

# 目录

一. OVI_1.....	3
(1) 示例线路图.....	3
(2) 示例程序.....	3
(3) 波形图.....	4
(4) 数据.....	5
二. PV3_4.....	5
(1) 示例线路图.....	5
(2) 示例程序.....	5
(3) 波形图.....	7
(4) 数据.....	7
三. ACS_5.....	8
(1) 示例线路图.....	8
(2) 示例程序.....	8
(3) 波形图.....	10
(4) 数据.....	11
四. QTMU_6.....	11
(1) 示例线路图.....	11
(2) 示例程序.....	12
(3) 波形.....	14
(4) 数据.....	14
五. DDD_7.....	15
(1) 示例线路图.....	15
(2) 示例程序.....	15
(3) 波形.....	17
(4) 数据.....	17
六. DOAL_8.....	18
(1) 示例线路图.....	18
(2) 示例程序.....	19
(3) 数据.....	24
七. DVI_9.....	25
(1) 示例线路图.....	25
(2) 示例程序.....	25
(3) 波形.....	26
(4) 数据.....	26
八. DVI_11.....	27
(1) 示例线路图.....	27
(2) 示例程序.....	27
(3) 波形.....	28
(4) 数据.....	29
九. OV2_12.....	29
(1) 示例线路图.....	29
(2) 示例程序.....	29

(3) 波形.....	31
(4) 数据.....	32
十. DVI3_13.....	32
(1) 示例线路图.....	32
(2) 示例程序.....	32
(3) 波形.....	34
(4) 数据.....	34
十一. CBIT_14.....	35
(1) 示例线路图.....	35
(2) 示例程序.....	35
(3) 波形.....	36
(4) 数据.....	36
十二. HVS_15.....	37
(1) 示例线路图.....	37
(2) 示例程序.....	37
(3) 波形.....	38
(4) 数据.....	38
十三. XVI_16.....	39
(1) 示例线路图.....	39
十四. MUX_20.....	39
(1) 示例线路图.....	39
(2) 示例程序.....	40
(3) 波形.....	42
(4) 数据.....	42
十五. QFVI_21.....	43
(1) 示例线路图.....	43
(2) 示例程序.....	44
(3) 数据.....	45

# DEMO\_BOARD

## 一. OVI\_1

(1) 示例线路如下图 1.1 。

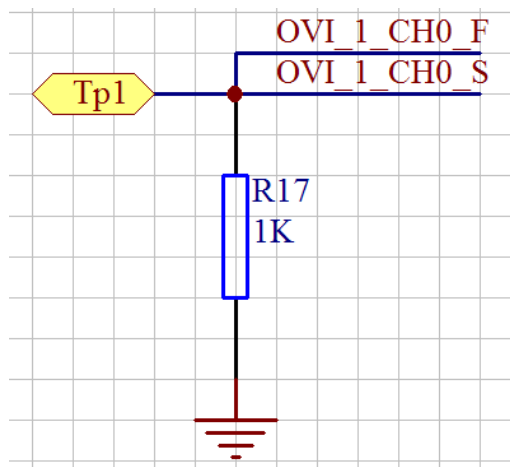


图 1.1 OVI 测试线路图

如上图 1.1 ，将 1K 欧姆的电阻一端接地，一端接 OVI 的 0 通道，OVI 加电压测试流过电阻的电流（加电流测电压亦可），再根据欧姆定律算出阻值，接近 1K 欧姆。

(2) 示例程序

```
float r,v,i,fail_bin = 5; // 变量声明
system_init();//测试卡初始化
delay(1);//“1”表示延时 1mS
ovi_1->set_voltage(OVI_CHANNEL_0,1.0f,RANGE_1_V);
//“OVI_CHANNEL_0”表示通道选择，“1.0f”表示给电压钳位 1.0V，
“RANGE_1_V”表示档位选择，为 1V 档
ovi_1->set_current(OVI_CHANNEL_0,3e-3f,RANGE_3_MA);
//“OVI_CHANNEL_0”表示通道选择，“3e-3f”表示电流钳位 3mA，
“RANGE_3_MA”表示档位选择，为 3mA 档
ovi_1->set_meas_mode(OVI_CHANNEL_0,OVI_MEASURE_CURRENT);//电
流测试模式设置语句，表示测量 0 通道的电流，“OVI_CHANNEL_0”表示通道
```

选择，“OVI\_MEASURE\_CURRENT”表示进入测试电流模式

```
delay(2);
i=ovi_1->measure_average(10);//返回 10 次 ADC 采样的平均值
ovi_1->set_meas_mode(OVI_CHANNEL_0,OVI_MEASURE_VOLTAGE);//电
压测试模式设置语句，表示测量 0 通道的电压，“OVI_CHANNEL_0”表示通道
选择，“OVI_MEASURE_CURRENT”表示进入测试电压模式
delay(2);
v=ovi_1->measure_average(10);//返回 10 次 ADC 采样的平均值
r=v/i;//欧姆定律公式
ovi_1->set_voltage(OVI_CHANNEL_0,0.0f,RANGE_1_V); //掉电
ovi_1->set_current(OVI_CHANNEL_0,1e-3f,RANGE_3_MA);
delay(2);
system_init();//测试卡初始化
delay(2);
/////显示/////
do_dlog(func,0,v,POWER_UNIT,fail_bin);// “0”表示显示的顺序，“v”表示
要输出的变量值，“POWER_UNIT”表示量程档位为 1，“fail_bin”表示错误返
回值，简称软件 BIN
do_dlog(func,1,i, POWER_MILLI,fail_bin);// “POWER_MILLI”表示量程档位
为 1e-3
do_dlog(func,2,r,POWER_UNIT,fail_bin);
```

### (3) 波形图

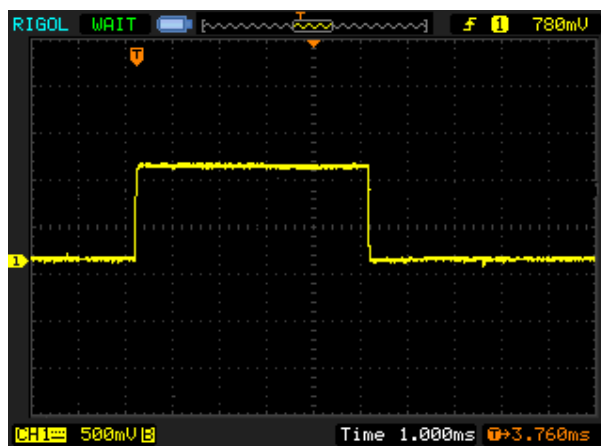


图 1.2 Tp1 处波形

如上图 1.2，选用 1K 的电阻，给 1V 的电压，理论上应该产生 1mA 的电流，程序里钳位在 3MA，所以此时电压钳位了，如上图 1.2 波形图显示的电压值为“1V”，与理论值相符。

#### (4) 数据

12/28/13 15:53:08 Serial Number: 41			Total Devices: 41			PASSED BIN: 1		
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.01.01	OVI_1	V	1.0032	1	V	none	none	
1.01.02	OVI_1	I	0.9969	1	mA	none	none	
1.01.03	OVI_1	R	1006.3957	1	ohm	990.0000	1010.0000	

12.725 mSecs Total=12.725 mSec

图 1.3 测试数据

如上图 1.3 显示, 电压值、电流值和计算得到的阻值分别为 1.0032V、0.9969 mA 和 1.006K 欧姆 (测试线路中非严格的开尔文连接, 阻值属于正常范围内)。

## 二. PV3\_4

(1) 示例线路如下图 2.1:

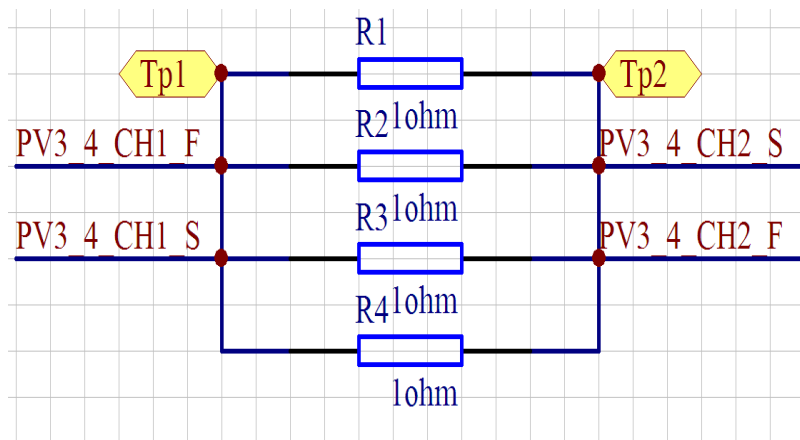


图 2.1 PV3 测试线路图

如上图 2.1, R1, R2, R3, R4 并联两端分别接 pv3 的 1 通道和 2 通道, 通过 PV3 给电阻加电流, 测量电阻两端的电压, 根据欧姆定律算出电阻阻值。

(2) 示例程序

```
float charge=0,fail_bin=6;//变量声明
int timeout=0;
float R;
float i=0,v=0;
```

```

pv3_4->init();//PV3 初始化
delay(2);//延时 2mS
charge=0;
timeout=0;//变量初始值
pv3_4->charge_on();//对电容器充电
pv3_4->set_meas_mode(PV3_CHARGE_S);
delay(1);
charge = pv3_4->measure();
while((charge < 63.0)  && (timeout < 200))//充电电容达到 63V 后停止充电
{
    delay(1);
    timeout++;
    charge = pv3_4->measure();
}
pv3_4->charge_off();//停止对电容器充电，允许以浮动模式使用电源
delay(2);
if(timeout >= 200) goto end1;//跳出，放弃 pv3 测试
pv3_4->set_voltage(0);
pv3_4->set_current(0);//给一个初始电压电流
delay(1);
pv3_4->close_switch(PV3_HIGH_FORCE_1);//关闭通道 1 的 force 继电器
pv3_4->close_switch(PV3_HIGH_SENSE_1);//关闭通道 1 的 sense 继电器
pv3_4->close_switch(PV3_LOW_FORCE_2);//关闭通道 2 的 force 继电器
pv3_4->close_switch(PV3_LOW_SENSE_2); //关闭通道 2 的 sense 继电器
delay(1);
pv3_4->fast_comp();//可以用于维持定时，使负载有效削弱控制回路的带宽
i=3;
pv3_4->set_current(i);//设定电流
pv3_4->set_voltage(2);//设定电压
pv3_4->set_meas_mode(PV3_MEASURE_VOLTAGE);//设定测试模式为电压
delay(1);
pv3_4->drive_on();//开始驱动
delay_us(280);// 延时 280uS
v=pv3_4->measure_average(10); //返回 10 次 ADC 采样平均值
R=v/i;
pv3_4->set_voltage(0); //源置 0
pv3_4->set_current(0);
delay(1);//掉电
pv3_4->charge_off();
delay(5);
pv3_4->open_switch(PV3_HIGH_FORCE_1);//打开用到的继电器
pv3_4->open_switch(PV3_HIGH_SENSE_1);
pv3_4->open_switch(PV3_LOW_FORCE_2);
pv3_4->open_switch(PV3_LOW_SENSE_2);

```

```

delay(5);
//////////
end1:
pv3_4->init();//PV3 初始化
delay(2);
//////////
//显示函数
do_dlog(func,0,v,POWER_UNIT,fail_bin);// “0”表示显示的顺序,“v”表示要输出的变量值,“POWER_UNIT”表示量程档位为1,“fail_bin”表示错误返回值,简称软件BIN
do_dlog(func,1,i,POWER_UNIT,fail_bin);
do_dlog(func,2,R,POWER_UNIT,fail_bin);

```

### (3) 波形图

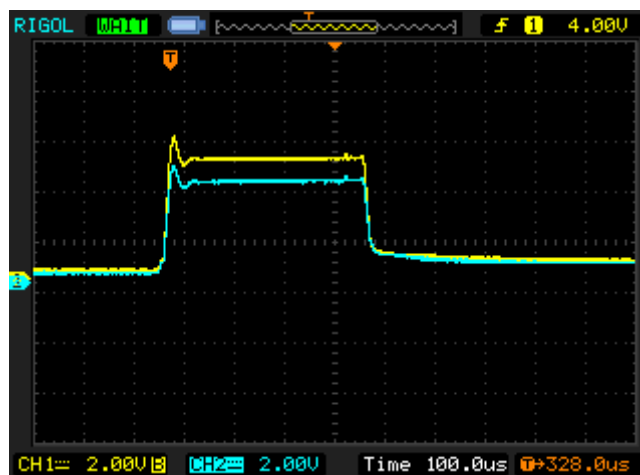


图 2.2 电阻两端波形图

如上图 2.2 所示,黄表笔接 Tp1,蓝表笔接 Tp2,程序中给 3A 的电流,4 个 1 欧姆电阻并联阻值为 0.25 欧姆,理论上电阻的压降应为 0.75V 从波形中可以看到符合理论值。

### (4) 数据

12/28/13 16:06:10 Serial Number: 94 Total Devices: 94 PASSED BIN: 1								
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.02.01	PV3_4	V	0.7523	1	V	none	none	
1.02.02	PV3_4	I	3.0000	1	A	none	none	
1.02.03	PV3_4	R	0.2508	1	ohm	0.2400	0.2600	
29.832 mSecs Total=29.832 mSec								

图 2.3 测试数据

如上图 2.3 显示，电压值、电流值和计算得到的阻值分别为 0.7523V、3A 和 0.2508 欧姆。

### 三. ACS\_5

#### (1) 示例线路如下图 3.1:

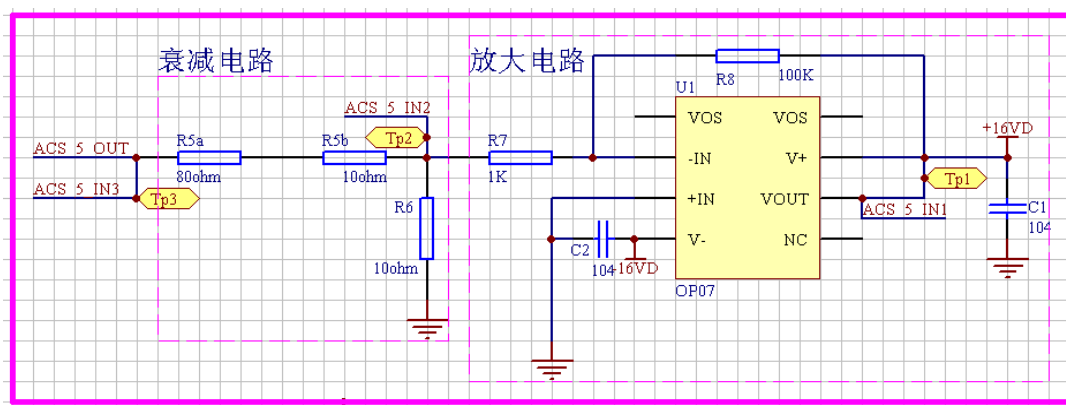


图 3.1 ACS 测试线路图

如上图 3.1，ACS 测试卡含有一个 OUT（输出）通道和四个 IN（输入）通道，输出通道输出交流信号；输入通道输入交流信号，且测量输入信号的有效值。我们先通过 ACS\_OUT 通道输出一段正弦信号，用 ACS\_IN3 通道来自测其有效值，R5a, R5b, R6 组成的衰减电路使信号缩小 10 倍，再用 ACS\_IN2 通道来测量经过衰减后信号的有效值，再利用 R7, R8, U1 组成的放大电路放大 100 倍，ACS\_IN1 通道来测量放大后信号的有效值。

#### (2) 示例程序

```
float IN1=0,t=0,IN2=0,IN3=0,fail_bin=7;//变量声明
acs_5->init();//ACS 初始化
delay(2);//延时 2mS
//正弦波
acs_5->ldram(0.5e3);//频率 500HZ
acs_5->set_path(ACS_BASE_FILTER,ACS_1_VOLT_RANGE); //插入过滤器
acs_5->set_level(0,0,ACS_PEAK_LEVEL);//0V 振幅, 0V 偏移, 峰形
acs_5->close_relay(D_SIG_OUT);//闭合输出连接继电器
acs_5->close_relay(D_RMS_CH1);//闭合输入 1 通道连接继电器
acs_5->open_relay(D_RMS_CH2);//断开输入 2 通道连接继电器
acs_5->open_relay(D_RMS_CH3);//断开输入 3 通道连接继电器
```



```

delay(2);
acs_5->set_meas_mode(ACS_CHANNEL_1,ACS_5V_RMS_RANGE,ACS_AC
_COUPLING);//选择测量的通道和电压范围, 和连接器
acs_5->set_level(0.5f,0.0f,ACS_PEAK_LEVEL);//0.5V 振幅, 0V 偏移, 峰形
acs_5->start();//启动波形
delay(10);
IN1=acs_5->measure();//返回 ADC 采样 10 次的平均值
delay(2);
acs_5->open_relay(D_RMS_CH1);//断开输入 1 通道连接继电器
acs_5->close_relay(D_RMS_CH2); //闭合输入 2 通道连接继电器
acs_5->open_relay(D_RMS_CH3);//断开输入 3 通道连接继电器
delay(2);
acs_5->set_meas_mode(ACS_CHANNEL_2,ACS_1V_RMS_RANGE,ACS_AC
_COUPLING);
delay(2);
IN2=acs_5->measure();
delay(2);
acs_5->open_relay(D_RMS_CH1);//断开输入 1 通道连接继电器
acs_5->open_relay(D_RMS_CH2);//断开输入 2 通道连接继电器
acs_5->close_relay(D_RMS_CH3);//闭合输入 3 通道连接继电器
delay(2);
acs_5->set_meas_mode(ACS_CHANNEL_3,ACS_1V_RMS_RANGE,ACS_AC
_COUPLING);
delay(2);
IN3=acs_5->measure();
delay(2);
acs_5->stop();//停止波形存储器输出
delay(5);
acs_5->init();//ACS 初始化
delay(2);
///显示函数
do_dlog(func,0,IN1,POWER_MILLI,fail_bin);// “0” 表示显示的顺序, “IN1”
表示要输出的变量值, “POWER_MILLI” 表示量程档位为 1e-3, “fail_bin” 表
示错误返回值, 简称软件 BIN
do_dlog(func,1,IN2, POWER_MILLI,fail_bin);
do_dlog(func,2,IN3,POWER_MILLI,fail_bin);

```

### (3) 波形图

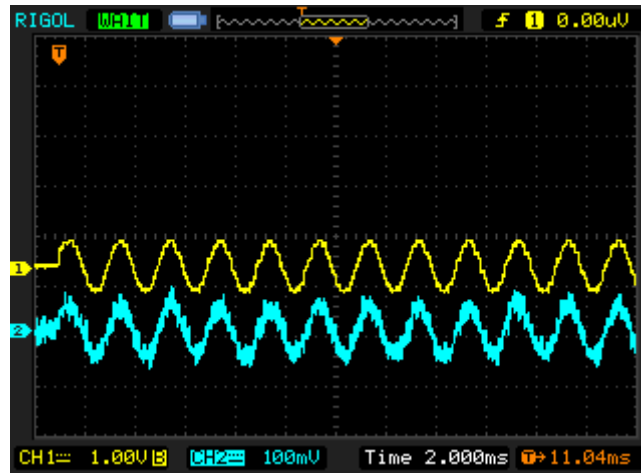


图 3.2 衰减电路波形图

如上图 3.2，黄色为  $T_{p3}$  处波形，蓝色为  $T_{p2}$  处波形，正弦信号经过衰减电路缩小十倍，图 3.2 中可以清楚的看到，幅值变为原来的十分之一，相位和频率不变。

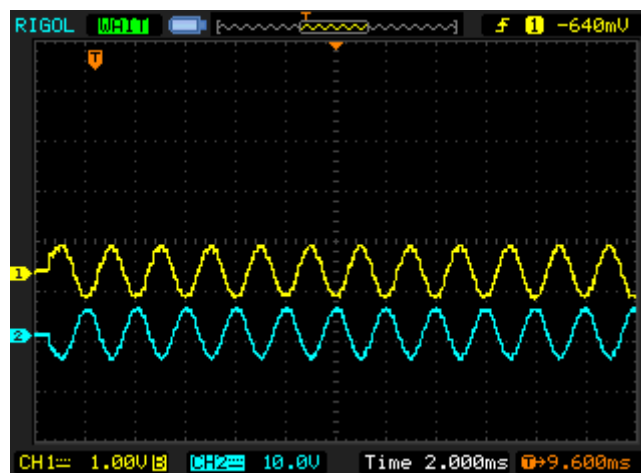


图 3.3 放大电路波形图

如上图 3.3，黄色为  $T_{p3}$  处波形，蓝色为  $T_{p1}$  处波形，经过衰减电路衰减 10 倍之后又经过放大电路反向放大 100 倍，图 3.3 中可以看到经过衰减之后的波形放大一百倍，相当于原来的波形放大了十倍，幅值扩大十倍，频率不变。

#### (4) 数据

12/28/13 16:10:42 Serial Number: 30		Total Devices: 30		PASSED BIN: 1				
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.03.01	ACS_5	IN1	3388.1345	1	mV	none	none	
1.03.02	ACS_5	IN2	33.2713	1	mV	none	none	
1.03.03	ACS_5	IN3	343.2905	1	mV	none	none	
60.780 mSecs Total=60.780 mSec								

图 3.4 测试数据

如上图 3.4 显示，IN3，IN2，IN1 处得有效值分别为 343.2905mV，33.2713mV，3388.1345mV，换算成峰值电压分别约为 0.49V，0.047V，4.79V。

## 四. QTMU\_6

(1) 示例线路如下图 4.1:

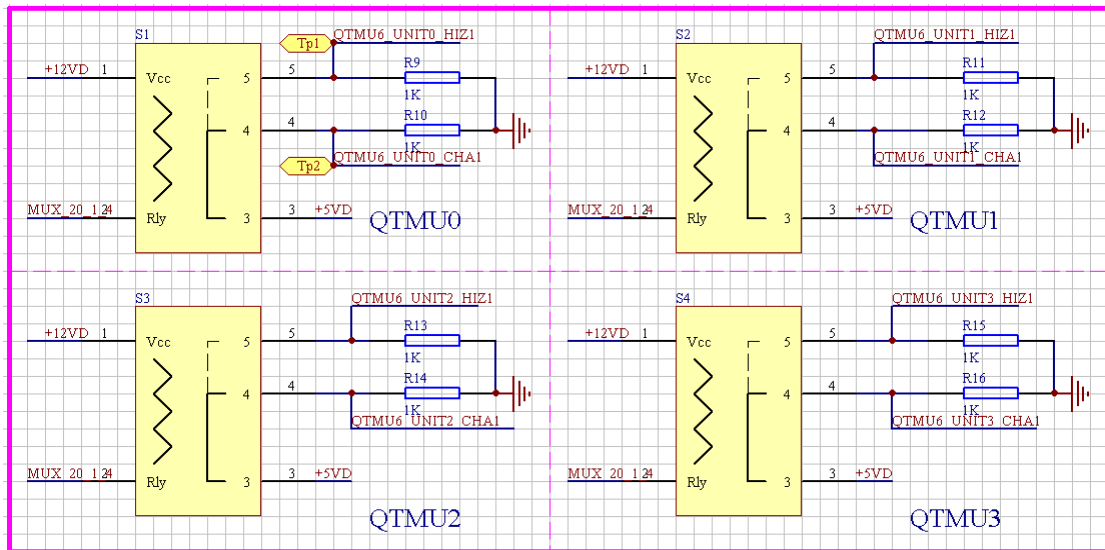


图 4.1 QTMU 测试线路图

QTMU 为 TMU 的升级版本，在 TMU 的基础上，多加了 3 个测试卡，可以 4 路并测。线路 4.1 中，选用 DS1E-M-DC12V 继电器来作为示例，测量其吸合时间。当继电器没闭合时，常闭位为高电平 5V，常开位为低电平，当继电器闭合时，常开位从 0V 上升到 5V，常闭位从 5V 下降至 0V，根据这一变化，可以用 QTMU 测出继电器的吸合时间。

## (2) 示例程序

```
float t[4]={0},fail_bin=8; //定义变量
system_init();//系统初始化
delay(2);//延时 2mS
mux_20->close_relay(MUX_BANK_1_2);
mux_20->close_relay(MUX_2_GND);//闭合 MUX 板上的继电器到地
delay(2);//延时 2mS
qtmu1_6->qtmu1_close_relay(QTMU1_UNIT_0,QTMU1_CHAN_A_DUT1);//
闭合 QTMU 的 TMU_0 的 CHAN_A1
qtmu1_6->qtmu1_close_relay(QTMU1_UNIT_1,QTMU1_CHAN_A_DUT1);//
闭合 QTMU 的 TMU_1 的 CHAN_A1
qtmu1_6->qtmu1_close_relay(QTMU1_UNIT_2,QTMU1_CHAN_A_DUT1);//
闭合 QTMU 的 TMU_2 的 CHAN_A1
qtmu1_6->qtmu1_close_relay(QTMU1_UNIT_3,QTMU1_CHAN_A_DUT1);//
闭合 QTMU 的 TMU_3 的 CHAN_A1
qtmu1_6->qtmu1_close_relay(QTMU1_UNIT_0,QTMU1_HIZ_DUT1);//闭合
QTMU 的 TMU_0 的 CHAN_HIZ1
qtmu1_6->qtmu1_close_relay(QTMU1_UNIT_1,QTMU1_HIZ_DUT1);//闭合
QTMU 的 TMU_1 的 CHAN_HIZ1
qtmu1_6->qtmu1_close_relay(QTMU1_UNIT_2,QTMU1_HIZ_DUT1);//闭合
QTMU 的 TMU_2 的 CHAN_HIZ1
qtmu1_6->qtmu1_close_relay(QTMU1_UNIT_3,QTMU1_HIZ_DUT1);//闭合
QTMU 的 TMU_3 的 CHAN_HIZ1
delay(2);
qtmu1_6->qtmu1_start_trigger_setup(QTMU1_UNIT_0,2.0f,QTMU1_NEG_SL
OPE,QTMU1_CHAN_A,QTMU1_IN_5V);//设置 QTMU 起始点为 TMU_0 的
CHAN_A1，触发电压为 2V 下降沿，量程为 5V
qtmu1_6->qtmu1_stop_trigger_setup(QTMU1_UNIT_0,2.0f,QTMU1_POS_SL
OPE,QTMU1_HIZ,QTMU1_IN_5V);//设置 QTMU 结束点为 TMU_0 的
CHAN_HIZ1，触发电压为 2V 上升沿，量程为 5V
qtmu1_6->qtmu1_start_trigger_setup(QTMU1_UNIT_1,2.0f,QTMU1_NEG_SL
OPE,QTMU1_CHAN_A,QTMU1_IN_5V);//同上
qtmu1_6->qtmu1_stop_trigger_setup(QTMU1_UNIT_1,2.0f,QTMU1_POS_SL
OPE,QTMU1_HIZ,QTMU1_IN_5V);
qtmu1_6->qtmu1_start_trigger_setup(QTMU1_UNIT_2,2.0f,QTMU1_NEG_SL
OPE,QTMU1_CHAN_A,QTMU1_IN_5V);
qtmu1_6->qtmu1_stop_trigger_setup(QTMU1_UNIT_2,2.0f,QTMU1_POS_SL
OPE,QTMU1_HIZ,QTMU1_IN_5V);
qtmu1_6->qtmu1_start_trigger_setup(QTMU1_UNIT_3,2.0f,QTMU1_NEG_SL
OPE,QTMU1_CHAN_A,QTMU1_IN_5V);

qtmu1_6->qtmu1_stop_trigger_setup(QTMU1_UNIT_3,2.0f,QTMU1_POS_SL
```

```

OPE,QTMU1_HIZ,QTMU1_IN_5V);
    delay(10); //延时 10mS
    qtmu1_6->qtmu1_arm(QTMU1_UNIT_0); //设置测量到的值存储在 TMU_0 储
寄存器中
    qtmu1_6->qtmu1_arm(QTMU1_UNIT_1);
    qtmu1_6->qtmu1_arm(QTMU1_UNIT_2);
    qtmu1_6->qtmu1_arm(QTMU1_UNIT_3);
    delay(1);
    mux_20->close_relay(MUX_1_4);    //闭合 MUX20-1-4 继电器
    delay(5);
    t[0]=qtmu1_6->qtmu1_read(QTMU1_UNIT_0); //读取 TMU_0 储存器中的值
    t[1]=qtmu1_6->qtmu1_read(QTMU1_UNIT_1);
    t[2]=qtmu1_6->qtmu1_read(QTMU1_UNIT_2);
    t[3]=qtmu1_6->qtmu1_read(QTMU1_UNIT_3);
    ////////////
    qtmu1_6->qtmu1_open_relay(QTMU1_UNIT_0,QTMU1_CHAN_A_DUT1); //
断开 QTMU 所有用到的通道的继电器
    qtmu1_6->qtmu1_open_relay(QTMU1_UNIT_1,QTMU1_CHAN_A_DUT1);
    qtmu1_6->qtmu1_open_relay(QTMU1_UNIT_2,QTMU1_CHAN_A_DUT1);
    qtmu1_6->qtmu1_open_relay(QTMU1_UNIT_3,QTMU1_CHAN_A_DUT1);
    qtmu1_6->qtmu1_open_relay(QTMU1_UNIT_0,QTMU1_HIZ_DUT1);
    qtmu1_6->qtmu1_open_relay(QTMU1_UNIT_1,QTMU1_HIZ_DUT1);
    qtmu1_6->qtmu1_open_relay(QTMU1_UNIT_2,QTMU1_HIZ_DUT1);
    qtmu1_6->qtmu1_open_relay(QTMU1_UNIT_3,QTMU1_HIZ_DUT1);
    mux_20->open_relay(MUX_1_4); //断开继电器
    mux_20->open_relay(MUX_BANK_1_2); //断开 MUX 板上接地的继电器
    mux_20->open_relay(MUX_2_GND);
    system_init(); //系统初始化
    delay(2); //延时 2mS
    ///显示函数
    for(int i=0; i < 4; i++)
    {
        do_dlog(func, i, t[i],POWER_MILLI, fail_bin); // “i” 表示显示的顺序,
“t[i]” 表示要输出的变量值, “POWER_MILLI” 表示量程档位为 1e-3, “fail_bin”
表示错误返回值, 简称软件 BIN
    }

```

### (3) 波形

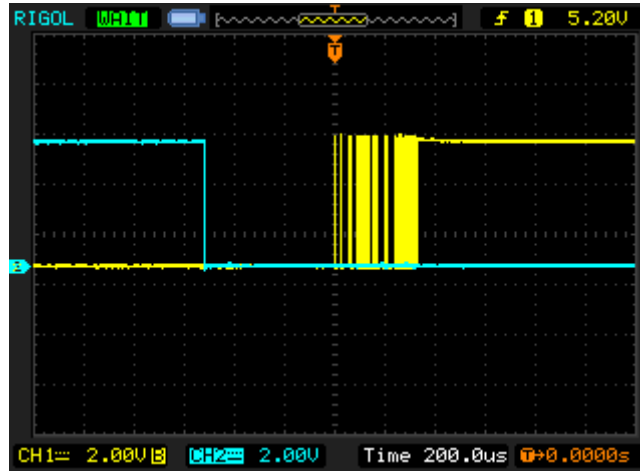


图 4.2 继电器吸合波形图

如上图 4.2 所示，黄色表笔接 Tp1 继电器的常开位，蓝色表笔接 Tp2 继电器的常闭位，测试继电器吸合时间，波形中可以看出 S1 的吸合时间约为 0.6mS，符合测试值。

### (4) 数据

12/28/13 16:33:23 Serial Number: 207		Total Devices: 207		PASSED BIN: 1				
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.04.01	QTMU_6	T1	0.5801	1	mS	none	2.0000	
1.04.02	QTMU_6	T2	0.8159	1	mS	none	2.0000	
1.04.03	QTMU_6	T3	0.9090	1	mS	none	2.0000	
1.04.04	QTMU_6	T4	1.0483	1	mS	none	2.0000	
28.882 mSecs Total=28.882 mSec								

图 4.3 测试数据

如上图 4.3 显示，S1, S2, S3, S4 的吸合时间分别为 0.5801mS, 0.8159mS, 0.9090mS, 1.0483mS，值在 SPEC 内。

## 五. DDD\_7

(1) 示例线路如下图 5.1:

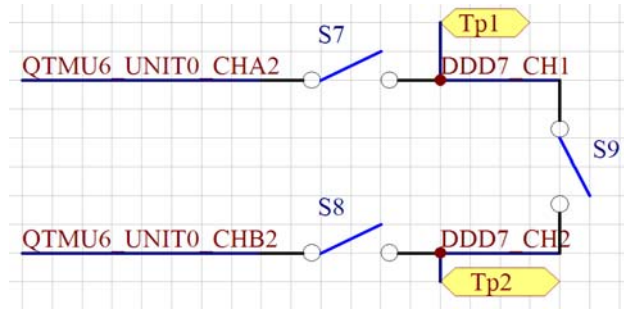


图 5.1 DDD 测试线路图

DDD 主要用来输出数字信号，线路图 5.1 中，把 DDD\_CH1，DDD\_CH2 通过继电器与 QTMU 的 CHA2，CHB2 相连，通过控制 S7，S8 的通断就可以利用 QTMU 来测量 DDD 输出波形信号的频率。

(2) 示例程序

```
float t1,t2,fail_bin=9;
float fre=1e6;
system_init();//系统初始化
delay(2);//延时 2mS
mux20_close();//MUX20 初始化
delay(2);//延时 2mS
ddd_7->ddd_set_clock_freq(fre);//设置时钟频率
ddd_7->ddd_set_voltage_ref(2.5f);//设置参考电平为 2.5V
ddd_7->ddd_set_hi_level(5.0);//设置高电平为 5V
ddd_7->ddd_set_lo_level(0.0);//设置低电平为 0V
ddd_7->ddd_set_delay(1, 5.0e-9f,0.5*(1/fre));//设置向量延时
ddd_7->ddd_load_pattern(0,9,"XXXXXX11","XXXXXX00");//在指定的向量
地址装载数据
ddd_7->ddd_load_pattern(10,11,"XXXXXXXX","XXXXXXXX");
ddd_7->ddd_end_pattern(12);//完成一定数目向量实行后，使连续循环模式跳
到 0 地址向量
delay(1); //延时 1mS
mux_20->close_relay(MUX_2_1); //闭合 MUX20-2-1 继电器
mux_20->open_relay(MUX_2_2); //断开 MUX20-2-2 继电器
delay(2); //延时 2mS
```

```

    qtmu1_6->qtmu1_close_relay(QTMU1_UNIT_0,QTMU1_CHAN_A_DUT2);
//闭合 QTMU 所用通道的继电器
    qtmu1_6->qtmu1_close_relay(QTMU1_UNIT_0,QTMU1_CHAN_B_DUT2);
    qtmu1_6->qtmu1_start_trigger_setup(QTMU1_UNIT_0,2.5f,QTMU1_POS_SL
OPE,QTMU1_CHAN_A,QTMU1_IN_5V);//
    qtmu1_6->qtmu1_stop_trigger_setup(QTMU1_UNIT_0,2.5f,QTMU1_POS_SL
OPE,QTMU1_CHAN_A,QTMU1_IN_5V);
    delay(2);
    qtmu1_6->qtmu1_arm(QTMU1_UNIT_0);
    delay(1);
    ddd_7->ddd_run_pattern(0,9);//在开始地址和结束地址之间运行
    t1=qtmu1_6->qtmu1_read(QTMU1_UNIT_0);
    delay(10);
    mux_20->close_relay(MUX_2_2);
    delay(2);
    qtmu1_6->qtmu1_close_relay(QTMU1_UNIT_0,QTMU1_CHAN_A_DUT2);
    qtmu1_6->qtmu1_close_relay(QTMU1_UNIT_0,QTMU1_CHAN_B_DUT2);
    qtmu1_6->qtmu1_start_trigger_setup(QTMU1_UNIT_0,2.5f,QTMU1_POS_SL
OPE,QTMU1_CHAN_B,QTMU1_IN_5V);
    qtmu1_6->qtmu1_stop_trigger_setup(QTMU1_UNIT_0,2.5f,QTMU1_POS_SL
OPE,QTMU1_CHAN_B,QTMU1_IN_5V);
    delay(2);
    qtmu1_6->qtmu1_arm(QTMU1_UNIT_0);
    delay(1);
    ddd_7->ddd_run_pattern(0,11);//在开始地址和结束地址之间运行
    t2=qtmu1_6->qtmu1_read(QTMU1_UNIT_0);
    delay(10);//延时 10mS
    qtmu1_6->qtmu1_open_relay(QTMU1_UNIT_0,QTMU1_CHAN_A_DUT2);
//断开 QTMU 所用的继电器
    qtmu1_6->qtmu1_open_relay(QTMU1_UNIT_0,QTMU1_CHAN_B_DUT2);
    mux_20->open_relay(MUX_2_1);
    mux_20->open_relay(MUX_2_2);
    delay(2);//延时 2mS
    mux20_close();//MUX20 初始化
    delay(2);//延时 2mS
    system_init();//系统初始化
    delay(2); //延时 2mS
    //显示函数
    do_dlog(func,0,t1,POWER_MICRO,fail_bin);// “0” 表示显示的顺序,“t1”
表示要输出的变量值,“POWER_MICRO” 表示量程档位为 1e-6,“fail_bin” 表
示错误返回值,简称软件 BIN
    do_dlog(func,1,t2,POWER_MICRO,fail_bin);

```



### (3) 波形

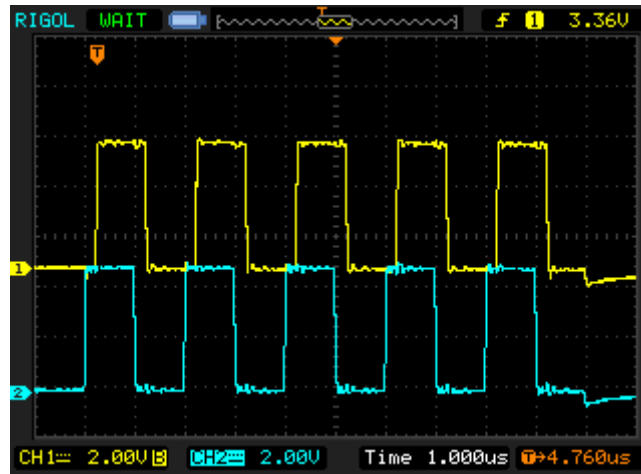


图 5.2 DDD 波形图

如上图 5.2 所示,黄色为 Tp1 处 DDD\_CH1 的波形,蓝色为 Tp2 处 DDD\_CH2 的波形,程序中 DDD 打的波形周期为 2uS,从波形中可以看到周期为 2uS 与理论值相符。

### (4) 数据

12/30/13 08:10:38 Serial Number: 13			Total Devices: 13			PASSED BIN: 1		
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.05.01	DDD_7	T_DDD1	1.9828	1	uS	1.9000	2.1000	
1.05.02	DDD_7	T_DDD2	1.9919	1	uS	1.9000	2.1000	
45.439 mSecs Total=45.439 mSec								

图 5.3 测试数据

如上图 5.3 显示,DDD\_CH1, DDD\_CH2 的周期分别为 1.9828uS,1.9919uS,与波形和理论值都相符。

## 六. DOAL\_8

(1) 示例线路如下图 6.1:

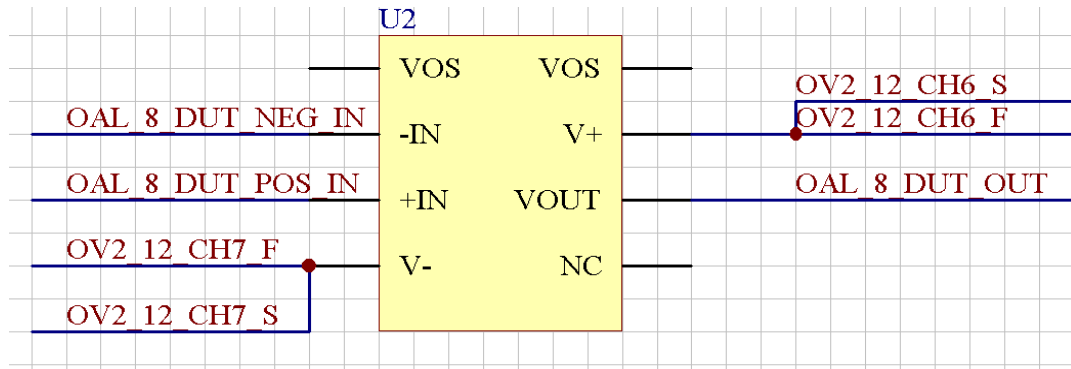


图 6.1 DOAL 测试线路图

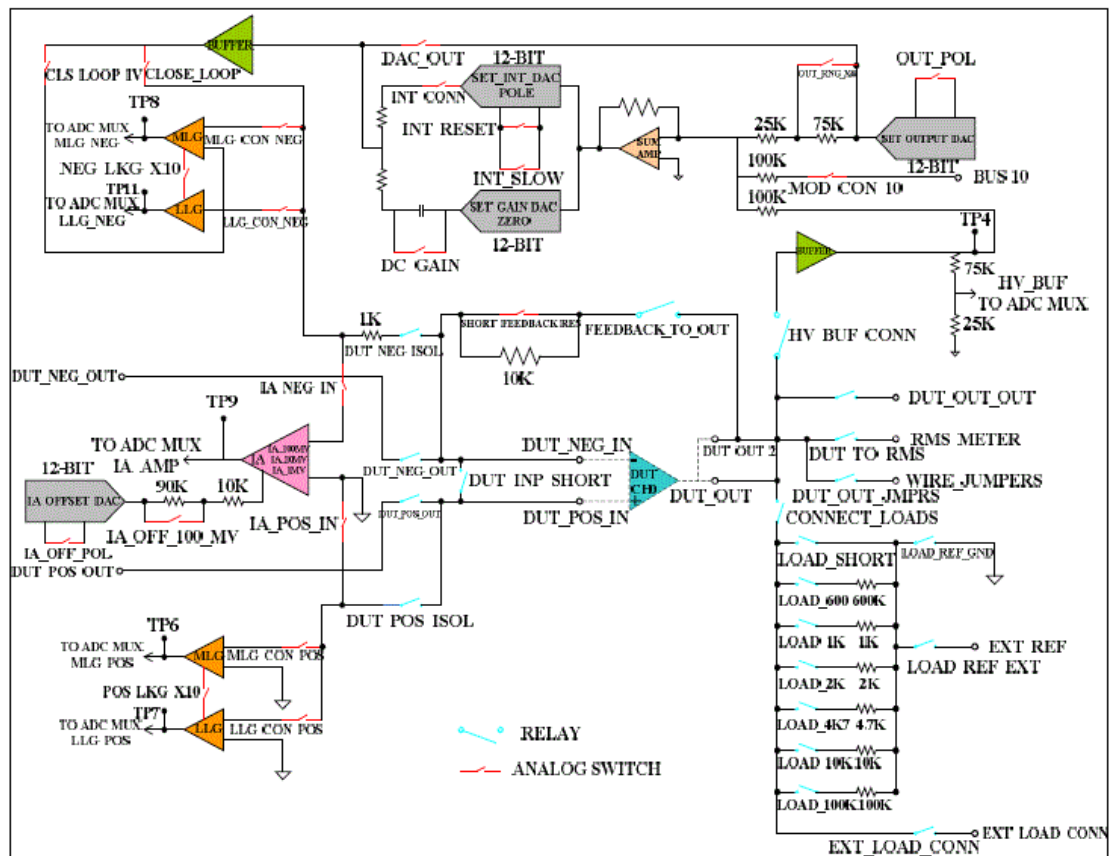


图 6.2 DOAL 原理图

如上图 6.1 所示，本测试案例是测量单运放 op07，故以 doal 功能板（doal 有 ch0 与 ch1 两个运放环路）的 ch0 环路为例进行阐述。

**输入失调电压**的测量过程：先空测运放反向输入端的电压，然后用 OV2\_12\_CH6， OV2\_12\_CH7 在电源端施加规定电压，再通过 doal 内部函数设置零极点使环路稳定，再设置运放输出端规定电压，并测量运放反相输入的电压，此时运放反相输入端的电压减去空测时的电压就是输入失调电压。

**输入失调电流和输入偏置电流**的测量过程：设置零极点使环路稳定，设置运放输出端规定电压，测量运放正反相输入的电流，此时运放正反相输入端的电流之差就是输入失调电流，此时运放正反相输入端的电流就是输入偏置电流。

**开环电压增益**的测试过程：先空测运放反向输入端的电压，然后用 OV2\_12\_CH6， OV2\_12\_CH7 在电源端施加规定电压，再设置零极点使环路稳定，然后通过内部函数设置运放输出端规定电压（两个），并用 doal 的设备内部 IA 测量运放反相输入的电压，那么输出两个规定电压之差除以出两个规定电压所对应的反向输入端电压之差的值，即是运放开环电压增益。

**共模抑制比**测试方法：用 OV2\_12\_CH6， OV2\_12\_CH7 在电源端施加的值是规定电压减输入共模电压，再设置零极点使环路稳定，然后通过内部函数设置运放输出端的值为规定电压（两个）减去输入共摸电压，并用 doal 的设备内部 IA 测量运放反相输入的电压，那么输入共模电压的值除以出两个规定电压所对应的反向输入端电压之差的值，及是共模抑制比。

**电源抑制比**测试过程：用 OV2\_12\_CH6， OV2\_12\_CH7 在电源端施加规定电压（两个），再通过 doal 内部函数 s 设置零极点使环路稳定，然后通过内部函数，设置运放输出端规定电压，并用 doal 的设备内部 IA 测量运放反相输入的电压，那么输出两个规定电源电压之差除以出两个规定电源电压所对应的反向输入端电压之差的值，及是电压抑制比。

## (2) 示例程序

```
////////////////////////////////////测量输入失调电压////////////////////////////////////  
//init  
float range=0.0f,fail_bin=10;//变量声明  
int adc_val[1]={0},adc_cal[1]={0};  
float vio_5V[1]={0};  
int i=0;  
system_init();//系统初始化  
ovi_1->disconnect(OVI_CHANNEL_4);//断开不需要用到的源  
ovi_1->disconnect(OVI_CHANNEL_5);  
ovi_1->disconnect(OVI_CHANNEL_6);  
delay(2);//延时2mS  
ov2_12->set_voltage(OV2_CHANNEL_6, 0.0f); //V+端初始化  
ov2_12->set_current(OV2_CHANNEL_6, 1e-9f);  
ov2_12->set_voltage(OV2_CHANNEL_7, 0.0f); //V-端初始化  
ov2_12->set_current(OV2_CHANNEL_7, 1e-9f);  
delay_us(10);  
range = select_range(150e-6f);//得到电压对应的值  
connect_out(FALSE,FALSE,FALSE);//不加负载,不输出  
connect_short_vio(); //输出短路连接
```

```

delay(3);
measure_ia(adc_cal,100,500);//测量得到第一组短路数据，空测的电压
oal_8->open_relay(DUT_INP_SHORT);//打开内部继电器
delay(1);
ov2_12->set_voltage(OV2_CHANNEL_6, 15.0f); //v+端加+15V电压
ov2_12->set_current(OV2_CHANNEL_6, 200e-3f);
ov2_12->set_voltage(OV2_CHANNEL_7, -15.0f); //V-端加-15V电压
ov2_12->set_current(OV2_CHANNEL_7, 200e-3f);
delay(10);
oal_8->close_switch(CLOSE_LOOP); //接通准备补偿校准
delay(5);
set_compensation(ours->pole_dac, ours->zero_dac);//设置零极点使环路稳定
delay(5);
oal_8->open_switch(INT_RESET);
delay(2);
oal_8->set_output_voltage(0.0f); //设置运放输出端规定电压
delay(3);
measure_ia(adc_val,100,1000);//得到补偿校准后的第二组数据，反向输入端
的电压
ov2_12->set_voltage(OV2_CHANNEL_6, 0.0f); //V+端掉电
ov2_12->set_current(OV2_CHANNEL_6, 1e-9f);
ov2_12->set_voltage(OV2_CHANNEL_7, 0.0f); //V-端掉电
ov2_12->set_current(OV2_CHANNEL_7, 1e-9f);
delay(4);
vio_5V[0] = -(adc_val[0]-adc_cal[0]) / range; //反向输入端的电压减去空测
的电压就是输入失调电压
system_init();
delay(2);
//////////测量开环增益，共模抑制比，电源抑制比//////////
//init
int tests=0;
int temp_val[1]={0};
int adc_val1[1][6]={0};
float gv1=0, psr1=0, cmr1=0, inp_change1=0;
float Vcm=0;
range=0;
i=0;
ovi_1->disconnect(OVI_CHANNEL_4);
ovi_1->disconnect(OVI_CHANNEL_5);
ovi_1->disconnect(OVI_CHANNEL_6);
delay(2);
ov2_12->set_voltage(OV2_CHANNEL_6, 0.0f);
ov2_12->set_current(OV2_CHANNEL_6, 1e-9f);
ov2_12->set_voltage(OV2_CHANNEL_7, 0.0f);

```

```

ov2_12->set_current(OV2_CHANNEL_7, 1e-9f);
delay_us(10);
range = select_range(150.0e-6f);
connect_out(TRUE,FALSE,FALSE); //connect load
connect_ia_in();//select IA
delay(1);
set_compensation(ours->pole_dac_avd, ours->zero_dac_avd);//准备补偿，设置0
极点使环路稳定
delay(1);
ov2_12->set_current(OV2_CHANNEL_6, 0.2f);
ov2_12->set_voltage(OV2_CHANNEL_6, 15.0f);
ov2_12->set_current(OV2_CHANNEL_7, 0.2f);
ov2_12->set_voltage(OV2_CHANNEL_7, -15.0f);
oal_8->open_switch(INT_RESET); //enable
oal_8->set_output_voltage(0); //设置运放输出端规定电压
delay(2);
for(tests=0; tests<6; tests++)
{
    switch(tests)
    {
        case 0:
            oal_8->close_relay(LOAD_2K);
            oal_8->set_output_voltage( -10.0f ); //设置输出电压为-10V
            break;
        case 1:
            oal_8->set_output_voltage( 10.0f ); //设置输出电压为10V
            break;
        case 2:
            oal_8->open_relay(CONNECT_LOADS); //打开内部继电器
            delay(1);
            ov2_12->set_voltage(OV2_CHANNEL_6, 3.0f); //V+设置为3V

            ov2_12->set_voltage(OV2_CHANNEL_7 ,-3.0f); //V-设置为-3V

            oal_8->set_output_voltage(0.0f); //设置输出电压为0V
            break;
        case 3:
            ov2_12->set_voltage(OV2_CHANNEL_6, 18.0f); //V+设置为18V
            ov2_12->set_voltage(OV2_CHANNEL_7, -18.0f); //V-设置为-18V

            break;
        case 4:
            Vcm=-13.0f;
            ov2_12->set_voltage(OV2_CHANNEL_6, 15.0f-Vcm );

```

```

        ov2_12->set_voltage(OV2_CHANNEL_7, -15.0f-Vcm);
        delay(10);
        oal_8->set_output_voltage(0.0f-Vcm);
        break;
    case 5:
        Vcm=13.0f;
        ov2_12->set_voltage(OV2_CHANNEL_6, 15.0f-Vcm);
        ov2_12->set_voltage(OV2_CHANNEL_7, -15.0f-Vcm);
        delay(10);
        oal_8->set_output_voltage(0.0f-Vcm);
        break;
    }
    delay(4);
    measure_ia(temp_val,100,1000); //测得数据
    adc_val1[0][tests] = temp_val[0]; //A
}

//discharge
ov2_12->set_voltage(OV2_CHANNEL_6, 0.0f); //掉电
ov2_12->set_current(OV2_CHANNEL_6, 1e-9f);
ov2_12->set_voltage(OV2_CHANNEL_7, 0.0f);
ov2_12->set_current(OV2_CHANNEL_7, 1e-9f);
delay(1);

for(i=0; i<1; i++) //计算GV
{
    inp_change1=0;
    inp_change1 = adc_val1[i][1] - adc_val1[i][0];
    inp_change1 = fabs(inp_change1 / range); // value in V

    gv1 = 20.0 * log10(20.0 / inp_change1);
    if(inp_change1 == 0.0) gv1 = 160.0;
}

for(i=0; i<1; i++) //计算CMR
{
    inp_change1=0;
    inp_change1 = adc_val1[i][5] - adc_val1[i][4];
    inp_change1 = fabs(inp_change1 / range);

    cmr1 = 20.0 * log10(26 / inp_change1);
    if(inp_change1 == 0.0) cmr1 = 160.0;
}

```

```

for(i=0; i<1; i++) //计算PSR
{
    inp_change1=0;
    inp_change1 = adc_val1[i][3] - adc_val1[i][2];
    inp_change1 = fabs(inp_change1 / range);

    psr1 = 20.0 * log10(15.0 / inp_change1);
    if(inp_change1 == 0.0) psr1 = 160.0;
}
system_init();
delay(2);

```

//////////////////////////////////// 测量输入失调电流，偏置电流  
////////////////////////////////////

```

char lkg_type, lkg_x10;
float ib_val[2]={0};
range=0;
Vcm = 0.0f;
i=0;
ovi_1->disconnect(OVI_CHANNEL_4);
ovi_1->disconnect(OVI_CHANNEL_5);
ovi_1->disconnect(OVI_CHANNEL_6);
delay(2);
ov2_12->set_voltage(OV2_CHANNEL_6, 0.0f);
ov2_12->set_current(OV2_CHANNEL_6, 1e-9f);
ov2_12->set_voltage(OV2_CHANNEL_7, 0.0f);
ov2_12->set_current(OV2_CHANNEL_7, 1e-9f);
delay(10);

```

range = select\_lg\_range(15e-9f,lkg\_type, lkg\_x10); //选择量程，得到模式，  
得到是否放大

```

connect_out(FALSE,FALSE,FALSE); //不接负载，不输出
connect_lg_in(lkg_type, lkg_x10, 0.0f); //根据模式，是否放大，输出电压
set_compensation(ours->pole_dac_i, ours->zero_dac_i); //补偿校准
ov2_12->set_voltage(OV2_CHANNEL_6, 15.0f); //V+
ov2_12->set_current(OV2_CHANNEL_6, 200e-3f);
ov2_12->set_voltage(OV2_CHANNEL_7, -15.0f); //V-
ov2_12->set_current(OV2_CHANNEL_7, 200e-3f);
delay(2);
oal_8->open_switch(INT_RESET); //enable
delay(2); //加适当延时
measure_lkg(ib_val, lkg_type, 300,1000); //根据模式测量得到电流值
ov2_12->set_voltage(OV2_CHANNEL_6, 0.0f); //掉电

```

```

ov2_12->set_current(OV2_CHANNEL_6, 1e-9f);
ov2_12->set_voltage(OV2_CHANNEL_7, 0.0f);
ov2_12->set_current(OV2_CHANNEL_7,1e-9f);
delay(1);
float A_N=0,A_P=0,IA_O=0;
for (i=0; i<2; i++)
{
    ib_val[i] = -ib_val[i]/range;      // value in A
}
A_N=ib_val[0];
A_P=ib_val[1];
IA_O=A_P-A_N;//输入失调电流就是输出电压为零（或规定值）时，流入两
输入端的电流之差。
system_init();
delay(1);

//////////////////////////////////数据显示//////////////////////////////////
do_dlog(func, 0, vio_5V[0],POWER_MICRO,fail_bin);//VOS
do_dlog(func, 1, A_P,POWER_NANO,fail_bin); //Ib
do_dlog(func, 2, A_N,POWER_NANO,fail_bin); //Ib
do_dlog(func, 3, IA_O,POWER_NANO,fail_bin); //Iio
do_dlog(func, 4,gv1,POWER_UNIT, fail_bin); //GV
do_dlog(func, 5, cmr1,POWER_UNIT, fail_bin); //CMR
do_dlog(func, 6,psr1,POWER_UNIT, fail_bin); //PSR

```

### (3) 数据

12/28/13 16:35:56 Serial Number: 234 Total Devices: 234 PASSED BIN: 1								
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.06.01	DOAL_8	vio	67.8711	1	uV	none	150.0000	
1.06.02	DOAL_8	ib_pos	-0.5493	1	nA	-12.0000	12.0000	
1.06.03	DOAL_8	ib_neg	-0.7446	1	nA	-12.0000	12.0000	
1.06.04	DOAL_8	I_SETOFF	0.1953	1	nA	none	6.0000	
1.06.05	DOAL_8	GV	105.5230	1	dB	100.0000	none	
1.06.06	DOAL_8	PSRR	124.0576	1	dB	90.0000	none	
1.06.07	DOAL_8	CMRR	123.1503	1	dB	94.0000	none	
235.245 mSecs Total=235.245 mSec								

图 6.3 测试数据

如上图 6.3 显示，测量所得的  $vio=67.8711\mu V$ ， $ib\_pos=-0.5493nA$ ， $ib\_neg=-0.7446nA$ ， $I\_SHTOFF=0.1953nA$ ， $GV=105.5230dB$ ， $PSRR=124.0576dB$ ， $CMRR=123.1503dB$ 。



## 七. DVI\_9

(1) 示例线路如下图 7.1:

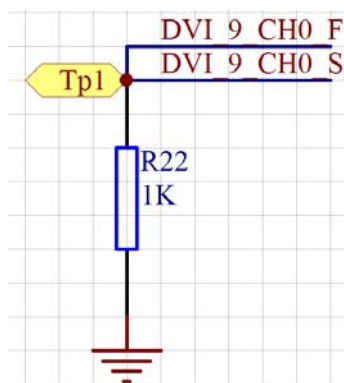


图 7.1 DVI 测试线路图

如图 7.1 所示，将 1K 电阻的一端接地，一端接 DVI9 的 0 通道，DVI9 通道 0 加电压来测量流过电阻的电流（也可加电流测电压），再根据欧姆定律算出电阻阻值。

(2) 示例程序

```
float r,v,i,fail_bin=11;//变量声明
system_init();//系统初始化
delay(1);//延时 1mS
dvi_9->set_voltage(DVI_CHANNEL_0,1.0f,RANGE_10_V); //设置 DVI 的通道
0 电压至 1V
dvi_9->set_current(DVI_CHANNEL_0,1.5e-3f,RANGE_3_MA);//设置 DVI 的
通道 0 电流至 1.5MA
dvi_9->set_meas_mode(DVI_CHANNEL_0,DVI_MEASURE_VOLTAGE);//测
量 DVI0 通道的电压
delay(2);//延时 2mS
v=dvi_9->measure_average(10);//返回 ADC 采样 10 次的平均值
dvi_9->set_meas_mode(DVI_CHANNEL_0,DVI_MEASURE_CURRENT);//测
量 DVI0 通道的电流
delay(2);
i=dvi_9->measure_average(10);
r=v/i;
dvi_9->set_voltage(DVI_CHANNEL_0,0.0f,RANGE_1_V); //用到的源置 0,
掉电
```

```

dvi_9->set_current(DVI_CHANNEL_0,1e-3f,RANGE_3_MA);
delay(2);
system_init();//系统初始化
delay(2);//延时 2mS
/////显示函数/////
do_dlog(func,0,v,POWER_UNIT,fail_bin);// “0”表示显示的顺序，“v”表示
要输出的变量值，“POWER_UNIT”表示量程档位为 1，“fail_bin”表示错误返
回值，简称软件 BIN
do_dlog(func,1,i,POWER_MILLI,fail_bin);
do_dlog(func,2,r,POWER_UNIT,fail_bin);

```

### (3) 波形

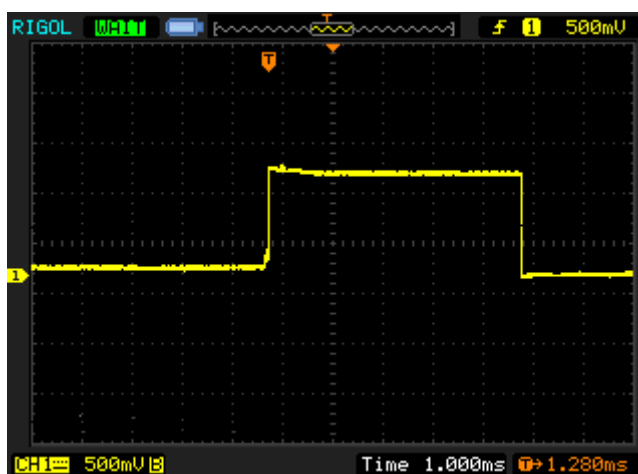


图 7.2 Tp1 处波形图

如上图 7.2 所示，选用 1K 的电阻程序中给 1V 的电压，测电流，波形中电压达到了 1V，只需测量此时的电流就可计算出电阻的阻值，这种测试方法叫做加压测流。

### (4) 数据

12/28/13 16:41:05 Serial Number: 290		Total Devices: 290		PASSED BIN: 1				
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.07.01	DVI_9	V	1.0013	1	V	none	none	
1.07.02	DVI_9	I	0.9959	1	mA	none	none	
1.07.03	DVI_9	R	1005.3962	1	ohm	990.0000	1010.0000	
12.864 mSecs Total=12.864 mSec								

图 7.3 测试数据

如上图 7.3 显示，测量所得的电压电流，计算所得的电阻分别为 1.0013V，

0.9959mA, 1005.3962 欧姆。

## 八. DVI\_11

(1) 示例线路如下图 8.1:

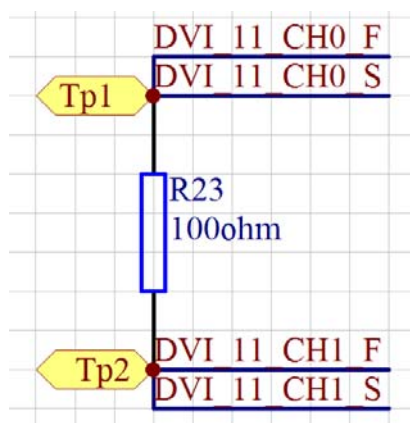


图 8.1 DVI 测试线路图

DVI\_11 是 DVI 的 2000 版本, 和 300 板相比, 电流的量程到 2A, 并且加入了差分测量功能。

如上图 8.1 所示, 将 100 欧姆电阻的两端分别接 DVI 的 0 通道和 1 通道, 给定适当的电流, 利用差分的方法来测电阻两端的压差, 再根据欧姆定律算出阻值。

(2) 示例程序

```
float r,v,i,fail_bin=12;//变量声明
system_init();//系统初始化
delay(5);
dvi_11->set_voltage(DVI_CHANNEL_0,0.0f,RANGE_20_V); //掉电
dvi_11->set_current(DVI_CHANNEL_0,30e-3f,RANGE_200_MA);
dvi_11->set_voltage(DVI_CHANNEL_1,0.0f,RANGE_10_V);
dvi_11->set_current(DVI_CHANNEL_1,30e-3f,RANGE_200_MA);
delay(2);//延时 2mS
i=100e-3f;
dvi_11->set_voltage(DVI_CHANNEL_0,20.0f,RANGE_20_V); //设置 0 通道的
电压
dvi_11->set_current(DVI_CHANNEL_0,i,RANGE_200_MA);//设置 0 通道的电
流
dvi_11->set_voltage(DVI_CHANNEL_1,10.0f,RANGE_20_V); //设置 1 通道的
```

电压

```
dvi_11->set_current(DVI_CHANNEL_1,i,RANGE_200_MA);//设置 1 通道的电
流
dvi_11->set_meas_mode(DVI_CHANNEL_0,DVI_MEASURE_DIFF);// 设置测
量，模式为差分，测量通道为 0 通道
dvi_11->set_diff_range(RANGE_10_V);//设置差分测量的档位
delay(5);//延时 2mS
v=dvi_11->measure_average(10);//返回 10 次 ADC 采样的平均值
r=v/i;
dvi_11->set_voltage(DVI_CHANNEL_0,0.0f,RANGE_20_V); //掉电
dvi_11->set_current(DVI_CHANNEL_0,30e-3f,RANGE_200_MA);
dvi_11->set_voltage(DVI_CHANNEL_1,0.0f,RANGE_10_V);
dvi_11->set_current(DVI_CHANNEL_1,30e-3f,RANGE_200_MA);
delay(2);//延时 2mS
system_init();//系统初始化
delay(5);
/////显示函数
do_dlog(func,0,v,POWER_UNIT,fail_bin);// “0” 表示显示的顺序，“v” 表示
要输出的变量值，“POWER_UNIT” 表示量程档位为 1，“fail_bin” 表示错误返
回值，简称软件 BIN
do_dlog(func,1,i,POWER_MILLI,fail_bin);
do_dlog(func,2,r,POWER_UNIT,fail_bin);
```

### (3) 波形

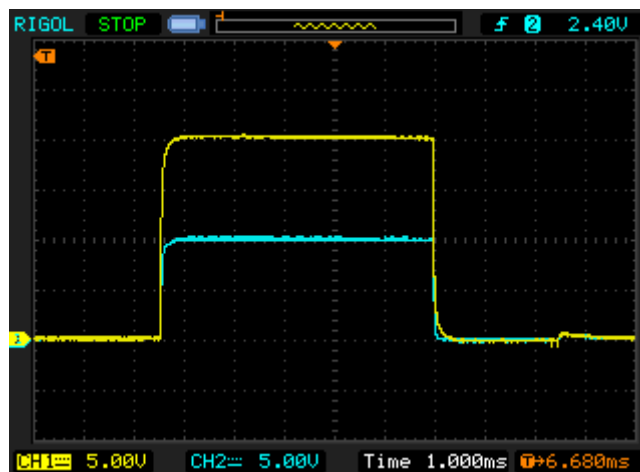


图 8.2 电阻两端波形图

如上图 8.2 所示，黄表笔接 Tp1，蓝表笔接 Tp2 处，程序中给 100mA 电流，如图波形中可以看到 Tp1 和 Tp2 之间的差值，即为电阻两端电压差，根据欧姆定律  $R = V / I = 10V / 100MA = 100$  欧姆，符合理论分析。

#### (4) 数据

12/28/13 16:51:49 Serial Number: 318		Total Devices: 318		PASSED BIN: 1				
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.08.01	DVI2_11	V	9.9529	1	V	none	none	
1.08.02	DVI2_11	I	100.0000	1	mA	none	none	
1.08.03	DVI2_11	R	99.5289	1	ohm	90.0000	110.0000	

12.009 mSecs Total=12.009 mSec

图 8.3 测试数据

如上图 8.3 显示，设定的电流值为 100mA，测量所得的电压为 9.9529V，计算所得的电阻为 99.5289 欧姆。

## 九. OV2\_12

(1) 示例线路如下图 9.1 和 9.2:

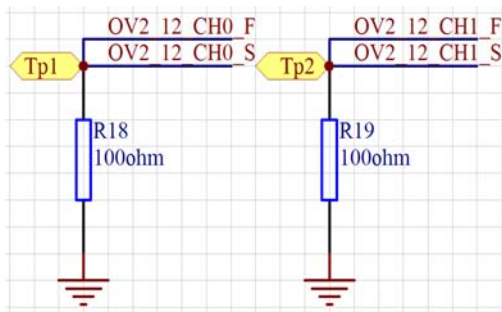


图 9.1 并测

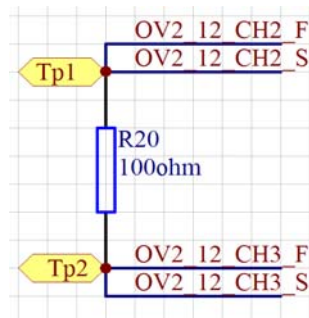


图 9.2 差分

OV2 为 OVI 的升级版，线路图 9,1 中 R20 的两端分别接 OV2 的 2 通道与 3 通道，给定电流，测 R20 两端的压差，这是 OV2 的差分测量方法。线路图 9.2 中，R18 与 R19 的一端接地，另外一端分别接 OV2 的 0 通道和 1 通道是使用了 OV2 含有两个 ADC,可以两路并测的能力，两路同时测，共用一个延时，可以节省测试时间。

(2) 示例程序

```
float v1,i1,r1,r2,r3,fail_bin=13;
system_init();//系统初始化
delay(1);//延时 1mS
```

```

//OV2 差分
i1=50e-3f;
ov2_12->set_voltage(OV2_CHANNEL_2,10.0f,RANGE_10_V); //设置通道 2
的电压
ov2_12->set_current(OV2_CHANNEL_2,i1,RANGE_50_MA); //设置电流
ov2_12->set_voltage(OV2_CHANNEL_3,0.0f,RANGE_10_V); //设置通道 3 的
电压
ov2_12->set_current(OV2_CHANNEL_3,i1,RANGE_50_MA);
ov2_12->set_meas_mode_diff_voltage(OV2_CHANNEL_2,OV2_CHANNEL_3,
RANGE_25_V); //选择差分测量的通道与量程
delay(2); //延时 2mS
v1=ov2_12->measure_average(10); //采样 10 次取平均数
r1=v1/i1;
//OV2 并测
ov2_12->set_voltage(OV2_CHANNEL_0,10.0f,RANGE_10_V);
ov2_12->set_current(OV2_CHANNEL_0,100e-3f,RANGE_500_MA);
ov2_12->set_meas_mode(OV2_CHANNEL_0,OVI_MEASURE_VOLTAGE); //
两个 meas, 同时测
ov2_12->set_voltage(OV2_CHANNEL_1,10.0f,RANGE_10_V);
ov2_12->set_current(OV2_CHANNEL_1,100e-3f,RANGE_500_MA);
ov2_12->set_meas2_mode(OV2_CHANNEL_0,OVI_MEASURE_VOLTAGE);
//两个 meas, 同时测
delay(2); //共用一个延时
r2=ov2_12->measure_average(10); //返回 ADC 采样 10 次的平均数
r2=r2/100e-3f;
r3=ov2_12->measure2_average(10); //返回 ADC2 采样 10 次的平均值
r3=r3/100e-3f;
ov2_12->set_voltage(OV2_CHANNEL_0,0.0f,RANGE_5_V); //所有用到的
源置 0, 掉电
ov2_12->set_current(OV2_CHANNEL_0,50e-3f,RANGE_50_MA);
ov2_12->set_voltage(OV2_CHANNEL_1,0.0f,RANGE_5_V);
ov2_12->set_current(OV2_CHANNEL_1,50e-3f,RANGE_50_MA);
ov2_12->set_voltage(OV2_CHANNEL_2,0.0f,RANGE_5_V);
ov2_12->set_current(OV2_CHANNEL_2,50e-3f,RANGE_50_MA);
ov2_12->set_voltage(OV2_CHANNEL_3,0.0f,RANGE_5_V);
ov2_12->set_current(OV2_CHANNEL_3,50e-3f,RANGE_50_MA);
delay(2);
system_init(); //系统初始化
delay(2); //延时 2mS
/////显示函数/////
do_dlog(func,0,v1,POWER_UNIT,fail_bin); // “0” 表示显示的顺序, “v1” 表
示要输出的变量值, “POWER_UNIT” 表示量程档位为 1, “fail_bin” 表示错误
返回值, 简称软件 BIN
do_dlog(func,1,i1,POWER_MILLI,fail_bin);

```

```
do_dlog(func,2,r1,POWER_UNIT,fail_bin);
do_dlog(func,3,r2,POWER_UNIT,fail_bin);
do_dlog(func,4,r3,POWER_UNIT,fail_bin);
```

### (3) 波形

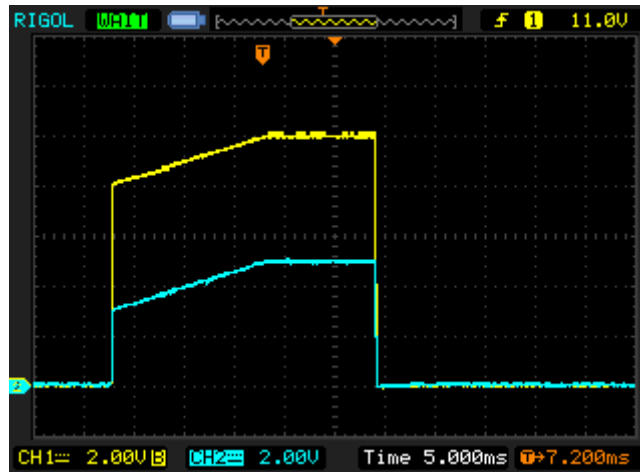


图 9.3 差分波形图

如上图 9.3 所示，黄表笔接 Tp1，蓝表笔接 Tp2，选用 100 欧姆电阻，程序给 50MA 的电流，从波形中可以看到 Tp1 和 Tp2 之间的差值，即为电阻两端电压差，根据欧姆定律  $R = V / I = 5V / 50MA = 100$  欧姆，符合理论分析。

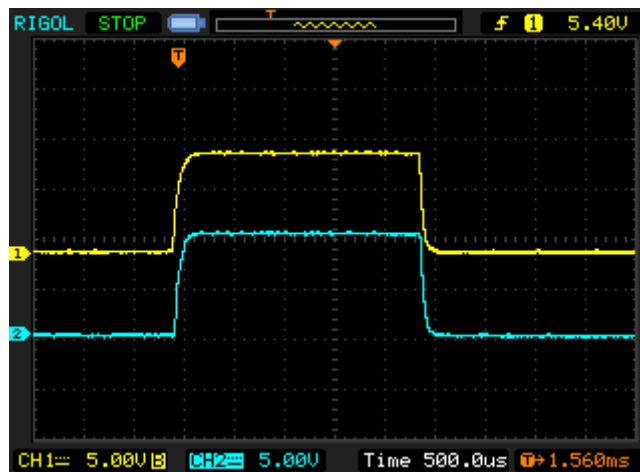


图 9.4 并测波形图

如上图 9.4 所示，黄表笔接 Tp1，蓝表笔接 Tp2，从波形中可以看到，R18 和 R19 同时给源同时掉电。

#### (4) 数据

12/28/13 16:57:56 Serial Number: 367 Total Devices: 367 PASSED BIN: 1								
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.09.01	ov2_12	V1	5.0002	1	V	none	none	
1.09.02	ov2_12	I1	50.0000	1	mA	none	none	
1.09.03	ov2_12	R1	100.0037	1	ohm	90.0000	110.0000	
1.09.04	ov2_12	R2	99.3629	1	ohm	90.0000	110.0000	
1.09.05	ov2_12	R3	99.3831	1	ohm	90.0000	110.0000	
34.238 mSecs Total=34.238 mSec								

图 9.5 测试数据

如上图 9.5 显示，差分测得的电压值为 5.002V，给定的电流值为 50mA，计算所得的 R1 为 100.0037 欧姆，R2，R3 分别为 99.3692 欧姆，99.3831 欧姆。

## 十. DVI3\_13

(1) 示例线路如下图 10.1:

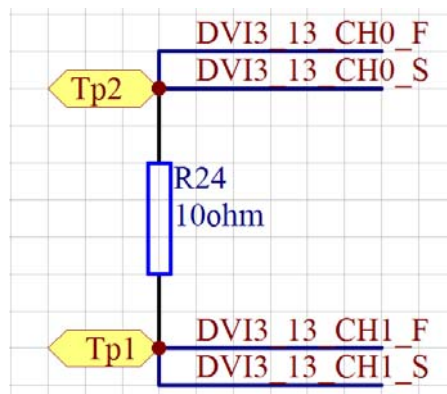


图 10.1 DVI 测试线路图

如上图 10.1 所示，将 10 欧姆电阻的两端分别接 dvi3 的 0 通道和 1 通道，可以使用 dvi3 的差分功能，dvi3 的最大电流为 20A，为了体现这一特点，选用阻值较小的电阻，通 1A 的电流，测电阻两端的压差，程序写法与 DVI11 类似。

(2) 示例程序

```
float r,v,i,fail_bin=14;//声明变量
system_init();//系统初始化
delay(2);//延时 1mS
```



```

dvi_13->set_voltage(DVI_CHANNEL_0,0.0f,RANGE_20_V); //源置 0
dvi_13->set_current(DVI_CHANNEL_0,100e-3f,AMP_2);
dvi_13->set_voltage(DVI_CHANNEL_1,0.0f,RANGE_20_V);
dvi_13->set_current(DVI_CHANNEL_1,100e-3f,AMP_2);
delay(2);
i=1;//设定电流
dvi_13->set_voltage(DVI_CHANNEL_0,20.0f,RANGE_20_V); //设置 0 通道的
电压值
dvi_13->set_current(DVI_CHANNEL_0,i,AMP_2);//DVI3 的最大档位支持
20A
dvi_13->set_voltage(DVI_CHANNEL_1,10.0f,RANGE_10_V);
dvi_13->set_current(DVI_CHANNEL_1,i,AMP_2);
dvi_13->set_meas_mode(DVI_CHANNEL_0,DVI_MEASURE_DIFF);//选择差
分测量模式
dvi_13->set_diff_range(RANGE_10_V);//选择差分测量的档位
delay(2);//延时 2mS
v=dvi_13->measure_average(10);//返回 10 次 ADC 采样的平均值
r=v/i;
dvi_13->set_voltage(DVI_CHANNEL_0,0.0f,RANGE_20_V); //源置 0
dvi_13->set_current(DVI_CHANNEL_0,100e-3f,AMP_2);
dvi_13->set_voltage(DVI_CHANNEL_1,0.0f,RANGE_20_V);
dvi_13->set_current(DVI_CHANNEL_1,100e-3f,AMP_2);
delay(2);//掉电
system_init();//系统初始化
delay(2);//延时 2mS
/////显示/////
do_dlog(func,0,v,POWER_UNIT,fail_bin);// “0”表示显示的顺序，“v”表示
要输出的变量值，“POWER_UNIT”表示量程档位为 1，“fail_bin”表示错误返
回值，简称软件 BIN
do_dlog(func,1,i,POWER_UNIT,fail_bin);
do_dlog(func,2,r,POWER_UNIT,fail_bin);

```

### (3) 波形

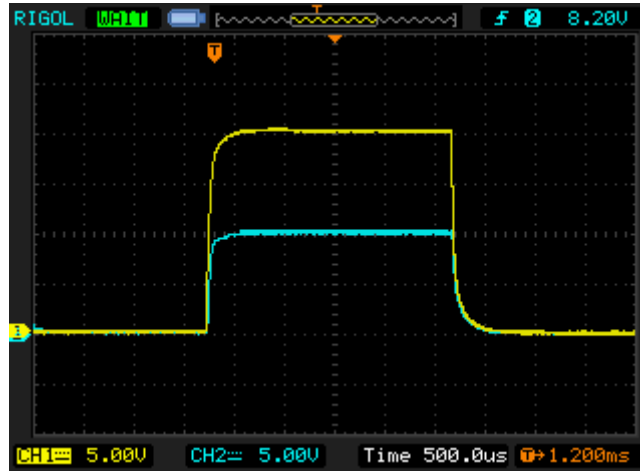


图 10.2 波形图

如上图 10.2 所示，黄色为 Tp1 处波形，蓝色为 Tp2 处波形，选用 10 欧姆电阻，程序中给 1A 电流，从波形中可以看到 Tp1 和 Tp2 之间的差值，即为电阻两端电压差，根据欧姆定律  $R = V / I = 10V / 1A = 10$  欧姆，符合理论分析。

### (4) 数据

12/30/13 08:23:35 Serial Number: 83			Total Devices: 83			PASSED BIN: 1		
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.10.01	DVI3_13	V	9.9700	1	V	none	none	
1.10.02	DVI3_13	I	1.0000	1	A	none	none	
1.10.03	DVI3_13	R	9.9700	1	ohm	9.0000	11.0000	

12.659 mSecs Total=12.659 mSec

图 10.3 测试数据

如上图 10.3 显示，差分测得的电压值为 9.97V，给定的电流值为 1A，计算所得的 R1 为 9.97 欧姆。

## 十一. CBIT\_14

(1) 示例线路如下图 11.1:

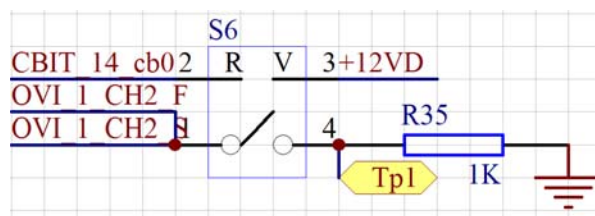


图 11.1 CBIT 测试线路图

CBIT 主要用来当继电器的驱动，与 MUX 类似。线路图 11.1 中，当继电器未闭合时，OVI 的 2 通道不管给多少电压，都无法检测到电流，而当继电器闭合时，用 OVI 源给 2V 电压，就能测量到 2MA 的电流，并且在 Tp1 处用示波器看到 2V 电压。

(2) 示例程序

```
float i,fail_bin=14;//定义变量
system_init();//系统初始化
delay(1);//延时 1mS
mux_14->close_relay(MUX_BANK_1_2);
mux_14->close_relay(MUX_2_GND);//闭合板载继电器
delay(2);
//当继电器驱动
mux_14->close_relay(MUX_1_1); //闭合 MUX14-1-1 继电器
delay(2);
ovi_1->set_voltage(OVI_CHANNEL_2,2.0f,RANGE_2_V); //OVI 通道 2 打 2V
电压，2V 档位
ovi_1->set_current(OVI_CHANNEL_2,5e-3f,RANGE_30_MA);//OVI 通道 2 打
5MA 电流，30MA 档位
ovi_1->set_meas_mode(OVI_CHANNEL_2,OVI_MEASURE_VOLTAGE);//测
量 OVI 通道 2 的电流
delay(2);
i=ovi_1->measure_average(10);//返回 10 次 ADC 采样的平均值
ovi_1->set_voltage(OVI_CHANNEL_2,0.0f,RANGE_1_V); ////将源置 0
ovi_1->set_current(OVI_CHANNEL_2,1e-3f,RANGE_3_MA);
delay(2);
mux_14->close_relay(MUX_BANK_1_2);
```

```

mux_14->close_relay(MUX_2_GND);//断开板载继电器
delay(2);//延时 2mS
system_init();//系统初始化
delay(1);
////////显示
do_dlog(func,0,i,POWER_MILLI,fail_bin);// “0” 表示显示的顺序,“i” 表示
要输出的变量值,“POWER_MILLI” 表示量程档位为 1e-3,“fail_bin” 表示错
误返回值, 简称软件 BIN

```

### (3) 波形

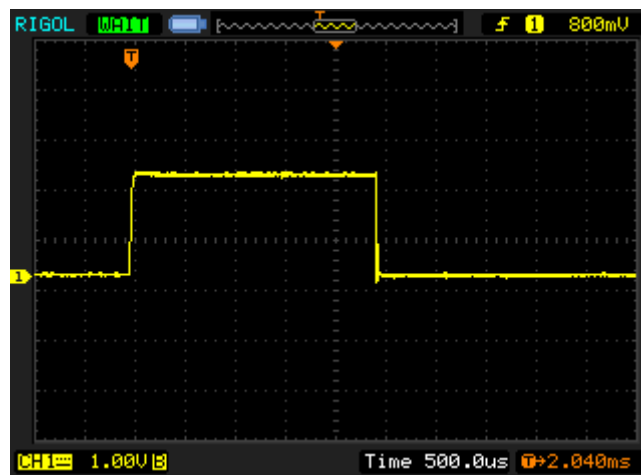


图 11.2 波形图

如上图 11.2 所示, 黄表笔在 Tp1 处能测到 2V 电压, 说明继电器已经闭合。

### (4) 数据

12/30/13 08:49:36 Serial Number: 17		Total Devices: 17		PASSED BIN: 1				
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.11.01	CBIT_14	I	1.9614	1	mA	none	none	I=V/1000, 导通
16.026 mSecs		Total=16.026 mSec						

图 11.3 测试数据

如上图 11.3 显示, 当继电器闭合时, 可以测得线路中的电流为 1.9614mA。

## 十二. HVS\_15

(1) 示例线路如下图 12.1:

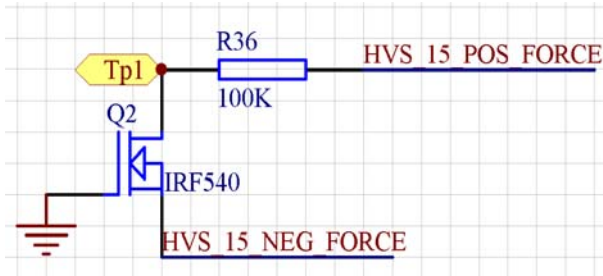


图 12.1 HVS 测试线路图

HVS 为高压源，如上图 12.1 所示，本示例选用 NMOS 管 IRF540 测量其 BVDSS，MOS 管的 D 级接 HVS 的 POS\_FORCE，S 级接 HVS 的 NEG\_FORCE，通 250UA 的电流，测 MOS 管的 VDS，在 HVS 的 POS 端接匹配的电阻，能消除 HVS 的纹波，使测量准确。

(2) 示例程序

```
float v,i,fail_bin=15;
system_init();//系统初始化
delay(1);//延时 1mS
i=250e-6f;
hvs_15->set_current(i,RANGE_1_MA);//设置 hvs 的电流为 250UA, 档位 1MA
hvs_15->set_voltage(0.0f);//继电器开关中无电压
hvs_15->close_relay(HVS_FORCE_POS);//使正输出经一条线连接至 DUT
hvs_15->close_relay(HVS_NEG_FORCE);//使负输出经一条线连接至 DUT
hvs_15->close_relay(HVS_SHORT_NEG_FS);//短路 force 线与 sense 线来允许
一条电线连接至 DUT
hvs_15->close_relay(HVS_SHORT_POS_FS);//短路 force 线与 sense 线来允许
一条电线连接至 DUT
hvs_15->close_relay(HVS_GND_NEG_SENSE);//使负输入短路至 DUT 地
delay(2);//延时 2mS
hvs_15->set_meas_mode(HVS_MEASURE_VOLTAGE);//设置测量模式
delay(5);//延时 5mS
hvs_15->set_voltage(150.0f);//设置电压
delay(20);//延时 20mS
```

```
v=hvs_15->measure_average(10)-25;//测量 10 次的平均值减去电阻上的分压
```

```
hvs_15->set_voltage(0.0); //源置 0， 断电
```

```
delay(5);//延时 5mS
```

```
system_init();//系统初始化
```

```
delay(2); //延时 2mS
```

```
////显示函数
```

```
do_dlog(func,0,i,POWER_MICRO,fail_bin);// “0” 表示显示的顺序，“i” 表示要输出的变量值，“POWER_MICRO” 表示量程档位为 1e-6，“fail_bin” 表示错误返回值，简称软件 BIN
```

```
do_dlog(func,1,v,POWER_UNIT,fail_bin);
```

### (3) 波形

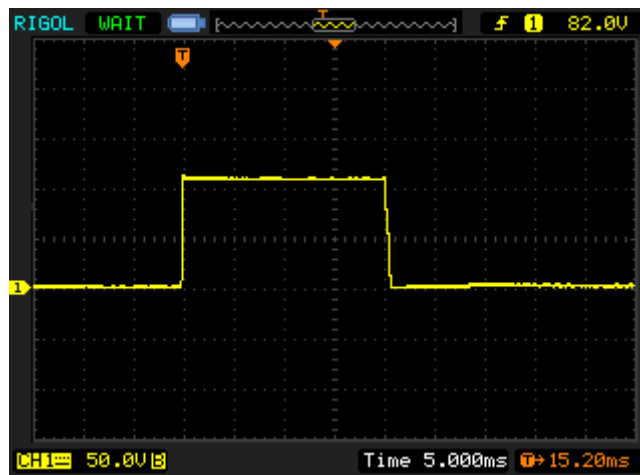


图 12.2 波形图

如上图 12.2 所示，Tp1 处波形中，可以看到电压约为 106V，波形与返回的测试值一致。

### (4) 数据

12/30/13 08:51:48 Serial Number: 22		Total Devices: 22		PASSED BIN: 1				
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.12.01	HVS_15	I	250.0000	1	uA	none	none	
1.12.02	HVS_15	V_DS	106.8765	1	V	100.0000	none	
47.636 mSecs Total=47.636 mSec								

图 12.3 测试数据

如上图 12.3 显示，测量所得的 VDS 为 106.8765V，程序中给的电流为 250uA。

### 十三. XVI\_16

(1) 示例线路如下图 13.1:

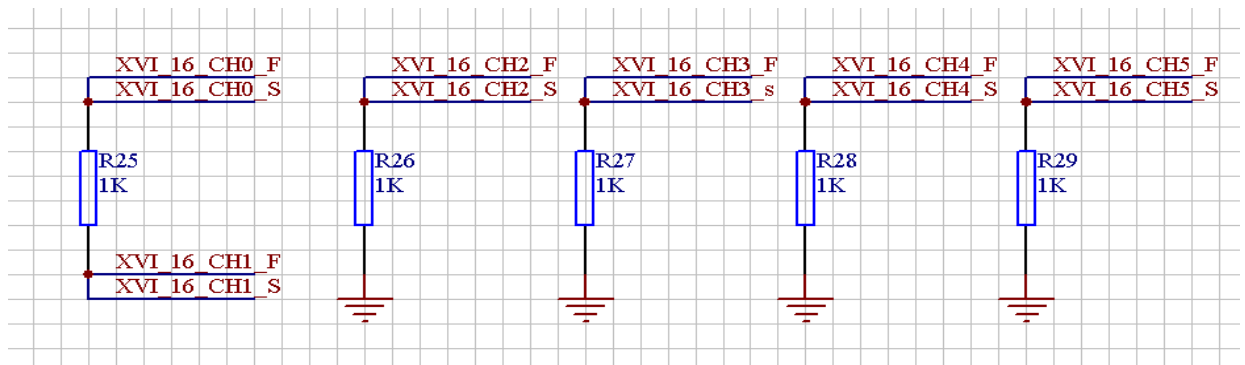


图 13.1XVI 测试线路图

### 十四. MUX\_20

(1) 示例线路如下图 14.1:

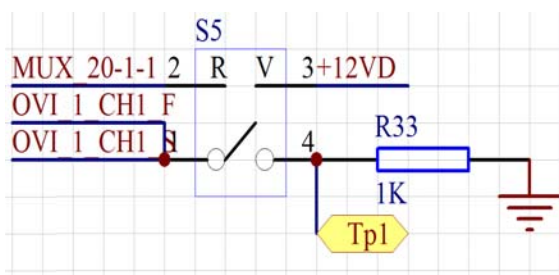


图 14.1MUX 当继电器驱动测试线路图

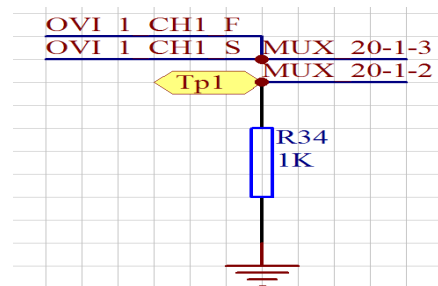


图 14.2MUX 当继电器测试线路图

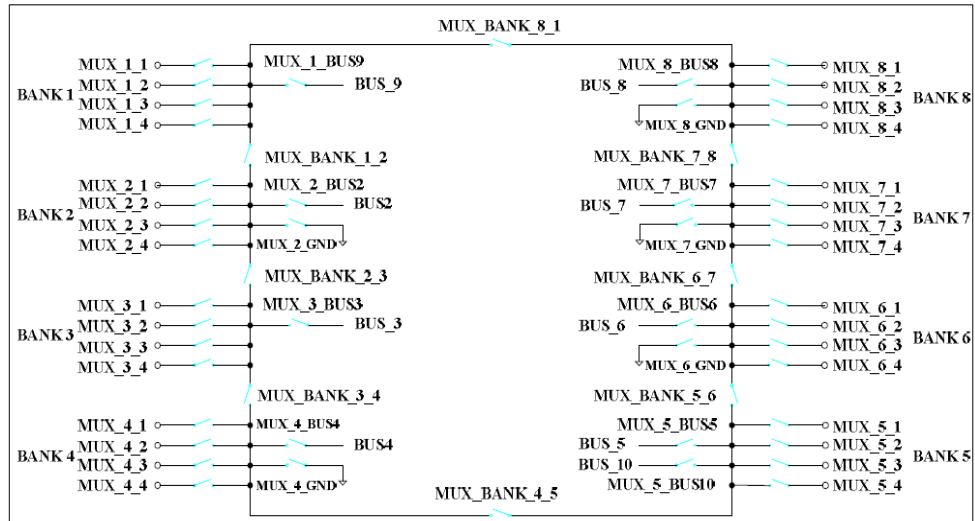


图 14.3 MUX 框图

MUX 板和 CBIT 板在当继电器的驱动的时候可以通用，不同的是，MUX 板可以直接作为继电器使用，线路图 14.1 中，S5 是使用了 MUX 作为继电器驱动时的功能，线路图 14.2 中，测 R34 阻值是使用了 MUX 内部的继电器，当 MUX20-1-3 和 MUX20-1-2 同时闭合时，且断开 MUX\_BANK\_1\_2，MUX\_BANK\_8\_1 这时内部导通，源加到电阻的一端，可以利用 OVI 测电阻的方法测出电阻的阻值。

## (2) 示例程序

```
float i1,r,fail_bin=16;//变量声明
system_init();//系统初始化
delay(2);//延时 2mS
//当继电器
mux_20->open_relay(MUX_BANK_1_2);//断开板子内部的继电器
mux_20->open_relay(MUX_BANK_8_1);
delay(2);
mux_20->open_relay(MUX_1_1); //断开 MUX20-1-1 继电器
mux_20->close_relay(MUX_1_2); //闭合 MUX20-1-2 继电器
mux_20->close_relay(MUX_1_3); //闭合 MUX20-1-3 继电器
delay(2); //延时 2mS
ovi_1->set_voltage(OVI_CHANNEL_1,1.0f,RANGE_1_V); //设置 OVI 通道 1
至 1V 电压
ovi_1->set_current(OVI_CHANNEL_1,3e-3f,RANGE_3_MA); //设置 OVI 通道
1 至 3MA 电流
ovi_1->set_meas_mode(OVI_CHANNEL_1,OVI_MEASURE_CURRENT); //测
量 1 通道的电流
delay(2);//延时 2mS
r=ovi_1->measure_average(10); //返回 10 次 ADC 采样的平均值
```



```

r=1/r;
ovi_1->set_voltage(OVI_CHANNEL_1,0.0f,RANGE_1_V); //掉电
ovi_1->set_current(OVI_CHANNEL_1,1e-3f,RANGE_3_MA);
//当继电器驱动
mux_20->close_relay(MUX_BANK_1_2);
mux_20->close_relay(MUX_2_GND); //闭合板子内部继电器
delay(2); //延时 2mS
mux_20->close_relay(MUX_1_1); //闭合 MUX20-1-1 继电器
mux_20->open_relay(MUX_1_2); //断开 MUX20-1-2 继电器
mux_20->open_relay(MUX_1_3); //断开 MUX20-1-3 继电器
delay(2);
ovi_1->set_voltage(OVI_CHANNEL_1,2.0f,RANGE_2_V); //设置 OVI 通道 1
至 2V 电压
ovi_1->set_current(OVI_CHANNEL_1,3e-3f,RANGE_3_MA); //设置 OVI 通道
1 至 3MA 电流
ovi_1->set_meas_mode(OVI_CHANNEL_1,OVI_MEASURE_CURRENT); //测
量 1 通道的电流
delay(2); //延时 2mS
i1=ovi_1->measure_average(10); //返回 10 次 ADC 采样的平均值
ovi_1->set_voltage(OVI_CHANNEL_1,0.0f,RANGE_2_V); //掉电
ovi_1->set_current(OVI_CHANNEL_1,3e-3f,RANGE_3_MA);
mux_20->open_relay(MUX_1_1); //断开所有用到的继电器
mux_20->open_relay(MUX_1_2);
mux_20->open_relay(MUX_1_3);
delay(2); //延时 2mS
mux_20->close_relay(MUX_BANK_1_2);
mux_20->close_relay(MUX_2_GND); //闭合板子内部继电器
delay(2); //延时 2mS
system_init(); //系统初始化
delay(2); //延时 2mS
////////显示函数
do_dlog(func,0,i1,POWER_MILLI,fail_bin); // “0” 表示显示的顺序, “i1” 表
示要输出的变量值, “POWER_MILLI” 表示量程档位为 1e-3, “fail_bin” 表示
错误返回值, 简称软件 BIN
do_dlog(func,1,r,POWER_UNIT,fail_bin);

```

### (3) 波形

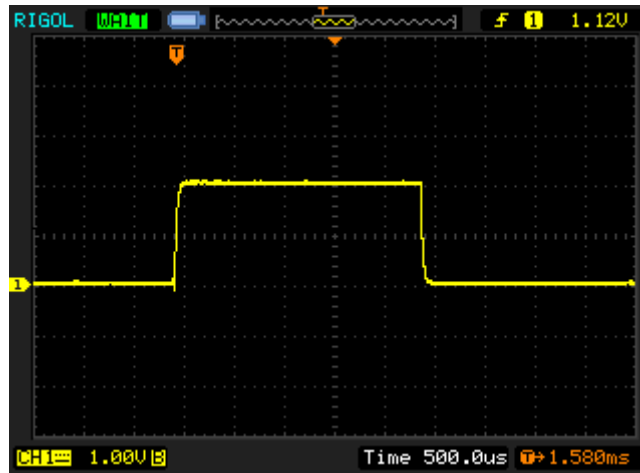


图 14.4 当继电器驱动波形图

如上图 14.4 所示，MUX 当继电器驱动时的波形，可在继电器 S5 的 Tp1 处检测到 2V 电压，说明继电器导通。

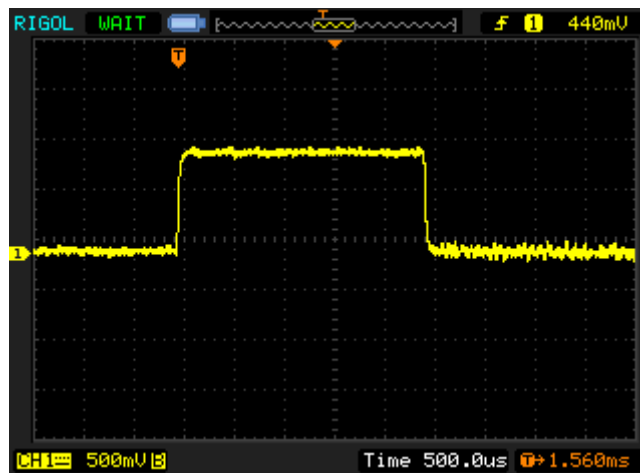


图 14.5 当继电器测电阻波形图

如上图 14.5 所示，MUX 当继电器测电阻时的波形，在 Tp1 处测到 1V 电压，可知 R31 此时与源导通。

#### (4) 数据

12/30/13 09:03:18 Serial Number: 78			Total Devices: 78			PASSED BIN: 1		
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.14.01	MUX_20	I	2.0019	1	mA	none	none	I=V/1000, 导通
1.14.02	MUX_20	R	1001.9857	1	ohm	990.0000	1100.0000	
20.520 mSecs Total=20.520 mSec								

图 14.6 测试数据

如上图 14.6 显示，测量所得的 I 为 2.0019mA，电阻 R 为 1001.9857 欧姆。

## 十五. QFVI\_21

(1) 示例线路如下图 15.1:

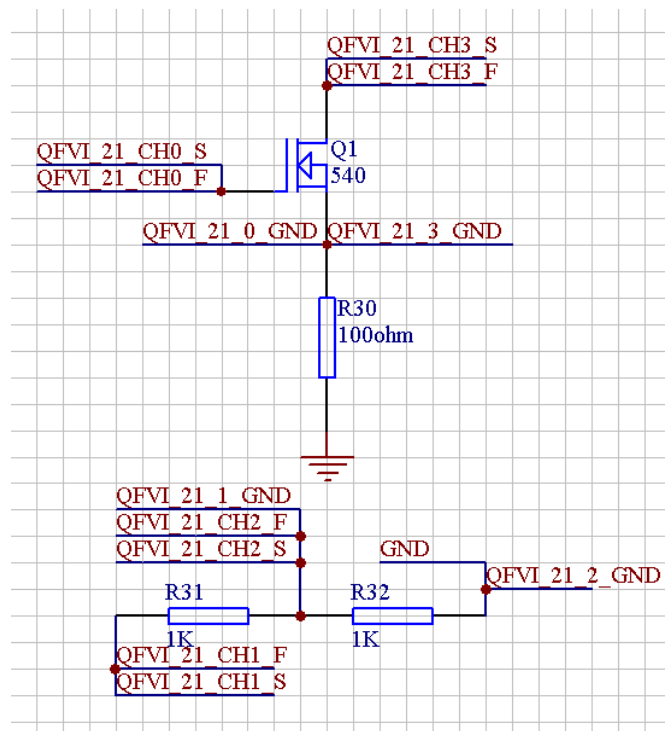


图 15.1 QFVI 测试线路图

如上图 15.1 所示，将 QFVI 的 0 通道的 0\_FS 和 0\_GND 端分别接在 MOS 管的 G,S 端，浮动源可以为 MOS 提供恒定的开启电压 VGS，通过 QFVI\_CH3 来检测 MOS 管 DS 两端的导通情况。另，用浮动源给 R31, R32 通 5MA 的电流，测电阻两端的压差都为 5V，可以说明浮动源测得是与参考地的压差。

## (2) 示例程序

```
float v,i,fail_bin=17;//变量声明
system_init();//系统初始化
delay(1);//延时 1mS
qvi_21->set_voltage(0,10.0f); //设置通道 0_F 到 0_GND 两端的电压 10V
qvi_21->set_current(0,5e-3f);//设置通道 0 电流为 5mA
delay(2);//延时 2mS
qvi_21->set_voltage(3,10.0f);
qvi_21->set_current(3,50e-3f);
qvi_21->set_meas_mode(3,QVI_MEASURE_CURRENT);//测量通道 3 的电流
delay(2);
i=qvi_21->measure_average(3,5);//采样通道 3 的值 5 次取平均值
qvi_21->set_voltage(2,10.0f);
qvi_21->set_current(2,1e-3f);
delay(2);
qvi_21->set_voltage(1,10.0f);
qvi_21->set_current(1,5e-3f);
qvi_21->set_meas_mode(1,QVI_MEASURE_VOLTAGE);//测量通道 1 的电压
delay(2);//延时 2mS
v=qvi_21->measure_average(1,5); //采样通道 1 的值 5 次取平均值
qvi_21->set_voltage(0,0.0f); //将源置 0,掉电
qvi_21->set_current(0,1e-6f);
qvi_21->set_voltage(1,0.0f);
qvi_21->set_current(1,1e-6f);
qvi_21->set_voltage(2,0.0f);
qvi_21->set_current(2,1e-6f);
qvi_21->set_voltage(3,0.0f);
qvi_21->set_current(3,1e-6f);
delay(2);
system_init();//系统初始化
delay(2);
/////显示/////
do_dlog(func,0,i,POWER_MILLI,fail_bin);// “0” 表示显示的顺序,“i” 表示
要输出的变量值,“POWER_MILLI” 表示量程档位为 1e-3,“fail_bin” 表示错
误返回值,简称软件 BIN
do_dlog(func,1,v,POWER_UNIT,fail_bin);
```

### (3) 数据

12/30/13 09:13:00 Serial Number: 121 Total Devices: 121 PASSED BIN: 1								
Test #	Function Name	Test Name	Value	P/F	Unit	Min Limit	Max Limit	Notes
1.15.01	QFVI_21	I	50.2262	1	mA	49.0000	51.0000	
1.15.02	QFVI_21	V	4.9511	1	V	4.9000	5.1000	
17.068 mSecs Total=17.068 mSec								

图 15.2 测试数据

如上图 15.2 显示，测量所得的 I 为 50.2262mA，说明 MOS 导通，测得的 V 为 4.9511V。