

DDS 原理及 AD9851 电路设计、测试

鼎峰电子 <http://www.dfans.net>

一. DDS (Direct Digital Synthesis) 直接频率合成技术概况

在频率合成 (FS, Frequency Synthesis) 领域中, 常用的频率合成技术有模拟锁相环、数字锁相环、小数分频锁相环 (fractional-N PLL Synthesis) 等, 直接数字合成 (Direct Digital Synthesis—DDS) 是近年来新的 FS 技术。单片集成的 DDS 产品是一种可代替锁相环的快速频率合成器件。DDS 是产生高精度、快速变换频率、输出波形失真小的优先选用技术。DDS 以稳定度高的参考时钟为参考源, 通过精密的相位累加器和数字信号处理, 通过高速 D/A 变换器产生所需的数字波形 (通常是正弦波形), 这个数字波经过一个模拟滤波器后, 得到最终的模拟信号波形。如图 2 所示, 通过高速 DAC 产生数字正弦数字波形, 通过带通滤波器后得到一个对应的模拟正弦波信号, 最后该模拟正弦波与一门限进行比较得到方波时钟信号。

DDS 系统一个显著的特点就是在数字处理器的控制下能够精确而快速地处理频率和相位。除此之外, DDS 的固有特性还包括: 相当好的频率和相位分辨率 (频率的可控范围达 μHz 级, 相位控制小于 0.09°), 能够进行快速的信号变换 (输出 DAC 的转换速率 300 百万次/秒)。这些特性使 DDS 在军事雷达和通信系统中应用日益广泛。

其实, 以前 DDS 价格昂贵、功耗大 (以前的功耗达 Watt 级)、DAC 器件转换速率不高, 应用受到限制, 因此只用于高端设备和军事上。随着数字技术和半导体工业的发展, DDS 芯片能集成包括高速 DAC 器件在内的部件, 其功耗降低到 mW 级 (AD9851 在 3.3V 时功耗为 650mW), 功能增加了, 价格便宜。因此, DDS 也获得广泛的应用: 现代电子器件、通信技术、医学成像、无线、PCS/PCN 系统、雷达、卫星通信。

二. DDS 工作原理:

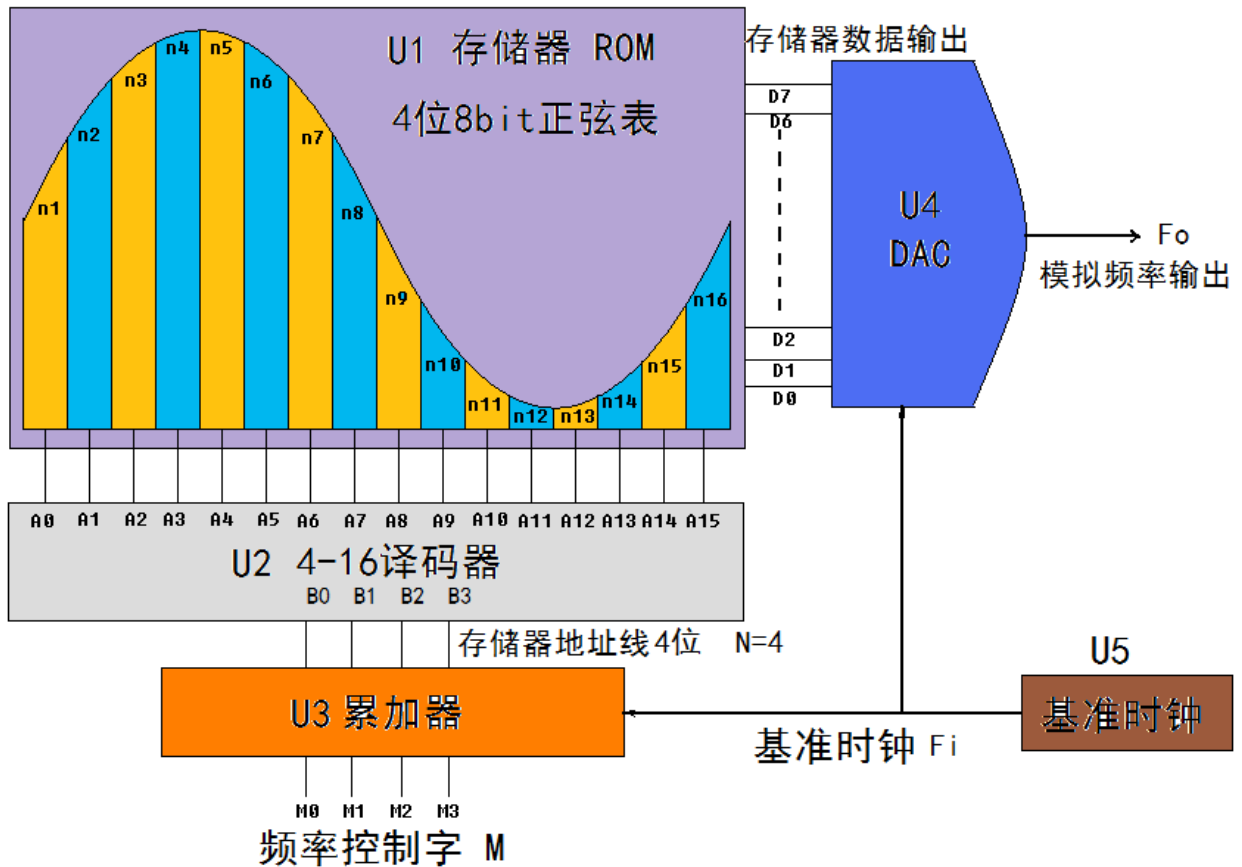


图 1

图 1 是一个 DDS 的基本内部结构。图中，U1 是一个容量 16 字节，按照一维排列的 8 位只读存储器，U1 的 16 个字节依次存放了完整的一个周期正弦波的量化数值 n_1 - n_{16} ，相当于把一个正弦波 $0-2\pi$ 的相位平分为 16 等份。我们把 ROM 中的 n_1 - n_{16} 波形数据称作正弦查找表，A0-A15 对应着正弦波的一个相位。当 A0-A15 地址选通线上某一位的电平为 0 时（同一时刻 A0-A15 中只能有一位为 0 电平），其对应 ROM 单元的存储值 n 将立即从 D0-D7 8 位总线输出。U2 为 4-16 线译码器，负责把从 B3-B0 输入的 4 位二进制码转变为 A0-A15 的 0 电平状态。

U4 是一个高速数模转换器（DAC），他负责把 D0-D7 输入的数字信号变成对应的模拟信号，在基准时钟的驱动下，每个基准时钟周期转换一次。例如最大值 n_4 对应模拟输

出 5V，中间值 n8 输出 2.5V，最小值 n12 对应输出 0V。

ROM 地址总线 B0-B3, ROM 地址选通线 A0-A15 以及 ROM 数据输出总线 D0-D7 的电平状态（逻辑关系）如下表:

ROM 地址总线 B3-B0	ROM 地址选通线 A0-A15	ROM 数据输出总线 D7-D0
0000	0111 1111 1111 1111	n1
0001	1011 1111 1111 1111	n2
0010	1101 1111 1111 1111	n3
0011	1110 1111 1111 1111	n4
0100	1111 0111 1111 1111	n5
0101	1111 1011 1111 1111	n6
0110	1111 1101 1111 1111	n7
0111	1111 1110 1111 1111	n8
1000	1111 1111 0111 1111	n9
1001	1111 1111 1011 1111	n10
1010	1111 1111 1101 1111	n11
1011	1111 1111 1110 1111	n12
1100	1111 1111 1111 0111	n13
1101	1111 1111 1111 1011	n14
1110	1111 1111 1111 1101	n15
1111	1111 1111 1111 1110	n16

U3 为相位累加器，它是 DDS 的结构中最关键的算法单元。相位累加器有一个时钟输入端，一组频率控制字输入总线 M3-M0, 一组相位地址输出总线 B3-B0。

相位累加器 U3 由 N 位加法器与 N 位累加寄存器级联构成。每来一个时钟脉冲 F0，加法器将频率控制字 M 与累加寄存器输出的累加相位数据相加，把相加后的结果送至累加寄存器的数据输入端。累加寄存器将加法器在上一个时钟脉冲作用后所产生的新相位数据反馈到加法器的输入端，以使加法器在下一个时钟脉冲的作用下继续与频率控制字相加。此过程的伪代码表述如下：

If(clock)

$B=B+M;$

这样，相位累加器在时钟作用下，不断对频率控制字进行线性相位累加。由此可以看出，相位累加器在每一个时钟脉冲输入时，把频率控制字累加一次，相位累加器输出的数据就是合成信号的相位，相位累加器的溢出频率就是 DDS 输出的信号频率。用相位累

加器输出的数据 B0-B3 产生波形存储器 (ROM) 的相位取样地址 A0-A15, 这样就可把存储在波形存储器内的波形量化值 (n, 二进制编码) 经查找表查出输出到数据总线 D0-D7, 完成相位到幅值转换。波形存储器的输出送到模数转换器 DAC, DAC 将数字量形式的波形幅值转换成所要求合成频率的模拟量形式信号。

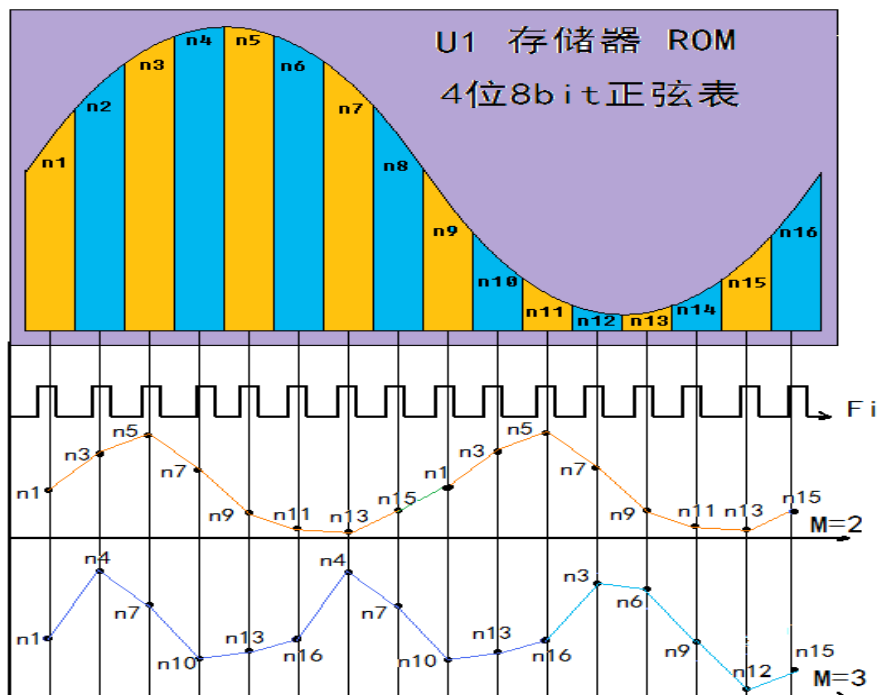
明白了 DDS 的工作过程后, 我们可以假定几个工作状态来分析其工作过程。(假定基准时钟连续产生, 频率为 16Hz, B0-B3 初始值为 0000)

M=0 时: $B=B+0$, 则 B 一直保持恒定的值, 意味着输出频率为 0。

M=1 时: $B=B+1$, B 按 1 递增, 依次将 ROM 中的 n1-n16 从 D0-D7 数据口输出。每 $2^N=16$ 个时钟, 相位累加器溢出一次, 从新开始累加。同时 DAC 输出一个完整的正弦波。此正弦波的频率为 $F_i/16=F_i/2^N=1\text{Hz}$ 。

M=2 时: $B=B+2$, B 按 2 递增, n1,n3,n5,n7……n15, 依次从 D0-D7 数据口输出。每 $2^N/2=8$ 个时钟, 相位累加器溢出一次, 从新开始累加。同时 DAC 输出两个完整的正弦波。此正弦波的频率为 $F_i \cdot M/2^N=2\text{Hz}$ 。

查表的过程如下:



三. DDS 相关计算

由上面的计算规律，我们可以总结出以下几个公式，各符号的定义为：**Fi**：基准频率；**Fo**：DDS 输出频率；**M**：频率控制字；**N**：相位累加器位数；**K**：DDS 每个输出周期的抽样点数；**Fmin**：DDS 最小输出频率（频率分辨率）；**Fmax**：DDS 最大输出频率。

$$F_o = (M/2^N) F_i \quad (1)$$

$$F_{\min} = (1/2^N) F_i \quad (2)$$

$$F_{\max} = (M_{\max}/2^N) F_i \quad (3)$$

$$K = 2^N / M \quad (4)$$

由此可以看出，当 **N** 比较大时，对于很大范围内的 **M** 值，DDS 系统都可以在一个周期内输出足够的点，保证输出波形失真很小。上图中 **N** 和 **M** 相差不大，处于同一量级，导致抽样点数极少。从合成出的波形来看，**M=3** 时已经不能很好复原出正弦波的样子。

四. DDS 频率合成的特点

（1）输出频率相对带宽较宽

输出频率带宽为 **50%Fi**（理论值）。但考虑到低通滤波器的特性和设计难度以及对输出信号杂散的抑制，实际的输出频率带宽仍能达到 **40%Fi**。

（2）频率转换时间短

DDS 是一个开环系统，无任何反馈环节，这种结构使得 DDS 的频率转换时间极短。事实上，在 DDS 的频率控制字改变之后，需经过一个时钟周期之后按照新的相位增量累加，才能实现频率的转换。因此，频率转换的时间等于频率控制字的传输时间，也就是一个时钟周期的时间。时钟频率越高，转换时间越短。DDS 的频率转换时间可达纳秒数量级，比使用其它的频率合成方法都要短数个数量级。

(3) 频率分辨率极高

若时钟 F_i 的频率不变，DDS 的频率分辨率就由相位累加器的位数 N 决定。只要增加相位累加器的位数 N 即可获得任意小的频率分辨率。目前，大多数 DDS 的分辨率在 1Hz 数量级，许多小于 1mHz 甚至更小。

(4) 相位变化连续

改变 DDS 输出频率，实际上改变的每一个时钟周期的相位增量，相位函数的曲线是连续的，只是在改变频率的瞬间其频率发生了突变，因而保持了信号相位的连续性。

(5) 输出波形的灵活性

只要在 DDS 内部加上相应控制如调频控制 FM、调相控制 PM 和调幅控制 AM，即可以方便灵活地实现调频、调相和调幅功能，产生 FSK、PSK、ASK 和 MSK 等信号。另外，只要在 DDS 的波形存储器存放不同波形数据，就可以实现各种波形输出，如三角波、锯齿波和矩形波甚至是任意的波形。当 DDS 的波形存储器分别存放正弦和余弦函数表时，既可得到正交的两路输出。

(6) 其他优点

由于 DDS 中几乎所有部件都属于数字电路，易于集成，功耗低、体积小、重量轻、可靠性高，且易于程控，使用相当灵活，因此性价比极高。

五. ADI 公司常用的 DDS 芯片性能对比

型号	最大时钟 频率 MHz	工 作 电 压 V	最 大 功 耗 mW	特 点
AD9830	50	5	300	经济，并行输入，内置 DAC
AD9831	25	3.3/5	120	低电压，经济，内置 DAC
AD9832	25	3.3/5	120	小型封装，串行输入，内置 DAC
AD9833	25	2.5-5.5	20	10 个管脚的 uSOIC 封装
AD9834	50	2.5-5.5	25	20 个管脚的 TSSOP 封装，内置比较器
AD9835	50	5	200	经济，小型封装，串行输入，内置 DAC
AD9850	125	3.3/5	480	内置 DAC 和比较器

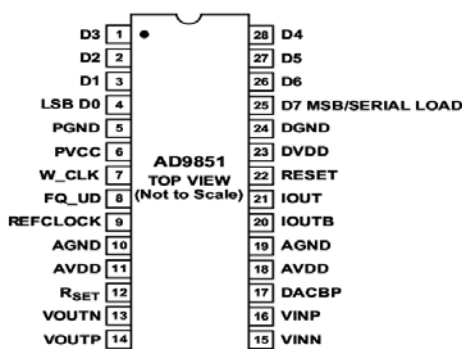
AD9851	180	3/3.3/5	650	内置 DAC,比较器, 时钟 6 倍频器
AD9852	300	3.3	1200	内置 12 位 DAC, 高速比较器, 线性调频和可编程参考时钟倍频器
AD9853	165	3.3	1150	内置可编程数字 QPSK/16-QAM 调制器
AD9854	300	3.3	1200	内置 12 位两路正交 DAC, 高速比较器和可编程时钟倍频器
AD9858	1000	3.3	2000	内置 10 位 DAC, 150MHz 相频检测器, 充电泵和 2GHz 混频器

六. AD9851 简介

1.AD9851 芯片是美国模拟器件公司采用先进 DDS 直接数字频率合成技术生产的高集成度产品。AD9851 是在 AD9850 的基础上, 做了一些改进以后生成的具有新功能的 DDS 芯片。AD9851 相对于 AD9850 的内部结构, 只是多了一个 6 倍参考时钟倍乘器, 当系统时钟为 180MHz 时, 在参考时钟输入端, 只需输入 30MHz 的参考时钟即可。如图 4 (AD9851 内部结构) 所示, AD9851 是由数据输入寄存器、频率/相位寄存器、具有 6 倍参考时钟倍乘器的 DDS 芯片、10 位的模/数转换器、内部高速比较器这几个部分组成。其中具有 6 倍参考时钟倍乘器的 DDS 芯片是由 32 位相位累加器、正弦函数功能查找表、D/A 变换器以及低通滤波器集成到一起。这个高速 DDS 芯片时钟频率可达 180MHz, 输出频率可达 70 MHz, 分辨率为 0.04Hz。

AD9851 可以产生一个频谱纯净、频率和相位都可编程控制且稳定性很好的模拟正弦波, 这个正弦波能够直接作为基准信号源, 或通过其内部高速比较器转换成标准方波输出, 作为精密可编程时钟发生器来使用。

其引脚图如下:



AD9851 的各引脚功能如下:

D0~D7: 8 位数据输入口, 可给内部寄存器装入 40 位控制数据。

PGND: 6 倍参考时钟倍乘器地。

PVCC: 6 倍参考时钟倍乘器电源。

W-CLK: 字装入信号, 上升沿有效。

FQ-UD: 频率更新控制信号, 时钟上升沿确认输入数据有效。

FREFCLOCK: 外部参考时钟输入。

CMOS/TTL 脉冲序列可直接或间接地加到 6 倍参考时钟倍乘器上。在直接方式中, 输入频率即是系统时钟; 在 6 倍参考时钟倍乘器方式, 系统时钟为倍乘器输出。

AGND: 模拟地。

AVDD: 模拟电源(+5 V)。

DGND: 数字地。

DVDD: 数字电源(+5 V)。

RSET: DAC 外部参考电阻连接端。

VOUTN: 内部比较器负向输出端。

VOUTP: 内部比较器正向输出端。

VINN: 内部比较器的负向输入端。

VINP: 内部比较器的正向输入端。

DACBP: DAC 旁路连接端。

IOUTB: “互补” DAC 输出。

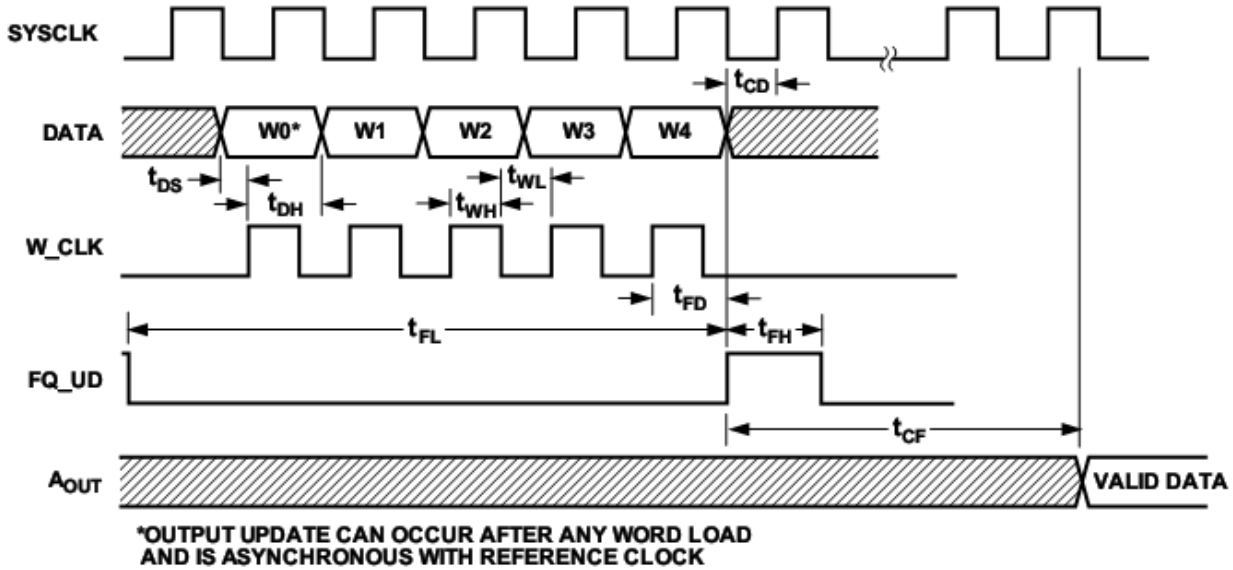
IOUT: 内部 DAC 输出端。

RESET: 复位端。低电平清除 DDS 累加器和相位延迟器为 0Hz 和 0 相位, 同时置数据输入为串行模式以及禁止 6 倍参考时钟倍乘器工作。

2.AD9851 的寄存器位置和并行写入时序如下

Word	Data[7]	Data[6]	Data[5]	Data[4]	Data[3]	Data[2]	Data[1]	Data[0]
W0	Phase-b4 (MSB)	Phase-b3	Phase-b2	Phase-b1	Phase-b0 (LSB)	Power-Down	Logic 0*	6× REFCLK Multiplier Enable
W1	Freq-b31 (MSB)	Freq-b30	Freq-b29	Freq-b28	Freq-b27	Freq-b26	Freq-b25	Freq-b24
W2	Freq-b23	Freq-b22	Freq-b21	Freq-b20	Freq-b19	Freq-b18	Freq-b17	Freq-b16
W3	Freq-b15	Freq-b14	Freq-b13	Freq-b12	Freq-b11	Freq-b10	Freq-b9	Freq-b8
W4	Freq-b7	Freq-b6	Freq-b5	Freq-b4	Freq-b3	Freq-b2	Freq-b1	Freq-b0 (LSB)

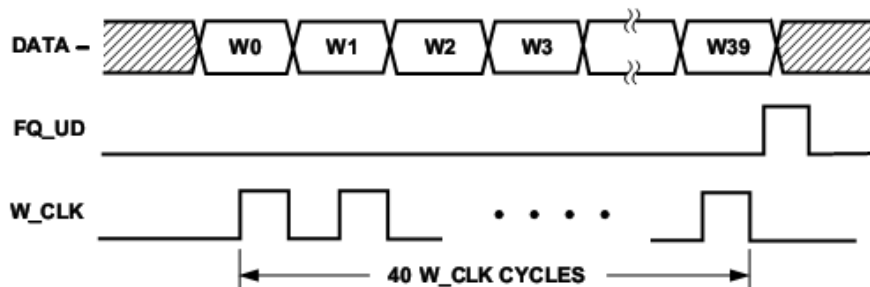
*This bit is always Logic 0 unless invoking the serial mode (see Figure 17). After serial mode is entered, this data bit must be set back to Logic 0 for proper operation.



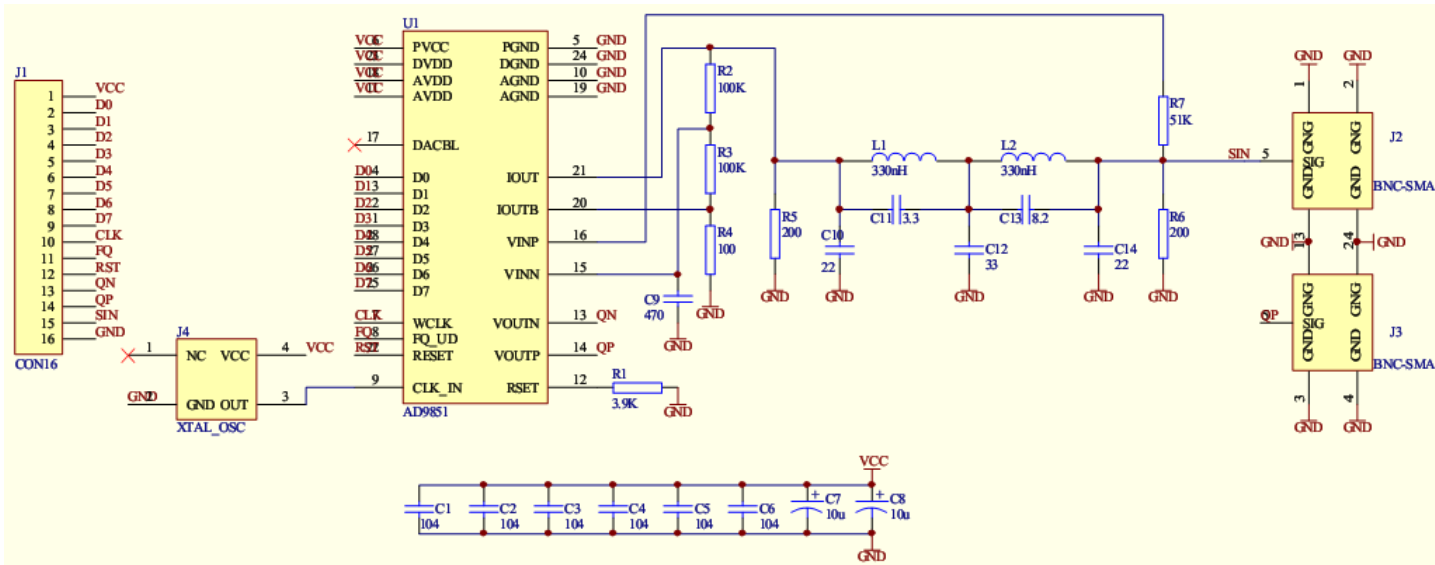
AD9851 的寄存器位置和串行写入时序如下

W0	Freq-b0 (LSB)	W13	Freq-b13	W27	Freq-b27
W1	Freq-b1	W14	Freq-b14	W28	Freq-b28
W2	Freq-b2	W15	Freq-b15	W29	Freq-b29
W3	Freq-b3	W16	Freq-b16	W30	Freq-b30
W4	Freq-b4	W17	Freq-b17	W31	Freq-b31 (MSB)
W5	Freq-b5	W18	Freq-b18	W32	6× REFCLK Multiplier Enable
W6	Freq-b6	W19	Freq-b19	W33	Logic 0*
W7	Freq-b7	W20	Freq-b20	W34	Power-Down
W8	Freq-b8	W21	Freq-b21	W35	Phase-b0 (LSB)
W9	Freq-b9	W22	Freq-b22	W36	Phase-b1
W10	Freq-b10	W23	Freq-b23	W37	Phase-b2
W11	Freq-b11	W24	Freq-b24	W38	Phase-b3
W12	Freq-b12	W25	Freq-b25	W39	Phase-b4 (MSB)
		W26	Freq-b26		

*This bit is always Logic 0.



3. AD9851 的基本运用原理如下（清晰版本见附件 PDF 文档）



4. 电路基本原理说明：

J1 为数据总线和控制线的引出端口，和外部 MCU,MPU 连接。

J4 为有源晶振，为能充分利用 AD9851 的高时钟频率特性，晶振选用 30.0000MHz。

C1-C8 为电源退耦电容，滤除 AD9851 各部分（数字，模拟，PLL）电源的高频寄生干扰。

R1 为 AD9851 内部 DAC 的参考电阻

R2,R3 为“跷跷板分压器”两个电阻的节点处电压时钟为 IOUT 和 IOUTB 两个倒相的正弦信号电压的平均值，也就是节点电压高度始终处于正弦波的中点处。

C9 为滤波电容，滤除 R2,R3 分压得到的中点电压，送往 AD9851 内部比较器反相输入端作为比较器参考电压。IOUT 信号经过 R7 后直接送进比较器同相输入端。这样，比较器在 DDS 输出任何频率，任何幅度的时候，都能保证输出的方波占空比接近 1：2。

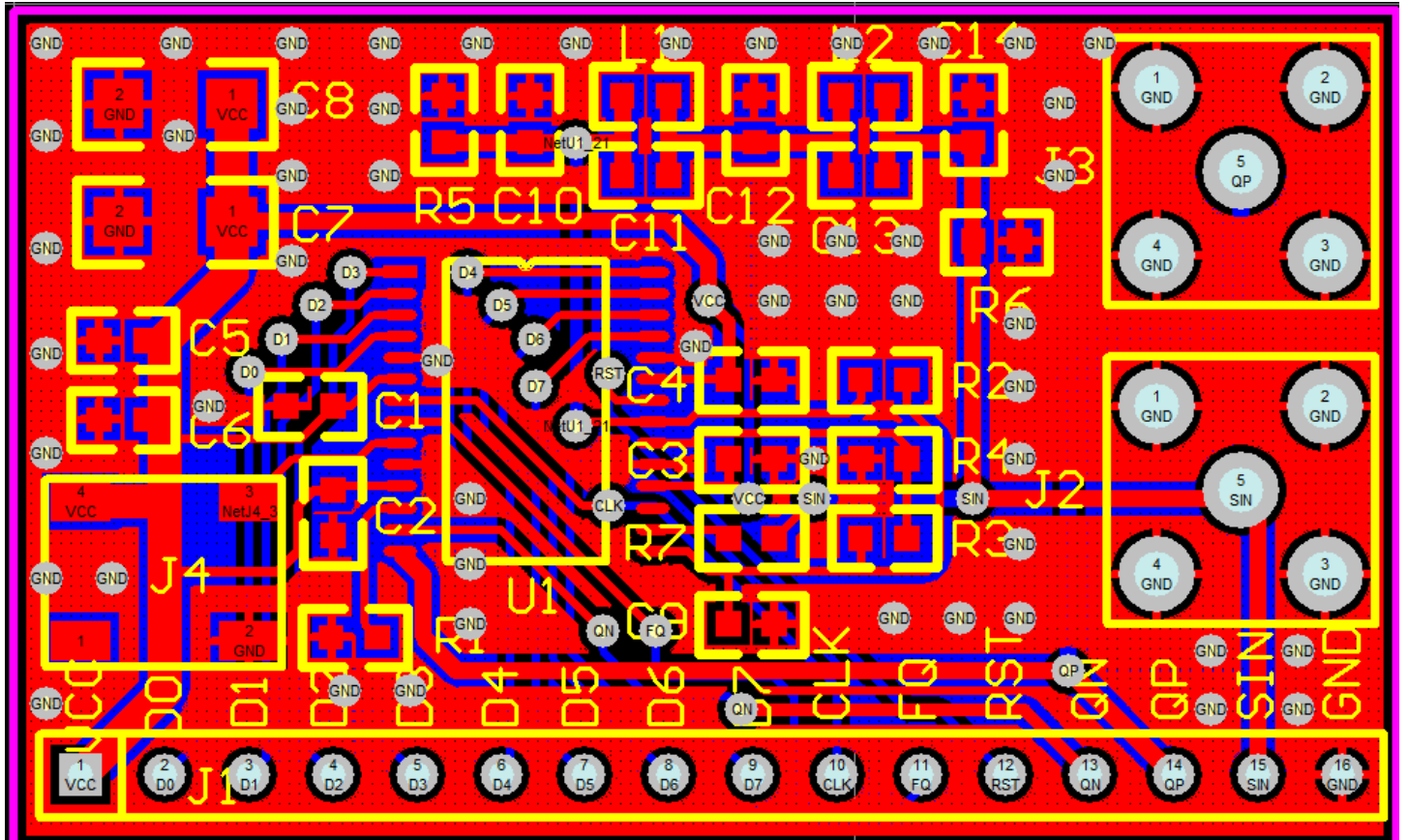
L1,L2 与周边的电容构成 2 阶椭圆低通滤波器，滤除 $1/2F_i$ 以上的频率成分。使输出频谱更干净。

R4,R5,R6 为 DDS 正弦输出的负载电阻。

J2,J3 分别为正弦信号输出 SMA 插座和方波信号 SMA 输出插座。

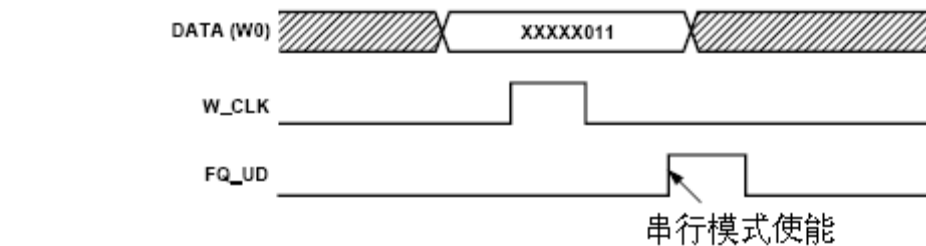
5.AD9851PCB 设计

在设计此类高频电路的时候，要注意良好接地，并避免走线角度过于尖锐，一般采用大面积覆铜来解决接地，散热和屏蔽的问题。



6.AD9851PCB 单片机驱动设计

AD9851 工作前必须对其进行初始化，AD9851 上电后默认为并行通信方式，如需采用串行通信方式，上电后应将 AD9851 的 D7~D0 置为二进制数 XXXXX011，紧接着依次往 W_CLK 和 FQ_UD 端输入宽度约 $1\mu\text{S}$ 的高电平脉冲就可以使 AD9851 工作在串行模式下。



*****AVR 单片机驱动*****

//AVR并行驱动，数据输出口用PA口，控制线放在PC口

//端口定义

```
#define   clk           0
#define   fq_ud1        2
#define   rst           2
unsigned char phase_word=0x00;    //相位控制字，使用低 5 位
unsigned char power_down=0x00;    //低功耗模式
unsigned char mult=0x01;          //6 倍频是否打开
unsigned char ww[5]={0x09,0x0E,0x38,0xE3,0x8E};
unsigned long freq,set_value;
//往 AD9851 写数据
void write_dds(unsigned long dds)
```

```
{
    unsigned char i;
    ww[0]=0x01;                //开启 6 倍频功能
    ww[0]=(phase_word<<3)|| power_down<<2 || mult;
    ww[1]=dds>>24;             //频率字最高字节
    ww[2]=(dds>>16)&0xFF;
    ww[3]=(dds>>8)&0xFF;
    ww[4]=dds&0xFF;           //频率字最低字节
    PORTC&=~(1<<fq_ud);      //FQ_UD 置低
    PORTC&=~(1<<clk);        //CLK 置低
    for(i=0;i<5;i++)          //送入 5 个字节的控制字
    {
        PORTA=ww[i];
        delay_us(2);
        PORTC|=(1<<clk);
        delay_us(2);
        PORTC&=~(1<<clk);
        delay_us(2);
    }
    PORTC|=(1<<fq_ud);        //使能信号
    delay_us(3);
    PORTC&=~(1<<fq_ud);
}
```

//写入频率

```
void write_freq(unsigned long freq)
{
    unsigned long dds;
    dds=23.860929422*freq;    //频率字计算,可进行微调
    write_dds(dds);
}
```

void init_dds(void)

```
{
    PORTC|=(1<<rst);          //高电平复位
    delay_us(10000);          //延时 10mS
    PORTC&=~(1<<rst);
    delay_us(10000);          //延时 10mS
}
```

//51并行驱动，数据输出口用PA口，控制线放在PC口

//端口定义

```
sbit clk      =P2^0;
sbit fq_ud    =P2^1;
sbit rst      =P2^2;
```

```
unsigned char phase_word=0x00;    //相位控制字，使用低 5 位
unsigned char power_down=0x00;    //低功耗模式
unsigned char mult=0x01;          //6 倍频是否打开
unsigned char ww[5]={0x09,0x0E,0x38,0xE3,0x8E};
```

```
unsigned long freq,set_value;
//往 AD9851 写数据
void write_dds(unsigned long dds)
{
```

```
    unsigned char i;
                                //控制字
    ww[0]=(phase_word<<3)|| power_down<<2 || mult;
    ww[1]=dds>>24;             //频率字最高字节
    ww[2]=(dds>>16)&0xFF;
    ww[3]=(dds>>8)&0xFF;
    ww[4]=dds&0xFF;           //频率字最低字节
    fq_ud=0;                   //FQ_UD 置低
    clk=0;                     //CLK 置低
    for(i=0;i<5;i++)          //送入 5 个字节的控制字
    {
        P0=ww[i];
        delay_us(2);
        clk=1;
        delay_us(2);
        clk=0;
        delay_us(2);
    }
    fq_ud=1;                   //使能信号
    delay_us(3);
    fq_ud=0;
}
```

//写入频率

```
void write_freq(unsigned long freq)
{
    unsigned long dds;
    dds=23.860929422*freq;    //频率字计算,可进行微调
    write_dds(dds);
}
```

void init_dds(void)

```
{
    rst=1;                     //高电平复位
    delay_us(10000);          //延时 10mS
    rst=0;
    delay_us(10000);          //延时 10mS
}
```

//AVR串行驱动，数据输出口用PA口，控制线放在PC口

//端口定义

```
#define   clk           0
#define   fq_ud1        2
#define   rst           2
#define   dat           3
unsigned char phase_word=0x00;    //相位控制字，使用低 5 位
unsigned char power_down=0x00;    //低功耗模式
unsigned char mult=0x01;          //6 倍频是否打开
unsigned char ww[5]={0x09,0x0E,0x38,0xE3,0x8E};
unsigned long freq,set_value;
```

//AD9851 初始化，在此之前请确认 D0,D1 和 D2 的电平状态

void init_dds(void)

```
{
    PORTC|=(1<<rst);          //高电平复位
    delay_us(10000);          //延时 10mS
    PORTC&=~(1<<rst);
    delay_us(10000);          //延时 10mS
    PORTC&=~(1<<clk);
    PORTC&=~(1<<fq_ud);
    delay_us(2);
}
```

```

PORTC|=(1<<clk);      //送入 W0
delay_us(2);
PORTC&=~(1<<clk);
delay_us(2);
PORTC|=(1<<fq_ud); //使能
delay_us(2);
PORTC&=~(1<<fq_ud);
delay_us(2);
}

//往 AD9851 写数据
void write_dds(unsigned long dds)
{
    unsigned char i,j,temp;

    ww[0]=dds&0xFF;          //频率字最低字节
    ww[1]=(dds>>8)&0xFF;
    ww[2]=(dds>>16)&0xFF;
    ww[3]=dds>>24;          //频率字最高字节
    ww[4]=(phase_word<<3)|| power_down<<2 || mult; //控制字

    PORTC&=~(1<<fq_ud);      //FQ_UD 置低
    PORTC&=~(1<<clk);          //CLK 置低
    for(i=0;i<5;i++)          //送入 5 个字节的控制字
    {
        temp=ww[i];
        for(j=0;j<8;j++)
        {
            PORTC&=~(1<<clk);
            if(temp&0x01)PORTC|=(1<<dat);
            else PORTC&=~(1<<dat);
            delay_us(2);
            PORTC|=(1<<clk);
            delay_us(2);
        }
    }
    PORTC&=~(1<<clk);
    PORTC|=(1<<fq_ud);        //使能信号
    delay_us(3);
    PORTC&=~(1<<fq_ud);
}

//写入频率
void write_freq(unsigned long freq)
{
    unsigned long dds;
    dds=23.860929422*freq;    //频率字计算,可进行微调
    write_dds(dds);
}

*****51 单片机驱动*****

//51 串行驱动
sbit clk      =P2^0;
sbit fq_ud    =P2^1;
sbit rst      =P2^2;
sbit dat      =P2^3;

unsigned char phase_word=0x00;    //相位控制字, 使用低 5 位
unsigned char power_down=0x00;    //低功耗模式

```

```

unsigned char mult=0x01;          //6 倍频是否打开
unsigned char ww[5]={0x09,0x0E,0x38,0xE3,0x8E};
unsigned long freq,set_value;

//AD9851 初始化, 在此之前请确认 D0,D1 和 D2 的电平状态
void init_dds(void)
{
    rst=1;          //高电平复位
    delay_us(10000); //延时 10mS
    rst=0;
    delay_us(10000); //延时 10mS
    clk=0;
    fq_ud=0;
    delay_us(2);
    clk=1;          //送入 W0
    delay_us(2);
    clk=0;
    delay_us(2);
    fq_ud=1;        //使能
    delay_us(2);
    fq_ud=0;
    delay_us(2);
}

//往 AD9851 写数据
void write_dds(unsigned long dds)
{
    unsigned char i,j,temp;

    ww[0]=dds&0xFF;          //频率字最低字节
    ww[1]=(dds>>8)&0xFF;
    ww[2]=(dds>>16)&0xFF;
    ww[3]=dds>>24;          //频率字最高字节
    ww[4]=(phase_word<<3)|| power_down<<2 || mult; //控制字

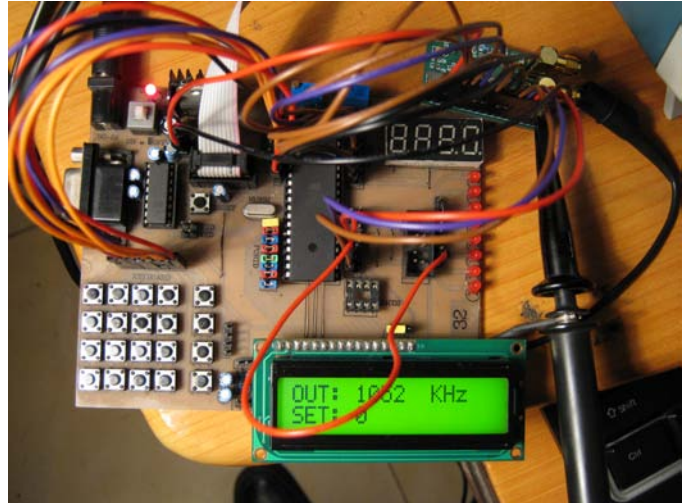
    fq_ud=0;          //FQ_UD 置低
    clk=0;            //CLK 置低
    for(i=0;i<5;i++)  //送入 5 个字节的控制字
    {
        temp=ww[i];
        for(j=0;j<8;j++)
        {
            clk=0;
            if(temp&0x01)dat=1;
            else dat=0;
            delay_us(2);
            clk=1;
            delay_us(2);
        }
    }
    clk=0;
    fq_ud=1;          //使能信号
    delay_us(3);
    fq_ud=0;
}

//写入频率
void write_freq(unsigned long freq)
{
    unsigned long dds;
    dds=23.860929422*freq;    //频率字计算,可进行微调
    write_dds(dds);
}

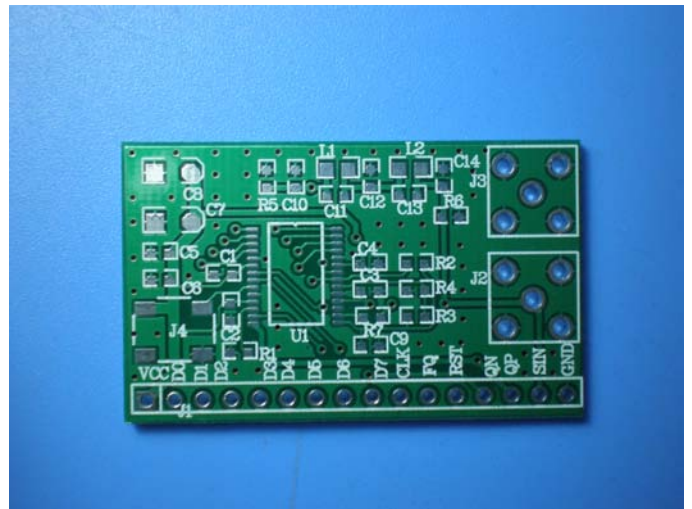
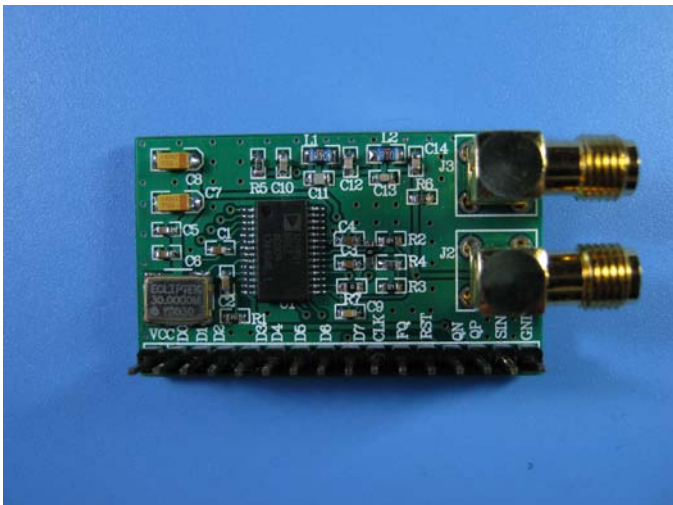
```

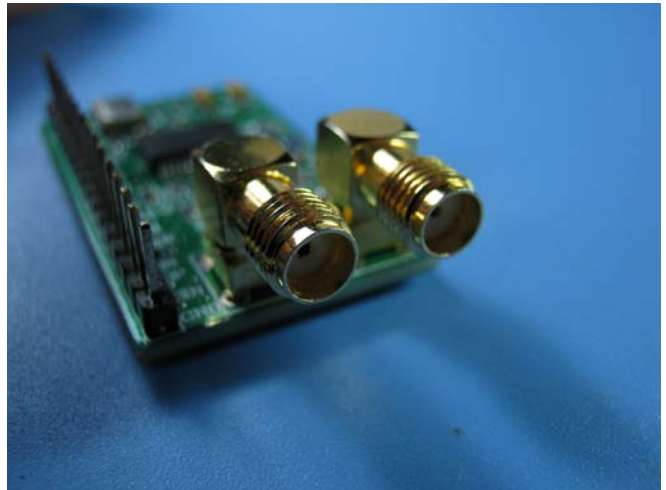

七. 实际测试

测试仪器 泰克 TDS2102B AVR 最小系统板



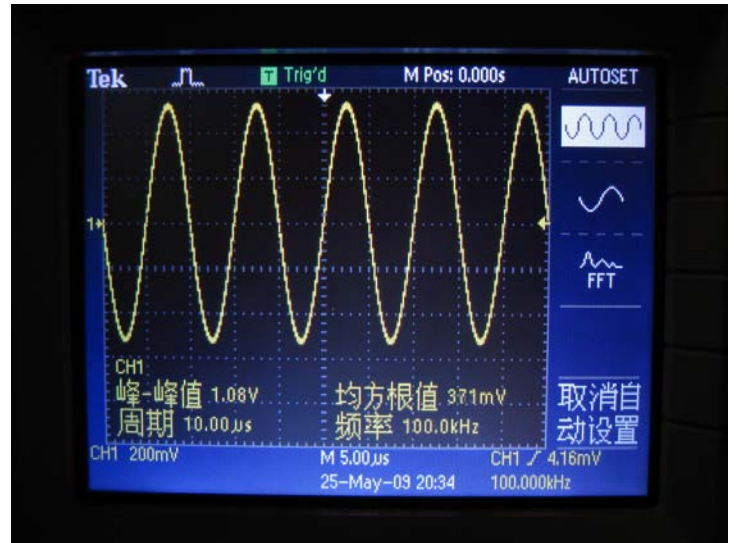
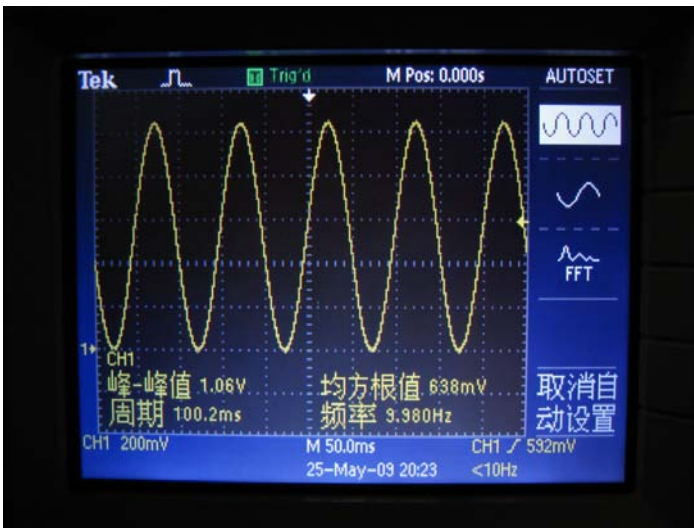
这是制作出来的 PCB

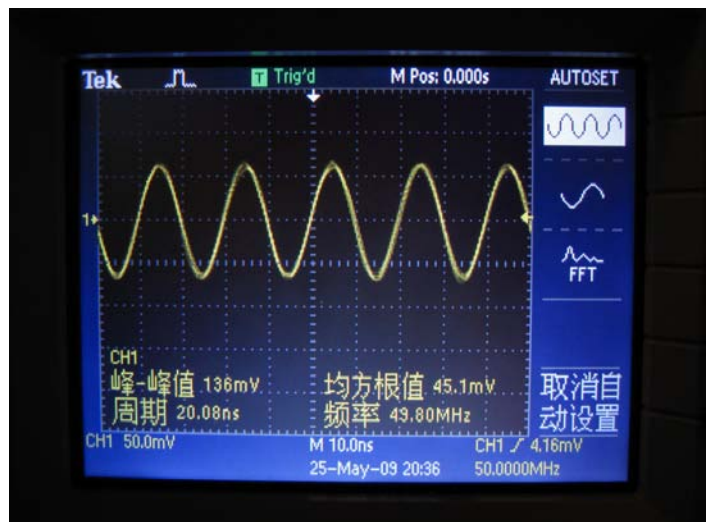
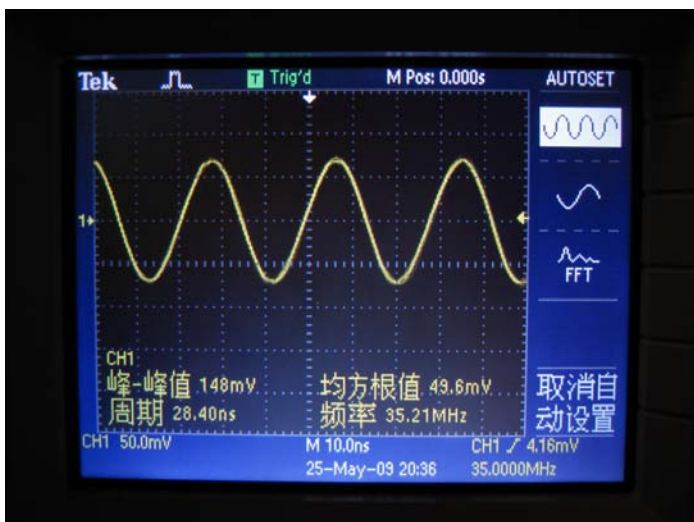
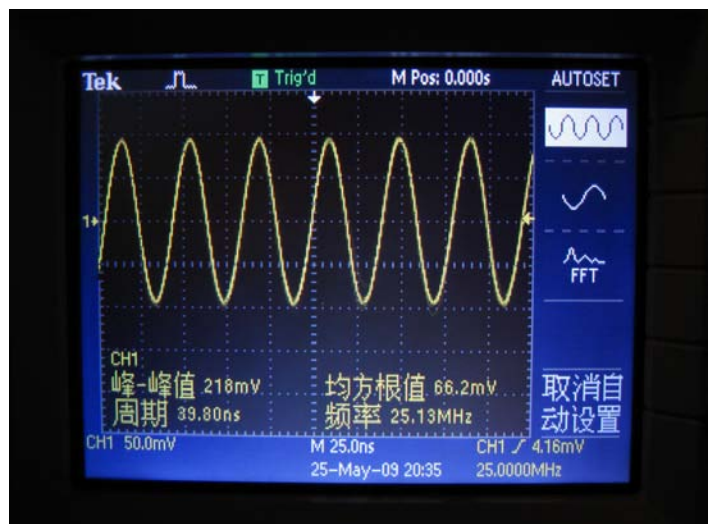
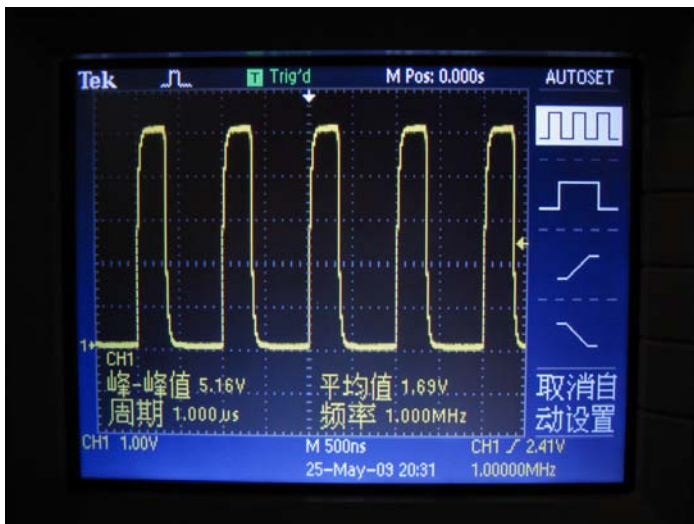
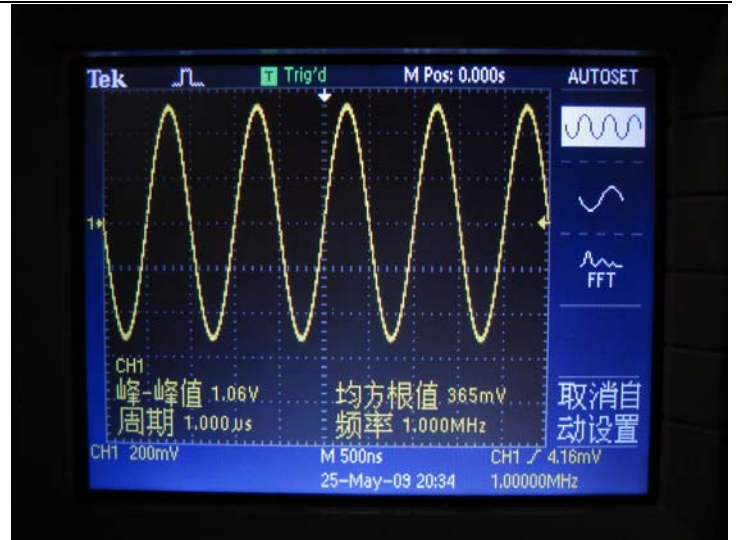
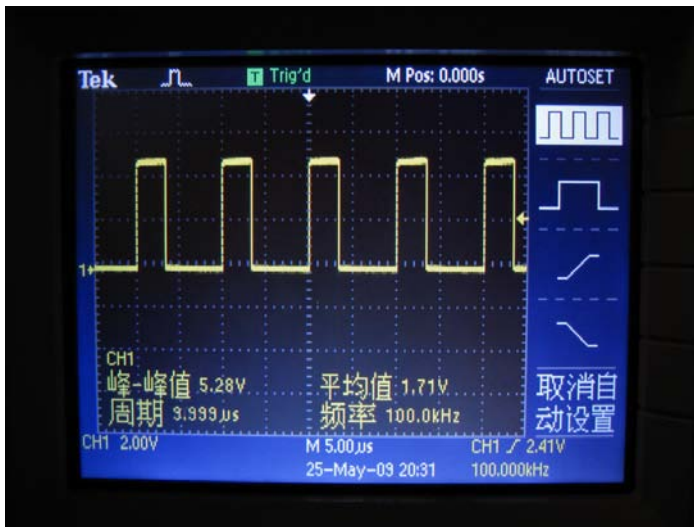


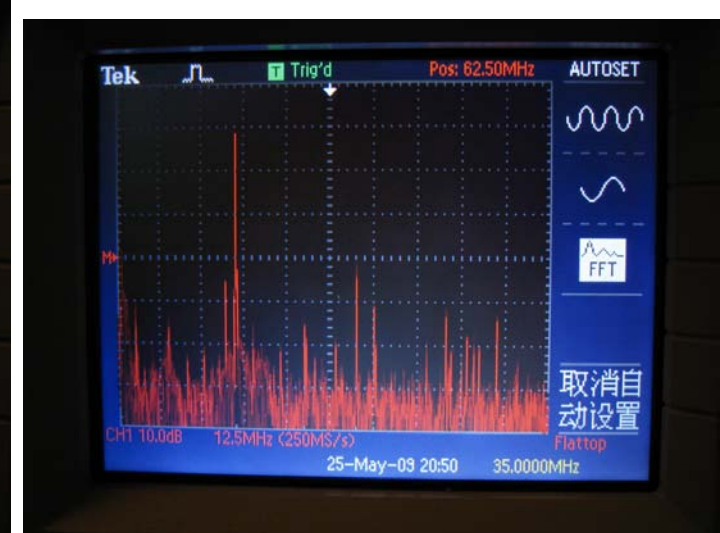
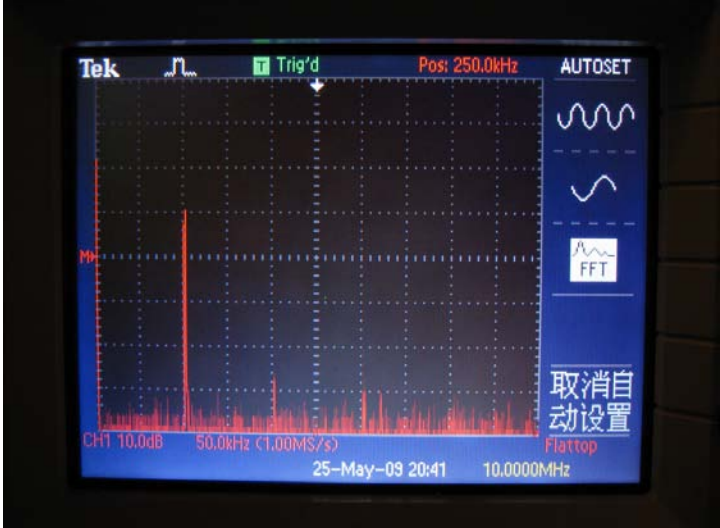
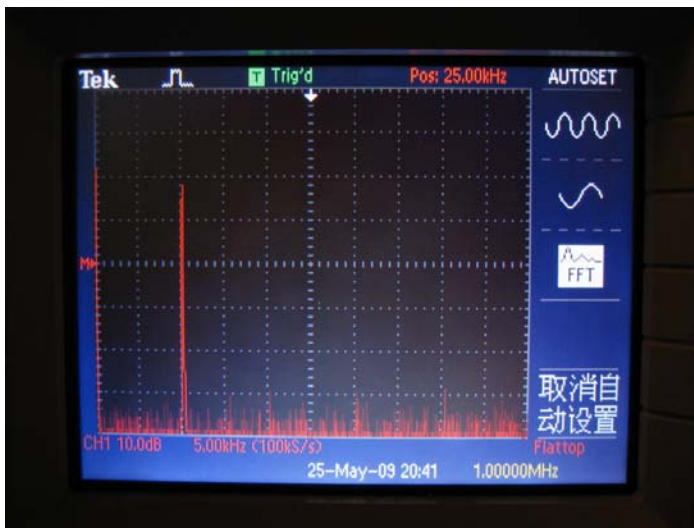
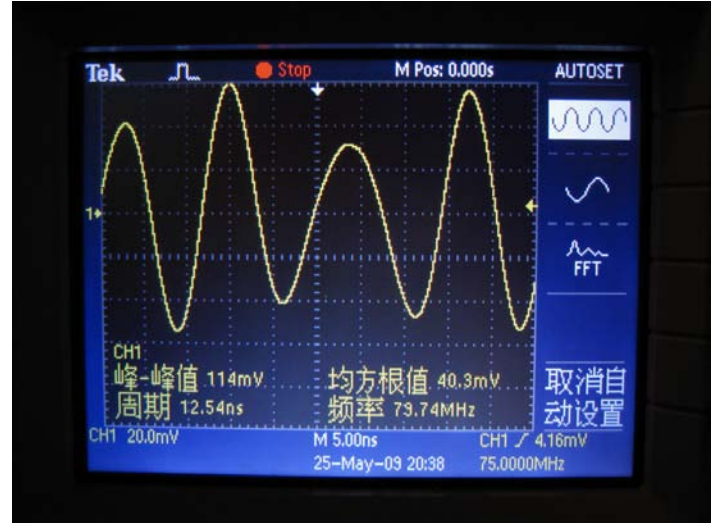
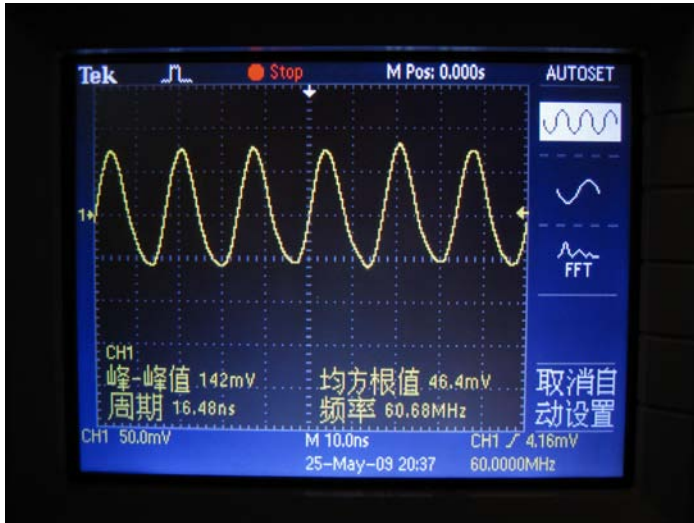


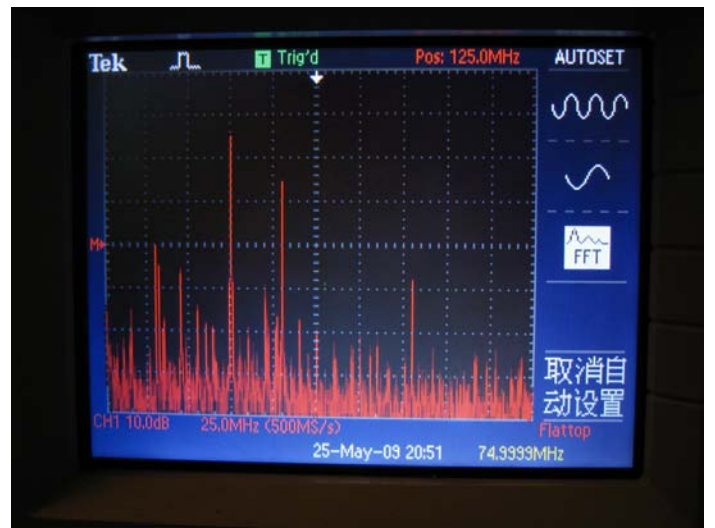
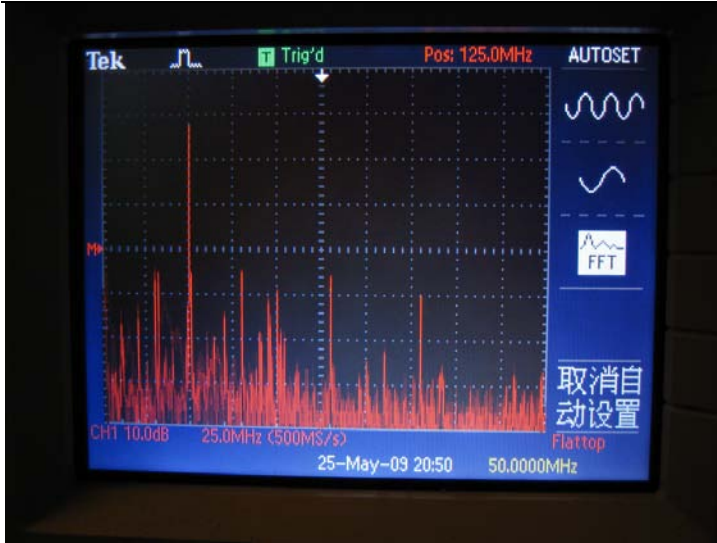
下面是 AD9851 模块设置输出不同频率的，波形和频率，频谱实测图：

可以看出，DDS 的输出频率十分精确和稳定。在 60MHz 以下，波形比较完美，频谱比较干净。越接近 $1/2F_i$ ，波形失真越严重，频谱的各次谐波逐渐增多。所以在实际使用的时候，尽量控制在 30% F_i 以内使用。









七. 经验之谈

1. 在 3.3V 供电时，内部时钟 6 倍频器工作不稳定
2. 串行模式的时候，注意数据口的状态设置
3. 可以通过修改程序中计算频率字算法 $dds=23.860929422*freq$; 中的 23.860929422，来微调输出频率。
4. AD9851 内部比较器的截止频率在 30MHz 左右，在 10MHz 以上，输出波形发生畸变。如果要用到更高频的方波，务必使用外部高速比较器。
5. DDS 正弦输出的幅度随频率增高而下降。低频端约 1Vpp, 高频端约 200mVpp。实际应用中应外加合适的宽带放大器。

欢迎大家光临 www.dfans.net 鼎峰电子论坛