



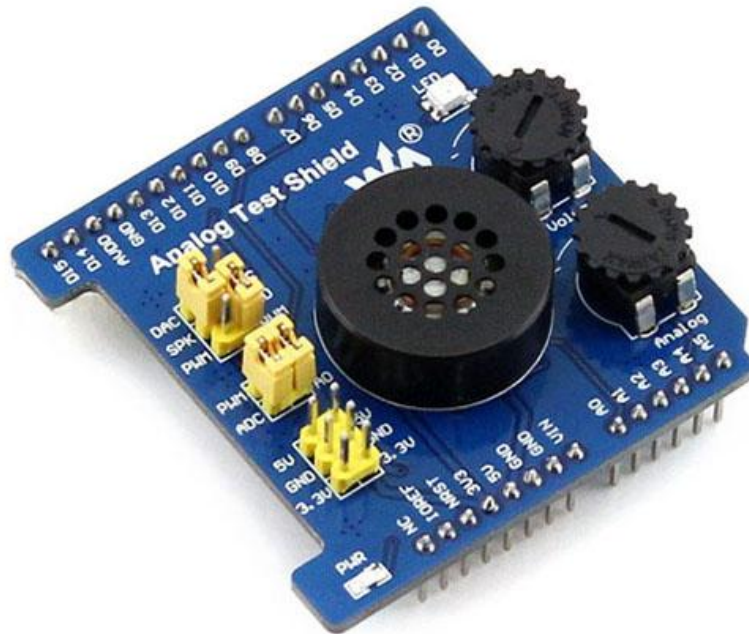
Analog Test Shield

用户手册

1. 产品概述

Analog Test Shield 是兼容 Arduino、NUCLOE 和 XNUCLEO 接口的扩展板，具备 AD 采集和 DA 输出功能。

- 基于 Arduino 标准接口设计，兼容 UNO、Leonardo、NUCLEO、XNUCLEO 开发板
- 旋转电位器，就能获得不同的 AD 采样数据
- 板载 12bit DAC 芯片 MCP4725，可以输出你想要的波形
- 通过板载的 LED 和喇叭，可以形象展示出 DAC 和 PWM 的效果



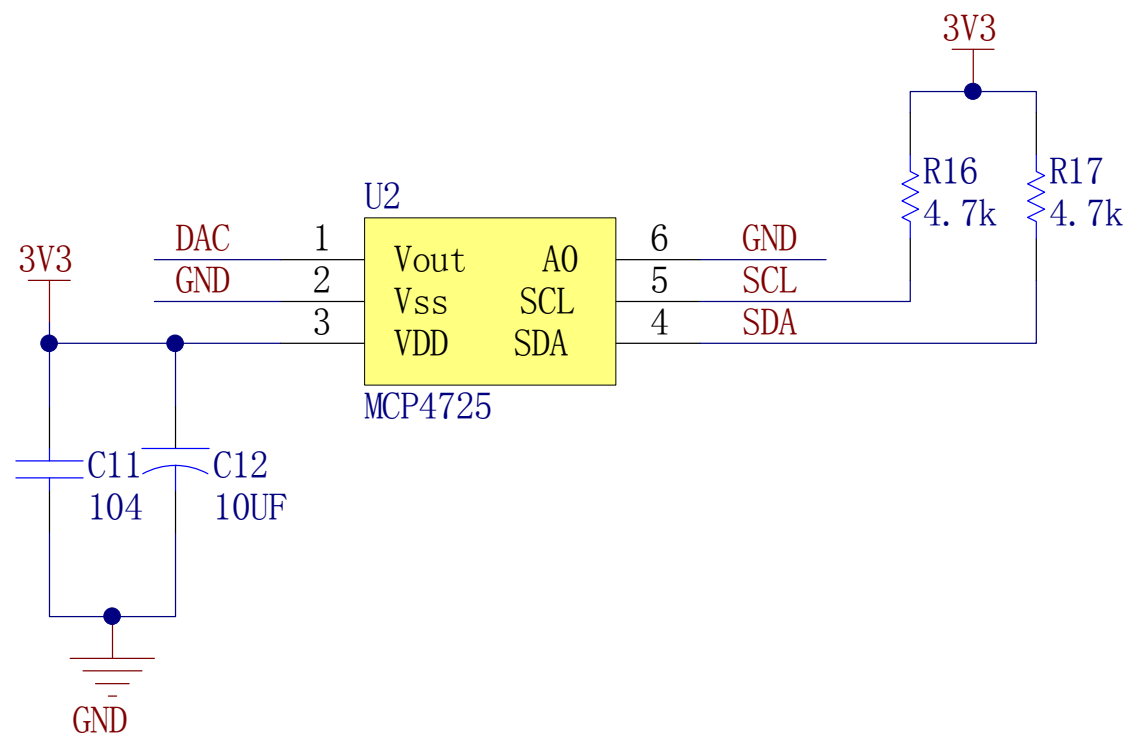
2. 硬件说明

该产品集成了 DAC 电路和功放电路。主要资源有：两个电位器旋钮，一个电源 LED，一个状态 LED 和一个外放喇叭。您可通过模数转换功能采集电位器的电压值；可通过数模转换或脉冲宽度调制（PWM，Pulse Width Modulation）控制 LED 亮度和驱动喇叭。这里只简要地说明一下产品 DAC 电路和功放电路原理。如果您需要了解更多芯片的数据，请参阅芯片的数据手册：[MCP4725-Datasheet.pdf](#) 和 [LM386-Datasheet.pdf](#)。

2.1. DAC 电路

MCP4725 是 12 位分辨率高精度的电压输出模拟转换器(DAC)，具有非易失性存储器(EEPROM)，通过 I2C 串行接口控制。单电源工作：2.7 至 4.5V。DAC 电路如下图所示：

DAC Circuit



引脚说明：

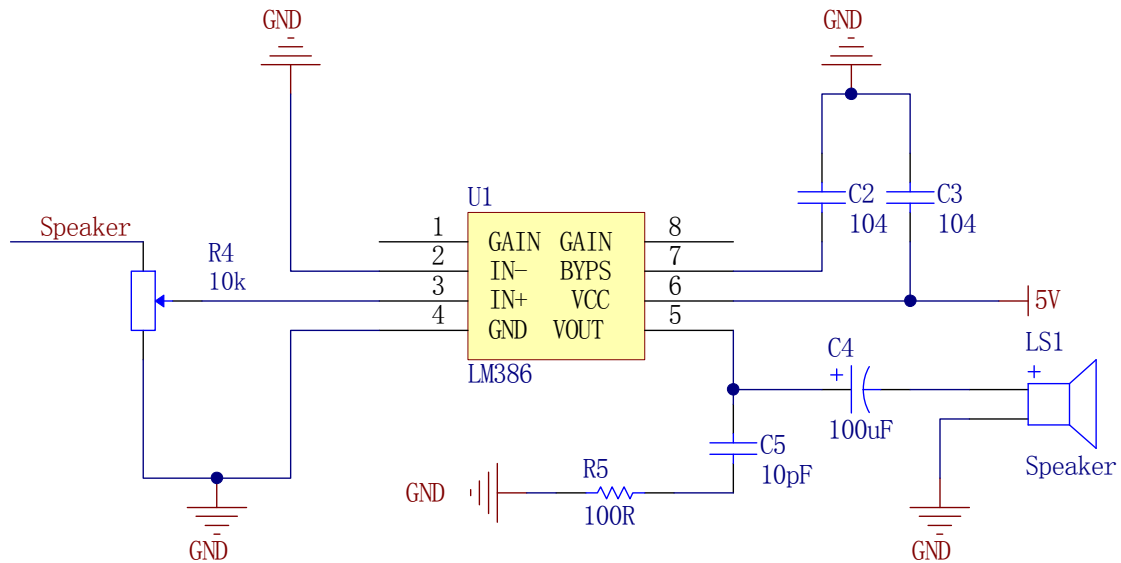
MCP4725	名称	说明
SOT-23		
1	V _{OUT}	模拟量输出电压

2	V _{SS}	电源地参考电压
3	V _{DD}	电源电压
4	SDA	I ² C 串行数据
5	SCL	I ² C 串行时钟输入
6	A0	I ² C 地址位选择引脚(A0 位)。该引脚可以连接到 V _{SS} 或者 V _{DD} ，或由数字逻辑电平有效驱动。该引脚的逻辑状态决定了 I2C 地址位的 A0 位。详情请参见 MCP4725-Datasheet.pdf 第 3.0 节引脚说明。

2.2. LM386 功放电路

LM386 是音频集成功率放大器，电压内置增益为 20，可通过 1 和 8 脚位间电阻电容的搭配，调节增益，增益最高可达 200。

Speaker Circuit



引脚说明：

引脚	名称	说明
1	GAIN	用于增益控制。
2	IN-	反相输入端
3	IN+	同相输入端
4	GND	接地
5	VOUT	电压输出端
6	VCC	电源电压
7	BYPS	用于旁路连接。请参见 LM386-Datasheet.pdf.
8	GAIN	用于增益控制

3. 程序原理

本产品的配套程序原理是，由 PWM 输出正弦波控制 LED 渐变闪烁，DAC 输出正弦波控制喇叭发声。每隔一秒读取一次模拟输入，并通过串口打印输出 ADC 转换的电压。

3.1. XNUCLEO 示例程序

该示例程序基于 mbed 平台。

ADC 转换实现方式：调用 mbed 中 AnalogIn 类 float read()函数读取 AD 转换的数据。

PWM 输出实现方式：调用 mbed 中 PwmOut 类的 void write(float value)函数，并且将 12 位的正弦数据转化为浮点形式。

DAC 输出实现方式：通过 mbed 中 I2C 类的 int write(int address, const char *data, int length)函数发送数据到 MCP4725。

3.2. UNO/Arduino 示例程序

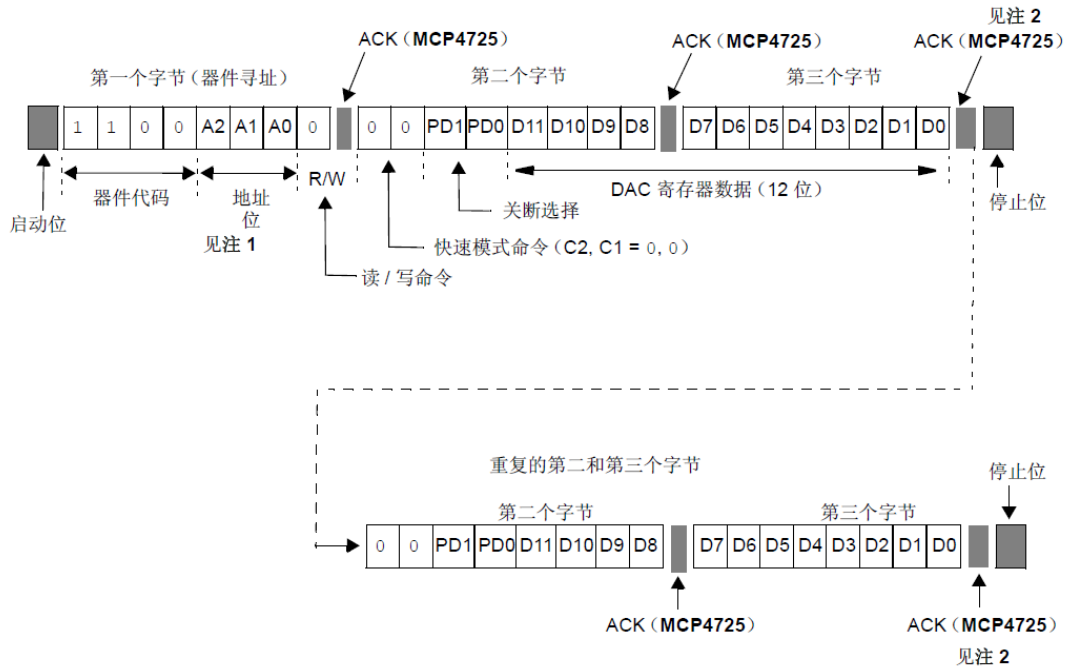
ADC 转换实现方式：调用 analogRead(pin)函数，读取模拟输入数值，返回值在 0-1023 之间。

PWM 输出实现方式：使用 analogWrite(pin, value)函数，value 的值在 0-255 之间。

DAC 输出实现方式：该方式基于 Wire 库，通过 I2C 接口控制 MCP4725 芯片输出模拟信号。

MCP4725 快速写命令如下图，传输三个字节，第一个为地址，第二第三个地址为数据。

使用快速模式写命令写 DAC 寄存器 (C2,C1) = (0,0)



- 注 1: A2 和 A1 位是在出厂前通过硬连线方式编程的, 而 A0 位由 A0 引脚的逻辑状态决定。
 注 2: 器件在第三个字节的 ACK 脉冲信号的下降沿更新 V_{OUT} 。

参见 MCP4725-Datasheet.pdf 图 6-3。

4. 快速入门

4.1. 准备工作

- Arduino 软件（请从这里下载：<http://www.arduino.cc/en/Main/Software>）
- Analog Test Shield x 1
- Xnucleo 或者 Arduino UNO PLUS 主板

4.2. 编译基本操作

产品附带的程序都需要编译和下载。基于 mbed 库的程序都可以通过 mbed 在线编译器或者离线编译器（例如 keil）进行编译。而基于 Arduino 的程序，都可以使用 Arduino IDE 进行编译。再把编译通过生成的文件下载到芯片，具体的下载方法则根据连接 shield 的开发板芯片的不同而定。如果您已经熟悉了 mbed 或者 Arduino 的编译方法，请直接跳过本节。

4.2.1. 使用 mbed 编译

本节主要说明 mbed 在线编译器（请从这里注册并登录：<http://developer.mbed.org/>）的基本操作。包括官方 mbed 库的导入、程序编译。关于 mbed 在线编译器的更多详情，请查阅 [ARM mbed 官网](#)。首次登陆，mbed 编译器会要求您选择开发板平台。选定之后，您也可以可以在 User's platforms 新增、删除或者更改您的开发板平台。请根据实际情况，选择您的开发板平台。

● 资源的导入

如需导入官方库：进入 mbed 在线编译器 -> 点击左上角的 **Import** -> 点击 **Libraries** 选项卡 -> 双击要导入的库，例如 mbed。

如需导入本地资源：进入 mbed 在线编译器 -> 点击左上角的 **Import** -> 点击 **Upload** 选项卡 -> 从本地电脑选择要导入的资源。在线编译器可以支持上传 zip 打包文件，也可以支持单个文件导入。详情请参考 mbed 在线编译器（<https://developer.mbed.org/compiler/>）右上角的 **Help** 关于 Importing 的内容。

● 程序编译

导入完成之后，选择相应的工程或者文件，点击 **Compile** 即可编译。编译成功的话，浏览器会提示您下载编译通过的 .bin 文件。把该 .bin 文件保存在本地，再用相应的方法把它烧录到开发板的芯片中即可。如果编译不通过，请根据编译器的提示进行除错。

注意，用户不一定要使用 mbed 在线编译工程，也可以使用离线的 IDE 例如 Keil。示例程序如果使用了 mbed 库的话，可以从 mbed 中导出 mbed 库到本地。或者直接使用配套软件自带的库。



4.2.2. 使用 Arduino IDE 编译

本节主要说明 Arduino IDE（请从这里下载：<http://www.arduino.cc/en/Main/Software>）的基本操作，包括库的导入、程序编译。关于 Arduino IDE 的更多详情，请查阅 [Arduino 官网](#)。请参照软件提示以安装软件。

● 库的导入

- 1) 将示例程序中的 libraries 文件夹复制到 Sketchbook location 所指定的位置，这个位置通常是 C:\Users\Username\Documents\Arduino。您可以通过设置首选项变更这个位置：点击 File -> Preferences。
- 2) 重启软件，选择 Sketch -> Import Library，查看是否包含之前导入的库。若有，说明库导入成功。

● 程序编译和下载

- 1) 程序可以使用 AVR 下载器通过 ISP 口下载，也可以通过 Bootloader 直接用 USB 接口下载。如果用 AVR 下载器下载的话，请参考下载器的相关文档。本文采用 Bootloader 方法下载。插上 USB 接口将开发板的 BOOT 开关（如果有的话）设置到 ON 档，VCC 电压设置为 5V。
- 2) 接入 Shield 模块，通过 USB 口连接 Arduino 开发板和电脑。
- 3) 这里以 Arduino IDE 自带的示例程序来描述程序的编译及下载方法。依次点击 File -> Example -> 01.Basics -> Blink 即可打开示例程序。
- 4) 点击 Tools -> Board -> Arduino Uno 选择 Arduino 开发板型号。
- 5) 点击 Tools -> Port:选择 Arduino 开发板的串口号，您可以通过设备管理器来查看对应的串口号。
- 6) 点击 Verify 按钮 ，开始编译程序。如果软件提示“Done compiling”，则说明编译成功。
- 7) 点击 Upload 按钮 ，开始下载程序。

4.3. 操作与现象

下面章节以本公司两款不同类型的开发板为例，描述具体操作步骤及实验现象。

4.3.1. 设置 Analog Test Shield 跳线

- 1) 引脚 D6 连接到 PWM，该引脚用于脉冲宽度调试。
- 2) 引脚 A0 连接到 ADC，该引脚用于模拟量输入。
- 3) 设置 PWM 和 DAC 引脚的选择跳线。LED 接到 PWM 输出，SPK 接到 DAC 输出。

4.3.2. 搭配 XNUCLEO-F103RB（主控芯片 STM32F103R）

- 1) 该示例程序基于 mbed 库。进入 mbed 在线编译器之后，导入 mbed 官方库。
- 2) 导入配套程序中的.\Test\src\main.cpp。
- 3) 编译这个 main.cpp 程序，并保存新生成的.bin 文件到本地。
- 4) 把这个.bin 文件烧录到 XNUCLEO-F103RB 开发板中。例如使用 [STM32 ST-LINK Utility](#) 软件烧录。
- 5) 烧录完成之后，可以看到 LED 渐变闪烁。
- 6) 接上 USB 电源，会听到喇叭响。调节 Volume 电位器可以改变声音大小。

注意，下载程序成功之后，如果喇叭不响，通常是由于供电不足。请使用外部供电或者接上 USB 电源。



- 7) 连接板载的 USB TO UART 接口到电脑，打开串口监视软件（例如 PuTTY, secureCRT 等），选择正确串口号并设置：


波特率：9600；数据位：8；停止位：1；校验位：None；控制流：None。

调节 Analog 电位器可以改变输入电压。串口不断显示当前模拟输入电压值，例如：

```
The analog input is :0mv
The analog input is :232mv
The analog input is :614mv
The analog input is :1198mv
The analog input is :1584mv
The analog input is :2465mv
The analog input is :3001mv
The analog input is :3248mv
The analog input is :3248mv
The analog input is :3250mv
```

4.3.3. 搭配 Arduino UNO PLUS

- 1) 通过 USB 口连接开发板到电脑。
- 2) 运行 Arduino 软件，打开 .\Analog_Test_Shield\Analog_Test_Shield.ino。
- 3) 点击 Verify 按钮 ，开始编译程序。如果软件提示“Done compiling”，则说明编译成功。
- 4) 点击 Tools -> Port:选择 Arduino 开发板的串口号，您可以通过设备管理器来查看对应的串口号。
- 5) 选择 Tools -> Board: -> Arduino Uno 选择 Arduino 开发板型号。只需要配置一次，之后操作都会默认使用该型号。
- 6) 点击 Upload 按钮 ，开始下载程序。

- 7) 烧录完成之后，可以看到 LED 渐变闪烁。
- 8) 接上 USB 电源，会听到喇叭响。调节 Volume 电位器可以改变声音大小。
注意，下载程序成功之后，如果喇叭不响，通常是由于供电不足。请使用外部供电或者接上 USB 电源。
- 9) 点击  打开 Arduino IDE 自带的串口监视器。选择正确串口号并设置为：

No line ending, 9600 baud

调节 Analog 电位器可以改变输入电压。串口不断显示当前模拟输入电压值，例如：

```
The analog input is :0mv
The analog input is :232mv
The analog input is :614mv
The analog input is :1198mv
The analog input is :1584mv
The analog input is :2465mv
The analog input is :3001mv
The analog input is :3248mv
The analog input is :3248mv
The analog input is :3250mv
```

5. 版本修订

版本号	修改处	发行时间
1.0	初稿	2015 年 5 月 28 日
1.1	STM32 程序更新 Arduino 程序更新 操作步骤更新	2015 年 6 月 19 日