



# Analog Test Shield

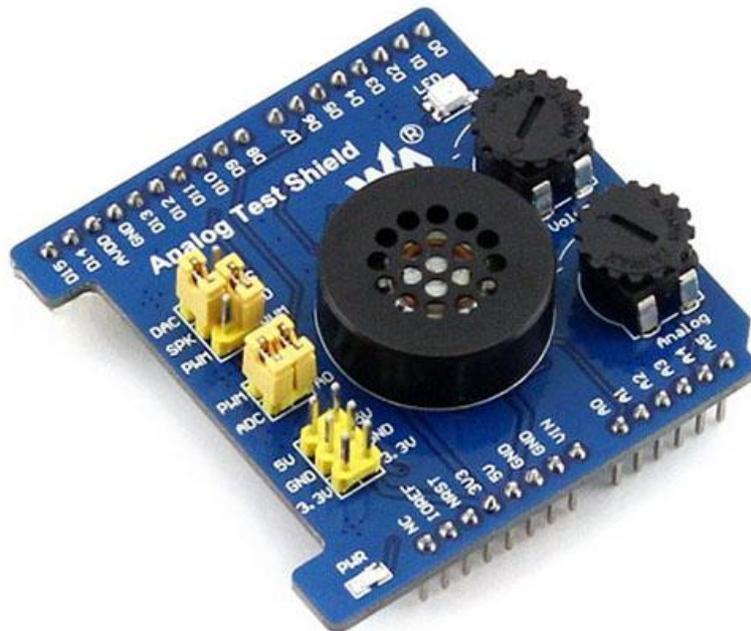
## User Manual

### 1. Overview

Analog Test Shield is an Arduino expansion board, provides functionalities of both AD acquisition and DA output. It's an easy way to know about the analog electronics.

#### Features

- Arduino standard interfaces, compatible with Arduino boards like Arduino UNO, Leonardo, NUCLEO, XNUCLEO
- Rotate the potentiometer to adjust AD sampling data
- Onboard 12bit DAC MCP4725, outputs required wave form
- Demonstrate the DAC/PWM effects via LED and buzzer



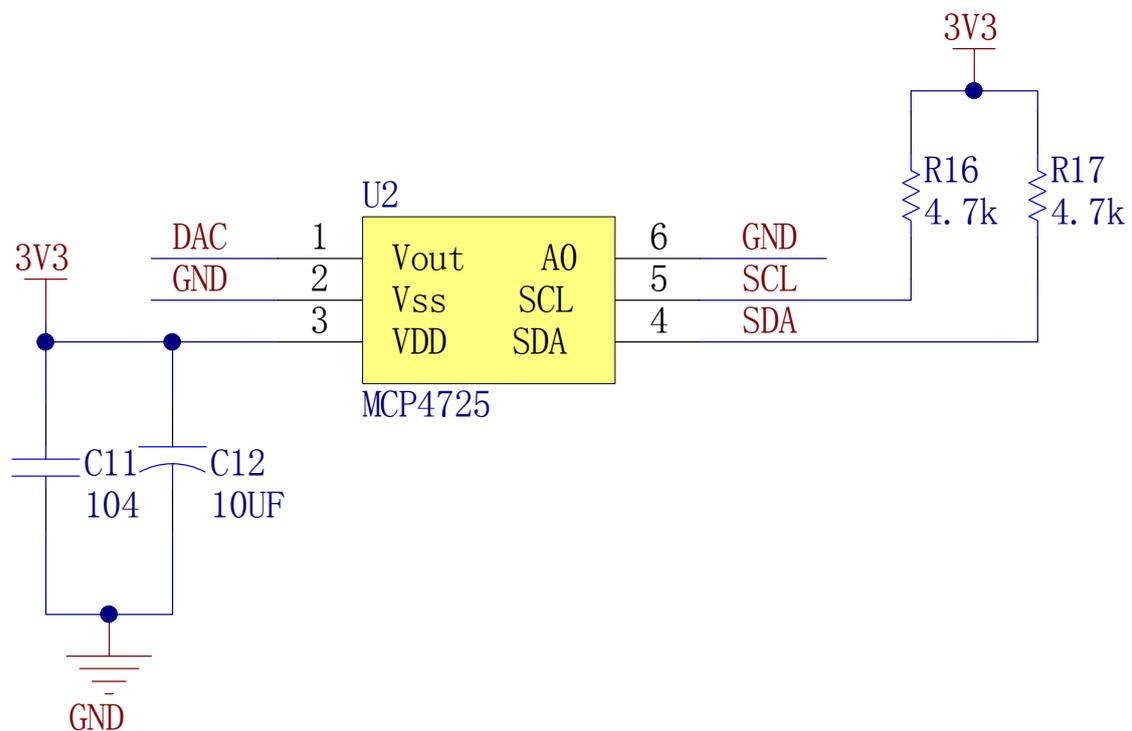
## 2. Hardware description

The Analog Test Shield integrates DAC circuit and amplifier circuit. The main resource on board: Potentiometer knob x 2, PWR LED x1, Status LED x1, Speaker x 1. You can collect the potentiometer's voltage by AD conversion. Control LED brightness and drive speaker by DA conversion or PWM (Pulse Width Modulation). Here's a brief description about the principle of DAC circuit and amplifier circuit. Please refer to MCP4725-Datasheet.pdf and LM386-Datasheet.pdf for more details.

### 2.1. DAC circuit

The MCP4725 is a low-power, high accuracy, single channel, buffered voltage output 12-bit Digital-to-Analog Converter (DAC) with non-volatile memory (EEPROM), which can be controlled by the I2C interface. The DAC circuit works on Single-Supply Operation: 2.7V to 4.5V. As shown in the following figure:

## DAC Circuit



#### Pin descriptions:

MCP4725	Name	Description
SOT-23		

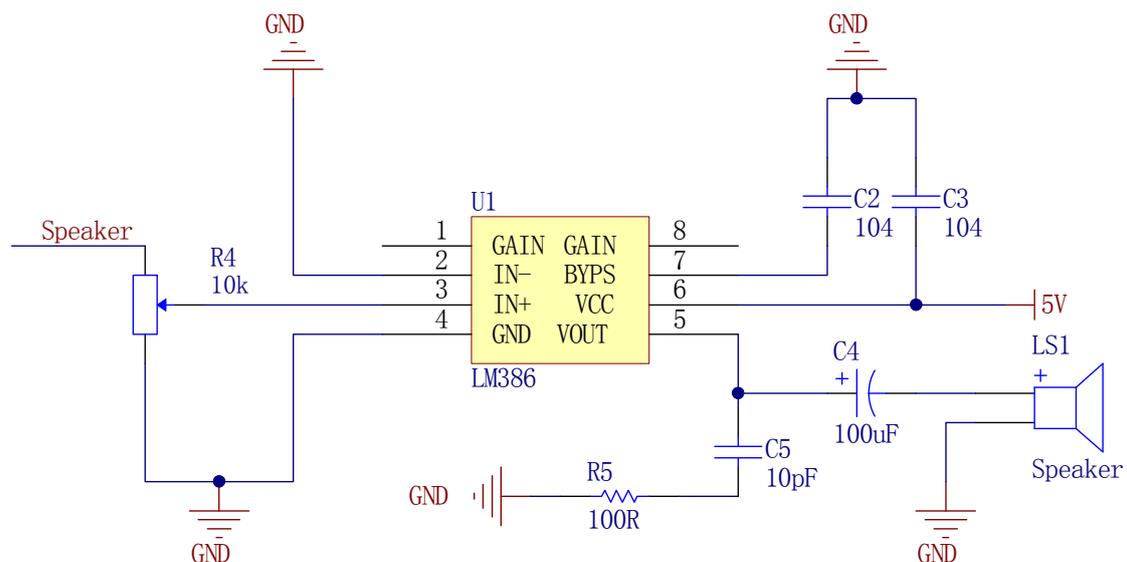
1	V <sub>OUT</sub>	Analog Output Voltage
2	V <sub>SS</sub>	Ground Reference
3	V <sub>DD</sub>	Supply Voltage
4	SDA	I <sup>2</sup> C Serial Data
5	SCL	I <sup>2</sup> C Serial Clock Input
6	A0	I <sup>2</sup> C Address Bit Selection pin (A0 bit). This pin can be tied to V <sub>SS</sub> or V <sub>DD</sub> , or can be actively driven by the digital logic levels. The logic state of this pin determines what the A0 bit of the I2C address bits should be.

Please refer to the section 3.0 of MCP4725-Datasheet.pdf for more details.

## 2.2. LM386 amplifier circuit

The onboard LM386 is used as an audio power amplifier. The gain is internally set to 20 to keep external part count low, but the addition of an external resistor and capacitor between pins 1 and 8 will increase the gain to any value up to 200.

### Speaker Circuit



#### Pin description:

Pin	Name	Description
1	GAIN	Provided for gain control.
2	IN-	Inverting input
3	IN+	Non-inverting input
4	GND	Ground
5	VOUT	Voltage output
6	VCC	Supply voltage
7	BYPS	Provided for bypassing connection. See the LM386 datasheet.
8	GAIN	Provided for gain control.

Please refer to the [LM386-Datasheet.pdf](#) for more details.

### 3. Demo description

The companion-programs-principle of the product is: The PWM outputs sine waves to control LED gradient. The DAC outputs sine waves to control the speaker. The analog inputs are read once per second. And they will be converted to digital output, printed through the serial,

#### 3.1. XNUCLEO demo

The demo is based on mbed platform.

How to implement ADC conversion: using mbed library-> Class AnalogIn -> Function float read() to read AD conversion data.

How to implement PWM output: using mbed library -> Class PwmOut -> Function void write(float value), and converting 12bits sine wave data to float form.

How to implement DAC output: using mbed library -> Class I2C -> Function int write(int address, const char \*data, int length) to send data to MCP4725.

#### 3.2. UNO/Arduino demo

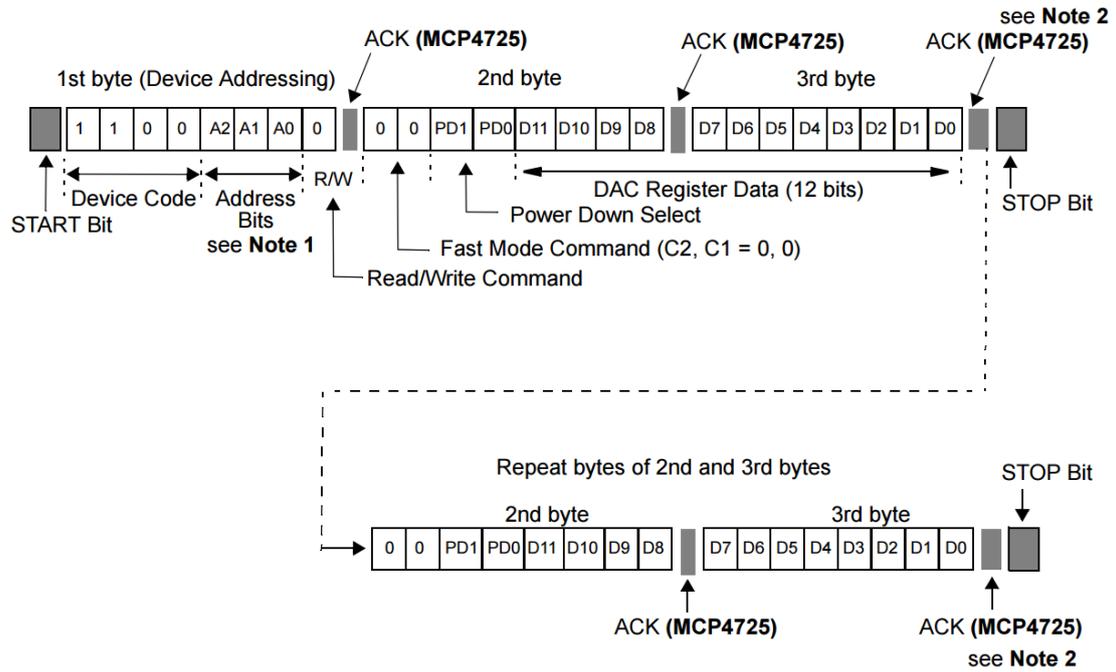
How to implement ADC conversion: using the function analogRead(pin) to read analog input value, return a number between 0 and 1023.

How to implement PWM output: using the function analogWrite(pin, value). The "value" is a number between 0 and 255.

How to implement DAC output: this manner is based on Wire library, using I<sup>2</sup>C to control analog output of MCP4725.

MCP4725 Fast Mode Write Command is shown in the following figure. 3 bytes transmission, 1<sup>st</sup> byte is Device Addressing, 2<sup>nd</sup> and 3<sup>rd</sup> bytes are Register Data.

**Write DAC Register using Fast Mode Write Command: (C2, C1) = (0, 0)**



**Note 1:** A2 and A1 bits are programmed at the factory by hard-wired, and A0 bit is determined by the logic state of A0 pin.

**2:** The device updates  $V_{OUT}$  at the falling edge of the ACK pulse of the 3rd byte.

Please see the figure 6-1 of MCP4725-Datasheet.pdf.

## 4. Getting started

### 4.1. Preparation

- Arduino software (Please download it from: <http://www.arduino.cc/en/Main/Software>)
- Analog Test Shield x 1
- Xnucleo or Arduino UNO PLUS board

### 4.2. Basic compiling operation

All of the program of the present product should be compiled and programmed. The program based on mbed library can be compiled via mbed online compiler or offline compiler (like: Keil). And all of the programs based on Arduino can be compiled by the software Arduino IDE. Then, you can program the generated file from the compiler to the chip. The program manner depends on the chip, which is connected to the shield, to be programmed. If you have mastered the compiled steps of mbed or Arduino, please skip this section.

#### 4.2.1. To use mbed online compiler

Here is a brief description of mbed online compiler (login or signup to <http://developer.mbed.org/>). This description contains mbed library import and program compiling. Please refer to [ARM mbed website](#) for more details. The mbed requires your development platform at your first login. After platform selected, you can add, remove or change it in the User's platforms window. Please choose your actual development platform.

- **Resource importing**

How to import official library: login mbed online compiler -> click **Import** on the top left -> click **Libraries** tab -> double click the library to be imported, like mbed.

How to import local resource: login mbed online compiler -> click **Import** on the top left -> click **Upload** tab -> browse in your PC and upload it. The online compiler allows zip archive files upload. And also allows single file upload. Please click the **Help** on the top right of mbed online compiler (<https://developer.mbed.org/compiler/>) to see the Importing content.

- **Compiling**

After sources imported, please choose the project or files accordingly, and click **Compile**. If done compiling, the browser will prompt to download a .bin files. Save it to local and program it to the chip on your development board. If compiling error, please debug according to the compiler hints.

Note, you haven't to compile your project using mbed. Anyway, many offline compilers can be chosen, (Keil, etc). Even If the mbed library is used by the example, you can export the library to local. Or use the products supporting program directly.

---

#### 4.2.2. To use Arduino IDE

Here is a brief description of Arduino IDE (Please download it from <http://www.arduino.cc/en/Main/Software>). This description contains library import and program compiling. Please refer to [Arduino web site](#) for more details. First of all, please install the software according to the prompt.

- **Library importing**

- 1) Copy the libraries directory of the demo to the position of "Sketchbook location". It's often in C:\Users\Username\Documents\Arduino and can be changed by setting preferences (File -> Preferences).
- 2) Restart the software, select Sketch -> Import Library, check whether the library has imported, if yes, the importing completes successfully.

- **Compiling and programming**

- 1) The chip can be programmed by AVR programmer through ISP interface, and it can also be programmed by Bootloader through USB port. If an AVR programmer used, please see related document. In the present manual, Bootloader is chosen for programming. Plug in USB port and set the BOOT switch (if exist) to ON, and set VCC to 5V.
- 2) Connect the Shield to your Arduino board and connect the Arduino board to a PC via USB port.
- 3) This document presents the method of program compiling and uploading with the Demo provided by Arduino IDE. Click File -> Example -> 01.Basics -> Blink, to open the Demo.
- 4) Click Tools -> Board -> Arduino Uno, to select relative Arduino development board.
- 5) Click Tools -> Port: to configure the serial port of the Arduino development board for uploading. Please take a note that the port number in used is various with different PCs, so you need to check the actual serial port number by Device Manager before making the setting.
- 6) Click  (Verify) and try to compile the project until it prompts "Done compiling".
- 7) Click  (Upload) to program the chip.

#### 4.3. How to operate

Here 2 development boards, made by Waveshare, are used to demonstrate the operation steps and corresponding phenomena.

#### 4.3.1. Set the jumpers on Analog Test Shield

- 1) The pin D6 is connected to PWM for Pulse Width Modulation.
- 2) The pin A0 is connected to ADC for Analog Input.
- 3) Set the jumpers PWM and DAC. Connect LED to PWM output, and SPK to DAC output.

#### 4.3.2. Using the shield with XNUCLEO-F103RB (onboard chip STM32F103R)

- 1) This demo is based on mbed library. Login mbed online compiler and import mbed official library.
- 2) Import the program `.\Test\src\main.cpp`.
- 3) Compile the `main.cpp`, and then save the generated `.bin` file to local PC.
- 4) Program the `.bin` file to your XNUCLEO-F103RB board e.g. you can program it by the software [STM32 ST-LINK Utility](#).
- 5) After programmed, the brightness of the onboard LED will be changed gradually.
- 6) The speaker will ring as long as USB power supplied. You can turn the Volume potentiometer to adjust the loudness.

**Note, if the speaker does not ring but the chip has been programmed, it's often due to the insufficient power supply. Make sure the board has been supplied by external power, or supplied by USB.**

- 7) Connect the onboard USB TO UART port to a PC, run a serial monitor software (PuTTY, secureCRT, etc.), choose the correct COM port and configure as:

Baud rate: 9600; Data bits: 8; Stop bits: 1; Parity: None; Flow control: None.

Adjust the Analog potentiometer to change input voltage. The serial port will print the present analog input continuously. For example:

```
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 :3250mv
```

### 4.3.3. Using the shield with Arduino UNO PLUS

- 1) Connect the board to a PC via USB port.
- 2) Run Arduino software, open .\Analog\_Test\_Shield\Analog\_Test\_Shield.ino.
- 3) Click  (Verify) and try to compile the project until it prompts “Done compiling”.
- 4) Click Tools -> Port: to configure the serial port of the Arduino development board for uploading. Please take a note that the port number in used is various with different PCs, so you need to check the actual serial port number by Device Manager before making the setting.
- 5) Click Tools -> Board -> Arduino Uno, to select relative Arduino development board. Since the setting will be saved automatically, you only need to configure it once.
- 6) Click  (Upload) to program the chip.
- 7) After programmed, the brightness of the onboard LED will slowly changing.
- 8) The speaker will ring as long as USB power supplied. You can turn the Volume potentiometer to adjust the loudness.

**Note, if the speaker does not ring but the chip has been programmed, it's often due to the insufficient power supply. Make sure the board has been supplied by external power, or supplied by USB.**

- 9) Click  (Arduino Serial Monitor) and choose the correct COM port and configure as:

No line ending, 9600 baud

Adjust the Analog potentiometer to change input voltage. The serial port will print the present analog input continuously. For example:

```
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 :3250mv
```

## 5. Revision history

Revision	Changes	Date
1.0	Initial release	May 28 2015
1.1	Updated STM32 program. Updated Arduino program. Updated operation steps.	June 19 2015