

2022 年江苏省大学生电子设计竞赛 (TI 杯)

题目： 信号调制度测量装置

题目编号： F 题

参赛队编号： NJ149

参赛队学校： 南京大学

参赛队学生： 纪宇豪 靳希睿 李水仙

二〇二二年七月

摘 要

本设计为一种信号调制度测量装置，可以实现对输入信号的调制方式进行识别并显示，输入调幅、调频信号解调，以及调制度测量等功能。系统通过 DDS (Direct Digital Synthesizer, 即直接数字频率合成) 技术进行扫频，并结合乘法器混频，调幅调频检波以及单片机 ADC 采样并进行 FFT 处理等技术，可以实现稳定、无失真、无干扰无噪声的检波以及准确的输入信号判断，同时测量出调频信号载波频率以及已解调信号的频率、幅度等信息。最终利用对鉴频曲线，调幅度-已检波信号幅度曲线，鉴频器频率响应曲线的拟合，对单片级 ADC 采集到的数据进行处理运算，最终实现准确的 AM、FM 波调制度的准确测量。系统稳定，判断准确，检波稳定，干扰噪声小，测量精准，且全部测量过程不需要人工干预，均有单片机控制完成，理想地达到了题目要求并在此基础上有所提高。

关键词：调制信号解调；调制度测量；高频信号测量

1. 方案论证

1.1 技术方案分析比较

(1) 输入信号类型判断

方案 1：通过扫频确定载波，将所有信号与载波混频并通过 100k 低通滤波，将输入信号搬移去除载波搬移到低频段，并通过 ADC 采样做 FFT 绘制频谱，若只有直流为单频信号，若除直流外只有一条谱线为调幅波，若除直流外有多条谱线为调频波。此方法再判断的同时不能实现检波，且需要增加一个混频器增加一次变频操作。

方案 2：将输入吸纳后直接经过 ADL5511 模块包络检波，对检波得到信号做 FFT，若有相应谱线则为调幅波；若未得到再进行扫频，将载波搬移到 10.7M 并进行检波，对检波得到信号做 FFT，若有相应谱线则为调频波。若两次均无相应谱线则为单频信号。

综合以上两种方案，选择方案 2。

(2) 调频波解调

方案 1：将输入信号下变频到 10.7MHz，采用 NE564 锁相环对调频波进行解调，但锁相环每次解调需要调节电位器进行载波对准锁相环锁定，检波得到的波形存在较大抖动，且存在一定的频响。

方案 2：将输入信号上变频到 56MHz，采用 AC21HHK835 射频解调模块，此模块在扫频未扫到载波时，输出波形峰峰值较大，容易烧坏 ADC。

方案 3：将输入信号下变频到 10.7MHz，采用 10.7M 鉴频器电路进行调频波解调，此方法得到的波形较为稳定，噪声小且无需调节电位器进行锁定，对 3kHz 的调制信号也能输出较大

且稳定的波形。

综合以上三种方案和实地试验，选择方案 3。

(3) 混频正弦信号的产生

方案 1: 锁相环 PLL 产生正弦波，但 PLL 锁定需要时间，不利于扫频，且存在较大杂散。

方案 2: DDS 数字频率合成技术，精度高，相位噪声小，频率切换快，可实现快速扫频。

方案 3: 利用有源晶振产生方波并低通滤波得到正弦波，此时频率固定，无法实现扫频。

综合以上三种方案，本题需要扫频且对扫频信号频率要求较高，因此选择方案 2。

(4) 频偏测量

方案 1: 直接用单片机 ADC 对输入信号进行采样，绘制频谱，通过测量带宽计算频偏。此方法对 ADC 采样率要求极高，且频偏计算易产生较大误差。

方案 2: 将输入信号检波后再进行 ADC 采样，读出检波后信号的幅度和峰峰值，利用调制信号幅度和频偏间线性关系拟合曲线计算频偏。此方法对 ADC 采样频率要求较低，且频偏计算误差较小。

综合以上两种方案，本装置 STM32F407 单片机使用 12 位 ADC，无法实现较高频率采样，因此选用方案 2。

1.2 系统结构工作原理

(1) DDS 原理

DDS 的结构主要由相位累加器、波形存储器、数模(D/A)转换器和低通滤波器等四个大的结构组成。其结构框图如下。

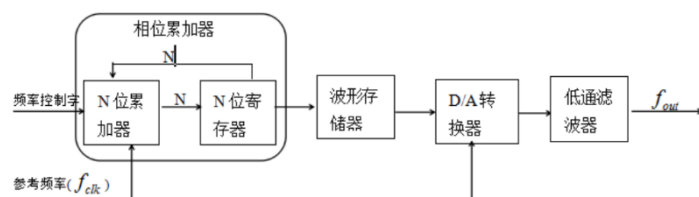


图 1 DDS 工作原理

在 DDS 模块中，输出频率的公式为

$$f_{out} = \frac{f_{clk}}{2^N} \times M \quad (1)$$

其中 M 为频率控制字，当频率控制字变化时，输出频率也跟着变化，从而可以实现调频的基本功能。

(2) 调幅原理

调幅的特点时载波的振幅收受调制信号的控制作用周期性的变化，这个变化的周期与调制信号的周期相同，而振幅变化则与调制信号的振幅成正比，对于简谐振荡的调制信号 $v_{\Omega} =$

$V_{\Omega}\cos\Omega t$, 用它对载波 $v = V_0\cos\omega_0 t$ 进行调幅, 则已调幅波的振幅为 $V(t) = V_0 + k_a V_{\Omega}\cos\Omega t$, 其中 k_a 为比例常数。

因此, 已调波可以用下式表示:

$$v(t) = V(t)\cos\omega_0 t = V_0(1 + m_a\cos\Omega t)\cos\omega_0 t \quad (2)$$

$$m_a = \frac{\frac{1}{2}(V_{\max} - V_{\min})}{V_0} \quad (3)$$

式中 $m_a = \frac{k_a V_{\Omega}}{V_0}$ 叫做调幅指数, m_a 的数值范围可以自 0(未调幅)至 1(百分之百调幅), 调幅波形如图 2 所示。

从上式可知, 在输入信号幅度 V_0 恒定且已知的情况下, 调幅度与包络检波得到的正弦波峰峰值 $V_{\max} - V_{\min}$ 成正比, 因此只需测量已检波形的峰峰值即可得到调幅指数 m_a 。本设计装置即通过此方法确定调幅度。

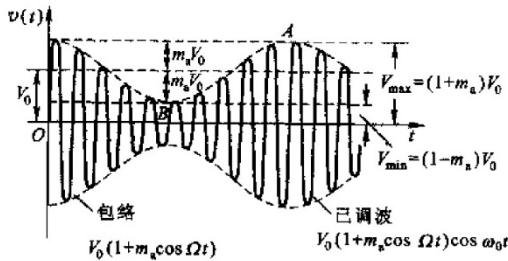


图 2 调幅波形

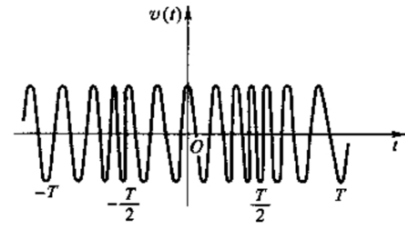


图 3 调频波形

(3) 调频原理

频率调制是用调制信号去控制高频载波的频率, 使其瞬时频率在原来基础上新增随调制信号线性变化的频率分量, 简称为调频, 记为 FM。调频时已调波的瞬时角频率为

$$\omega(t) = \omega_0 + k_f v_{\Omega}(t) \quad (4)$$

式中, ω_0 为正弦载波的固有频率, $v_{\Omega}(t)$ 为调制信号, k_f 为电路变换指数, $\Delta\omega(t) = k_f v_{\Omega}(t)$ 表示瞬时角频率相对于载频的频偏, 简称角频偏。 $\Delta\omega(t)$ 的最大值 $\Delta\omega$ 叫做最大频移, 也称为频偏。

$$\Delta\omega = k_f |v_{\Omega}(t)|_{\max} \quad (5)$$

若调制信号为 $v_{\Omega} = V_{\Omega}\cos\Omega t$, 则可以得到调频波的数学表达式

$$a_f(t) = A_0 \cos(\omega_0 t + m_f \sin\Omega t) \quad (6)$$

式中调频波的调制指数 $m_f = \frac{k_f V_{\Omega}}{\Omega}$, 可以得到调频指数与最大频偏和调频信号频率间的关系

$$m_f = \frac{\Delta f}{F} \quad (7)$$

1.3 系统方案描述

综合以上预期目标，技术方案分析比较以及工作原理分析，最终确定系统方案如下：

输入信号放大后分成两路，初始模拟开关置于调幅路输出。信号在调幅路经过包络检波，ADC 采样并做 FFT，若得到非直流的谱线，则确定输入信号类型为 AM 波，通过谱线高度与固定幅度信号调幅度线性拟合，读取谱线高度即可确定 AM 波调幅度。

若没有在调幅路输出检测到非直流谱线将开关掷于调频路输出。同时单片机控制 DDS 进行 20.7MHz~40.7MHz 扫频，经过与放大过的输入信号混频，再经放大和压电陶瓷滤波，将载波搬移到 10.7MHz。通过 10.7MHz 鉴频器检波得到解调波形，经过各级滤波器，ADC 采样并做 FFT，若在扫频过程中得到非直流的谱线，则确定输入信号为调频波，并确定载波频率。通过读取 FFT 结果，可以得到调制信号频率 F 以及谱线高度，利用固定调制频率下谱线高度与频偏间的线性关系拟合，可以确定频偏并最终确定 FM 波调频度。

若在以上过程中均无非直流谱线，则确定输入信号为单载波信号。

2. 理论分析与计算

2.1 关键器件性能分析

(1) 单片机 STM32F407，12 位 ADC 最高采样频率 2.4M，可以实现对检波后 1~10kHz 的调制信号进行充分采样。

(2) DDS 直接数字频率合成方法采用高性能芯片 AD9959，具有高频率分辨率、高频率切换速度，其工作频率高达 500 MHz，可以轻松实现 10M~30M 的连续扫频。

2.2 系统相关参数设计

(1) 滤波器设计

此装置中滤波器参数需要进行理论计算选择，由于工作频率较低，为追求通带内平坦以及截止频率准确性，选用 TL081 芯片制作有源滤波器。调幅波调制信号频率要求 1~10kHz，因此包络检波后的低通滤波器选用截止频率 10kHz。调频波频率要求 3~10kHz，调频度 m_f 要求 1~6，因此整个实验中最大频偏为 60kHz。同时由于鉴频电路输出波形有 1kHz 一下低频干扰，因此选用截止频率 2kHz 高通滤波器和截止频率 100kHz 低通滤波器串联制作成带通滤波器，再用程控低通滤除掉带通通带内的高频分量。另外，模拟开关引入了高频干扰，为使输出波形稳定，在示波器前又接入了跟随器和 60kHz 低通滤波器，1~10kHz 的已解调信号均能通过。有源滤波器设计可以选用 TI 公司的滤波器设计软件 Filter Designer 进行设计，为保证通带内平坦，选用巴特沃兹类型；为降低器件灵敏度，选用 MFB 型滤波器；为保证下降速度，选用 4 阶滤波器；同时选择 E24 类型电阻电容便于焊接。其中一个有源滤波器设计结果如下。

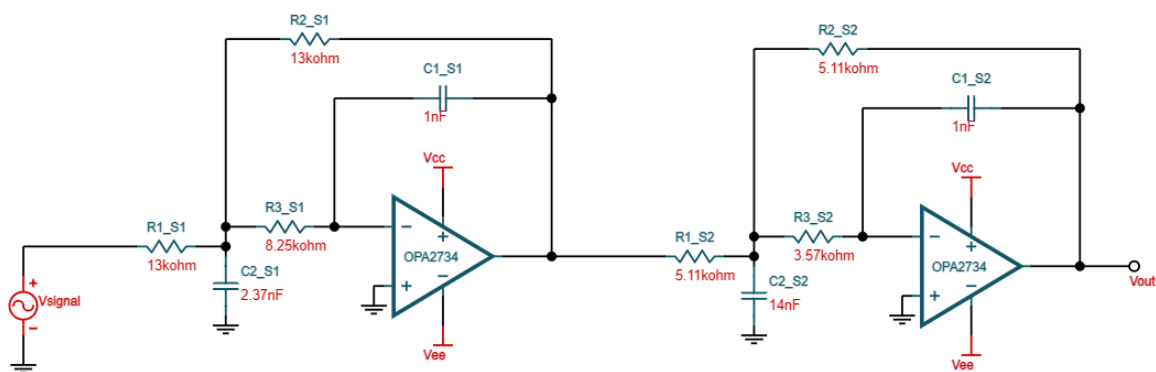


图 4 4 阶有源 10kHz 低通滤波器参数设计

(2) 10.7M 压电陶瓷滤波器阻抗匹配

未经过阻抗匹配的压电陶瓷滤波器在中心频率处下凹，且频带宽度不符合题目要求。本题由于载波在 $10 \sim 30\text{M}$ 之间以 0.5M 为步进，因此滤波器需要保证 10.5M 的载波不能通过。同时调频波最大频偏为 60k ，因此滤波器的带宽应在 $120\text{k} \sim 400\text{k}$ 之间（所以此装置不能采用石英晶体滤波器，且需要保证选用的压电陶瓷带宽合适），通过压电陶瓷滤波器输入输出电阻测量并利用阻抗匹配网站进行 L 型匹配网路参数计算，得到合理的参数（电感 1970nH ，电容 70.2pF ），实现了带通滤波器的通带平坦以及带宽要求。

3. 电路与程序设计

3.1 系统组成与原理框图

系统结构框图如图 2 所示，图中包络检波采用 ADL5511，混频器采用 AD835，比较器采用 TLV3501，程控滤波器采用 LTC1069，模拟开关采用 CH440G，低噪放采用 SPF5189，单片机使用 STM32F407，DDS 采用 AD9959。

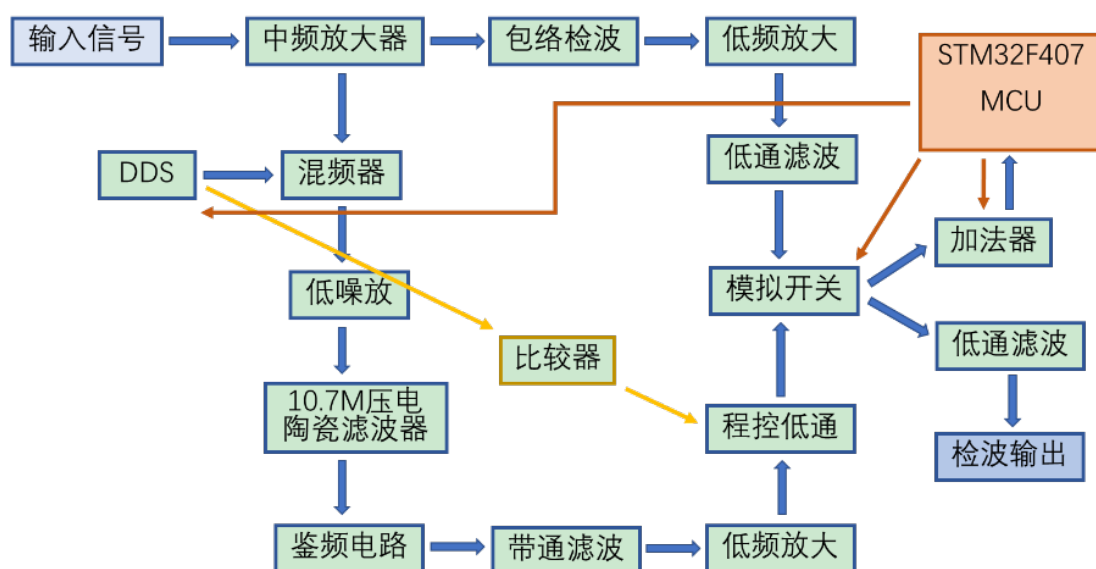


图 5 系统方案框图

3.2 关键模块电路原理

(1) 包络检波模块 ADL5511

ADL5511 为射频 RF 检波器，芯片内部电路如下。峰值检波器通过放大器、二极管及电容器实现对输入射频信号峰值的捕获和保持。具体而言，放大器使得输入信号对电容器充电，而二极管用于防止该电容器放电。

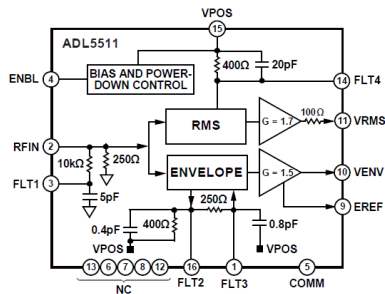


图 6 ADL5511 芯片内部电路

(2) 鉴频电路

该鉴频电路为电容耦合相位鉴频器，由频相转换电路和鉴相器两部分组成。输入的调频信号加到放大器 T 的基极上，放大管的负载是频相转换电路，是通过电容 C_4 、 C_5 耦合的双调谐回路。初级和次级都调谐在中心频率 1/10.7MHz 上。初级回路电压 S 直接加到次级回路中的串联电容 C_4 、 C_5 的中心点上，作为鉴相器的参考电压；同时，S 又经电容 C_4 、 C_5 耦合到次级回路，作为鉴相器的输入电压，即加在 L_2 两端用 U 表示。电容 C_4 、 C_5 上没有压降，二极管 D_4 两端电压为 U^* 。 U^* 是初级回路通过电容耦合到次级回路两端的电压，两者幅度相同，但相位差随着输入调频信号瞬时频率变化。

鉴相器采用两个并联二极管检波电路，电压传输系数完全相等，输出信号为两个检波电路的输出电压差。当 $f = f_0$ 时输出电压比输入滞后 90° ，此时鉴频器输出为 0；当 $f > f_0$ 时滞后小于 90° ，此时鉴频器输出大于 0；当 $f < f_0$ 时滞后大于 90° ，此时鉴频器输出小于 0。

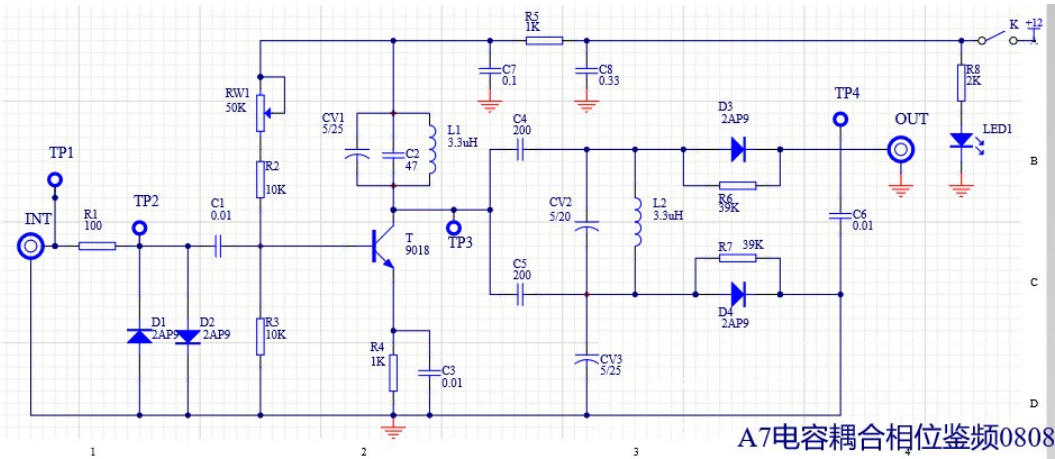


图 7 鉴频特性曲线

3.3 系统软件与流程图

(1) 软件流程图

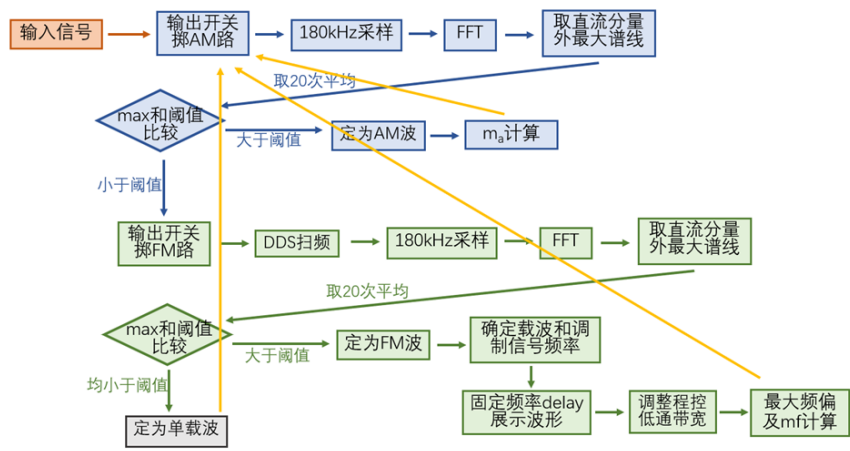


图 8 TLV3501 迟滞比较器 V-A 特性

(2) 主要模块程序设计

代码清单（具体代码见附录）：① 判断波形是否为调幅波；② 判断波形是否为调频波；③ 找频域最大谱线；④ FFT 波形绘制；⑤ 拟合 m_f 代码

4. 测试方案与测试结果

4.1 测试环境

①固纬程控直流电源（GPD3303）；②鼎阳 500M 示波器（SDS3052）；③信号发生器（DG1032）；④鼎阳 3.2G 频谱仪（SA8300B-E）；⑤STM32F407 开发板

4.2 测试方案

（1）首先输入单频载波，看装置能否准确判断波形。（2）再输入 100mV 调幅波，改变载波频率，调制信号频率，输入信号调幅度，看是否能准确判断波形，检波得到波形是否稳定噪声小，显示得到的调幅度是否准确。（3）最后输入 100mV 调频波，改变载波频率，调制信号频率，最大频偏，看能否准确判断波形，检波得到波形是否稳定噪声小，显示得到的最大频偏和调频度是否准确。

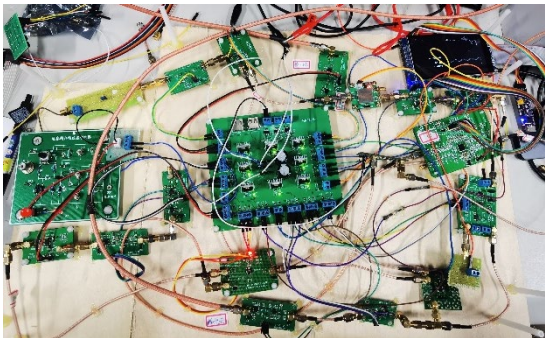


图 9 实际电路

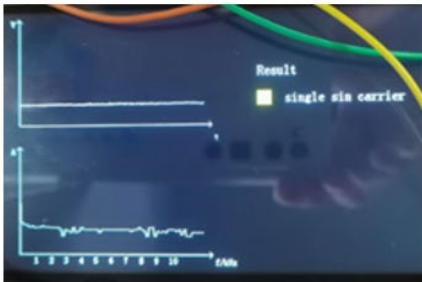
4.3 测试结果

(1) 波形判断

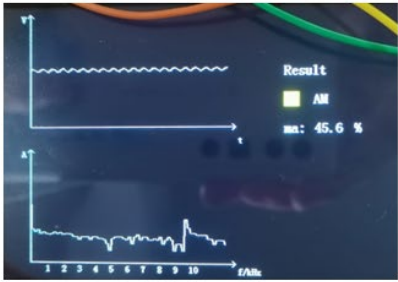
如图 10 中 a~c 所示，装置可以对输入波形进行正确判断并显示。

(2) 解调信号波形

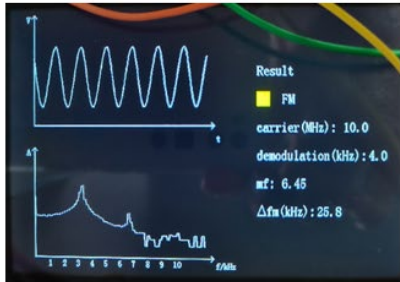
如图 10 中 d e, 分别为调频波解调结果，可以看到该装置解调得到的波形幅度大小适中，无明显失真和干扰，且波形较为稳定，完美完成了题目要求。



a 单载波波形判断



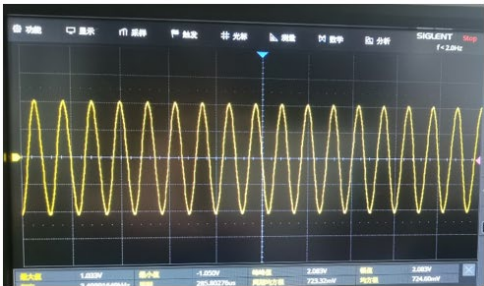
b 调幅波波形判断



c 调频波波形判断



d 调幅波解调波形



e 调频波解调波形

(3) 调幅度及调频度测试

控制载波频率为 10MHz，改变调制信号频率和调制度，得到调制度测量值如下。

表 1 调幅度测试表

调幅度真实值/%	调制信号频率/kHz								
	1	2	3	5	6	7	8	9	10
20	20.5	19.5	19.3	19.2	18.3	17.2	17	22.3	25
30	29.7	31	32.7	28.8	27.4	34.1	33.2	26.6	35.3
40	40.4	39.2	38.2	37.3	43	44.3	36.5	44.2	45
50	50.3	51.2	51.3	52.4	46.7	47.2	53.2	54.4	46.5
60	59.5	62.3	58	63.4	56.4	64.7	65.3	56.7	55.6
70	69.7	67.8	68.4	72.1	71.3	68.9	65	67.4	73.8
80	80.2	78.2	77.8	82.9	78.4	83.5	76.4	85.3	77.8
90	90.4	88.5	89.6	90.8	88.4	87.5	93.2	86.7	94.2
100	99.3	100.3	98	102	97.2	97.6	96.3	94.2	104.3

如上表所示，调幅度测量误差均控制在 0.1 之内，但误差有随调制信号频率增大而增大，随调幅度增大而增大的趋势。当调幅度较大时，检波得到的波形不可避免地失真，且由于放大器等器件频响不平坦，拟合时采用二次曲线，因此较大的调制信号频率处很可能产生误差。

如下表所示，调频度误差均控制在 0.3 之内，但在调频度较高时误差较大。由于放大器，滤波器等器件频响不平坦，因此也采用二次曲线拟合，且 DDS 输出信号幅度随上电时间有一定变化，因此在调频度较大时会存在较大误差。

表 2 调频度测试表

调频度真实值	调制信号频率/kHz							
	3	4	5	6	7	8	9	10
1	0.98	1.03	0.99	1.02	0.96	0.97	0.98	0.99
2	1.97	1.96	2.04	2.05	2.02	1.96	1.93	2.06
3	2.95	3.06	3.22	2.74	2.93	3.04	3.25	3.16
4	4.01	3.89	3.97	3.86	4.04	4.16	3.85	4.27
5	4.82	4.78	4.76	4.81	5.24	4.76	5.26	4.81
6	5.78	5.75	5.81	5.74	5.76	5.77	6.1	5.88

5. 参考资料

[1] 康华光. 电子技术基础(模拟部分) [M]. 北京: 高等教育出版社, 2013.

[2] 黄智伟. 《全国大学生电子设计竞赛训练教程》. 电子工业出版社, 2005 年第 1 版

[3] 张肃文. 高频电子线路(第四版) [M]. 北京: 高等教育出版社, 2004.

附录

1. 判断波形是否为调幅波

```
float type_judge()
{
    int num=0;
    float maxpower=0;
    for(int m=0;m<20;m++)
    {
        for(int j=0;j<NPT;j++){
            lBufInArray[j]=(signed short)Get_Adc(ADC_Channel_5);
        }
        for(int i=0;i<NPT;i++)
        {
            lBufInArray[i] = (lBufInArray[i]) << 16;
        }
        cr4_fft_1024_stm32(lBufOutArray, lBufInArray, NPT);
        GetPowerMag();
        maxpower+=get_maxpower(&num);
    }
    maxpower/=20;
    if (maxpower>0)
    {
        return maxpower;
    }
    else
    {
        return 0;
    }
}
```

2. 判断波形是否为调频波

扫频，确定载波，频谱判断

```
int get_carrier()
{
    int freq0=20700000;
    int freq;
    int num=0;
    float max=0;
    int first=0;
    int maxp;
    float power_first;
    float power;
    WriteFreq(1,1000000,1);
    for(int i=0;i<=40;i++)
    {
        freq=freq0+500000*i;
        WriteFreq(0, freq, 1);
        WriteAmplitude(0, get_amp(freq/1000000.0), 1);
        delay_ms(50);
        for(int j=0;j<NPT;j++){
            lBufInArray[j]=(signed short)Get_Adc(ADC_Channel_5);
        }
        for(int i=0;i<NPT;i++)
        {
            lBufInArray[i] = (lBufInArray[i]) << 16;
        }
        cr4_fft_1024_stm32(lBufOutArray, lBufInArray, NPT);
        GetPowerMag();
        power_first=get_maxpower(&maxp);
        if(power_first>max){max=power_first;num=freq;}
    }
    power_first=0;
}
```

```

if (max>40)
{
    WriteFreq(0, num, 1);
    WriteAmplitude(0, get_amp(num/1000000.0), 1);
    delay_ms(50);
    for(int m=0;m<5;m++){
        for(int j=0;j<NPT;j++){
            lBufInArray[j]=(signed short)Get_Adc(ADC_Channel_5);
        }
        for(int i=0;i<NPT;i++)
        {
            lBufInArray[i] = (lBufInArray[i]) << 16;
        }
        cr4_fft_1024_stm32(lBufOutArray, lBufInArray, NPT);
        GetPowerMag();
        power=get_maxpower(&maxp);
        power_first+=power;
    }
    return num-10700000;
}
return 0;
}

```

3. 找频域最大谱线

```

void GetPowerMag()
{
    signed short lX,lY;
    float X,Y,Mag;
    unsigned short i;
    for(i=0; i<NPT/2; i++)
    {
        lX = (lBufOutArray[i] << 16) >> 16;
        lY = (lBufOutArray[i] >> 16);

        X = NPT * ((float)lX) / 32768;
        Y = NPT * ((float)lY) / 32768;

        Mag = sqrt(X * X + Y * Y) / NPT;
        if(i == 0)
            lBufOutArray1[i] = (long) (Mag * 32768);
        else
            lBufOutArray1[i] = (long) (Mag * 65536);
    }
}

```

4. FFT 波形绘制

```

void Draw_FFTwave(void)
{
    int i;
    LCD_Display_Dir(1);
    double y0,y1;
    LCD_Fill(51,256,409,449,BLACK);
    for(i=0;i<70;i++)
    {
        if(lBufOutArray1[i]<=0)
        {
            y0=-1;
        }
        else
        {
            y0=(log10(lBufOutArray1[i]))/2.5*100;
        }
        if(lBufOutArray1[i+1]<=0)
            y1=-1;
        else
            y1=(log10(lBufOutArray1[i+1]))/2.5*100;
        if(y0>190)y0=190;
        if(y1>190)y1=190;
        LCD_DrawLine(50+5*i,440-(int)y0,50+5*i+2,440-(int)y1);
    }
}

```

5. 拟合 m_f 代码

```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
from numpy import polyfit, polyid

if __name__ == '__main__':
    x = [204, 404, 595, 780, 952, 1110,
182, 365, 540, 698, 848, 969,
157, 297, 439, 558, 669, 760,
195, 385, 558, 706, 836, 950,
220, 426, 608, 763, 894, 1015,
184, 352, 490, 610, 708, 802,
162, 303, 413, 509, 595, 664,
180, 329, 447, 546, 630, 708,
119, 219, 291, 352, 408, 454,
127, 230, 305, 372, 427, 471]

    y = [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6 \
, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]

    coe3 = polyfit(x[0:5], y[0:5], 2)
    coe35 = polyfit(x[6:11], y[6:11], 2)
    coe4 = polyfit(x[12:17], y[12:17], 2)
    coe45 = polyfit(x[18:23], y[18:23], 2)
    coe5 = polyfit(x[24:29], y[24:29], 2)
    coe6 = polyfit(x[30:35], y[30:35], 2)
    coe7 = polyfit(x[36:41], y[36:41], 2)
    coe8 = polyfit(x[42:47], y[42:47], 2)
    coe9 = polyfit(x[48:53], y[48:53], 2)
    coe10 = polyfit(x[54:59], y[54:59], 2)
    print("double coe3[3]={} {}, {}, {}".format("{", coe3[0], coe3[1], coe3[2], "}"))
    print("double coe35[3]={} {}, {}, {}".format("{", coe35[0], coe35[1], coe35[2], "}"))
    print("double coe4[3]={} {}, {}, {}".format("{", coe4[0], coe4[1], coe4[2], "}"))
    print("double coe45[3]={} {}, {}, {}".format("{", coe45[0], coe45[1], coe45[2], "}"))
    print("double coe5[3]={} {}, {}, {}".format("{", coe5[0], coe5[1], coe5[2], "}"))
    print("double coe6[3]={} {}, {}, {}".format("{", coe6[0], coe6[1], coe6[2], "}"))
    print("double coe7[3]={} {}, {}, {}".format("{", coe7[0], coe7[1], coe7[2], "}"))
    print("double coe8[3]={} {}, {}, {}".format("{", coe8[0], coe8[1], coe8[2], "}"))
    print("double coe9[3]={} {}, {}, {}".format("{", coe9[0], coe9[1], coe9[2], "}"))
    print("double coe10[3]={} {}, {}, {}".format("{", coe10[0], coe10[1], coe10[2], "}"))
```