



Using Arduino in Engineering Education: Motivating Students to Grow from a Hobbyist to a Professional

Mr. Brian M. Wood, Keysight Technologies

Brian Wood has worked at all incarnations of Hewlett-Packard - HP, Agilent Technologies, and now Keysight Technologies - since 1973, with a few years off (2006-13) for "retirement." He's been an R&D design engineer, production engineer, application engineer and is now in marketing, providing worldwide support for Keysight's BenchVue instrument control software. In his spare time, he enjoys ham radio and electronic tinkering.

Dr. Alexander O. Ganago, University of Michigan

**USING ARDUINO IN ENGINEERING EDUCATION:
MOTIVATING STUDENTS TO GROW FROM A HOBBYIST
TO A PROFESSIONAL**

Using Arduino in Engineering Education: Motivating students to grow from a hobbyist to a professional

Abstract

Motivation is key to human development, including professional growth. Quite often, the rewards for student learning seem to be vague and remote; years in classroom are not fun. Effectiveness of education skyrockets if cool outcomes of learning are within reach.

Arduino – microcontroller boards popular with hobbyists – can be used to help both the teacher and the student find new answers to the key question: Why should we learn about “X”? Instead of traditional preaching that “X” would probably be useful in your profession, the answer becomes: “Because it helps you do cool things with Arduino this week.”

In the context of Engineering Education, at any level from high school to college, for Electrical Engineering (EE) and non-EE majors alike, “X” may be any of the following: Sensors and calibration; Ohm’s law and voltage division; Current and power; Sine and square waves; Transient responses; Analog and digital signals; Sampling rate, bits, and accuracy; Control theory; Programming; Protocols for wireless communication; Test/measurement procedures; How motors work, etc.

Use of Arduino-based projects as motivators for learning is beneficial to both the student and the instructor: (a) To a student, clear goals for advanced learning; (b) To a teacher, a fresh look at what makes students learn; (c) To both, they bring joy of achievement.

In this report, a collaboration between Keysight Technologies, an industry leader in Test and Measurement, and the University of Michigan, one of the premier engineering schools in the U.S., we share our experience at several levels: (i) Short summer camps for high-school students; (ii) Crash course of lectures and lab experiments on basics of EE; (iii) Description of student achievements; (iv) Advanced studies for EE majors in the industrial setting; (v) Simulating a commercial DMM and measuring performance with Keysight BenchVue software.

We believe that the experiments reported here would help our colleagues who work on course development at various levels of Engineering Education. Further details may be provided per request.

At all levels of learning, Arduino-based projects are multidisciplinary (from programming to electronics to mechanics and design) and meet ABET Outcomes 3 a, b, c, d, e, g, i.

Introduction

Arduino is a microcontroller invented for non-engineers (students at an art school in Ivrea, Italy), simple but powerful, immensely popular among hobbyists around the world [1], and widely used in STEM education [2]. Arduino provides one of the most accessible ways to engineering, from motor control to programming to wireless communications, etc. If used properly, Arduino may also facilitate student learning at deeper levels of Engineering Education.

On the surface, it looks easy for a hobbyist:

- Get an Arduino with a protoboard;
- Follow instructions available in the style of a cook book;
- Connect a few gadgets with wires;
- Obtain the functionality, which is new to you, for instance, vary the speed (RPM) of a DC motor by simply turning the knob.

Success of a project is a great motivator; also, it opens many ways to the future. For instance, applications for motors, which can be run at variable speed, are endless. Many web sites and books offer description of cool projects [3], which can be within reach for everyone, on a small budget and without need for advanced learning.

However, the distinction between a hobbyist and a professional is in the depth of understanding. Becoming a professional requires development of a vision of what happens below the surface. Relating to the motor control, professionals ask many questions, such as the following:

- Where do the controlling signals come from?
- Are these signals analog or digital?
- What is the accuracy of analog-to-digital conversion?
- How does the microcontroller process the input signals?
- Are the microcontroller's output signals analog or digital?
- What is the accuracy of digital-to-analog conversion?
- How is the actuator's response (motor RPM) related to the output signal?

Without reaching these deeper levels of understanding, the enthusiasts remain mere hobbyists.

A 2013 article in Popular Science [4] reports:

“A 2011 study published in the journal *Science Education* showed that high-school seniors who express an interest in pursuing science, technology, engineering, and math (STEM) are three times more likely to complete college degrees in those subjects. The key to getting students to that tipping point, says lead author Adam Maltese, an assistant professor of science education at Indiana University, seems to be exposing them early to a STEM experience that sparks their interest, then providing them with a way to maintain it.”

Students are enthusiastic about cool projects. Engineering educators may take advantage of the students' enthusiasm: use it as a driving force for deeper learning. Thus, students are led from the obvious to the essential: from the observation of a motor, which slows down or speeds up, to learning the background necessary for answering the questions such as listed above.

Notably, this approach requires good equipment, which few hobbyists have on their desks. For example, when students relate their observation of how fast a motor runs, when the knob is turned, to the waveforms, which they see on oscilloscope screen, the role of pulse-width modulation (PWM) for motor control becomes easier to appreciate. Availability of equipment (which is costlier than Arduino itself) becomes a necessity for deeper learning.

Due to the deeper learning, students develop clearer vision and a broader view of their projects:

- Understand the limitations of the device, which they built (power, speed of response, etc.)
- Know what alternative options can lead to similar functionalities;
- Assess the alternatives in order to choose the most optimal for the particular project.

Such transition, from the superficial or obvious to the essential or hidden from view, can be fostered at several levels of engineering education. In this report, we focus on two examples at very different levels:

- At the introductory level, inform high-school students of the signals and processes, which are involved in their first Arduino-based projects.
- At the professional level, scrutinize the distinctions between Arduino, which can serve as a measurement device, and the commercial digital multimeters (DMM), which measure the same voltages and currents.

Specifically, for the introductory level, in the context of a short summer camp for high-school students, we describe:

- Simple lab experiments, which students do before they start working with Arduino
 - Calibrate a position sensor, which is based on a linear potentiometer used as a voltage divider.
 - Learn to use a waveform generator to produce, and an oscilloscope to monitor square waves, which serve as prototypes of Arduino's PWM output signals.
- Brief video recordings, which were used in lectures to demonstrate the relationships between observed phenomena (movements of motors, lights produced by LEDs) and the controlling signals, which were monitored with 2- and 4-channel oscilloscopes.
- Overview of Arduino-based projects, which were chosen and completed by high-school students during the summer camp.

For the advanced level, in the context of student internships at Keysight Technologies, a set of specially developed laboratory experiments, which are targeted at the following topics:

- Communications between DMM and computer (drivers, interfaces, writing software to execute a single command);
- Accuracy and precision, with direct comparison of Arduino and DMM;
- Input protection against noise and electrostatic discharge;
- Various sources of electronic noise;
- Several types of analog-to-digital converters, including that used in Arduino;
- Expansion of capabilities of Arduino, in particular, to measure electric currents;
- Direct comparison of performances of Arduino and Keysight 3446xx DMM.

Summer camp for high-school students

In July 2017, Alexander Ganago led a 5-day-long summer camp for high-school students. At the beginning, only one of 26 participants knew what electric resistance is (according to informal survey in Lecture 1). The camp activities started with a general introduction:

- Lectures reviewed the basics of Electrical Engineering
 - Voltages, currents, resistances
 - Building circuits on a solderless prototyping board
 - Voltage division
 - Resistive sensors
 - Analog and digital signals
 - Microcontrollers
 - Arduino specifications
 - The basics of programming Arduino
 - Advice for creating a good project.
- Lectures were short (30 to 60 minutes each) and followed with Labs (2 – 3 hours each; two or three Labs every day). Several Lab experiments are described below.
- Test/measurement instruments used in the Labs included the following:
 - E3631A triple power supply
 - 34450A digital multimeter
 - 33220A function generator
 - DSO-X-2012A oscilloscope
 - All instruments listed above were made by Hewlett-Packard, Agilent, or Keysight; this is one company, which changed its name during the years.
 - VirtualBench® made by National Instruments. Within one box, this instrument combines the functionalities of all four, which are listed above; comparison of their features is beyond the scope of this report.
- Arduino-compatible, RedBoards by SparkFun were provided, along with several sets of sensors, electronic components, wires for circuit building, etc.

Students organized themselves into 6 teams (4 or 5 students in each) to work on their projects. Every team had their own SparkFun board, VirtualBench®, desktop computer with necessary software; access to sensors and other components, tools and accessories. During their work on the projects, students were encouraged to apply their learning: monitor the signals with the oscilloscope and digital multimeter of the VirtualBench®.

At the end of the camp, every team of students presented their project to the instructors and a group of parents. For these presentations, students prepared slides with brief descriptions and photographs, some of which are shown below. Every project included:

- Sensors, which produced analog or digital signals;
- Arduino for communication, calculations, making decisions, commands for actuators;
- Actuator (motor, light, sound, display of data, etc.).

Every project was successful. Every project worked. The camp provided every student with an opportunity for rapid professional growth.

Lab experiments for high-school students

Lab Experiment 1: Calibrate the position sensor.

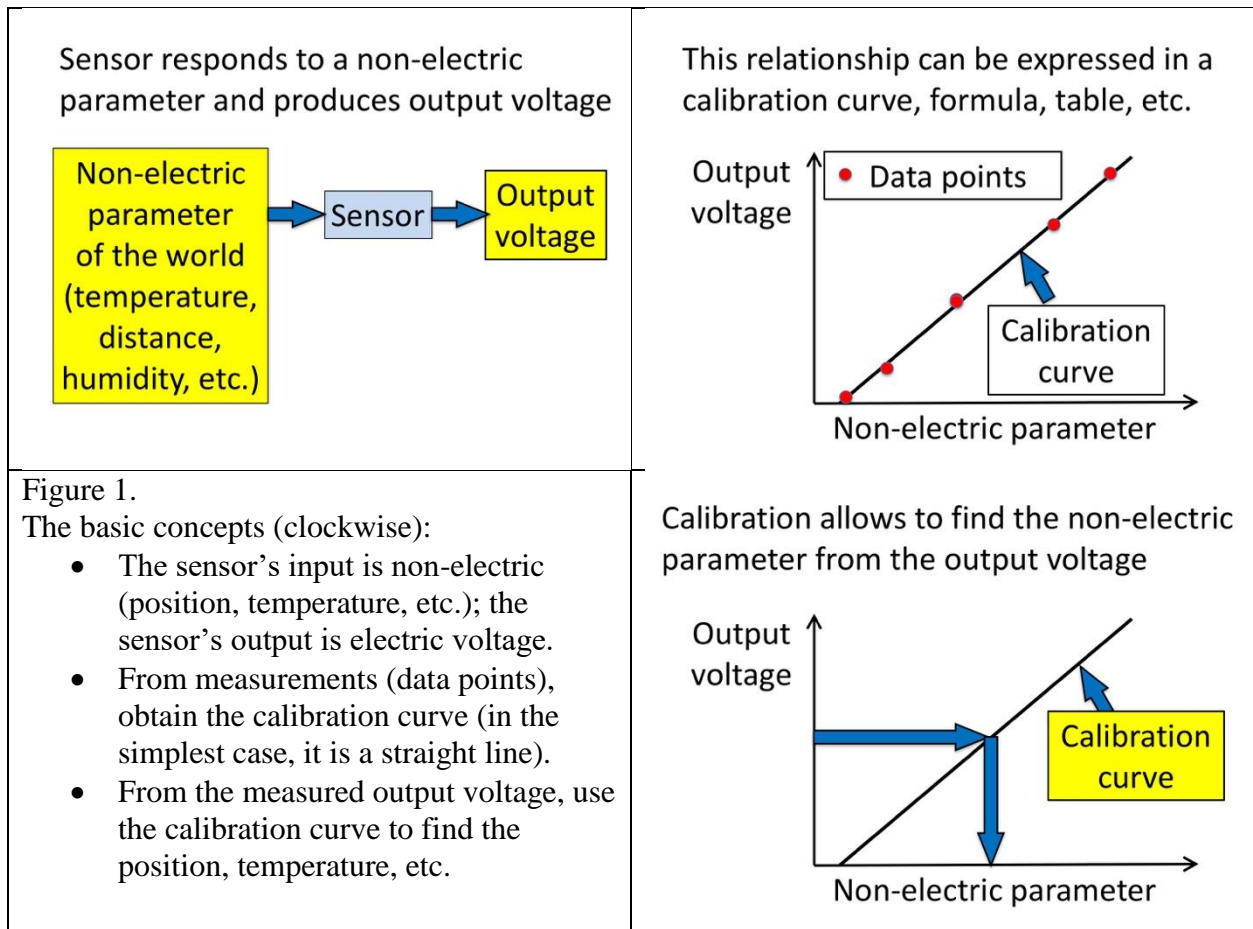
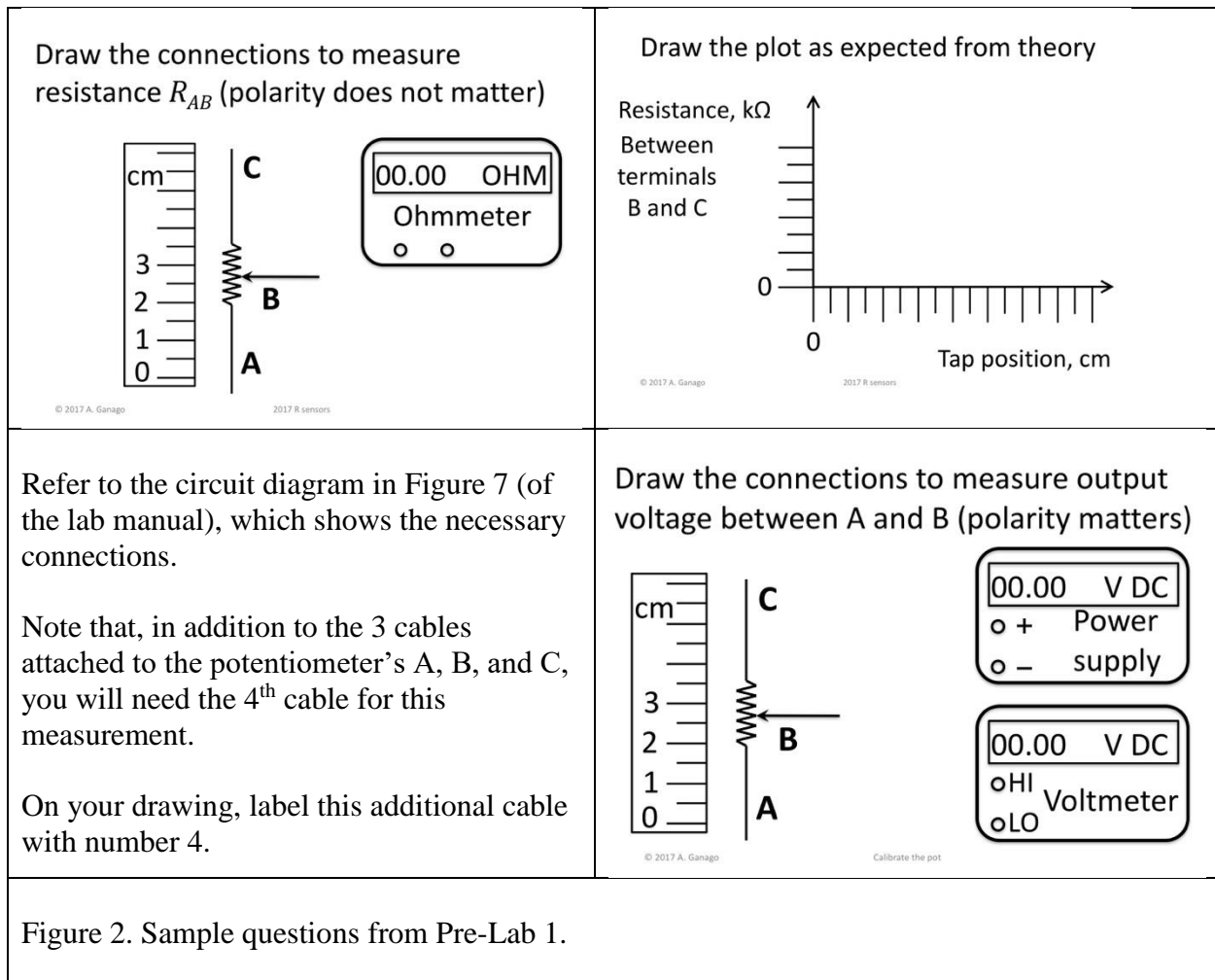


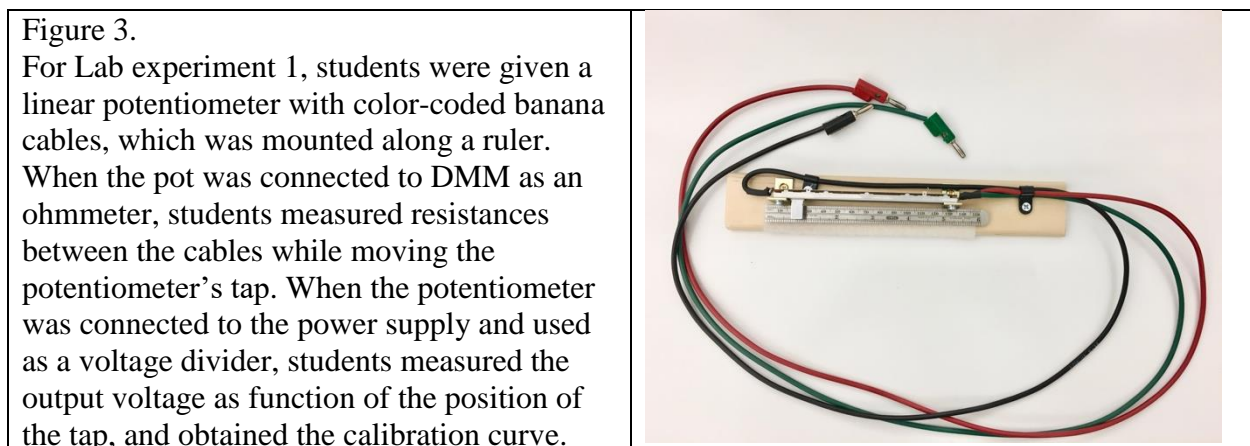
Figure 1 from the lab manual for students presents the outline of Lab 1. Before doing this experiment, students were introduced to the following background information:

- Arduino is a microcontroller, which uses electric signals from sensors. It has pins for analog and digital inputs, for power connections, and for digital outputs.
- Voltage, current, resistance.
- Electric resistance depends on the geometric factors such as the length of resistive sensor.
- Potentiometer; its use as a variable resistor and as a voltage divider.
- Measurements of voltages and resistances with DMM.
- Calibration of a sensor; possible errors in calibration of a position sensor.

Before doing the experiment, students were encouraged to do a brief Pre-Lab; Figure 2 shows two typical questions from this Pre-Lab.



For experiments, students were given a 10-k Ω linear potentiometer 10 cm long: See Figure 3.



After a student team obtained their calibration, they were suggested to play the game between the teammates for 1 – 2 minutes. The goal is to determine whether the calibration was good:

- Hide the potentiometer from sight (behind a small piece of paper such as a 3x5 card);

- One partner moves the tap, not showing its new position.
- The other partner measures the output voltage.
- From this measurement and the calibration data, determine the position of the tap in mm.
- Look at the actual position of the tap; compare with the prediction based on the calibration; calculate the error
$$Error (mm) = (Actual\ position) - (Expected\ position)$$
- Repeat for other positions of the tap (total, up to 3 positions for each teammate).
- If all errors are within 5 mm (0.5 cm, or 5% of the total length of the potentiometer), the calibration is good.

This experiment is easy and fundamental: it trains the students to perform measurements, and introduces several key concepts, which relate to all measurements with sensors. In our opinion, this and similar experiments are necessary for the growth of a hobbyist to a professional.

Lab Experiment 2: Use square waves.

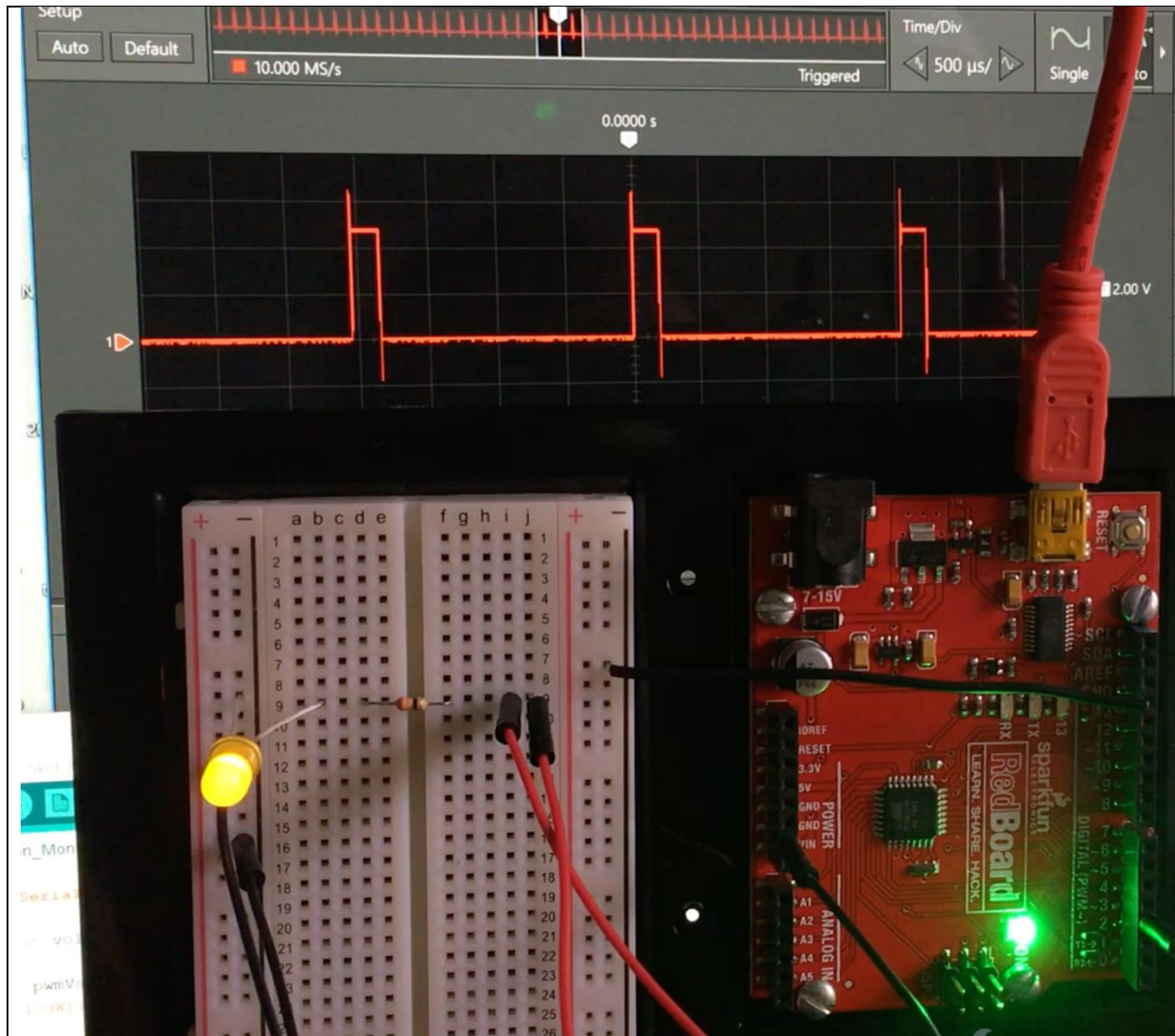


Figure 4.

The background of this photograph is the screen of a computer, which is connected to *VirtualBench*® – modern instrument for field engineers, equivalent to power supply, DMM, function generator and oscilloscope all in one box that fits in a briefcase. The white plastic rectangle with many holes is a solderless prototyping board, on which circuits are built. *RedBoard*® microcontroller on the right is connected to both the circuit and *VirtualBench*®. (Both the Figure and its legend are copied from the lab manual for students.)

In Experiment 1, students worked with signals, which can be used as inputs for Arduino. Now, they look at the output signals. Figure 4 from the introduction to this Lab, shows real outputs of Arduino-compatible RedBoard by SparkFun.

Figure 5 shows the oscilloscope screen with the results of automatic measurements on a square wave produced with a function generator.

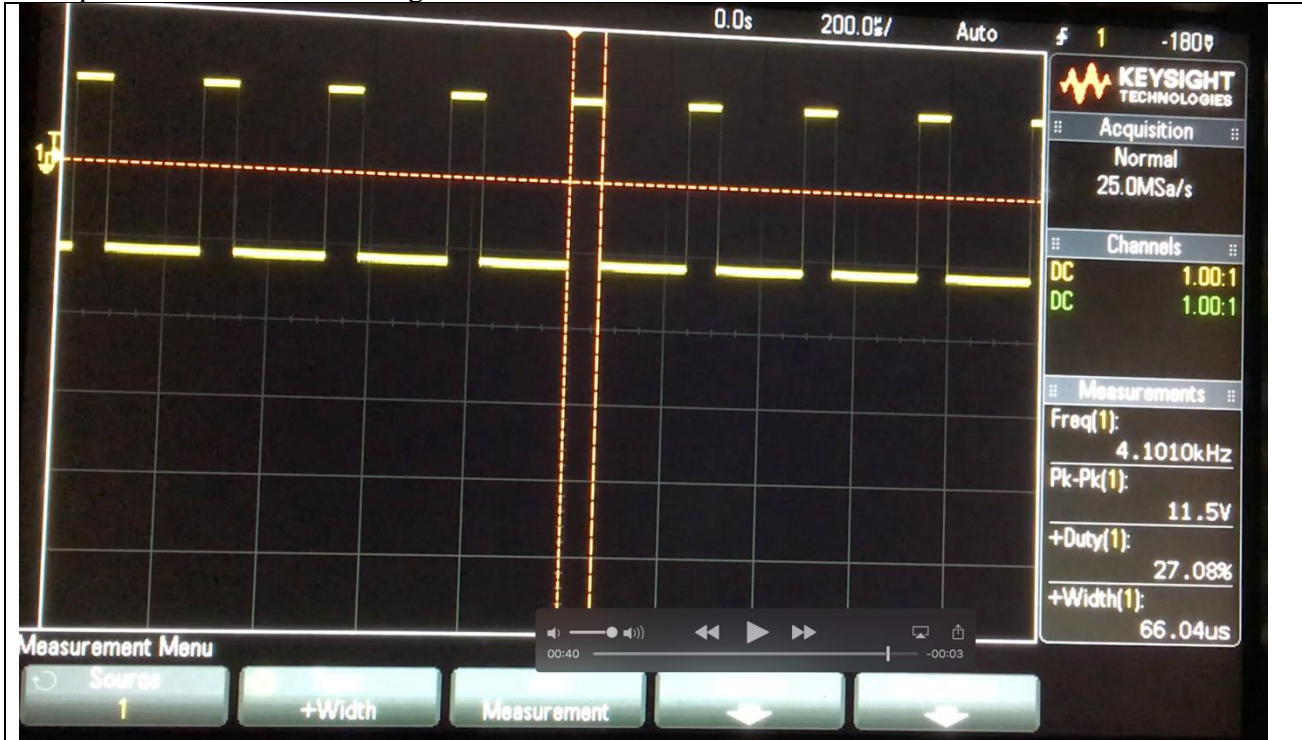


Figure 5.

Example of automatic measurements performed with Agilent/Keysight DSO-X 2012A oscilloscope. A square waveform signal is connected to input channel 1 of the oscilloscope; thus, all parameters listed in the right sidebar are labeled (1); parameters of another signal, which could be connected to input channel 2, would be labeled (2). **Freq** stands for frequency; **Pk-Pk** denotes peak-to-peak amplitude; **+Duty** is the duty cycle; **+Width** is the duration of positive pulse. (Both the Figure and its legend are copied from the lab manual for students.)

In the Introduction to Lab experiment 2, students learn about three parameters of a square wave – the amplitude, frequency, and duty cycle. In the Pre-Lab, they practice to relate the frequency and duty cycle to the duration of the signals. See Figure 6 for a typical question.

Pre-Lab question 3

Servo motor moves to 90° position and stays there if it receives positive pulses 1.5 ms long. Microcontroller produces the square wave at 400 Hz. What duty cycle is used to position the servo at 90°?

- A. 50%
- B. 20%
- C. 40%
- D. 60%
- E. Impossible to obtain, because the period is shorter than 1.5 ms.

Figure 6. Pre-Lab question, which trains students to relate frequency and duty cycle to the duration of signals.

In Lab experiment 2, students learn the basics of measurements of time-dependent signals. For example, see Figures 7 and 8, which are taken from the lab manual for students.

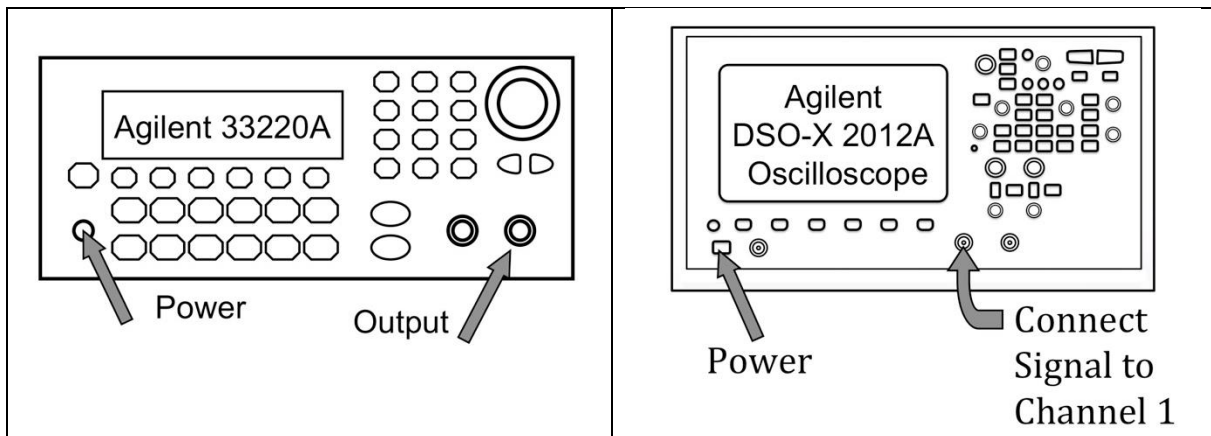
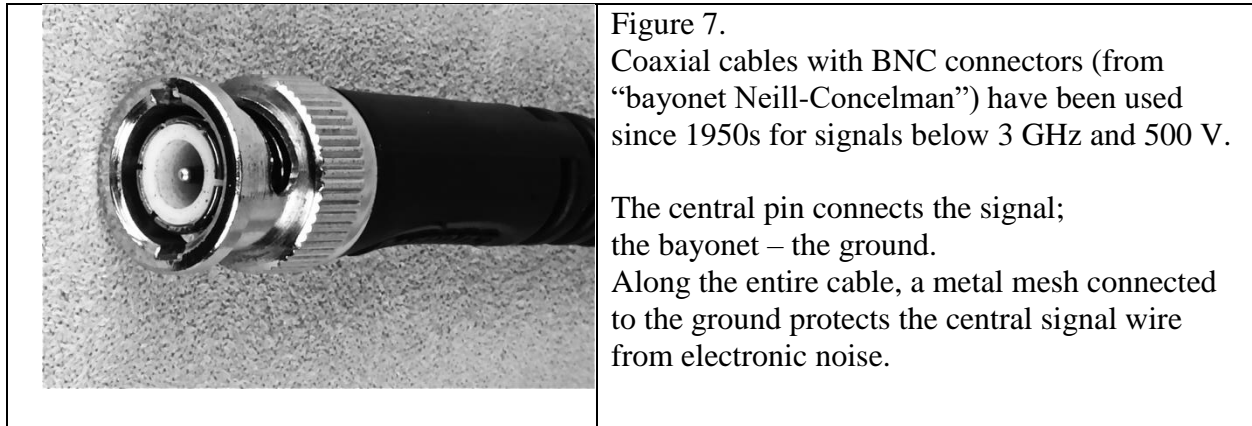


Figure 8.
Power both instruments and connect them with a BNC cable.
(Both the Figure and its legend are copied from the lab manual for students.)

Students learn to operate both instruments, to set and measure the parameters of waveforms. The following exercise puts their new skills to test.

In Pre-Lab question 3 (Figure 6 in this file), you predicted the duty cycle, which is needed to obtain $+Width = 1.5 \text{ ms}$ using a square wave at 400 Hz.

Now, verify your prediction:

- Set a real square wave at 400 Hz. Keep 5 Vpp, typical of microcontroller output.
- Adjust Duty Cycle on your function generator; monitor the $+Width$ reading on your oscilloscope; your goal is to achieve $+Width = 1.5 \text{ ms}$.
- After you achieved $+Width = 1.5 \text{ ms}$, read Duty(1) on your oscilloscope.
- Compare the result of your measurement with your answer to Pre-Lab question 3.
- Vary Duty Cycle on your function generator by $\pm 1\%$: see by how much the oscilloscope readings change in response to these variations.

Parameters of the waveform	Theoretical prediction	Lab data: Measured with oscilloscope
Square wave at 400 Hz; +Width = 1.5 ms	Duty cycle =	Frequency = +Width = Duty Cycle =
Your conclusion on the accuracy of your calculations and of your measurements: Do the results agree?		

Lab experiment 2 teaches the students what happens “under the hood” when Arduino produces PWM output. Brief videos shown in lectures provide examples of applications: see Figures 9, 10.

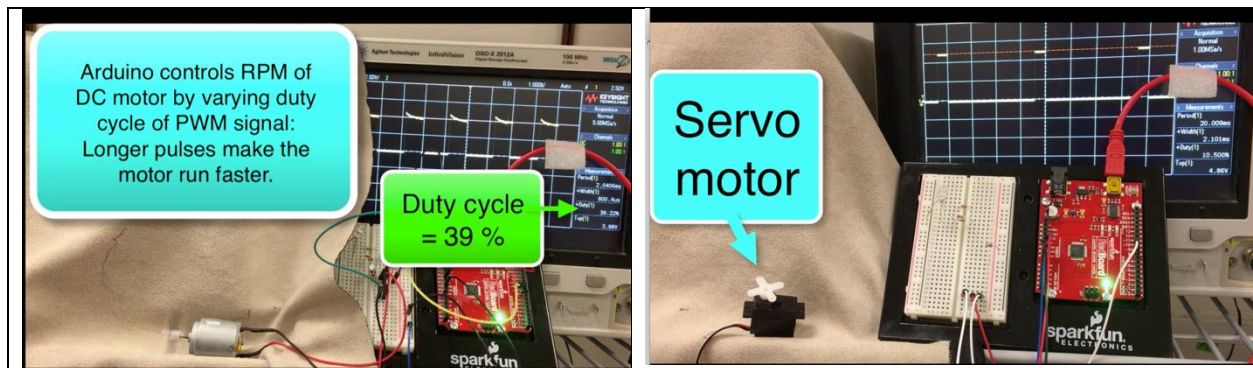


Figure 9. Screenshots of brief videos, which were created for lectures, demonstrate the applications of Arduino’s PWM output for motor control.

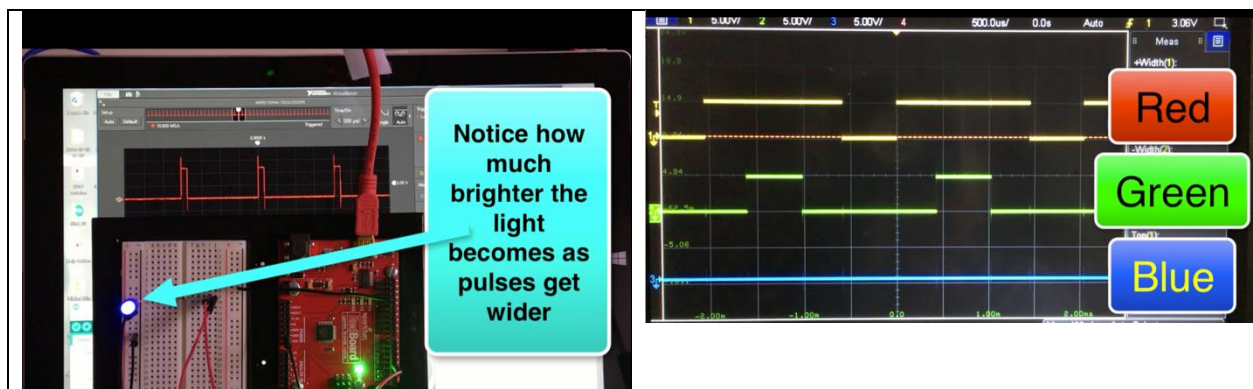


Figure 10. Screenshots of brief videos, which were created for lectures, demonstrate the applications of Arduino’s PWM output for control of a single LED (left panel) and or an RGB LED (right panel).

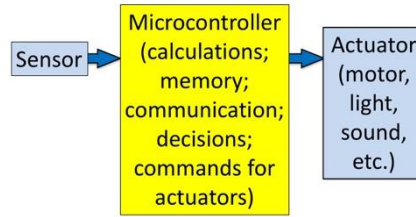
Combination of hands-on experience in the lab with video demonstrations in lectures helps students develop deeper, intuitive understanding of Arduino’s output signals and their uses.

Arduino-based projects created by high-school students

Figure 11.

Every student project was based on this block diagram. The descriptions of individual projects were created by the students; comments and arrows are added for clarity. At this level of learning, students were expected to understand whether the input signals were analog or digital; what Arduino did; how the outputs worked.

Microcontroller is the “brain” connected to both the sensors and the actuators



© 2017 A. Ganago 2017 Sense-it Overview

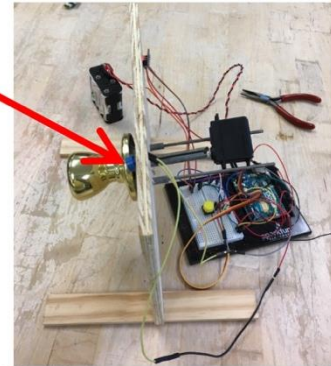
Combination Door Lock

Liz Howden
Miranda Makowski
Ben Kleindorfer
Michael Bochenek

© 2017 A. Ganago 2017 Sense-it Overview

Press the button

Only the correct combination opens the door for you.

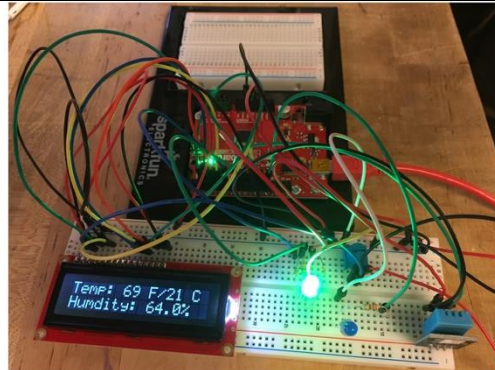


© 2017 A. Ganago 2017 Sense-it Overview

Mini Weather Display

Gurbaz Singh
Suneet Singh
Vikram Duvvur
Zamod Wellington

© 2017 A. Ganago 2017 Sense-it Overview



© 2017 A. Ganago 2017 Sense-it Overview

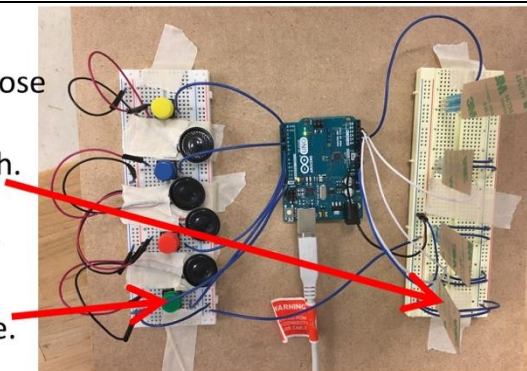
The Electric Ukulele

Molly Maloy
Rachel Imbrunnone
Tal Damelin
Tom Ni

© 2017 A. Ganago 2017 Sense-it Overview

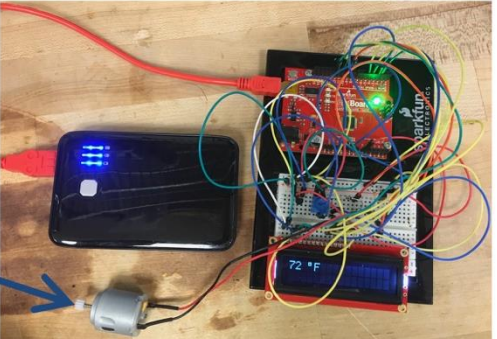
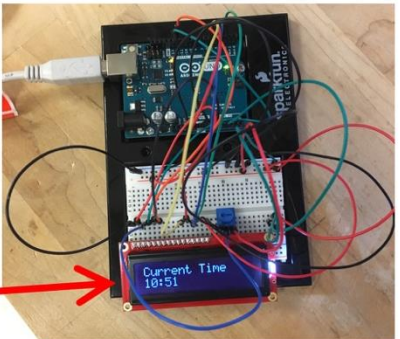
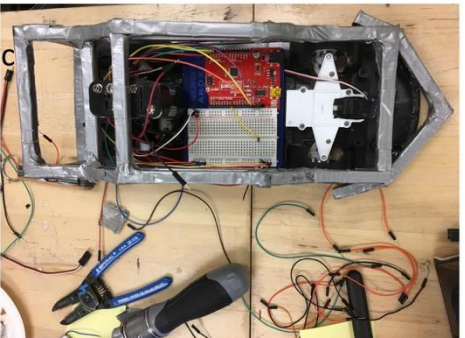
Choose the pitch.

Play the note.



© 2017 A. Ganago 2017 Sense-it Overview

Figure 11, continued.

<p>Temperature-controlled fan</p> <p>Andrew Cummings Jacob Klinger Sathuik Bombareddy Justin Oshiemi</p> <p><small>© 2017 A. Ganago 2017 Sense-it Overview</small></p>	<p>When it gets too hot, the fan runs.</p>  <p><small>© 2017 A. Ganago 2017 Sense-it Overview</small></p>
<p>Weather clock</p> <p>James Xu Benjamin Zhao Himan Yerrakalva Pranav Rakkappan Hubert Zhang</p> <p><small>© 2017 A. Ganago 2017 Sense-it Overview</small></p>	<p>Displays temperature in °F and °C.</p> <p>Then, displays time.</p>  <p><small>© 2017 A. Ganago 2017 Sense-it Overview</small></p>
<p>The Ultrasonic Sensors</p> <p>Cort Toschlog-Green Diego Rangel Tim Wang Julian Reinhart</p> <p><small>© 2017 A. Ganago 2017 Sense-it Overview</small></p>	<p>Ultrasonic sensors assist driving and/or parking the car.</p>  <p><small>© 2017 A. Ganago 2017 Sense-it Overview</small></p>

For the summer camp, where the high-school students were building their first Arduino projects, we developed lectures and Lab experiments, which provide necessary background for deeper learning, geared to Engineering Education. Before students actually started using Arduinos, we introduced them to rigorous measurements, comparisons of lab data with theoretical predictions, analysis of possible sources of errors, and other concepts and strategies, which distinguish a professional from a hobbyist.

Without Arduino as motivator, this learning might remain dry, unnecessary, and undesired; weaving the operation and applications of Arduino in this context, facilitated the student learning. On the other hand, without learning of the background concepts and strategies, tinkering with Arduino might remain a pastime or hobby, which does not foster personal and

intellectual growth. Brought together, rigorous learning and joy of creation appeal to the students, develop their creativity, and, hopefully, help them thrive in the future.

Every student team created a presentation about their final project, in which they demonstrated their learning of the essential concepts of Electrical Engineering, as well as their achievements in building things that actually worked. Every report included a technical description of the project and records of the actual functionality of the built device (photographs and/or video recordings). At the end of the camp, all teams made presentations of their projects to the audience that included instructors and parents; students demonstrated their creations in action, and answered technical questions about their work.

We consider this as an important step in the growth from the hobbyist level to the professional level. The collections of all student reports on the camp web site sets a high standard for future student teams.

In the tightly packed schedule of the 5-day summer camp, we did not dedicate special time for collecting feedback from students or using other assessment tools. We encouraged all student teams to include comments on their self-assessment into the final reports on their projects; however, their comments were very few and brief, possibly, because they had not been familiar with self-assessment at school.

Advanced studies for EE majors in the industrial setting

How do you turn a Do-It-Yourselfer (DIYer) into a good industry engineer? What distinguishes a DIYer from a practicing engineer? With the advent of inexpensive microprocessors and related interfaces, it has become possible for anyone with an interest to tinker, and even to create marvels. But it's a big leap from being able to write a few lines of C code for an Arduino and blink some lights to being part of a design team that's tasked with, say, making a Digital Multimeter go from 7.5 digits to 8.5 digits of resolution. We'll explore the characteristics of a good engineer and show how the Arduino can be used as a beginning step leading toward a EECS career.

First, what characteristics does industry expect of a good engineer? They are:

1. **Ability to work on a team.** While there are exceptions for extremely gifted people, usually a design team will have anywhere from 3 to 15 people working on a project, not including all the support services, such as PC board layout, parts coordination, marketing, and so on.
2. **A good understanding of the fundamentals of electronics and software/firmware.** One does not generally need a terrific understanding of the physics of materials to use off-the-shelf parts or to program custom logic arrays. But a good understanding of transistor biasing, fan-in, fan-out, propagation delays, effects due to electrical noise, RFI, and component tolerances is essential.
3. **Perseverance.** Some engineering problems can take weeks to figure out. One must have the ability to stick with a problem until it is solved, often with long evening and weekend hours.
4. **Understanding of how to work the tools.** An engineer's desk is usually littered with a dozen or more instruments, from a simple DMM to a Vector Network Analyzer. It's essential that time not be wasted learning how to use basic stimulus and measurement tools.
5. **How to make design tradeoffs.** If cost of the end product is a major concern, a consideration of when to use a less expensive part over one that may be pricier but better for the job has to be weighed. Fortunately, many companies have a preferred parts catalog, and the more experienced engineers will be available to help. The young engineer must drink in all this knowledge and not be too parochial about his or her design.
6. **Passion.** It's possible to do a good job without enjoying it, but it's much easier (on everyone) if an engineer truly loves the work. Holding a vacuum tube in one hand and marveling that our civilization can make something so wondrous, and doing the same with an SMT MOSFET in the other hand and getting that same feeling is what it's all about.

Many of these characteristics are part of a DIYer's basic nature. But converting the DIYer into an engineer who's qualified to be part of an industry design team is not trivial. What we propose to bridge the gap is to use Arduino to emulate a commercial instrument and then use its performance as the basis for an analysis of why a \$1000 commercial DMM has better speed and accuracy, and is impervious to external noise sources, than a \$35 Arduino board. This serves to teach important engineering concepts that are relevant to the industry, while still doing a series of fun experiments.

Seven Fun lab experiments.

There are complete science courses available [5] that delve into Arduino as a way of learning about science in a general way. Others have focused on students' natural inclination to be attracted to things that move, so courses have been devised focusing on mechatronics and robotics using Arduino [6]. We felt that an approach tuned to helping students learn things directly applicable to their jobs was needed. To that end, a series of seven labs were created and offered free by Keysight Technologies to explore the nature of the problems encountered when designing a commercial instrument. The labs require the use of an Arduino Uno. Similar processors could be used, but the Uno is well suited for the job, since it has a 10-bit A/D inside, as well as some GPIO pins.

Keysight BenchVue software was used to control and monitor a real Keysight DMM such as a 34461A and an Arduino coded to emulate a Keysight 34460A DMM. This was done by enabling the IO Monitor function in Keysight Connection Expert software and writing down all the commands that BenchVue sent out and the typical responses. A simple C program was written for the Arduino to look for these commands and generate equivalent responses. So, for example, a "READ?" command causes the Arduino code to take an A/D reading from analog channel 0, convert it to ASCII and then send it back to the computer.

Here are the labs that we created:

Lab 1: Get Connected!



Figure 12. Back end of a typical instrument, showing signal connections (left), I/O to the PC (middle), and AC power (right)

What kinds of interfaces are available? The interfaces (LAN, USB, GPIB) are highlighted using Keysight's "Connection Expert" software, which allows all instrument interfaces to be defined.

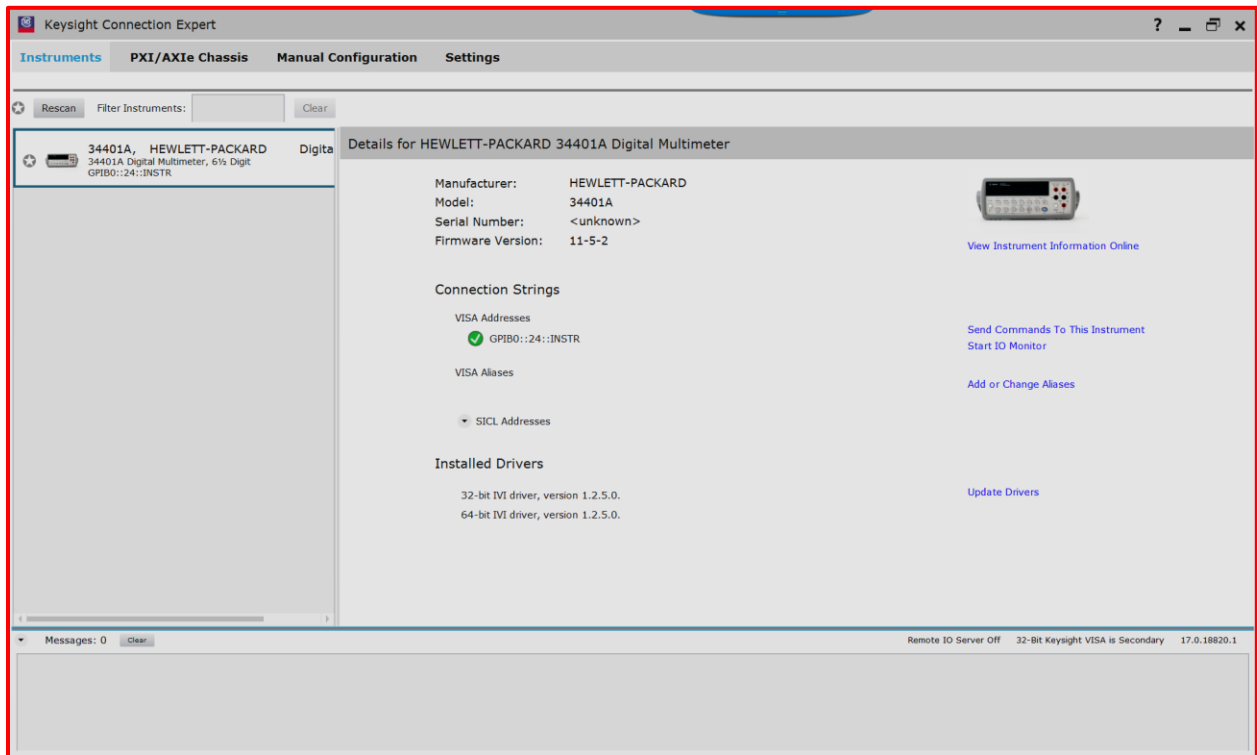


Figure 13. Keysight Connection Expert discovers instruments both automatically and manually

Students are then asked to write a simple program using Keysight BenchVue “Test Flow”, which is similar in concept to MIT Scratch, a well-known programming language for beginning programmers. The goal is to see how long it takes to send and receive data on three different interfaces while getting familiar with BenchVue and Test Flow.

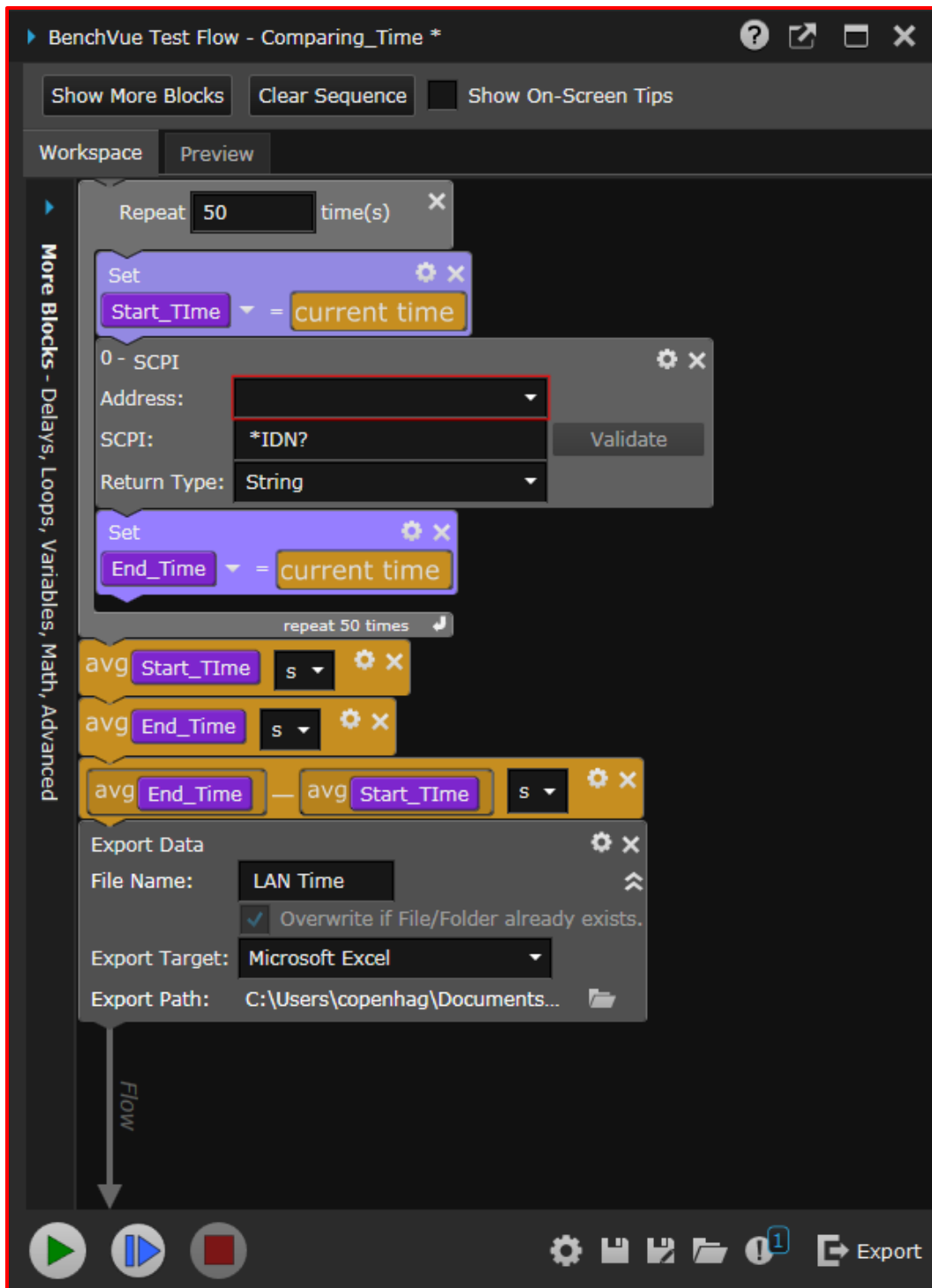


Figure 14. Test Flow sequence that measures time interval

Students then learn a little bit about the SCPI (Standard Commands for Programmable Instruments – pronounced “Skippy”) instrument control language used with almost all Keysight instruments and many other companies. This step is crucial to being able to communicate with the Arduino board that will “pretend” to be a DMM that speaks SCPI.

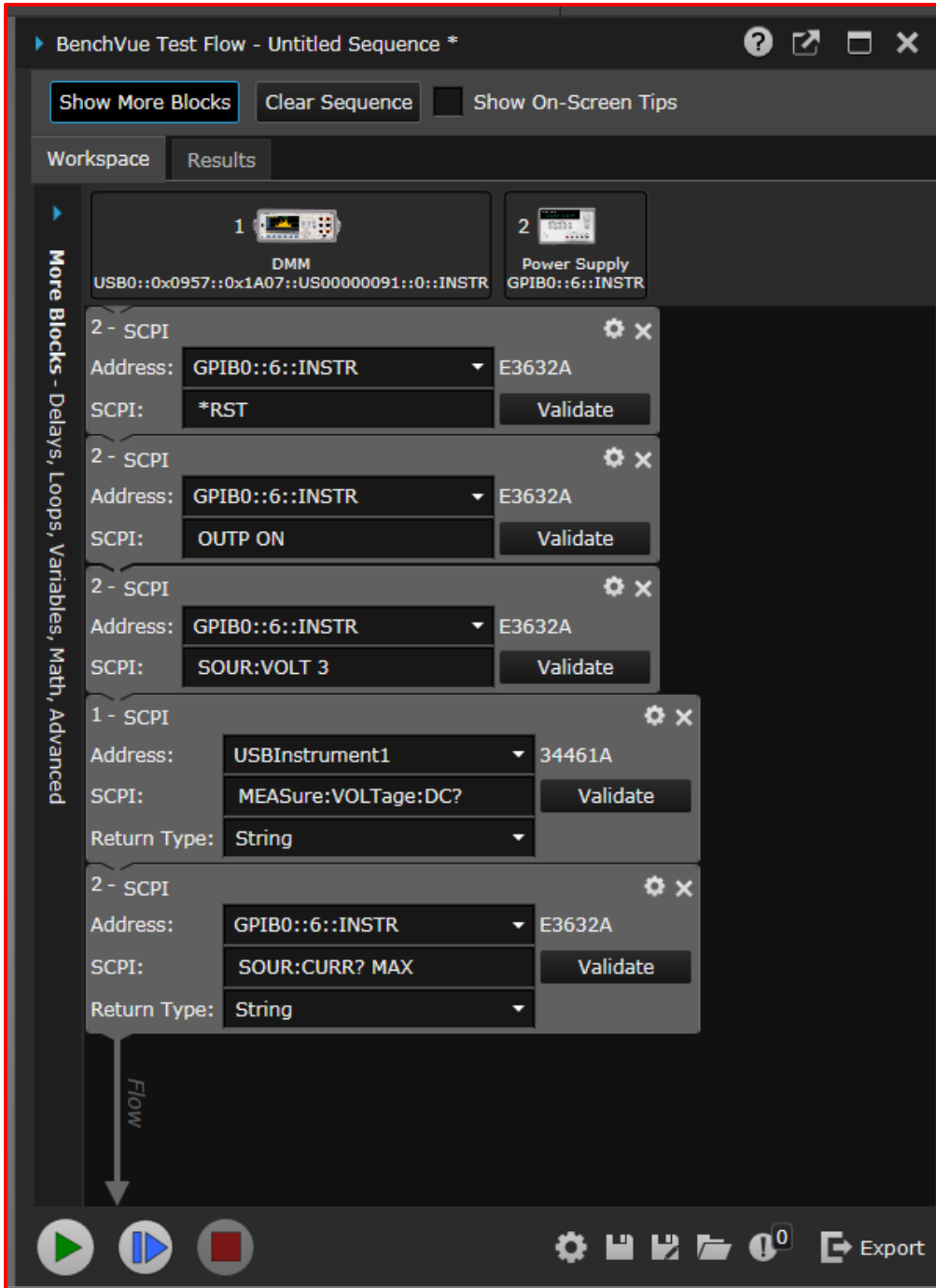


Figure 15. Test Flow sequence to send common SCPI commands to a DMM

Next, using Arduino’s Integrated Development Environment (IDE), called “Sketch”, students create the software to process a command. Students with little or no programming experience can be given this code, but it is instructive for the teacher to explain how it works.

```
// Idle loop
void loop() {

int i,c;

const String newline=F("\n");

boolean termchar_received;

char s[100];

do {
  i = 0;
  termchar_received = false;
  // READ A STRING FROM THE USB PORT
  do {
    while (Serial.available()<=0);
    c = Serial.read();

    if (c == 10) { // if termchar, null term string
      s[i] = '\0';
      termchar_received = true;
    }
    else if (c!=13){
      s[i] = (char) c; // cast int to char, put in buf
      i++;           // bump index
    }
  } while (!termchar_received);

  ss = (String) s;
  ss.toUpperCase();
}
```

Figure 16. Arduino “C” code to read from serial port and re-create the command string.

At this point, students can be given the remainder of the program, which takes the received string and compares it with all SCPI commands that might be sent, and generates appropriate responses. Advanced students may even want to try creating the entire program.

Why is Lab 1 important? I/O interfaces are always evolving and industry engineers are expected to understand the benefits of each kind of interface so that they can make smart choices for what to put in an instrument. Knowing something about the current standards can help students “hit the ground running” when they get that first job.

Lab 2. Accuracy vs. Precision

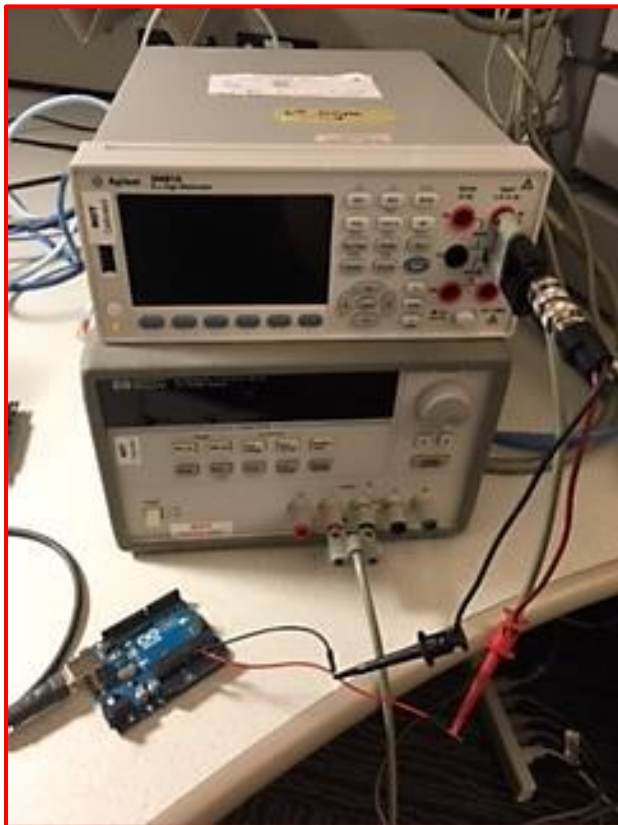


Figure 17. Power supply feeds a DC voltage less than 5V into both the Arduino and the DMM

Students are then asked to load the control program into the Arduino that makes it act like a Keysight 34460A DMM. A power supply feeds a DC voltage into analog input port A0 on the Arduino. Measurements are then taken using BenchVue on both a 3446xA DMM and the Arduino, side by side. The differences in accuracy and precision are immediately apparent. A portion of the lab exercise is shown below (from the answer key):

6. **Calculate** the total measurement error and the percent of input error for a 5V input to the DMM on a 10V range (use 90-day accuracy, and an NPLC of 100)

Total Error: 150uV

Percent of Input Error: 0.003%

*For a 34460 DMM and
 Reading Error = 0.0020% * 5V = 100uV
 Range Error = 0.0005% * 10V = 50uV
 Total Error = 100uV + 50uV = **150uV**
 % of input error = 150uV/5V = **0.003%**

7. Open BenchVue and launch the DMM, simulated DMM, and Power Supply
 8. Set the Voltage Range to be 10V
 9. Set the number of power line cycles for both DMMs to 100
 10. Output 20mV from the Power Supply

Record both measurements:

DMM: 0.020 V

Arduino DMM: 0.00977 V

11. For the Arduino DMM:

Calculate the Expected Accuracy: 75.6% $\left(\frac{0.02 - \frac{5}{2^{10}}}{0.02} \right)$

Calculate the Actual Accuracy: 48.85% $\left(1 - \frac{|\text{Measured} - \text{Actual}|}{\text{Actual}} \text{ or } 1 - \frac{|0.00977 - 0.02|}{0.02} \right)$

Figure 18. A portion of Lab 2 showing calculations for accuracy

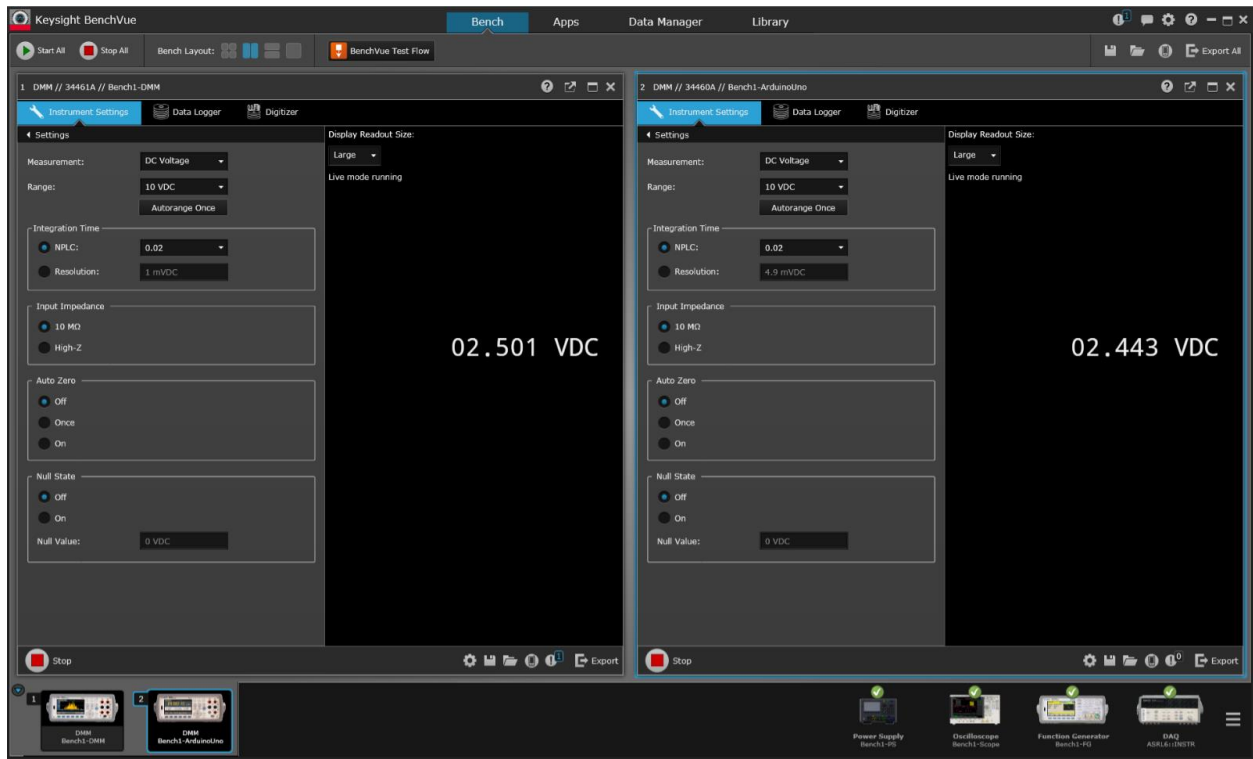


Figure 19. BenchVue software, with the Keysight DMM on the left side, and the Arduino DMM on the right. The power supply was putting out 2.500V.

Lab 2 is important because it:

1. Reinforces the difference between accuracy and precision, two very important engineering concepts
2. Shows off software commonly used in industry to interface to instruments

Lab 3. Input protection.

Why can you put almost any voltage up to 1000V into a commercial DMM without destroying it, even reversing the polarity? How can we protect the Arduino so that abuse will not fry it?

We first explore the use of a Zener diode (Figure 20 and 21).

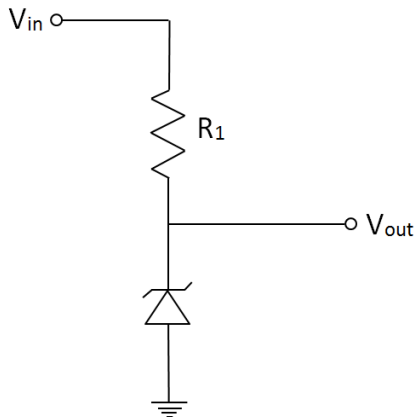


Figure 20. Zener protection

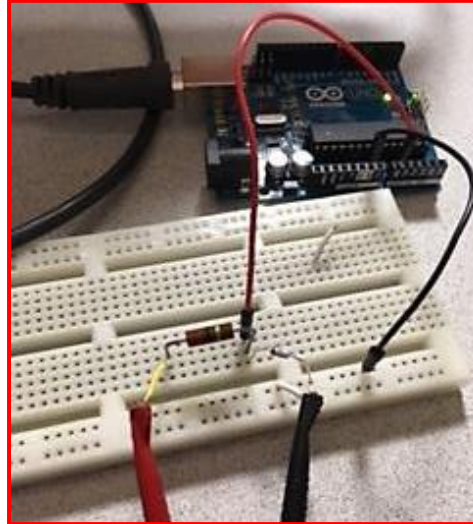


Figure 21. Zener and R1 wired into the circuit

Students make a chart of DMM reading while feeding in voltages higher than the Arduino would normally be able to take:

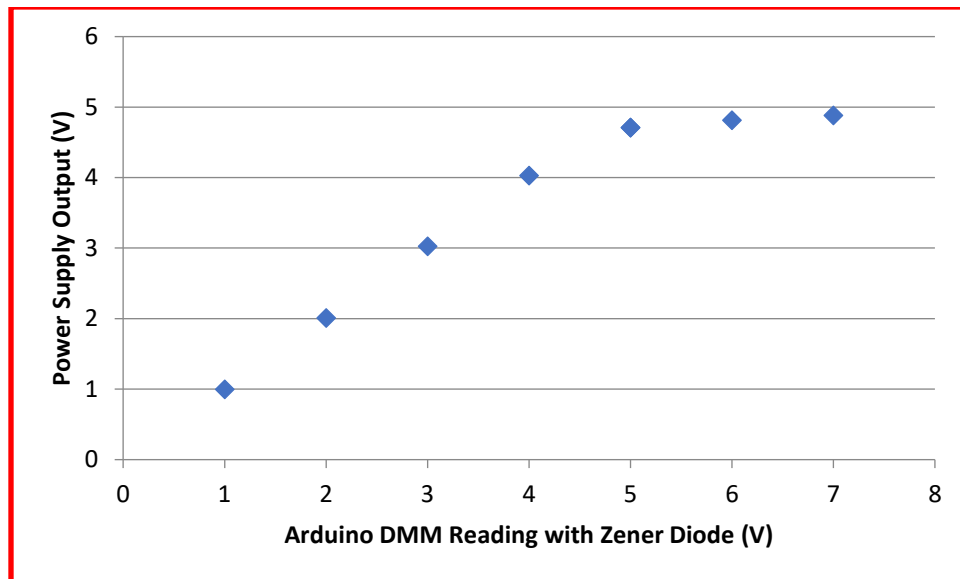


Figure 22. Chart showing effect of adding a Zener diode to the input for protection

While this works, it changes the accuracy of the measurement. So we next explore the use of a resistive voltage divider and show how to modify the code so that it can generate correct voltages despite measuring lower voltages.

Next, students learn how to use an operational amplifier to handle both positive and negative voltages. The circuit shown in Figure 23 is wired on a prototyping board and connected to two supplies, one positive and one negative. No matter what polarity of signal comes in, the output is always positive.

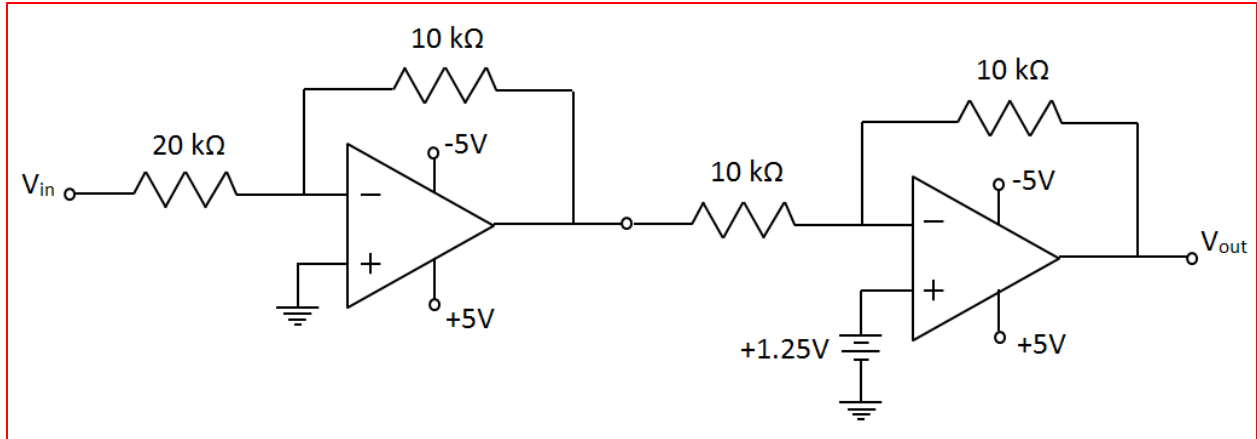


Figure 23. Op-amp circuit used as protection and with ability to handle forward and reversed polarity

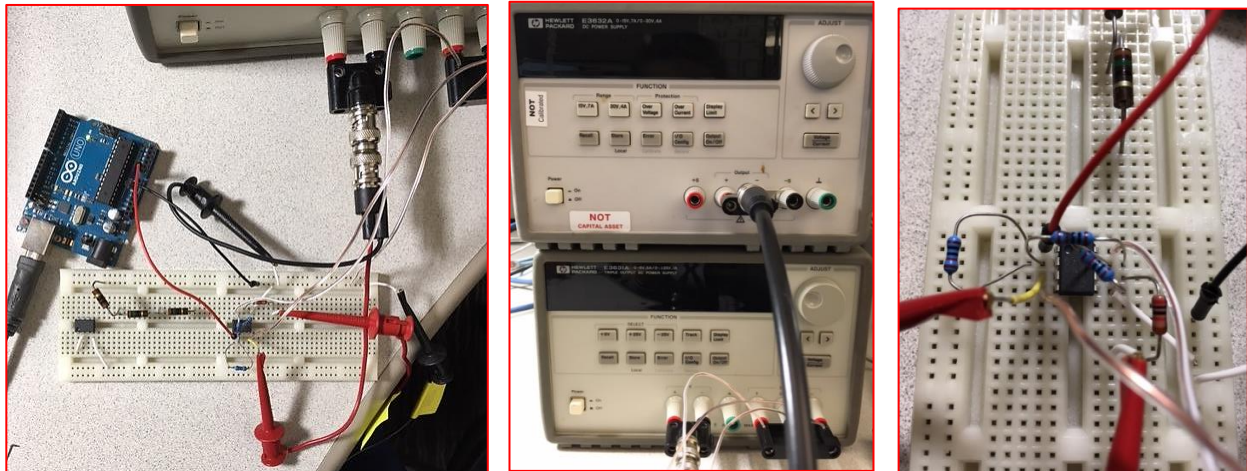


Figure 24. Connections to the power supplies and wiring of op-amp circuit

The op-amp circuit changes the voltage read by the Arduino, so the program must be modified to compensate. The modified program is shown in Figure 25.

```

void do_fetch ()
{
  float x;
  String vs, es;

  x = ((5 * v / 1023)/10*2)-5;
  if (x > range*1.2) Serial.print(F("+9.90000000E+37\n"));
  else if (x < -range*1.2) Serial.print(F("-9.90000000E+37\n"));
  else {
    if (x >= 0) vs=F("+"); else vs=F(" ");
    if (x > 1.0) {
      x = x / 10.0;
      es = F("E+01");
    }
    else es = F("E+00");
    vs = vs + String(x,8) + es + "\n";
    Serial.print(vs);
  }
}

```

Figure 25. Modified Arduino code to handle the changed readings caused by the op-amp circuit

Lab 3 conveys a very important concept – input protection is not just for positive signals! When a negative voltage is present, it can also damage electronics that were not designed to accommodate it.

Lab 4. Sources of Noise.

Internally-generated noise and external sources can also affect reading accuracy in a DMM. This lab shows how power-line noise (50 or 60 Hz) can be canceled out by “integrating” the signal for one whole power-line cycle. A portion of the lab is shown in Figure 26. In this lab, a function generator is used to produce a DC voltage with an AC signal “riding” on it.

Measurements

6. Set the range for both instruments to 10V
7. **Output** 2V from the Power Supply
8. **Record** the measurements in the table below for each of the given integration times

Integration Time (NPLC)	3446xA Digital Multimeter (V)	Arduino DMM (V)
0.02	2.004	2.008
0.2	2.0046	2.0088
1	2.0045	2.0088
10	2.00458	2.00879
100	2.00457	2.00879

9. **Connect** the Arduino to the Function Generator
10. **Output** a noise signal from the function generator with a 50 mV_{pp} amplitude and the offsets listed below

Voltage Offset	3446xA Digital Multimeter	Arduino DMM (V)
0V	0.134 mV	0.000 V
1V	0.999512 V	0.992 V
2V	1.99866 V	2.013 V

Figure 26. Effects of noise on DMM readings

Noise is a complicated topic, one that bears considerably more attention than is typically taught. In industry, removing noise from measurements is important because the goal is to see the signal, not the noise. This concept extends into the RF world as well!

Lab 5. Types of A/D Converters

The rubber hits the road in this lab! There are many types of analog to digital converters that have been used to measure voltage. These are the typical ones:

- Successive approximation (used in Arduino – very slow)
- Flash (cheap, uses a lot of parts)
- Dual-slope (used in DMMs starting in the 1970s – very accurate)
- Multi-slope (used in DMMs starting in the 1980s – more accurate)
- Delta-Sigma (used in Keysight DMMs today – best)

Figure 27 is a portion of lab 5 showing how dual-slope A/D converters work.

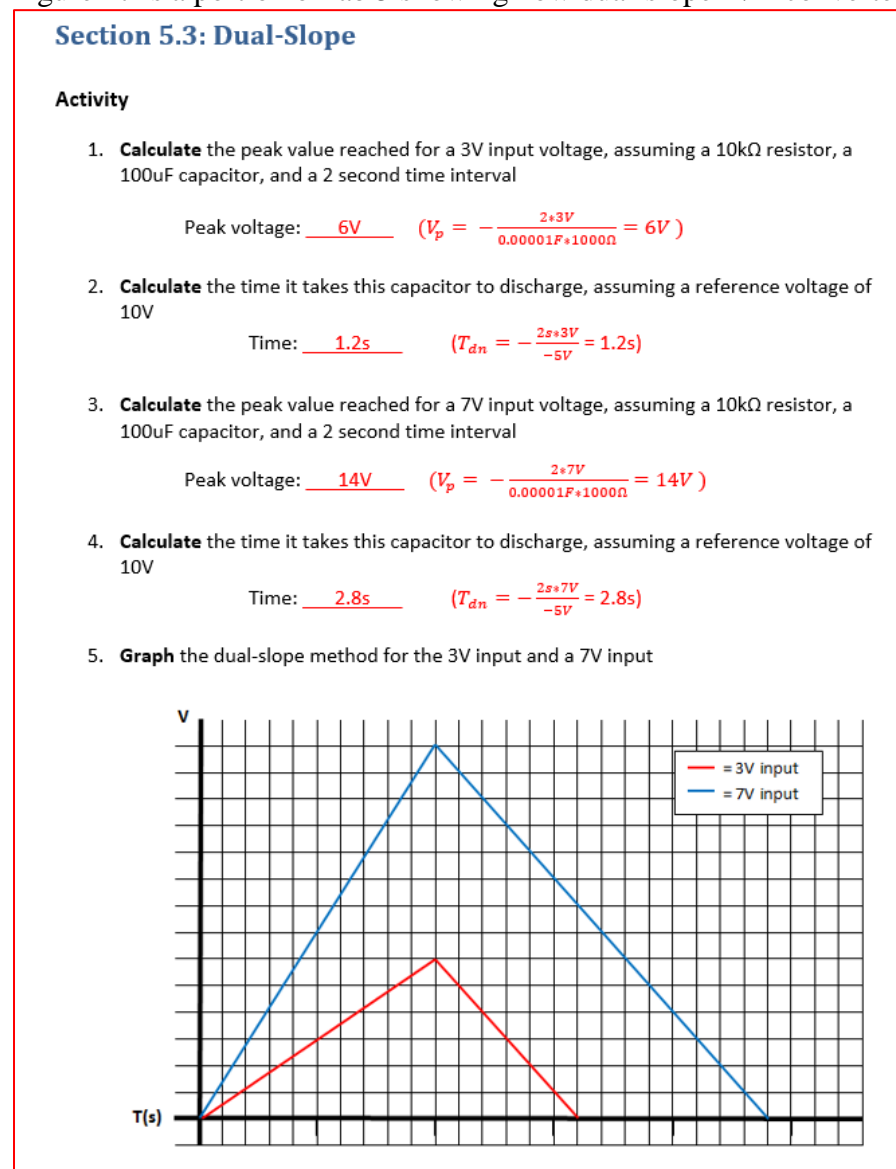


Figure 27. A portion of Lab 5 showing dual-slope A/D.

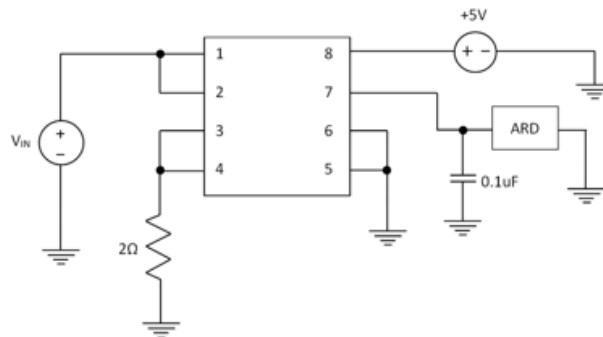
Lab 5 is important because as the electronics industry moves more and more analog “stuff” into software, the front-end becomes more important. The term “garbage-in, garbage-out” comes to mind. Fast, clean, accurate circuitry is needed so that the analog world can be accurately conveyed to the software world. Because of the trend toward doing everything in software, analog designers are becoming a rare breed. That makes them invaluable to industry! An analog designer who also understands A/D conversion is even more valuable.

Lab 6. Extend the Arduino’s measurement capabilities

With a few changes and some additional circuitry, Arduino can measure current, AC voltage, Frequency and Resistance.

For current measurement, we add a Hall Effect current sensor to the breadboard as shown in Figure 28.

4. **Connect** the ASC723 to the Arduino using the breadboard and wires (refer to datasheet)
 - a. Use a $0.1\mu\text{F}$ capacitor to protect the Arduino (VIOUT) and a 2Ω , 10W resistor for the current that is being measured (IP-)



5. Before connecting the Arduino, use a DMM to check the voltage output
 - a. **Output** 2V and record the voltage measured from the DMM
Voltage: 0.836 V
 - b. **Record** the current output from the power supply

Figure 28. Adding current measurement to Arduino using a Hall Effect current transducer

Lab 6 is important because it shows that one instrument can be tuned for different tasks with very little effort. A few additional parts and a few tweaks to the software, and we've turned a DVM into a DMM. The concept can be applied to any design project.

Lab 7. Measurement Speed.

Most modern DMMs have an output signal called "VM Complete" that is a pulse that is the duration of a measurement. By looking at the time between measurements, we can see how long the measurement took as well as how many readings we can get per second. In this lab, we add a "Voltmeter complete" output to digital output 7 of the Arduino that is normally high, goes low at the start of an A/D conversion and goes high at the completion. It is then fed to an oscilloscope and compared it to the "VM Complete" output from the 3446xA.

```
// Setup
void setup() {
  analogReference(DEFAULT);
  pinMode(7, OUTPUT);
  digitalWrite(7, HIGH); //initialize pin to high for pulse
  Serial.begin(9600);
  Serial.setTimeout(30000);
}

v = analogRead(0);
if (autorange == 1) {
  x = 5*v/1023;
  for (i=1; i<=4; i++) {
    if (x > range*1.2) range = range * 10.0;
    else if (x < range*0.1) range = range*0.1;
    if (range >= 1000.0 || range <= 0.1) break;
  }
}

digitalWrite(7,LOW); //send pulse after reading is done
digitalWrite(7,HIGH);
```

Figure 29. Arduino code to generate a "VM Complete" signal so that measurement speed can be seen on a scope.

Typical Scope results are shown in Figure 30:

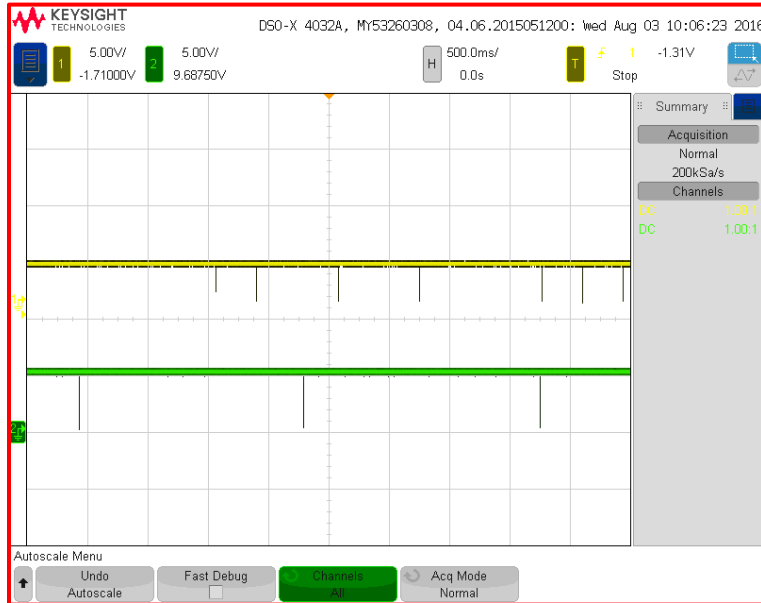


Figure 30. Voltmeter complete signals are shown for both DMMs. (Resolution was low on this screen capture, resulting in an incomplete picture)

Lab 7 explores a topic that is crucial in manufacturing test. While it may not matter much how fast a DMM can take a reading at an engineer's lab bench, it is of paramount importance in manufacturing! The faster a test system can take measurements, the faster a product can be tested. That means more products can be tested in a smaller time, making it possible to deliver products faster and at lower cost.

Development and use of the advanced Lab experiments

The lab experiments described above were developed in collaboration with and tested by an intern from the University of Colorado, between her Junior and Senior years in college. There has been no opportunity to test them in class teaching as yet. Several of our faculty contacts expressed interest in adding these labs to their courses but, to the best of our knowledge, these plans have not been materialized so far. One of the goals of our report at ASEE Conference is to reach a broader audience of educators who might be interested in the implementation of this set of experiments in their teaching. Detailed descriptions may be provided upon request.

Conclusions.

Arduino has plenty of advantages: it provides all basic functionalities of a microcontroller at affordable price; it easily connects to a wide variety of sensors (also, affordable) and actuators; literature on cool projects is abundant. However, there seems to be a gap, due to an obvious reason.

The books on Arduino usually teach how to build simple circuits (connect sensors and actuators); how to understand, reuse, and write your own programs, etc. The gap in instructions seems to be in measurements: readers are seldom informed how to monitor the input signals from sensors and the output signals sent to actuators, etc., although such measurements would definitely help in troubleshooting and modifications of projects. The cause is clear: a suite of test/measurement instruments is ~100-fold costlier than Arduino itself. This price barrier keeps many beginners at the DIY level. However, in the context of Engineering Education, with access to already equipped labs, this gap can and should be bridged.

Bridging this gap may open the road from DIY level to the professional level.

For instance, many books on Arduino include the diagrams that illustrate PWM, or pulse-width modulation, which plays the role of DAC to allow an Arduino output signal to control an actuator [7]. However, without test/measurement instruments at hand, such information might remain abstract. In the university setting, we provide the beginners with access to industry-standard workstations at every desk; encourage them to monitor the signals while working on the projects; use measurements for troubleshooting, etc. Thus, the motivation sparked by cool Arduino projects is geared to foster development of the understanding of how things actually work, which distinguishes professionals from amateurs. Work in this direction is gaining momentum [8 – 10].

The lab experiments detailed here can be tailored to introductory classes by reducing the amount of programming (by providing the programs ready to go); shifting the focus from software to hardware and actual signals used in measurement and control. On the other hand, the higher-level labs can be extended to senior classes where programming is second nature to students. These labs provide a seldom-seen glimpse into the world of commercial instrument development, while using tools that are very familiar to the DIYer. All labs are offered by Keysight free of charge.

We believe that the experiments reported here would help our colleagues who work on course development at various levels of Engineering Education. Further details may be provided per request.

DIYers can be turned into professionals by helping them understand that after graduation, real-world projects require much more rigor and attention to detail than the simple experiments that are typically meant to show concepts. The student who can adapt to the demands of the commercial world will be much better off when starting a career. We believe these labs show that bridging the gap is doable.

References

[1] For example, see the following web sites:

<https://www.quora.com/Why-is-Arduino-so-popular>

<https://makezine.com/2011/02/10/why-the-arduino-won-and-why-its-here-to-stay/>

<https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>

<https://www.baldengineer.com/engineers-hate-arduino.html>

<https://www.elprocus.com/different-types-of-arduino-boards/>

[2] For example, see the following web sites:

<https://www.arduino.cc/en/Main/Education>

<http://www.mindsensors.com/32-arduino>

<https://www.youtube.com/watch?v=TTftjNB9PyE>

<https://www.compadre.org/advlabs/bfy/Detail.cfm?id=4738>

<http://stemify.weebly.com/arduino.html>

[3] For example, see the following web sites:

<https://create.arduino.cc/projecthub>

<https://circuitdigest.com/arduino-projects>

<https://howtomechatronics.com/arduino-projects/>

<https://makezine.com/2015/03/28/20-projects-celebrate-arduino-day/>

<https://www.digitaltrends.com/cool-tech/best-arduino-projects/>

[4] S. Moran, Popular Science, August 19, 2013, “A 12-Year-Old’s Quest To Remake Education, One Arduino At A Time”, [Online] <https://www.popsci.com/technology/article/2013-08/short-circuit#page-5> [Accessed Jan. 25, 2018]

[5] L. Harris, “Arduino in Science education” [Online] Available: <http://moodle.asta.edu.au/course/view.php?id=57> [Accessed Jan. 25, 2018].

[6] A. Tatoglu & I. Russell, University of Hartford, 2016 ASEE Mid-Atlantic Section Conference, “IMPLEMENTING SELF LEARNING SKILLS WITH MULTIDICIPLINARY ROBOTICS COURSES”, [Online] <https://www.hofstra.edu/pdf/academics/colleges/seas/asee-fall-2016/asee-midatlantic-f2016-tatoglu.pdf> [Accessed Jan. 25, 2018]

[7] For example, see M. Geddes “Arduino Project Handbook”, San Francisco, No Starch Press, 2016, p. 31.

[8] H. Rashtian. “A New Application-Oriented Electronic Circuits Course for non-Electrical Engineering Students Using Arduino and NI VirtualBench.”
<https://engineering.ucdavis.edu/eelc/a-new-application-oriented-electronic-circuits-course-for-non-electrical-engineering-students-using-arduino-and-ni-virtualbench/>

[9] P. E. Hertzog. A. J. Swart. “Arduino — Enabling engineering students to obtain academic success in a design-based module.” Published in: [Global Engineering Education Conference \(EDUCON\), 2016 IEEE](#). INSPEC Accession Number: 16005127

[10] Mohammed El-Abd. “A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions.” iJEP – Vol. 7, No. 2, 2017, pp. 79 – 93.