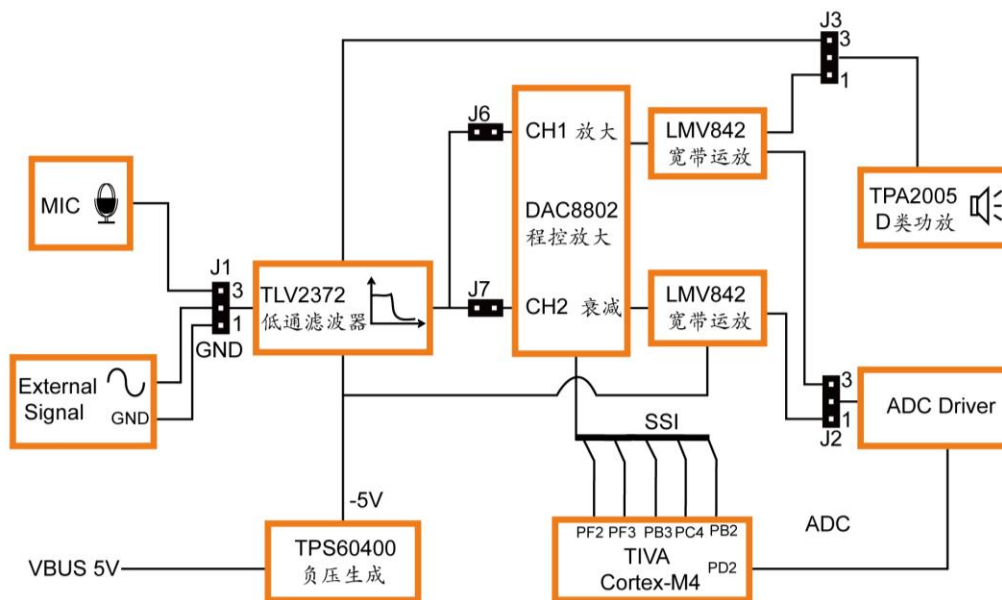


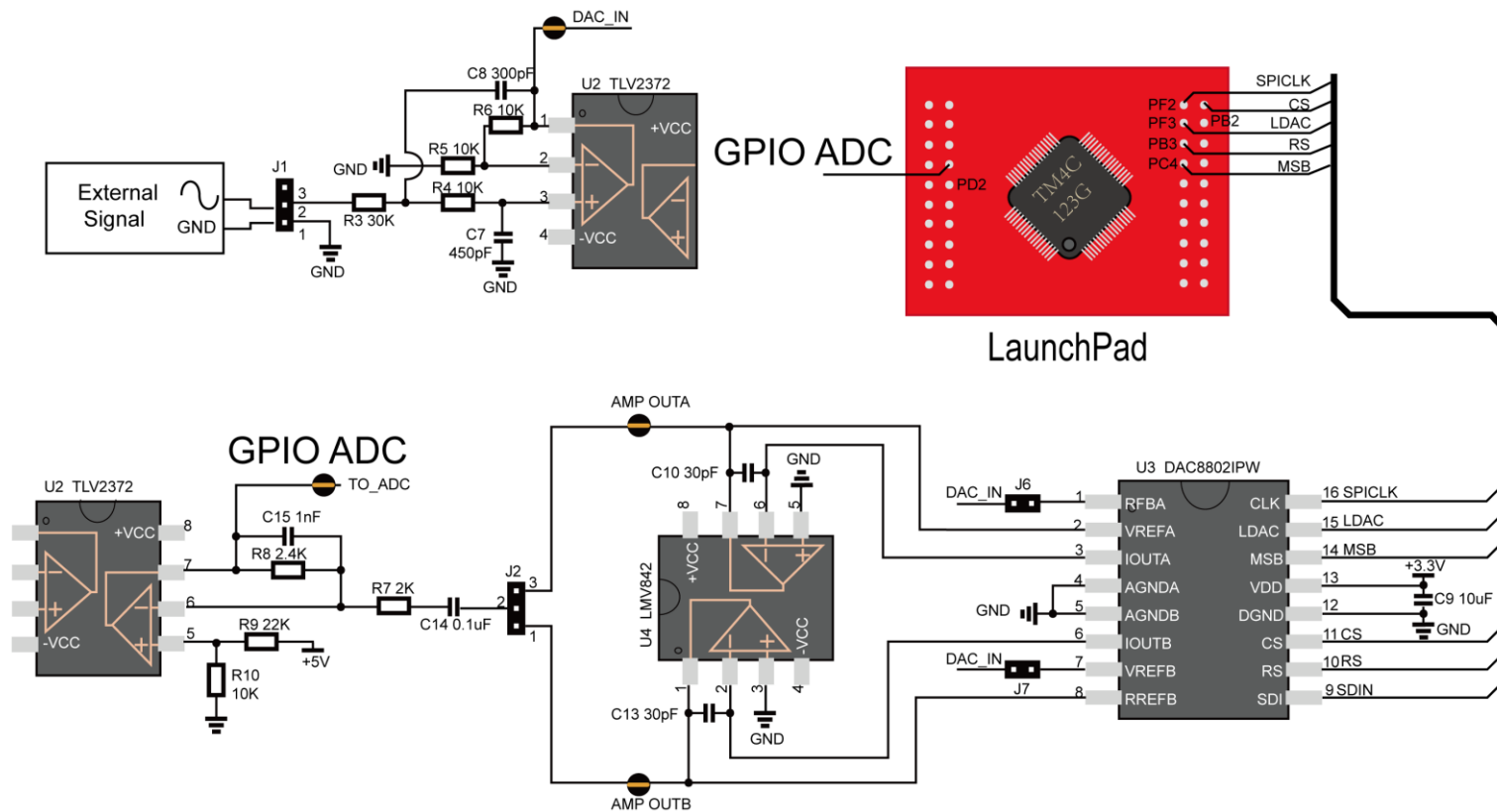
第五章. 程控增益放大模块

MDAC程控放大模块介绍

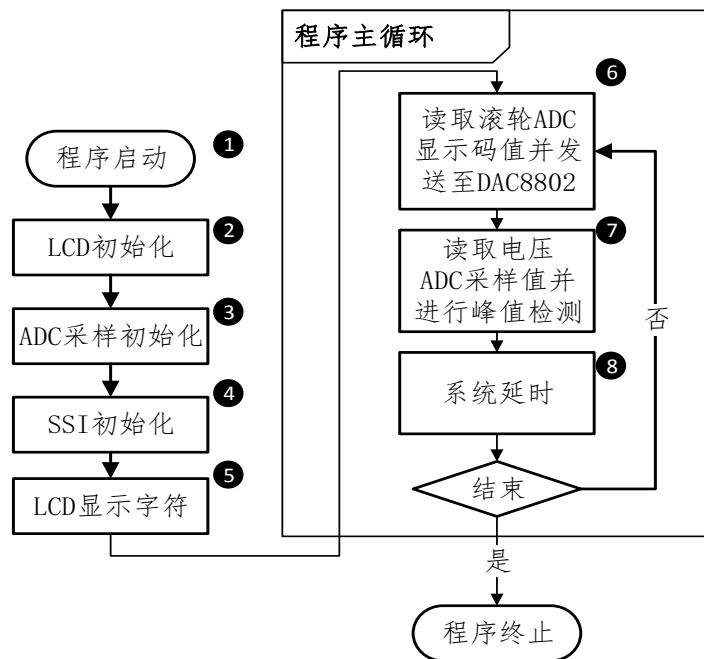
实验简介

本模块的核心部分为由一片14位双通道串行接口乘法数模转换器DAC8802芯片组成的程控放大和衰减部分，这部分中DAC8802芯片通过SSI总线与TIVA实现通信，由TIVA传输的CODE值决定放大/衰减幅值。另外，还可选择将外部信号或者程控放大衰减获得的信号作为输入，经过TPA2005D芯片得到音频输出信号，并通过喇叭外放。





- ① 程序成功烧写后，TIVA LaunchPad上电后可启动程序运行。
- ② LCD初始化包括LCD端口使能、SSI通信协议配置、LCD配置初始化、LCD清屏4个步骤。每个步骤详情请见本书第三章。
- ③ ADC采样初始化有两个部分：滚轮电阻采样初始化(PE0)和峰值检测模块电压采样初始化(PD2)。ADC采样初始化步骤：1、使能ADC模块外设 2、配置相关GPIO为ADC功能 3、ADC采样序列配置 4、ADC采样序列步进配置 5、使能采样序列并清除中断标志。
- ④ SSI初始化负责完成与DAC8802之间通信的所有信号线的配置。包括PF2、PF3、PF1、PB2、PB3、PC4，其中PF2、PF3、PF1配置成SSI通信端口，其余这使能为端口输出。SSI初始化步骤：1、使能SSI外设模块。 2、配置相关GPIO复用功能为SSI模块功能并为SSI模块通信使用。 3、SSI通信模式、时钟频率设置和数据位设置。 4、使能SSI。
- ⑤ 在LCD上显示Peak detection voltage: xx V，等信息，便于实验过程中观察实验数据变化。
- ⑥ 在程序主循环中，读取滚轮ADC采样值，TIVA的AD是12-Bit，而DAC8802的DA是14-Bit，需要通过相应的转换得到DAC8802对应的码值并将码值发送至DAC8802，同时将码值显示到LCD上。读取ADC采样值的程序部分详见本书第三章。



- ⑦ 读取电压 (PD2) ADC采样值，通过相应的转换公式转换成实际电压值并进行峰值检测，将检测到的峰值显示在LCD上，便于观察。此处读取ADC采样值的程序部分可参考本书第三章。
- ⑧ 为防止LCD过快重复刷新，系统进行延时将主循环控制在一个合理的执行速度内。

http://www.hpati.com

在程序设计过程中主要涉及到 LCD 显示，ADC 采样以及 SSI 通信，其中 LCD 显示和 ADC 采样功能设置可参考本书第三章相关内容。

Xxx SPI(SSl)通信

DAC8802 采用 SPI（相当于 Tiva M4 的 SSI 协议）通信协议进行数据传输。DAC8802 是 14-bit 的 DAC，而串行数据锁存在 DAC8802 的串行输入寄存器 (serial input register) 中，该寄存器为 16-bit 即两个字节长度（2-bit 的地址: A1-A0，以及 14 的 DA 数据: D13-D0）。寄存器数据格式如下：

表 xx SPI 寄存器数格式

Bit	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0(LSB)
Data	A1	A0	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

DAC8802 内部有两个 DAC，分别为 DAC A 和 DAC B，其中 DAC A 为放大功能，DAC B 为衰减功能，程序控制 DAC8802 的放大和衰减功能就是选择使能其中一个 DAC 模块。该两个模块的选择通过串行输入寄存器中地址位: A1 和 A0 进行选择。

表 xx 地址位设置

A1	A0	使能 DAC 模块
0	0	None
0	1	DAC A
1	0	DAC B
1	1	DAC A 和 DAC B

程序上的实现可以采用#define 宏定义，然后在发送数据时将定义好的地址值加上待发送数据即可。宏定义代码如下：

```
#define DAC_A 0x4000 //DAC A
#define DAC_B 0x8000 //DAC B
#define DAC_AB 0xC000 //DAC A和DAC B
```

SPI (SSl) 通信配置函数

DAC8802 通过 CS（低电平有效），SDI，SCK 三线控制数据的传输，其对应 Tiva M4 中的 SSIFss, SSITx, SSIClk 线，相应配置代码如下：

```
/******
* @brief SSI模块使能，并且设置相关端口初始状态
```

```

* @param none
* @return none
*
*
*          _____
*          |
*          PF2 (SSI1Clk) |-->SPICLK  时钟信号端
* TIVA     PF3 (SSI1Fss) |-->SYNC    帧信号端
*          PF1 (SSI1Tx)  |-->SDIN     SSI数据发送端 (LM4F120->DAC8802)
*          PB2 (GPIO)   |-->LDAC
*          PB3 (GPIO)   |-->RS
*          PC4 (GPIO)   |-->MSB
*          _____
*
***** /
void ssi_en()
{
    //使能外设SSI1模块
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_SSI1);
    //使能SSI1使用的外设GPIOF
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);

    //SSI1端口功能使能
    //PF2复用功能配置为SSI1CLK, 时钟线
    ROM_GPIOPinConfigure(GPIO_PF2_SSI1CLK);
    //PF3复用功能配置为SSI1FSS, 片选线

```

```
ROM_GPIOPinConfigure(GPIO_PF3_SSI1FSS);  
//PF1复用功能配置为SSI1TX, 数据发送线  
ROM_GPIOPinConfigure(GPIO_PF1_SSI1TX);  
  
//LDAC置高  
ROM_GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_2);  
ROM_GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_2, GPIO_PIN_2);  
//RS置高  
ROM_GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_3);  
ROM_GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_3, GPIO_PIN_3);  
//MSB置高  
ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOC);  
ROM_GPIOPinTypeGPIOOutput(GPIO_PORTC_BASE, GPIO_PIN_4);  
ROM_GPIOPinWrite(GPIO_PORTC_BASE, GPIO_PIN_4, GPIO_PIN_配置PF1, PF2, PF3供外设SSI1使用  
ROM_GPIOPinTypeSSI(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3);  
//端口模式:1M,16位数据  
ROM_SSIConfigSetExpClk(SSI1_BASE, ROM_SysCtlClockGet(),  
                        SSI_FRF_MOTO_MODE_0, SSI_MODE_MASTER, 1000000, 16);  
  
//使能SSI  
SSIEnable(SSI1_BASE);  
}
```

程序还涉及到 DAC8802 另外三根线的配置: LDAC 置高, RS 置高, MSB 置高。其中 LDAC 和 RS 都是低电平有效。LDAC 控制 DAC8802 的输出, RS 和 MSB 信号线连接至 DAC8802 内部的上电复位模块, 复位时若 MSB=0, 则所有寄存器值为 0x0000, 若 MSB=1, 则所有寄存器值为 0x2000。

DAC8802 的时序图如下:

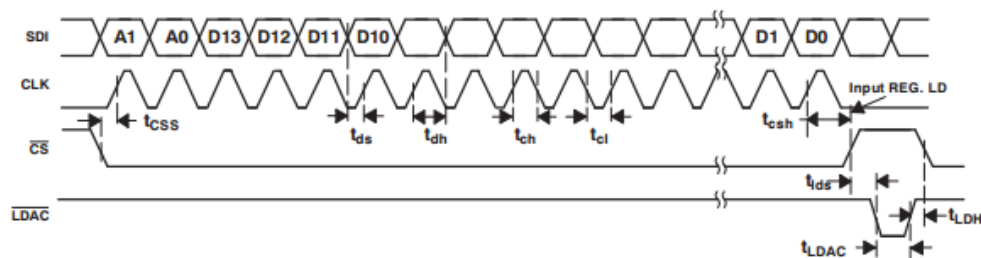


图 xx 时序图

根据时序图可以完成 DAC8802 跟 Tiva M4 之间的数据传输。在 SPI (SSI) 配置程序中，时序图中的 SDI，CLK 以及 CS 线都配置成了 Tiva M4 中的 SSI 功能，在数据传输过程中这三根信号线上的电平变化都由 Tiva M4 的 SSI 模块自行控制。而 LDAC 线配置成普通的 GPIO 功能，则改线上的电平变化需要自行控制。SPI (SSI) 传输程序代码如下：

```

/*****
*   @brief    向dac8802发送数据
*   @param    unsigned long val, 取值范围0~16384
*   @return    0, 参数不正确;
*             1, 传输成功;
*****/
unsigned char ssi_send_2_dac8802(unsigned long val)
{
    if (val > 16384) return 0;

    ROM_SSIDataPut (SSI1_BASE, DAC_AB + val);    //发数据+
    while (ROM_SSIBusy (SSI1_BASE));             //等待发送完成

    //数据发送结束时，LDAC线需要一个电平的跳变 (H->L->H)
}
    
```



```
ROM_GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_2, 0);  
delay();  
ROM_GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_2, GPIO_PIN_2);  
delay();  
return 1;  
}
```

程序传输中使用 ROM_SSIDataPut 就可完成数据发送，发送的数据格式地址+数据即程序中的 DAC_AB+val, 表示的是使能 DAC A 和 DAC B 同时使用两个 DAC 模块，val 即为需要发送的数据。此时同时完成放大和衰减，如果只要单独使用放大或者衰减，则发送数据时自需要变成 DAC_A+val 或 DAC_B+val。在数据发送完成后需要完成一个 LDAC 信号线的电平转变，完成 DAC8802 模块的输出。

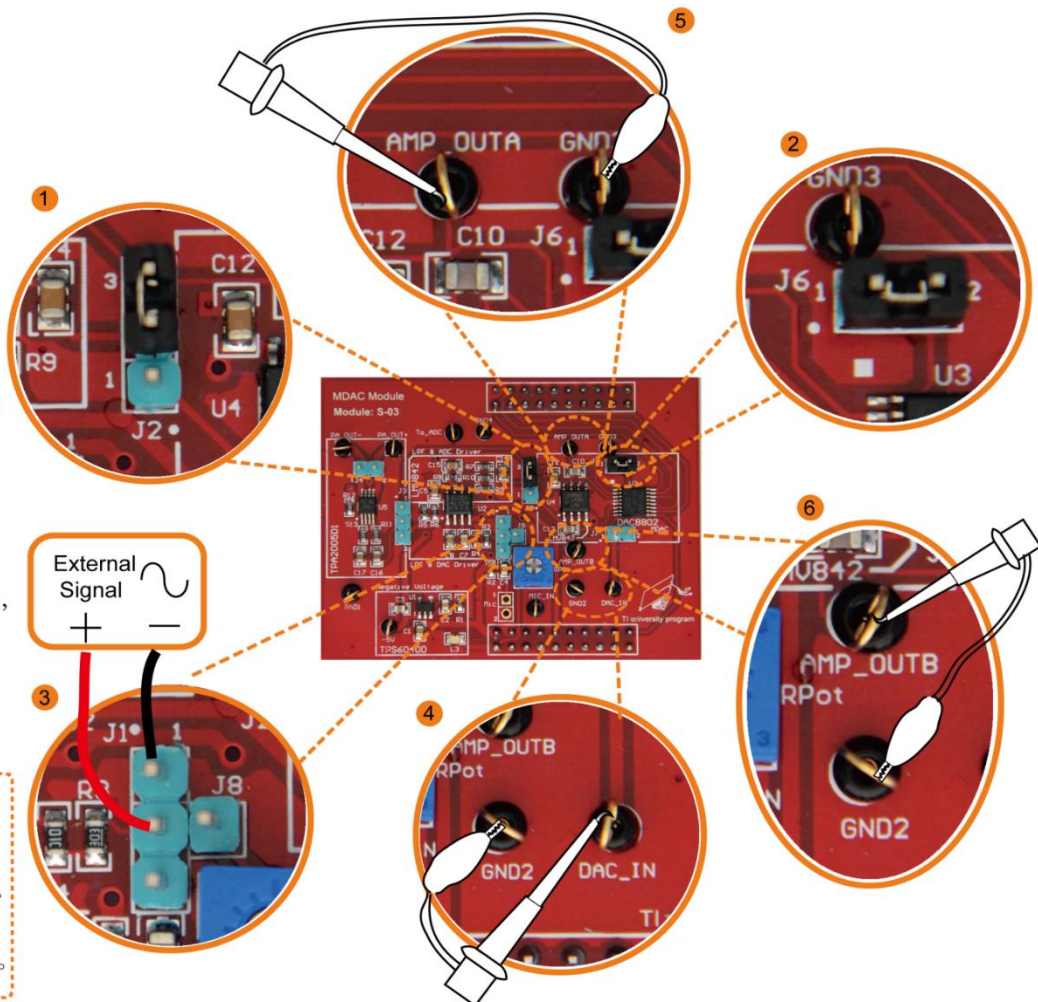
程控放大和衰减

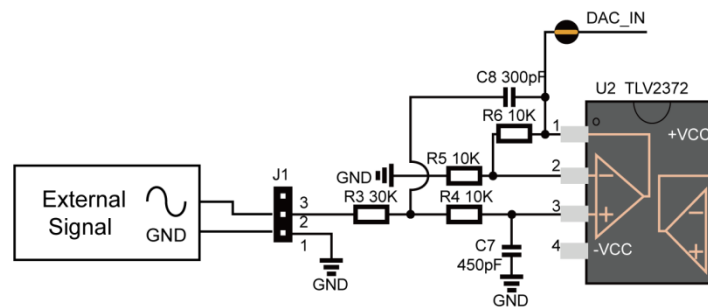
- 1、理解原理图以后编写Launchpad代码，代码可参考网上资源。然后烧写代码。
- 2、在母板上TIVA、液晶、MDAC模块连接完成，准备实验。
- 3、在MDAC模块上完成跳线连接，如图 ①, ② 所示，短接J2的 ②, ③ 以及J6。
- 4、用杜邦线连接信号源，信号正极接J1的2，信号负极接J1的1，如图 ③ 所示。输入信号以1V，25Hz为宜。
- 5、打开TIVA开关，用示波器双踪观察DAC_IN以及AMP_OUTA，如图 ④, ⑤ 所示。调节滚轮，观察波形变化。
- 6、断电后，短接J2的 ①, ②，拔掉J6，短接J7。
- 7、用示波器双踪观察DAC_IN以及AMP_OUTB，如图 ⑥ 所示。调节滚轮，观察波形变化。



注意

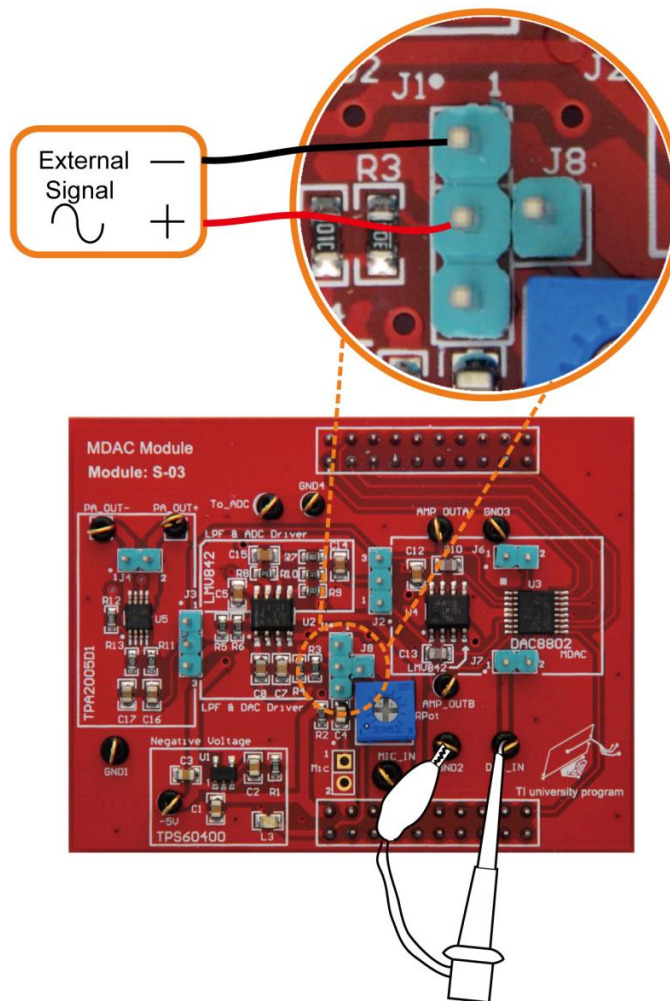
连接仪表及跳线时断开电源。并且电路初始状态输入到TIVA的管脚可能存在负压会导致TIVA的运行问紊乱。因此所有跳线完成之前系统不能上电。

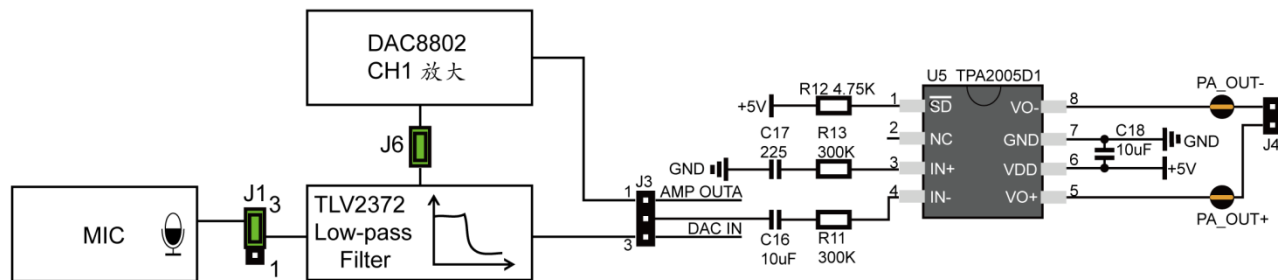




音频输入的信号调理

- 1、理解原理图以后编写Launchpad代码，代码可参考网上资源。然后烧写代码。
- 2、在母板上TIVA、液晶、MDAC模块连接完成，准备实验。
- 3、用杜邦线连接信号源，信号正极接J1的2，信号负极接J1的1，如右图所示。
- 4、打开TIVA开关，用示波器观察DAC_IN的波形。
- 5、改变输入信号频率，用示波器观察输出的结果。





音频输出功率推动

- 1、理解原理图以后编写Launchpad代码，代码可参考网上资源。然后烧写代码。
- 2、在母板上TIVA、液晶、MDAC模块连接完成，准备实验。
- 3、在MDAC模块上完成跳线连接，如图①，②，③，④短接J6，J2的2,3，J3的2,3，J1的2,3。
- 4、用杜邦线将话筒连接到MIC，如图⑤所示。将扩音器连接到J4，如图⑥所示。
- 5、打开TIVA开关，用示波器双踪观察PA_OUT+，PA_OUT-以及两个通道相减的波形。
- 6、断电后，短接J3的①，②，打开TIVA开关。调节滚轮，听声音的变化。



注意

连接仪表及跳线时断开电源。

