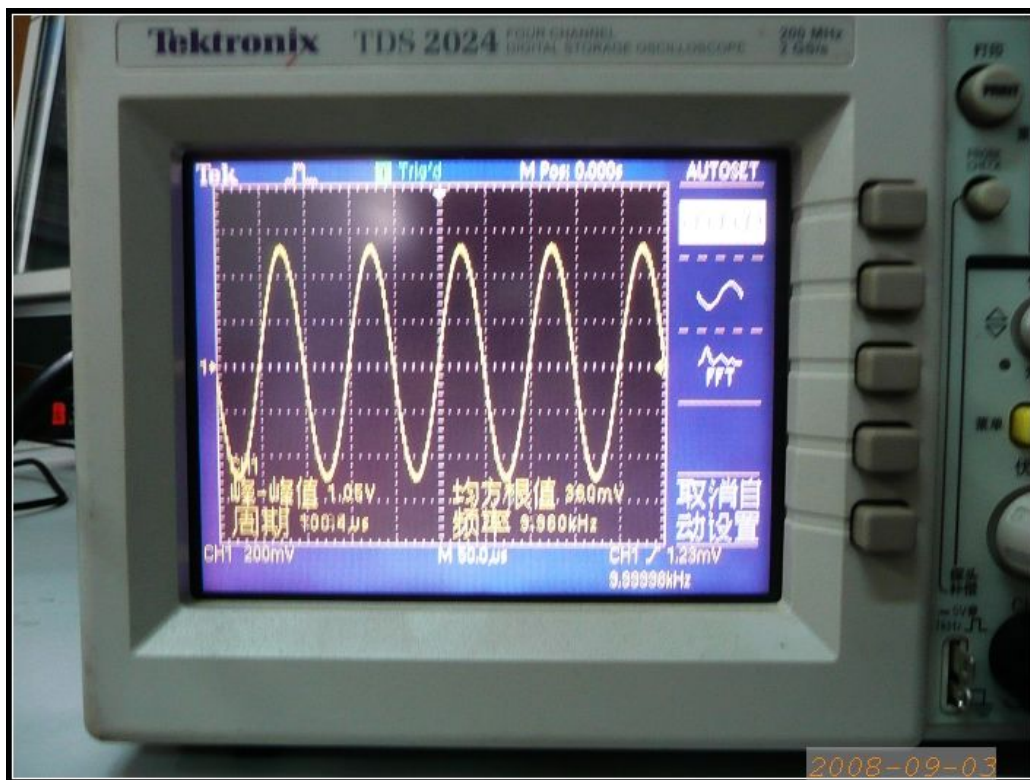
100HZ 波形输出



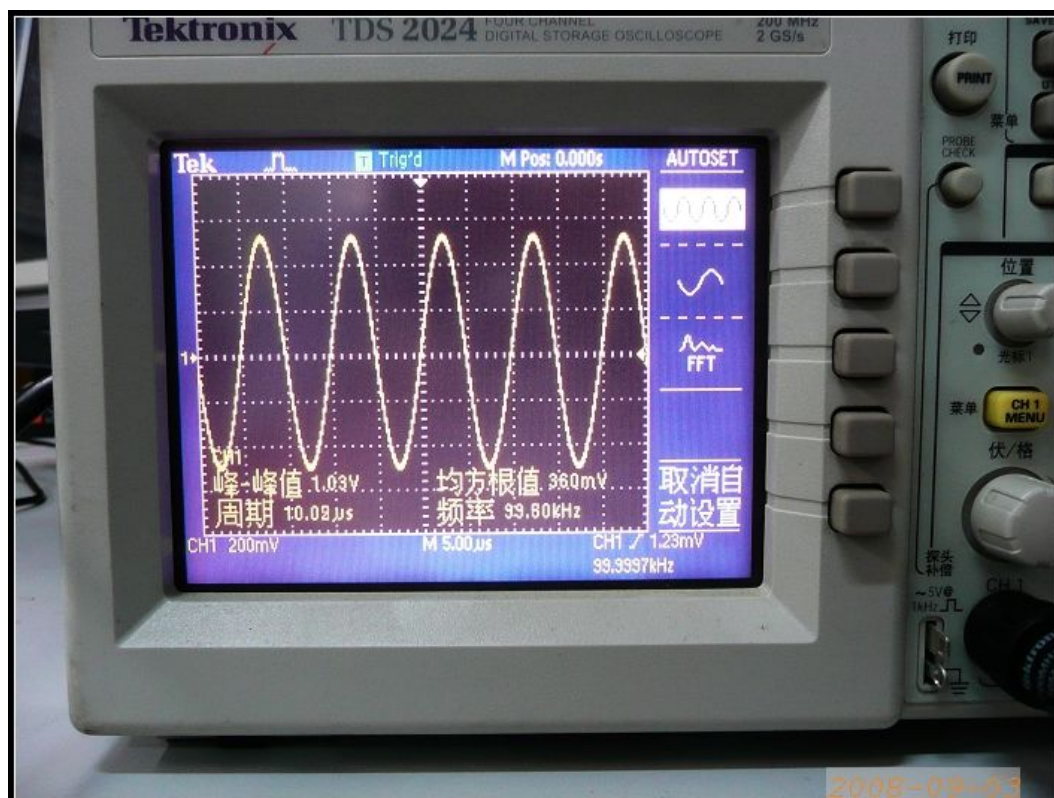
1KHZ 波形输出



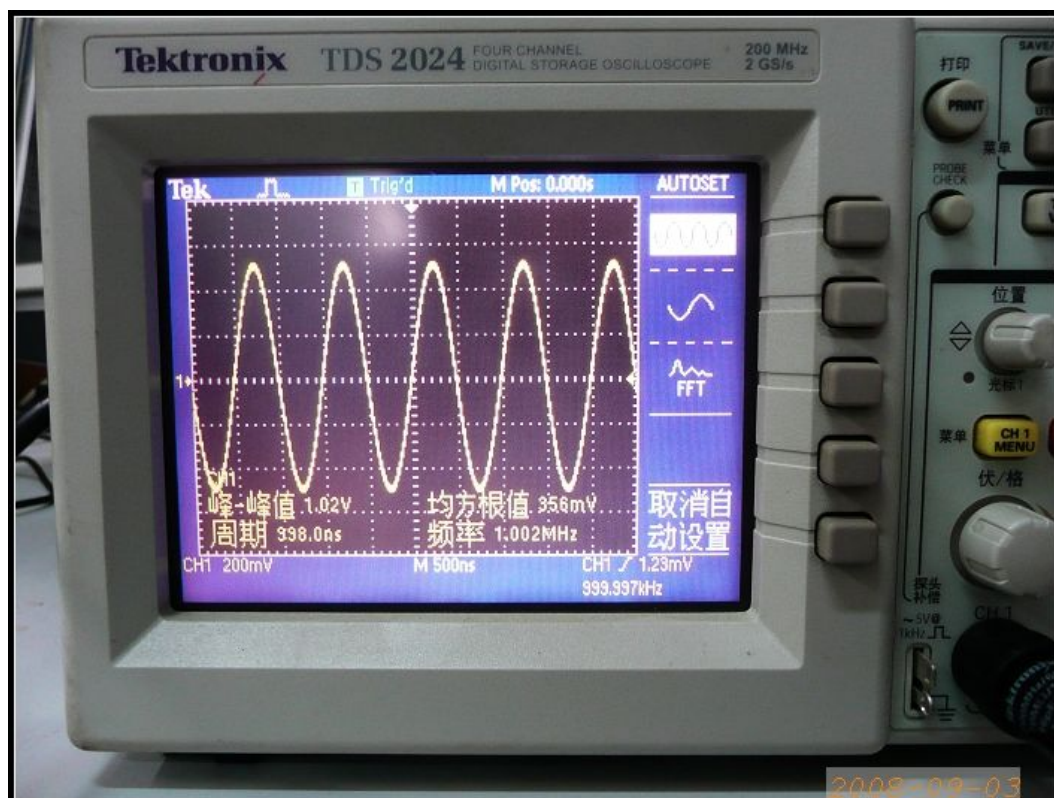
10KHZ 波形输出



100KHZ 波形输出



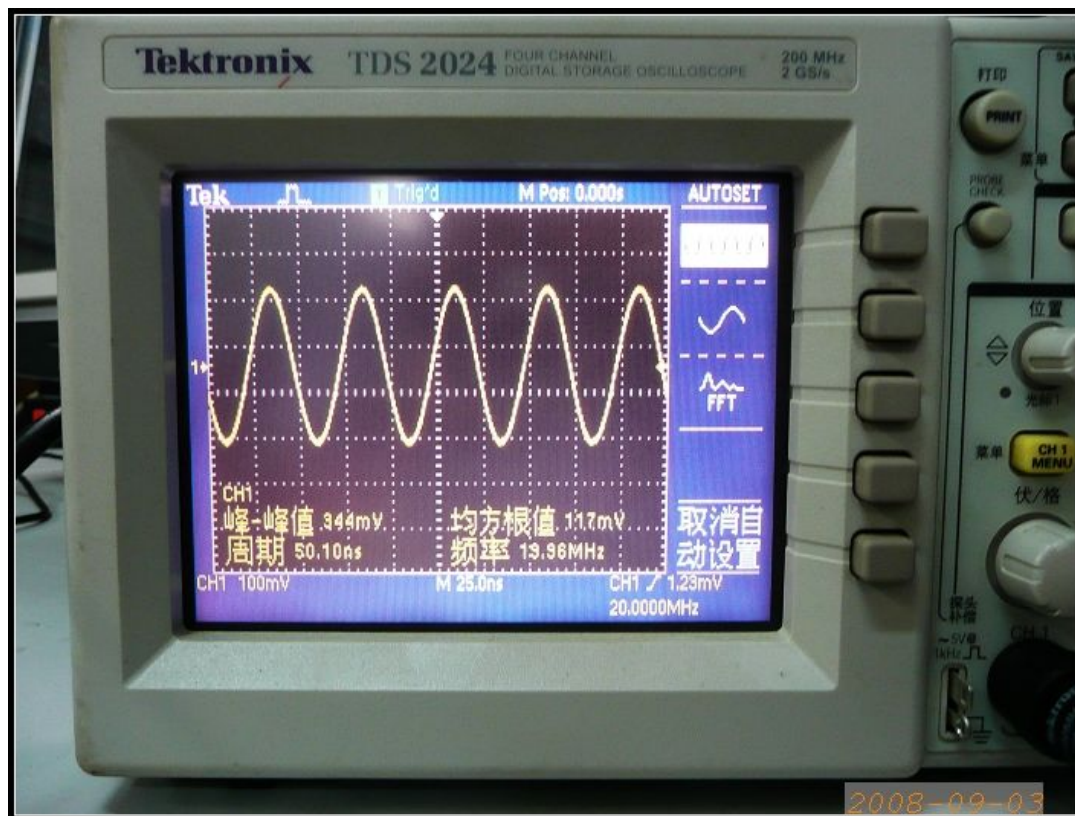
1MHZ 波形输出



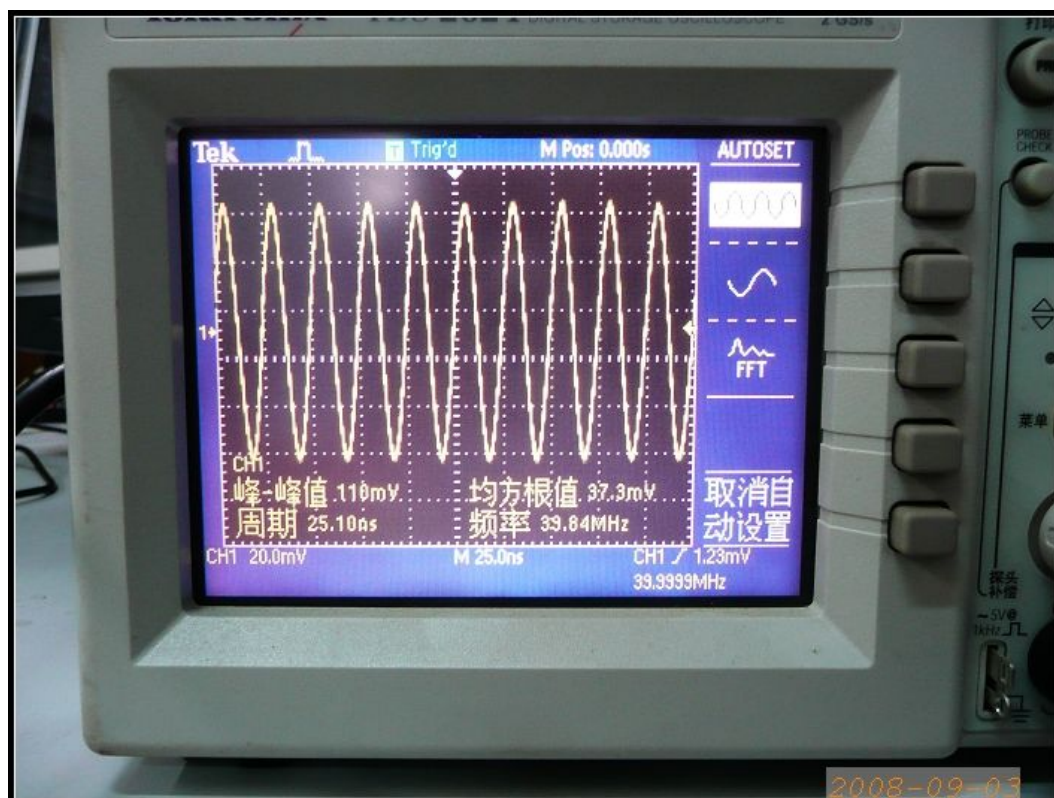
10MHZ 波形输出



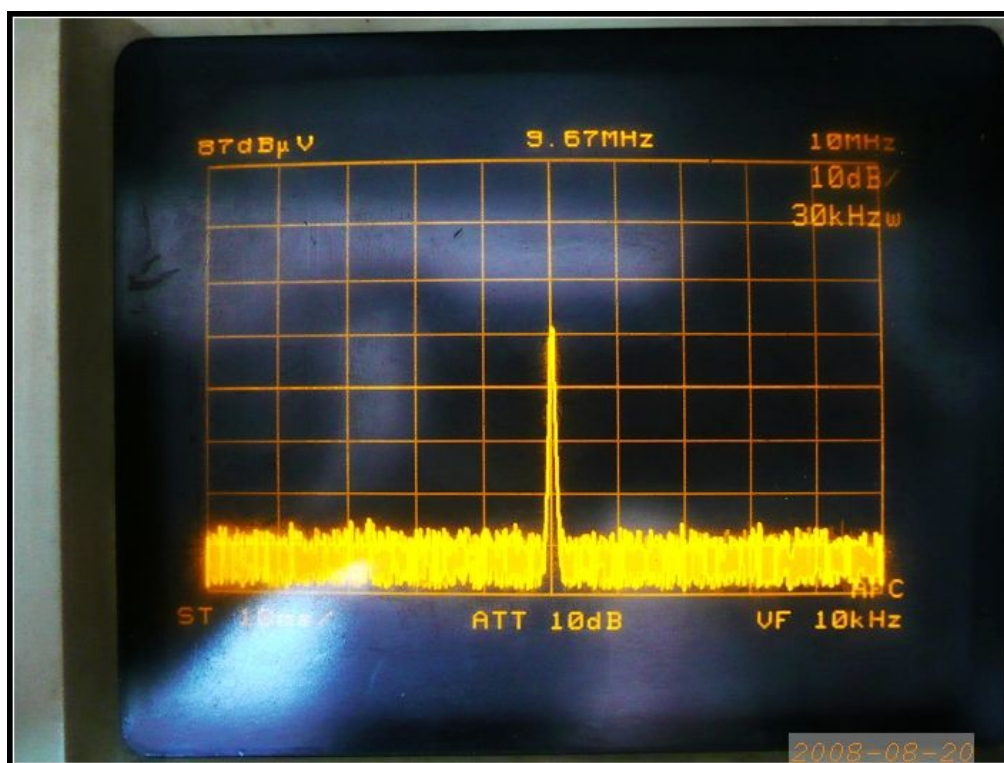
20MHZ 波形输出



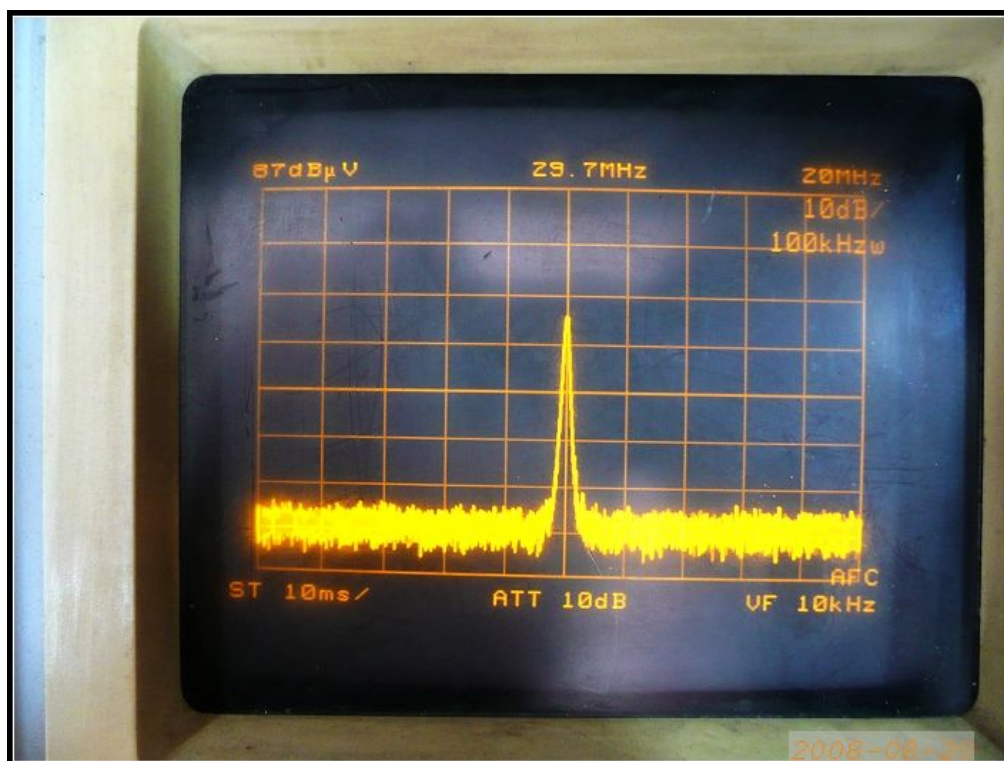
30MHZ 波形输出



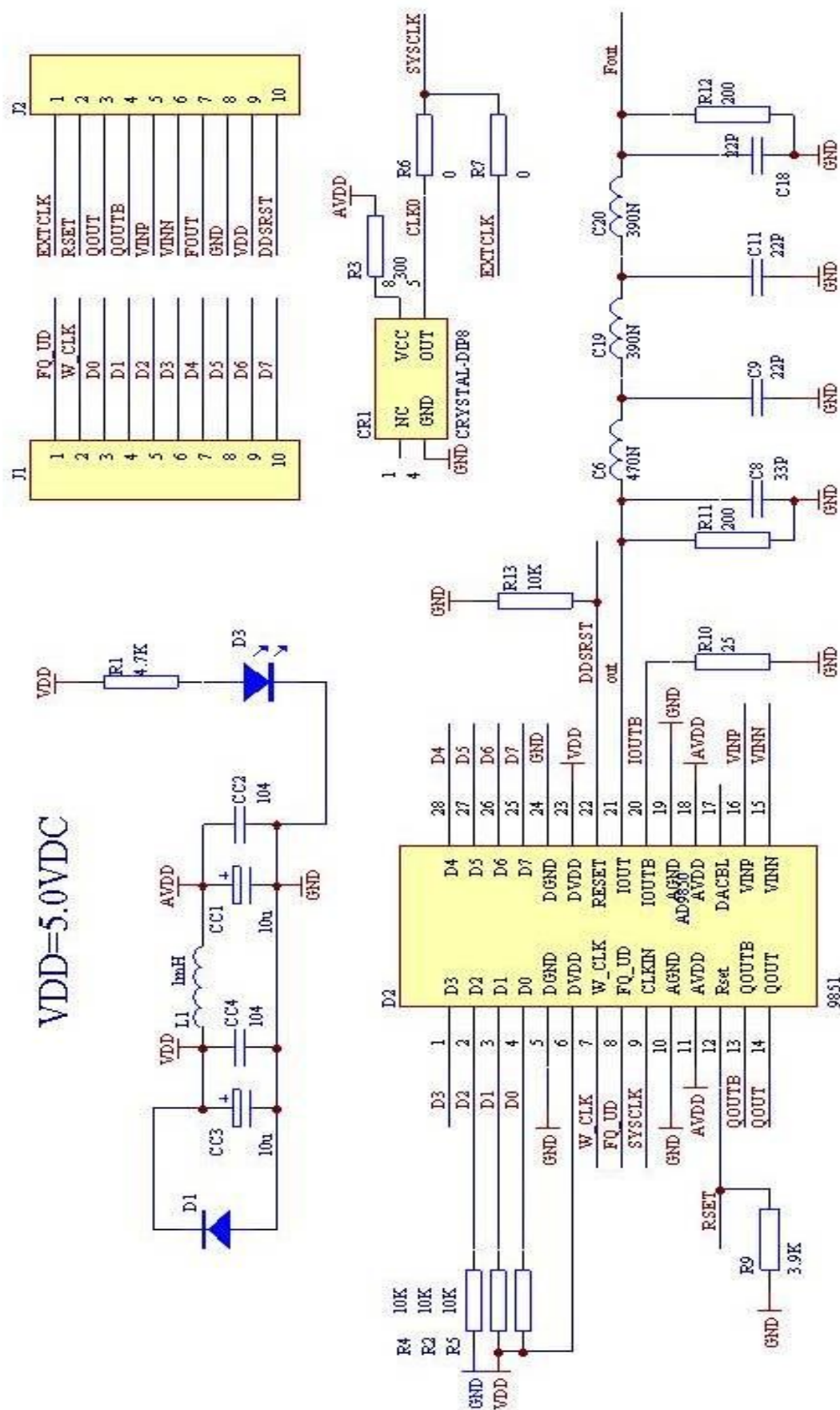
10MHZ 输出频谱特性



30MHZ 输出频谱特性

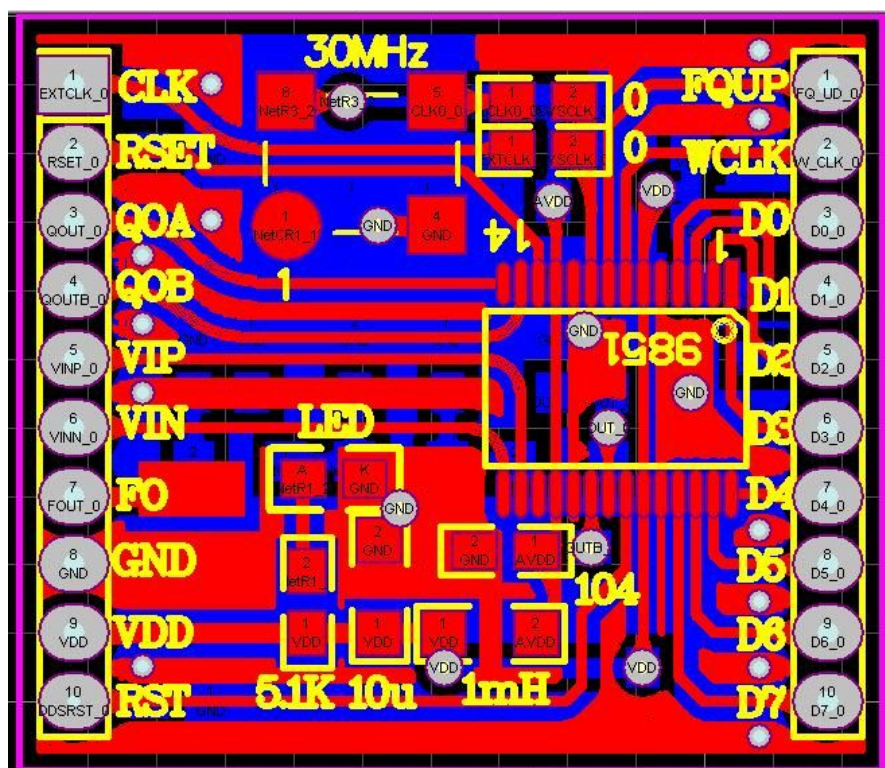


7、参考原理图：



8、参考 PCB 图

顶层



底层

