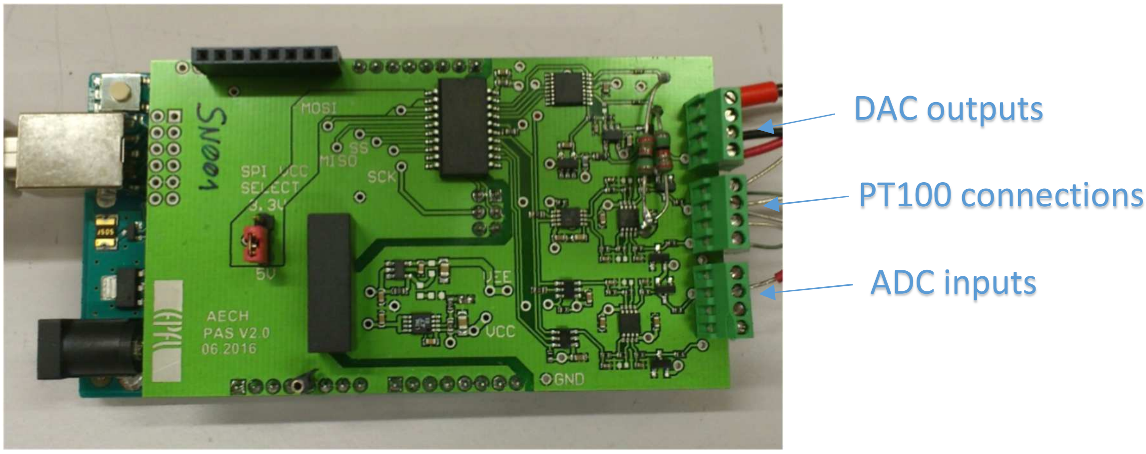
Precision Arduino Shield

EPFL SB ISIC-GE AECH, Benjamin LeGeyt, August 2016

The Precision Arduino Shield board is a small precision measurement board designed to be mounted on and controlled/read out by an Arduino. It provides 2 16-bit DAC outputs, two single-ended 16-bit ADC inputs, and a pseudo-differential ADC input and precision current source which can be used to measure temperature with a PT100 sensor. The DAC and all the ADCs can be operated at a rate of 100ksps.



The inputs and outputs are listed below, along with their ranges. The pin numbering convention is that pin #1 is the left-most pin when looking at the connector from the side where the wires enter.

ADC Input	Pin name	In/Out	Comment
1	ADC3	In	0-5V range
2	Ground	-	
3	ADC2	In	0-5V range
4	Ground	-	
PT100			
1	ADC1 +	In	Input range 0-5V Differential range 0 – 2.5V
2	ADC1 -	In	
3	Ground	-	
4	Current Source	Out	1mA
DAC Output			
1	DAC1	Out	0-5V range
2	Ground	-	
3	DAC2	Out	0-5V range
4	Ground	-	

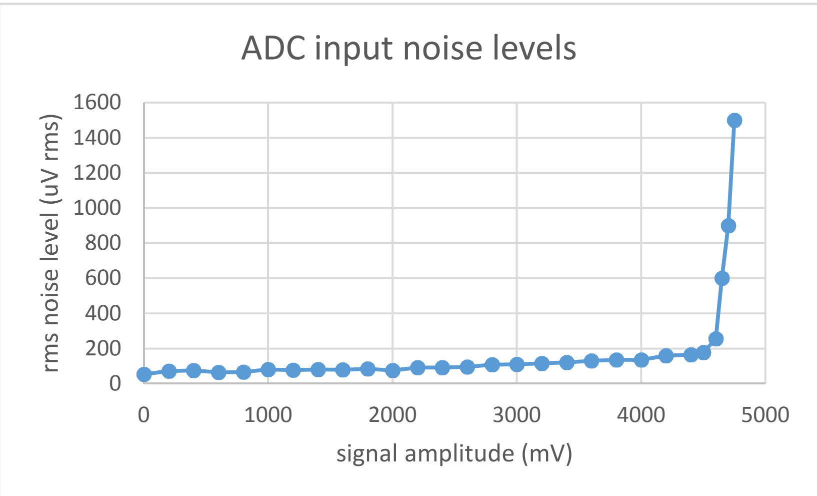
Theory of operation and performance

The Precision Arduino Shield (PAS) has 3 ADCs and one dual DAC all on a single SPI bus. The measurement side of the board (right side in picture) is galvanically isolated from the arduino's power supply via an isolated DC-DC converter. The SPI controls pass through a 'SPIsolator' which allows to select from 4 separate slaves with 2 address bits. The ADC inputs are buffered with precision amplifiers which are set to a gain of 1. These gains could in principle be changed to allow more precision for smaller signals.

The power is taken from the host board (Arduino or FPGA). There is an SPI voltage select jumper which allows to move between 5V-based systems (Arduino AVR) and 3.3V-based systems (Arduino SAM and digilent PMOD).

The ADCs have a relative accuracy of about 1 LSB but they have a zero offset of 1-2mV which needs to be calibrated out. The DACs are more accurate but also have a slight zero offset, which is addressed in the 'Modification' section. The performance, both with and without software calibrations is reported in the table below. All tests were performed at a -3dB bandwidth of 15kHz:

Spec	Normal	64x 'HiRes' Mode
Max sample rate	100ksps	1.56ksps
ADC inputs 2,3		
Zero-scale offset (typ, uncal)	1.5mV	1.5mV
Zero-scale offset (typ, cal)	10µV	3µV
Zero-scale noise level	56µV	7.3µV
1V scale error (typ, cal)	150µV	150µV
1V scale noise level	57µV	6.8µV
Full scale error (typ, cal)	1.5mV	1.5mV
Full scale noise level (4500mV)	250µV	20µV
Full scale noise level (4800mV)	20000µV	700µV
Useful range	0-4600mV	0-4800mV
Resolution (LSB)	76µV	1.2µV
RTD input		
Zero-degree offset (typ, uncal)	0.5°C	0.5°C
Zero-degree offset (typ, cal)	0.2°C	0.2°C
Typical noise level	0.18°C	0.02°C
DAC Outputs		
Zero-scale offset (typ, uncal)	300µV	-
Zero-scale offset (typ, cal)	10µV	-
Zero-scale noise level	35µV rms	-
1V scale error (typ, cal)	200µV	-
1V-scale noise level	144µV rms	-
Full-scale error (typ, cal)	1250µV	@4500mV
Full-scale noise level	600µV rms	@4500mV
Full-scale error (typ, cal)	10mV	@4850mV
Full-scale noise level	700µV rms	@4850mV



The ADC noise level is shown here in detail versus the signal amplitude to illustrate an effect that occurs near full scale. The noise level increases dramatically above 4.6V. This effect is likely the fault of the buffer amplifier rather than the ADC itself.

Modification

The DAC output is single-ended and it gives an output of 300µV when the DAC is set to zero. It was important for the LED controller application to have a value just below zero so a 1.5Ω 0402 resistor was added in series with the outputs and a 15kΩ resistor was connected from each DAC output to minus 5V in order to pull the output just slightly below zero.

Interface

The board was designed to be connected to an Arduino which supplies the SPI controls to drive the board. There is a connector underneath which mates to the Arduino SPI header and the connectors on the edges are used for power and chip select + address. The board can also be used as a digilent PMOD device, via an optional right-angle header, which would be on the left-hand side in the picture. This would provide access to the full sampling and data rate of the board, which the Arduino cannot access due to serial data limitations.

An Arduino sketch which provides access to the functionalities of the board via serial communication is available. In addition, a set of labview routines has been written to facilitate access to these functions. These are available in the 'Software' directory of the board documentation.