

Analog Shield: First Time Setup

Revised June 2, 2013

Author: William Esposito

This document was edited and adapted for MPIDE by Digilent. The original document was prepared by William Esposito at Stanford.

Installing the Libraries

Before the Digilent Analog Shield can be used, the specific libraries need to be installed. To install the library, first open Arduino IDE if you are using an Arduino™, or MPIDE if you are using a chipKIT™. Open up the preferences file by navigating to **File -> preferences**. It should look similar to Fig. 1.

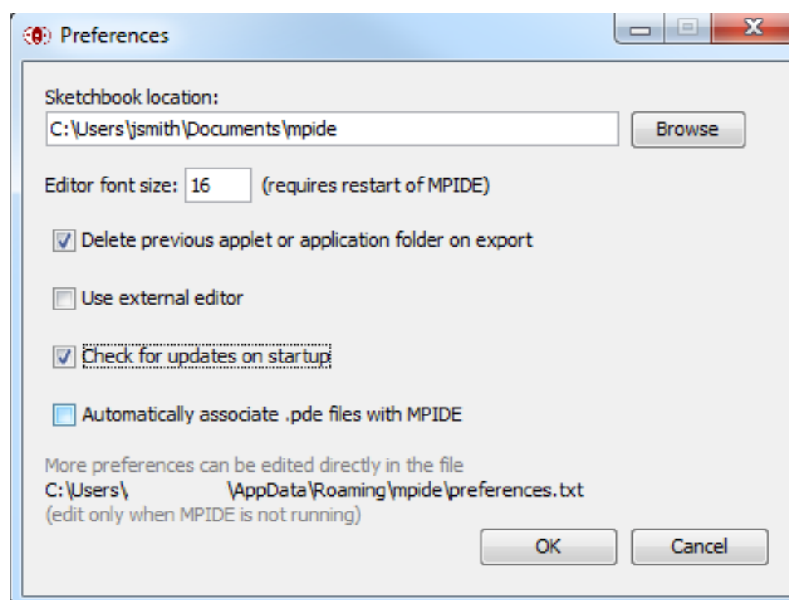


Figure 1. Preferences menu.

Screenshot of MPIDE running on Microsoft Windows 7.

Take note of the sketchbook location. Close the IDE and navigate to where the sketchbooks are located. If there is not a folder labeled as “libraries” in the sketchbook’s location, then you will need to create one. Navigate into the libraries folder and copy the contents of the library zip file to the library directory.

Once the folder has been copied to the libraries folder, open the IDE. The new library should be available under **File -> sketchbook -> libraries**.

There are demo projects bundled with the Analog Shield library, which can be found under **File -> sketchbook -> libraries -> AnalogShield**.

Any user installed library will go into the “libraries” folder. Do not be surprised if there are libraries already in the “libraries” folder. The file structure is important to the IDE and should look similar to Fig. 2.

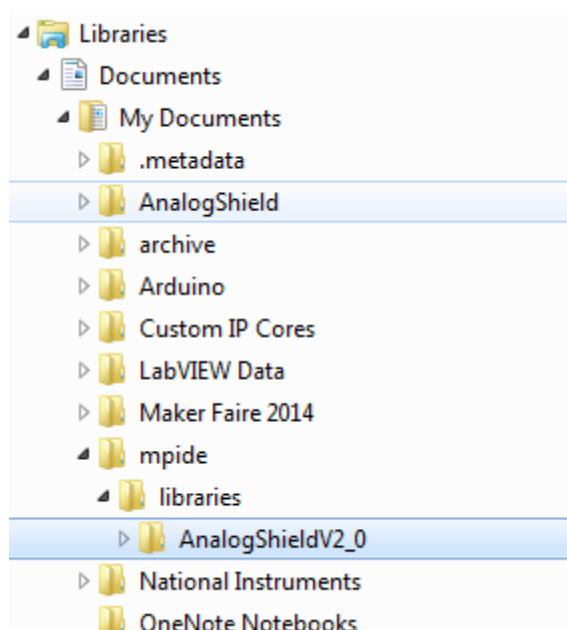


Figure 2. File structure of user defined libraries.

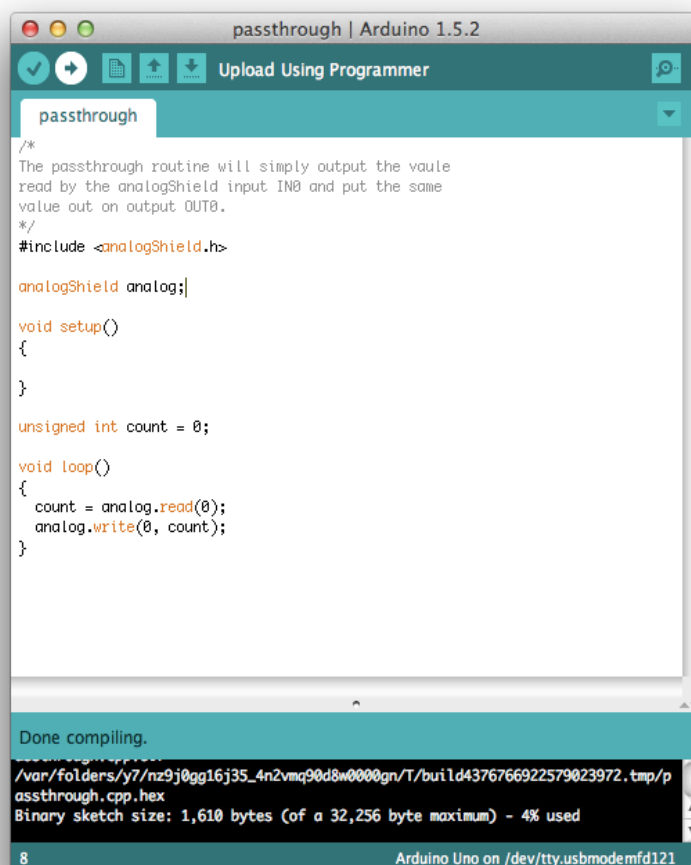
Screenshot of directory on Microsoft Windows 7.

Note that there are several other folders depicted here, each one an unrelated Arduino/MPIDE sketch. It is perfectly fine to have other folders within the “libraries” folder. If analogShield.cpp is not on the first folder level below AnalogShield, then the Arduino IDE will likely be unable to find the files and fail to compile your sketch. If there is not already a ‘libraries’ folder within your Arduino/MPIDE sketch folder, create a folder called “libraries” and put the code in it. Arduino/MPIDE will find it once the IDE is restarted.

First Sketch

To start using the Analog Shield in your sketches, a `#include<analogShield.h>` statement is required the top of the sketch, like any other library. Once the analogShield library is included, the read and write to the inputs and outputs of the shield can be read by calling the `analog.read()` and `analog.write()` methods, respectively.

An extremely simple sketch is included as an example with the Analog Shield library, called “passthrough,” and can be found within the example’s dropdown list once the library has been installed. This pass through library takes a voltage measured on input pin A0 of the Analog Shield and replicates it on the output pin D0. Figure 3 below shows an example of the pass through sketch.



```
passthrough | Arduino 1.5.2
Upload Using Programmer

passthrough
/*
The passthrough routine will simply output the value
read by the analogShield input IN0 and put the same
value out on output OUT0.
*/
#include <analogShield.h>

analogShield analog;

void setup()
{
}

unsigned int count = 0;

void loop()
{
  count = analog.read(0);
  analog.write(0, count);
}

Done compiling.
/var/folders/y7/nz9j0gg16j35_4n2vmd90d8w0000gn/T/build4376766922579023972.tmp/passthrough.cpp.hex
Binary sketch size: 1,610 bytes (of a 32,256 byte maximum) - 4% used
8 Arduino Uno on /dev/tty.usbmodemfd121
```

Figure 1. Simple pass through example.

Screenshot of Arduino IDE running on Mac OS.

Try uploading this sketch to the Analog Shield and applying a test voltage from the Analog Shield's onboard supply using a breadboard wire. Measure the output using a voltmeter connected between D0 and ground.

A number of other, more complex examples have been developed, and are discussed in more detail in a series of "Instructable" style application notes.