1300 Henley Court
Pullman, WA 99163
509.334.6306
www.digilentinc.com

Analog Shield Reference Manual

Revised June 5, 2014
Author: William Esposito

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

Overview

The Analog Shield offers a high fidelity and easy way to connect your Arduino™ or chipKIT™ to analog circuits. In order to do this, the Digilent Analog Shield* provides:

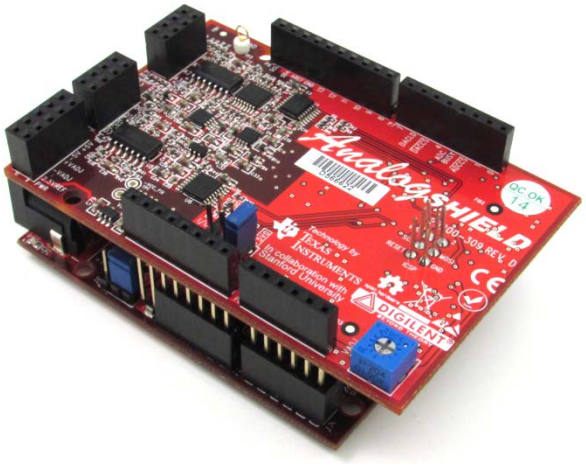

Figure 1. The Analog Shield.

- 4 Channel, 16 bit 100 ks/s SAR ADC.
- 4 Channel 16 bit 100 ks/s string DAC.
- Variable +/-7.5V Supply.
- Fixed +/-5V Supply.
- A small breadboard.
- Software read and write to the ADC/DAC with a single line of C.

In addition to providing many of the features available on other prototyping or breadboard shields, this shield provides substantially more sensitive inputs and outputs than the Arduino Uno™ or the chipKIT UNO32™, as well as providing the bipolar inputs and power supplies needed to drive many useful analog circuits.

The Analog Shield communicates with the Arduino using the SPI communication protocol. The Analog-to-Digital input converter (ADC) and Digital-to-Analog output converter (DAC) both use the standard Arduino form factor SPI bus pins and use independent chip selects.

** The Analog Shield was developed in cooperation with Texas Instruments® and the Kovacs / Giovangrandi lab at Stanford University.*

1 Applications

The Analog Shield allows better resolution for smaller and more sensitive signals that cannot reliably be recorded or generated using the standard pins on the Arduino.

A few demos using the Analog Shield have been developed:

- Sine wave / function generator (using direct digital synthesis).
- FFT spectrum analyzer.
- Polyphonic Music generation.
- 4-channel low bandwidth oscilloscope.
- Lissajou diagram generator.
- XY-Analog scope display driver.

These demos are each explained in their own associated documentation.

2 Hardware

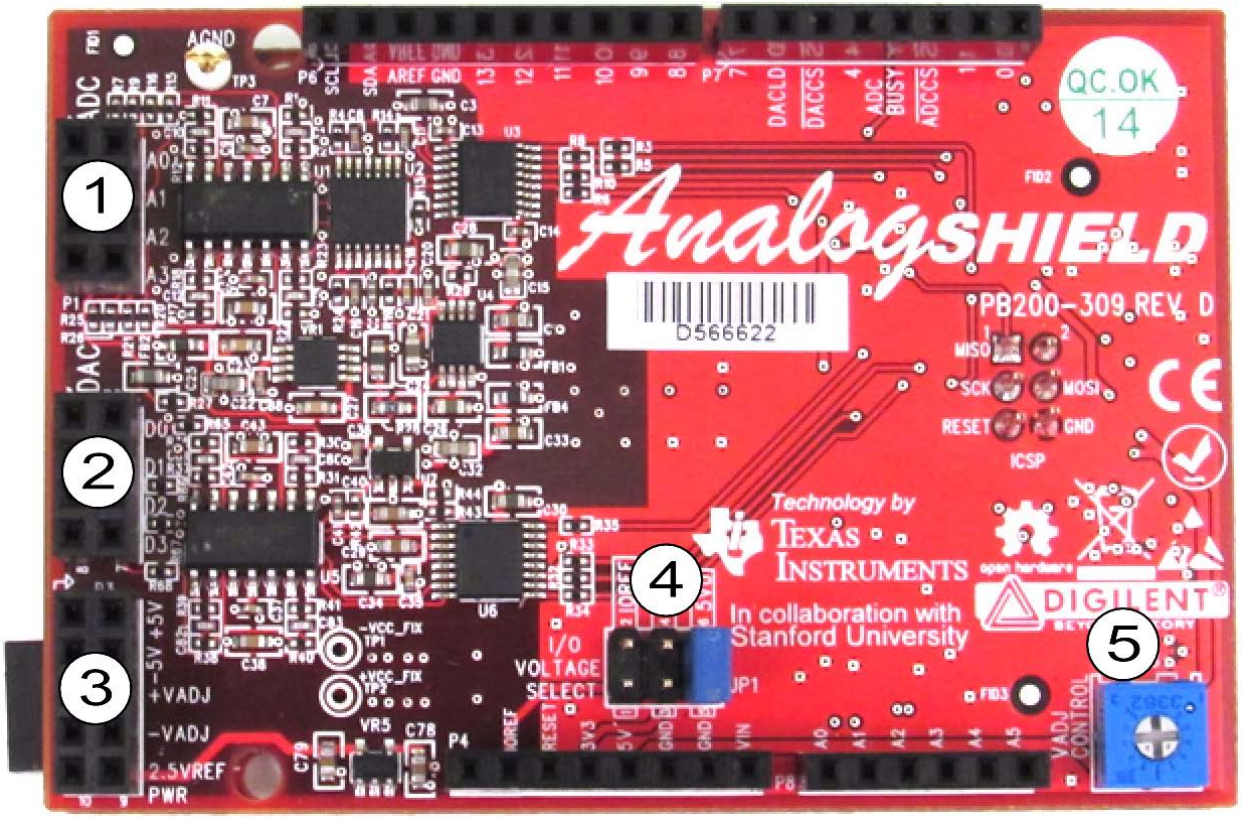

Figure 2. The analog shield with Hardware callouts.

1. Analog to Digital Converter (ADC) Header

The 2x4 header has the four ADC channels and four pins for the common ground on the board. The 4 ground pins are on the outside row of the header, while the 4 ADC channels are on the inner row of the header. The ADC can read 4-channels of data at 16-bits per channel using the analog.read() function, which are labeled with A0- A3. For more detail on the op-amp circuitry ahead of the ADC, please refer to the shield schematic.

2. Digital to Analog Converter (DAC) Header

The 2x4 header has the four DAC channels and four pins for the common ground on the board. The 4 ground pins are on the outside row of the header, while the 4 DAC channels are on the inner row of the header. The DAC modifies the analog signal produced from an Arduino or chipKIT.

The standard Arduino/chipKIT analogWrite() function produces a pulse-width modulated (PWM) square wave that approximates the desired overall value over time. PWM is acceptable for many applications, like dimming an LED or driving a motor, but it is not adequate for more sensitive analog circuits. Once a value is set to a DAC channel, the channel will produce a steady voltage corresponding to the provided value. Setting values on the channels is facilitated with the analog.write() function, and the channels are labeled D0-D4. For more detail on the op-amp circuitry behind the DAC, please refer to the shield schematic.

3. Power Header

The 2x5 header has a positive 5V rail, a negative 5V rail, a positive variable voltage rail, a negative voltage rail, 2.5V reference voltage rail, and five pins for the common ground on the board. The 5 ground pins are on the outside row of the header while the 5 power supplies are on the inner row of the header.

The +/-5V rail can vary due to USB power specification. The +/-V_{adj} rails can be adjusted using the variable voltage potentiometer located at the lower right corner of the shield. Adjusting the potentiometer will change the voltages of the +/-V_{adj} rails. There is also a +2.5V rail to be used as a reference voltage.

4. Voltage Select Jumper

The jumper needs to be set for whether the connected board uses IOREF, 3V3, or 5V0 for the I/O for the SPI interface.

5. Variable Voltage Potentiometer

The potentiometer is can be used to adjust the +/-V_{ADJ} power supply rails.

3 Software

Included with the Analog Shield is a simple library that is built to optimize readability and performance. It attempts to provide a similar interface to the standard Analog read and write functionality already present in the IDE. Instructions on installing the library are included in the library folder and in the "Analog Shield – 02 First Time Setup" document. After successfully installing the library into the IDE, the library can be invoked with the command:

```
#include <analogShield.h>
```

Once the analogShield library is included, a class variable called `analog` will be accessible. The analog variable will allow access to the analogShield library functions.

To sample one of the analog input channels, use the function below. read() returns the voltage read on the channel. The voltage is returned in a binary representation from 0 to 65535 for -5V to 5V.

```
unsigned int read(int channel, bool mode = false);
```

To read from channel 0 (without differential mode) the syntax will look like this:

```
unsigned int data;
data = analog.read(0);
```

This method samples the selected analog input. 'Mode' is an optional parameter. **False** (the default value) is the normal, single-ended mode where each analog pin is referenced to ground. **True** enables 'differential' mode, where the value returned is the difference between the voltages on pairs of adjacent inputs (i.e., A0-A1 and A2-A3). signedRead() acts the same at read() but returns a signed integer.

```
signedRead(int channel, bool mode = false);
```

To read from channel 0 (without differential mode) the syntax will look like this:

```
int data;
data = analog.read(0);
```

To write data to the DAC, there is a function called write(). This function allows the user to write to any channel: channel 0 and 1; channel 0, 1, and 2; or channel 0, 1, 2, and 3.

```
write(int channel, unsigned int value);
write(unsigned int value0, unsigned int value1, bool mode);
write(unsigned int value0, unsigned int value1, unsigned int value2,
      bool mode);
write(unsigned int value0, unsigned int value1, unsigned int value2,
      unsigned int value3, bool mode);
```

To write a value just for channel 1, the syntax will look like this:

```
unsigned int data = 65535;
analog.write(1, data);
```

To write multiple values to channels 0, 1, and 2 and have them update at the same:

```
unsigned int data0 = 65535;
unsigned int data1 = 256;
unsigned int data2 = 0;

analog.write(data0, data1, data2, true);
```

Simultaneously updating methods are useful for higher speed applications where a small delay between channel outputs will produce unwanted signals or timing issues. For example, to use the shield to drive the X-Y input of an oscilloscope to draw a circle, a delay between writing the X and the Y channels produces a ghost 'ellipse' drawn during the interim between the X and Y DAC updates.

This is a simple but powerful library to help the user get started. It is well documented to ease modification of the library for any individual project. The power supply is entirely analog, and as such, needs no software to operate. However, the ADC (part number DAC8564) and DAC (part number ADS8343) are both SPI devices. Their datasheets are freely available on the manufacturer's webpage.

Summary

Unlike the native analog input of the Arduino, the Analog Shield provides a 16 bit ADC, (as opposed to 10bits provided by the Arduino UNO and the ChipKIT UNO32) offering greater precision (about 25dB) than the Arduino or chipKIT. Additionally, the ADC works as a bipolar input, meaning that sense signals in the range of +/-5V can be sensed without any additional hardware.

The onboard DAC provides a similar improvement in precision. The Analog Shield provides 16 bits of precision, and offers as much as 25dB better signal-to-noise ratio when driving a sinusoidal output at low frequency.

The Analog Shield provides a compact integrated power supply with fixed and adjustable outputs to allow a variety of analog circuits to be assembled with a minimum of wasted space and complexity.

Special thanks to Gregory Kovacs, Fernando Mujica, Clint Cole, and their respective teams at Stanford, TI, and Digilent for supporting this project to its completion!