A Geothermal Reservoir Monitoring System 'Anyone' Can Program

By Randy Normann Feb 10, 2018

Abstract: *This paper examines the use of Arduino systems for developing geothermal well monitoring systems and logging tools. The Arduino is a low cost, public domain micro processing system commonly used for teaching machine programming at the high school level. These systems enjoy open domain support and many levels of self-help books. The Arduino enjoys worldwide acceptance for teaching high school students how to build and program robotic machines. The Arduino system is popular. Ebay lists 2,244 "Arduino Kits" with kits for robotics, 3D printers, sensors and other. This paper examines using the Arduino system for scientific measurement purposes. An application demonstrating the Arduino to operate and record a 275°C downhole well monitoring system. This paper highlights two new Arduino Shields developed for scientific measurements: a 22Bit Analog-to-Digital Converter (AD22B) and a 12Bit Digital-to-Analog Converter (DAC12B) low noise programmable power supply for powering sensors.*

Introduction

The author has been working in geothermal for approximately 25 years as an electric hardware engineer designing logging and drilling tools. The geothermal market is very small. The most popular logging tool is the well production logger (Pres, Temp and Spinner). The market for the production logger is estimated 20 at tools a year worldwide by the author. In the world of electronic manufacturing, a build of 20 is equal to 0. In fact, the oil industry builds around 200 production loggers. This is also considered near 0 by the electronics manufacturing industry. As such, even the much larger oil industry is a niche industry in the world of electronics. The best means for reducing costs within a niche market is to adapt technology from much larger markets. Geothermal needs to look beyond the oil industry for cost saving technology.

Now that the obvious is cleared away, time to talk about well monitoring. The Perma Works geothermal well monitoring tool is based around the same electronics developed for commercial aircraft engines. These electronics have a 5 year, 225°C operating specification. The future aircraft engine market is approximately 500,000 per year[1]. However, to support the downhole tool, we custom designed surface electronics to provide power, tool communication, data storage and downloading a laptop PC.

The cost of the surface electronics and developing software is significant. I estimate the software development for the surface electronics was well over $100K. Because of the small market and the cost of maintaining software companies can charge $20K just for surface hardware and software.

As an engineer hardware designer, I got tired of going to the programmers to make simple changes in the surface software to test a new tool upgrade. So, I designed a simple manual interface with push buttons and a voltage meter for manual reading of tool data. This works in the lab.

Enter the Arduino system. The Arduino system is based on 8bit microcontrollers from Microchip Technology, costing around $2 each. The Arduino people created a simple programming environment and provide a blog for users to share their work. They also released the Arduino board design as public domain. To learn more go to: https://www.arduino.cc/

The Arduino enjoys worldwide acceptance for teaching high school students how to build and program robotic machines. The Arduino system is popular in most high schools. For example, Ebay lists 2,244 "Arduino Kits" for robotics, 3D printers, sensors and other directed at high school students. Amazon.com lists more than 20 help books on working with the Arduino system. Many books are targeting high school age students while other are more technical.

Playing with the Arduino Uno (basic system), I realized programming a real world interface is simple. For example, reading one of the 6 analog-to-digital converters (these are 10bit, 0 to 5V) required only one programming statement.

```
int data1 = analogRead(0);
```

Here data1 is a 16 bit integer inside the program: 'int data1'. The program sets the variable of data1 equal to the 10bit binary reading from the Arduino analog input 0: 'data1 = analogRead(0)'. The ';' simply ends the statement. The Arduino Uno has 6 analog inputs 0-5.

To use a digital output as a switch to control some function is also easy. To turn on a digital pin to 'high' (~5V) and 'low' (~0V) with a delay of 20mS is illustrated below.

```
pinMode(2, OUTPUT);
…
digitalWrite(2,HIGH);
delay(20);
digitalWrite(2,LOW);
```

Here the pinMode statement is used in a part of the Arduino program which is run only once during boot-up. This tells the microcontroller that pin 2 is an output pin otherwise most pins are disabled at boot-up. We could have also defined this pin as an input pin.

After boot, the processor starts running the main program. Within the running program, setting pin 2 to high requires the 'digitalWrite(2,HIGH)' statement. The same statement is used to set pin 2 to a low value, replacing HIGH with LOW. The 'delay(20)' statement tells the microcontroller to wait 20mS. We could ask it to wait 200mS or 20000mS, if desired.

This created a simple idea, why not use the Arduino to control the manual surface electronics used to run the downhole geothermal well monitoring tool? So, the photo shows such a system working with a high temperature geothermal well monitoring tool.

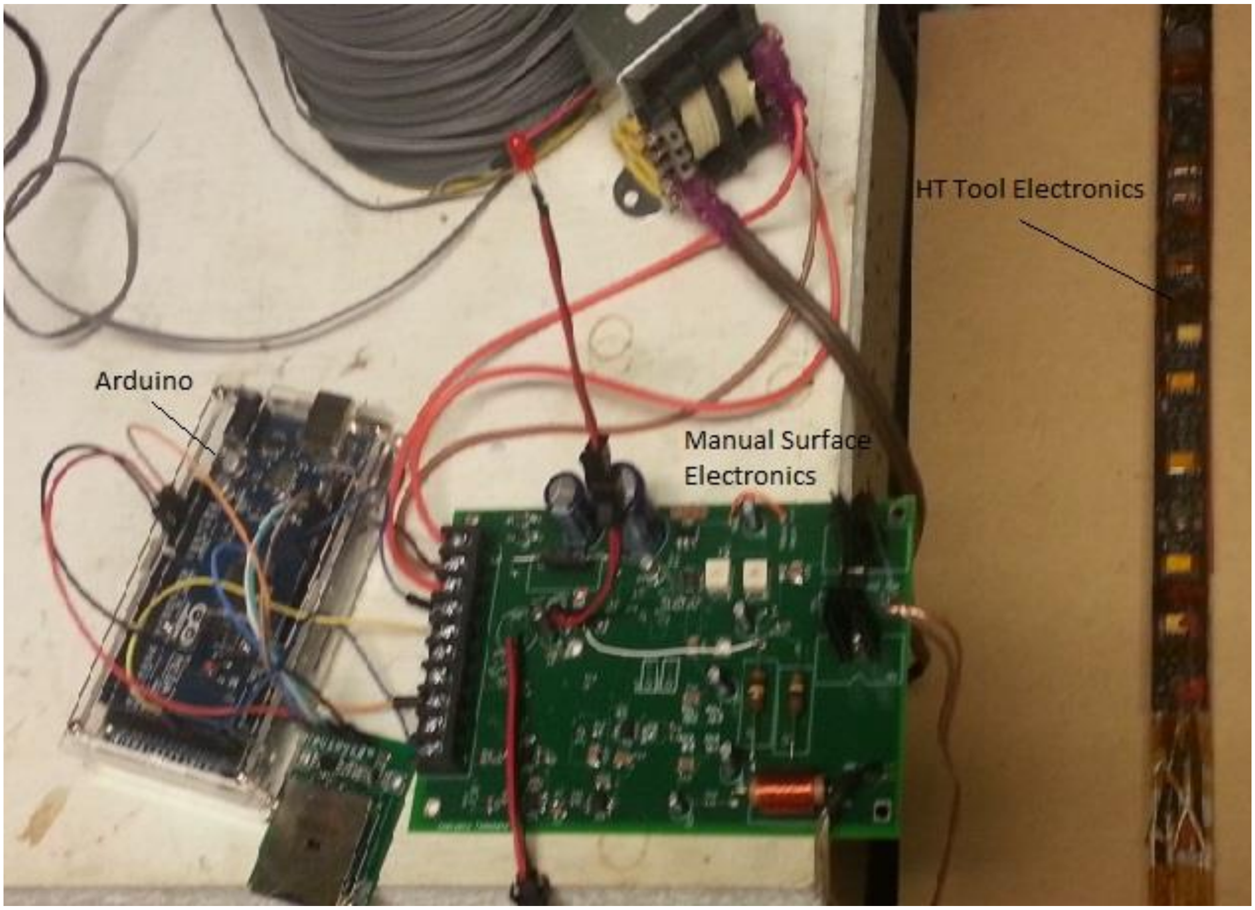


Figure 1First use of an Arduino system to operate the surface electronics for monitoring a geothermal well monitoring tool.

Using the Arduino for Scientific Measurements

The Arduino is a microcontroller and not a microprocessor which makes it great for interfacing to the real world. However, it's not the best choice for making scientific measurements. For controlling a robot or 3D printer, a 10bit Analog-to-Digital converter (ADC) is more than good enough but falls short for most scientific applications.

In the world of Arduino 'shields' small electronic circuits used to add functions to the Arduino. The author has created two new shields to enable the Arduino system to make scientific measurements needed by the surface electronics used to monitor and record geothermal well data. These shields are a AD22B (22Bit Analog to Digital Converter) and the DAC12B (12Bit Digital-to-Analog Converter). [*Note: perhaps 99% of Arduino shields are simply very low cost electronic toys – cheep 2 layer circuit boards and low cost passive components as 1-5% resistors with 100-200ppm/C drift. The two new shields demonstrated in this report are actually built on 4 layer circuit boards with complete uninterrupted ground planes and use high quality 0.1% resistors with 25ppm/C drift rates as needed for solid performance*.]

The DAC12B is used to create a low noise, programmable power supply. It can supply the well monitoring tool power down a cable. It can also be used to power surface well head sensors as temperature, pressure, flow or seismic. This is a very versatile shield for many scientific projects. The new AD22B greatly increases the resolution of the Arduino for making very sensitive measurements. The ADC also has a built in filter to reduce local 60Hz or 50Hz noise.

Both of these new shields come with example programs for use with the Arduino system. Using one of these example programs, the DAC12B is used to create small changes in voltage which is then measured by the AD22B shield and compared to the internal 10Bit ADC of the Arduino. The results of such a test are shown below.

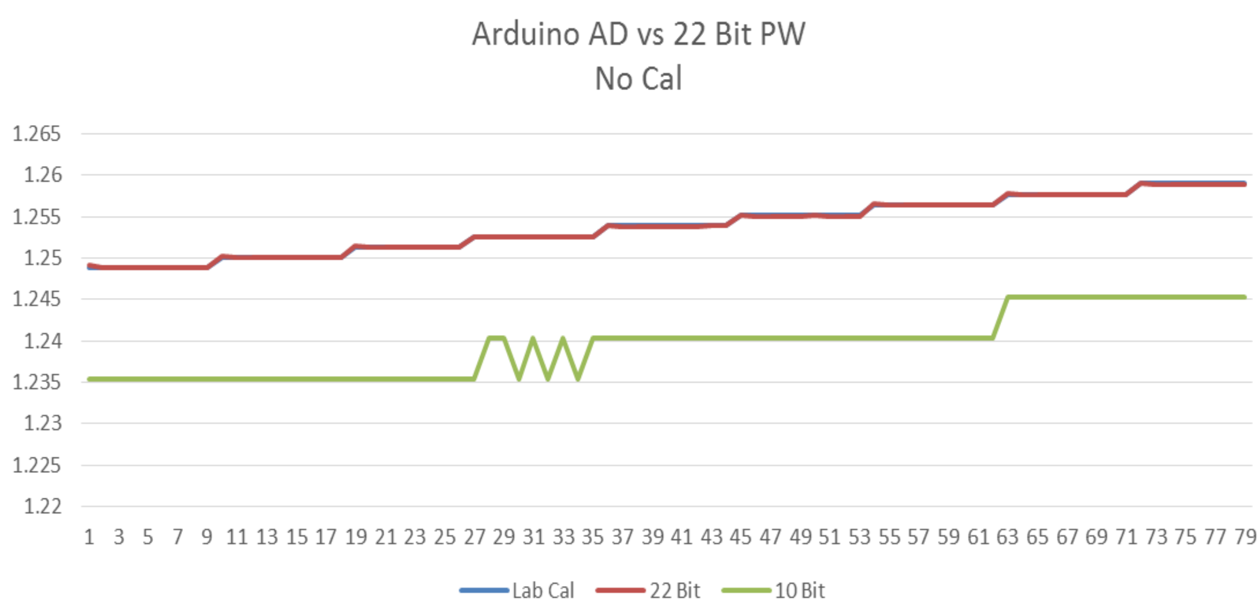


Figure 2 Actual voltage measurements taken both by the Arduino internal ADC, the external 22Bit shield and a 5.5digit calibrated voltage meter. The supplied test voltage is from the 12Bit DAC shield.

In the above data plot, there are three sets of reading from a precision lab voltage meter, the 22Bit ADC shield and the 10Bit internal Arduino ADC. The DAC12B is used to create the test voltage has a 0 to 10V output with 12bits of resolution. For this set of measurements, the DAC output was divide in half as the two ADC converters are limited to only 0 to 5V inputs. The Arduino program increased the output voltage in 1 bit increments or approximately 1.2207mV per DAC step value.

The internal 10Bit ADC only has ¼ the resolution of the 12Bit DAC. So the 10Bit ADC is only changing values every four increments of the 12Bit DAC. The lab cal meter and AD22B shield readings are too close to see a difference at this plotted resolution.

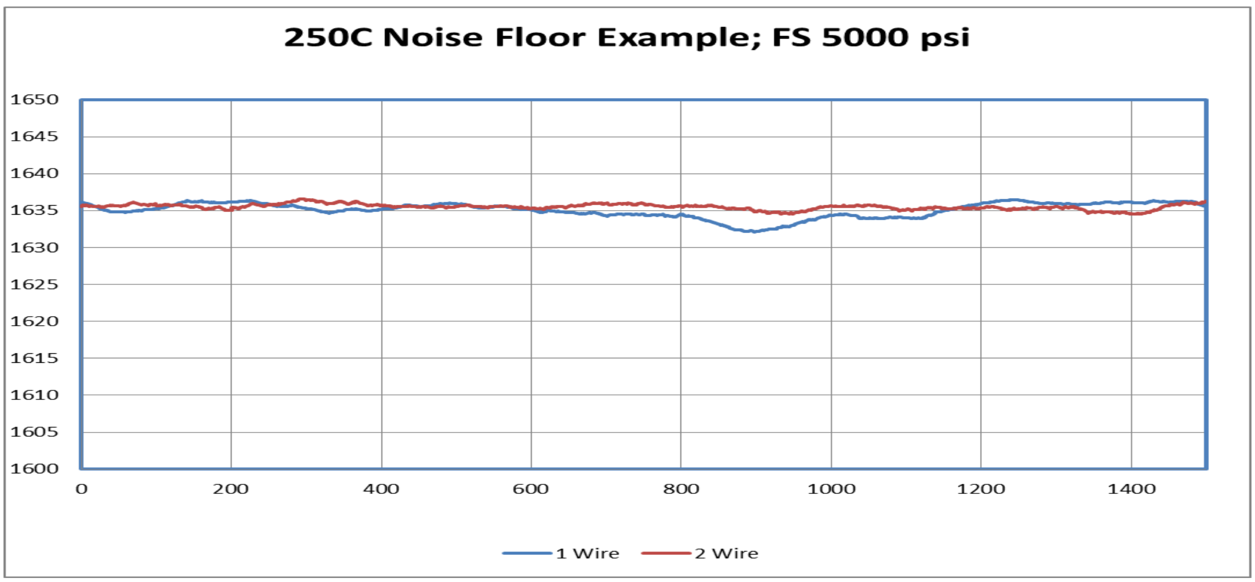
In short, the internal 10Bit ADC of the Arduino has only slight bit chatter at transition voltages. The Arduino 10 bit ADC is doing very well. The system noise floor is below the 10Bit or 4.88mV resolution. This may not be good enough resolution for many scientific applications. However using the 10 bit on-board ADC is good for secondary measurements supporting the primary scientific measurements. For examples, room temperature or system battery voltage are often needed to aid in calibration or validation of the primary scientific measurements.

The AD22B shield was run without any calibration factors or averaging. Its performance here was approximately +/-20uV off the laboratory measured values or about 0.0016% error. The AD22B has its own precision voltage reference where the Arduino microcontroller is using its DC 5V supply as the reference. The microcontroller used in the Arduino cost about $2 each which is a great deal for all of what it can do. The AD22B shield has a 22bit MCP3550 ADC and a 2.50V precision voltage reference costing a total of $7. When it comes to making electronic scientific measurements, it's always worth investing in precision. The Arduino Uno and AD22B shield give you the best of both worlds; programmable high speed data and high resolution data.

Now that we have two Arduino Shields designed for making scientific measurements, let's look at an example. The 275°C geothermal well monitoring tool.

High Temperature Geothermal Well Monitoring Tool

Below is a lab calibration test looking at the noise floor of the geothermal well monitoring tool operating at 250°C and fixed input pressure sensor reading of 1635 psi.



Here we are looking at the surface readout of an analog signal tied to well pressure. The well pressure measurement was used because the pressure transducers has a very small signal. The high temperature pressure transducer has only 4uV/psi output. The downhole tool amplifies the signal 200X using high temperature Perma Works amplifier circuits. There is a lot of opportunity for noise in the system with high amplification and high temperatures. In the above plot, the blue trace (1 wire) is pressure measured as a function of tool current. This allow the tool to operate using standard logging cable found on 90% of all logging trucks. As the tool's power and tool pressure signal are on the same wire, there is an increase in the noise level as seen by the 22Bit ADC.

If a second wire is available for the analog voltage signals, the noise floor is reduced. The AD22B has a high input impedance and a user defined noise filter. The second wire can be directly attached to the AD22B input without any problems.

Real Time Clock

There are several low cost, real time clock (RTC) shields for the Arduino. There are at least 4 different types. We chose the RTC shield based on the Maxim DS3232. Like the other RTC shields, it tracks Year, Month, Day, Hour, Second, milli-second including leap year. However, the DS3232 has an internal crystal with temperature compensation. This allows for greater accuracy over extended time while deployed in remote locations.

The RTC has a programmable alarm for waking up the Arduino to take measurements every hour or every day. The RTC can be used to time Arduino data reading up to 500 samples per second.

Get Demonstration Arduino Programs

Examples for programming the AD22B, DAC12B and the DS3232 RTC can be found in the book, "Arduino for Projects in Scientific Measurement" by Randy A Normann or at the web site, www.onmeasurement.com.

The Arduino can be synchronized with a PC clock using a program written by Joe Henfling called RTC_Software_Rev2. This is PC software connecting the Arduino to a PC via a USB cable. The program was tested under Windows 10 from Microsoft. Along with synchronizing the RTC to the PC time, it also starts recording measurement data from the Arduino for real time data collection.

Future Scientific Shields

The DAC12B and AD22B shields were built and tested by the author. The RTC clock shield is available from several commercial sources. These are the key components for making scientific measurements using the Arduino system. However, other shield are being considered as:

1. Natural Gamma Counter
2. Large Solid-Sate Memory (to replace SD cards)
3. Logging Tool Communication both Surface and Inside the Tool
4. Spinner Counter for Flow Measurement

Conclusion

The Arduino system is very low cost, in the public domain and used to teach high school students basic machine programming as in robotics. This system is easy to program and is well supported with books written for beginners and advanced programmers. The author has developed two key shields (circuits) to make the Arduino system useful for scientific measurements. As example using a 275C rated geothermal well monitoring tool was successful.

The two shields are the 22Bit Analog-to-Digital Converter (AD22B) shield and a low noise, 12Bit programmable power supply shield for powering sensors (DAC12B). These are professional grade, high resolution Arduino shields. Together, these two circuits are controlled by the Arduino microcontroller to create a basic scientific measurement system.

Programming the Arduino is simple using a text based public domain compiler available from Arduino.cc. By working with a public domain system as Arduino, the development costs and time to construct well monitoring systems or future logging tools can be reduced. Also, the future availability of new programmers needed to support the geothermal industry might be attending robotic classes in to today's high schools.

Reference

1 First High-Temperature Electronics Products Survey 2005, Randy A. Normann, Sandia Report SAND2006-1580, Sandia National Laboratories, April 2006