Introductory engineering design and computing using Arduino with MATLAB.

Eric Davishahl, Hung Le, Whatcom Community College, Bellingham, WA

Introduction

Freshman level engineering courses seek to accomplish several things:

- Orient students to the engineering disciplines and career opportunities
- Develop students' analytical and academic skills
- Expose students to engineering practice (design process, report writing)
- Practice and motivate communication skills (teamwork, oral presentations)
- Introduce computer programming

This poster presents a work-in-progress to integrate these goals in a freshman level design and computing course at Whatcom Community College. The course teaches MATLAB® programming in the context of Arduino® microcontroller applications to engineering design.

Why Arduino?

Several recent ASEE papers identify microcontrollers as a promising platform for freshman-level engineering [1], [2], [3], [4], [5]

- Arduino microcontrollers and associated electronics are inexpensive
- Microcontrollers offer an engaging way to introduce students to programming.
- Teaching programming in the context of design can motivate effort.
- Provides an introduction to basic electrical engineering concepts.

Why MATLAB?

- High level interpreted programming language that is relatively easy to learn
- Prepare students for MATLAB projects in later engineering courses
- Prevents students from "hacking" together Arduino C code they find on the internet and do not understand

Others have noted that learning two programming languages at once can be too much for freshman [4]. We limit code writing to MATLAB to avoid overwhelming novice programmers

```
%Setup
%Define a string variable named sensorPin and assign it 'A0' to reference
%analog pin 0 on the Arduino that the pot is conected to.
sensorPin = 'A0';
%Define a string variable named ledPin and assign it 'D13' to reference
%digital pin 13 on the Arduino that the LED is wired to.
ledPin ='D13';
%The infitinte loop below will run the commands inside forever.
%Use CTRL-C to break.
while(1)
    %read voltage in sensorPin and store the value in sensorValue
    %will range 0 to 5V depending on position of knob
    sensorValue = readVoltage(a, sensorPin);
    %Turn the LED on
    writeDigitalPin(a,ledPin, 1);  %Turn pin 13 ON
    pause(sensorValue);   %pause for the value of sensorValue
    writeDigitalPin(a,ledPin, 0);  %Turn pin 13 OFF
    pause(sensorValue/2);   %pause for half the value of sensorValue
end
```

Figure 1. Example MATLAB code to control the blinking rate of an LED with a potentiometer. From *Sparkfun Tinker Kit for MATLAB*

Figure 2. Example Arduino C code to control the blinking rate of an LED with a potentiometer. [6]

```
//Create global variables (variables that can be used anywhere in our sketch)
// Here we're creating a variable called "sensorPin" of type "int"
// and initializing it to have the value "0," which is the analog input pin for the pot
int sensorPin = 0;
// Variable for storing the pin number that the LED is connected to
int ledPin = 13;
// this function runs once when the sketch starts up
void setup()
{
  //set ledPin (13) as an OUTPUT
  pinMode(ledPin, OUTPUT);
}
// this function runs repeatedly after setup() finishes
void loop()
{
  //create a local variable (variable that can only be used inside of loop() to store
  //a sensor value called sensorValue
  int sensorValue;
  //use the analogRead() function to read sensorPin and store the value in sensorValue
  sensorValue = analogRead(sensorPin);
  // Turn the LED on
  digitalWrite(ledPin, HIGH);
  delay(sensorValue);
  // Turn the LED off
  digitalWrite(ledPin, LOW);
 //delay for the value of sensorValue
  delay(sensorValue);
  //loop back to the top
}
```

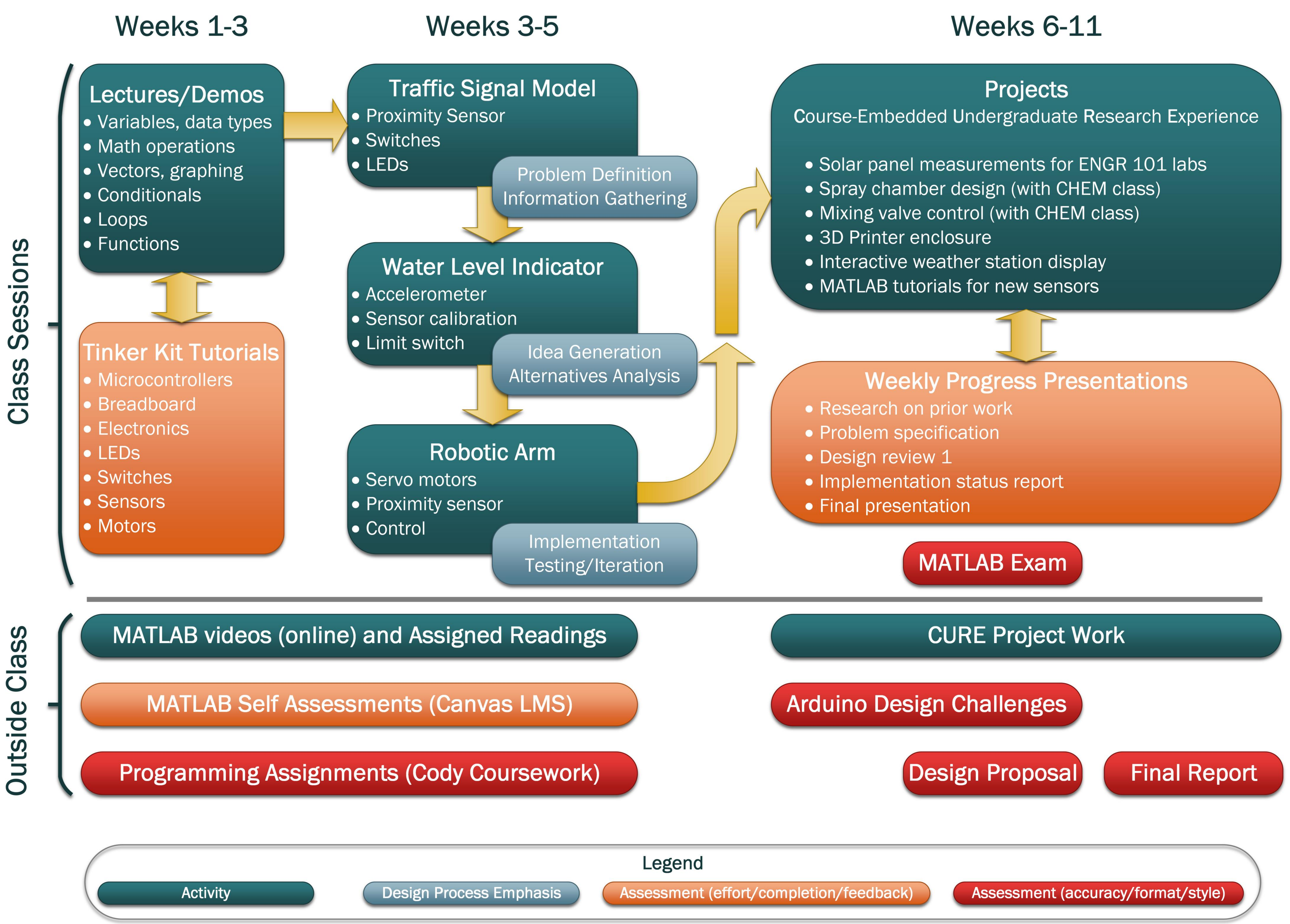
How it Works

MATLAB has offered native support for Arduino hardware since 2014. The software communicates via serial port with an I/O server running on the Arduino board. MATLAB commands send input to or receive output from the Arduino board while the control code runs in a standard MATLAB session.

Limitations/Drawbacks

- MATLAB is not free open-source software
- No stand-alone microcontroller operation because the serial interface requires connection between PC and Arduino to run control code.
- Communication latency can cause problems for some sensors and/or control strategies

Course Design



Short Structured Design and Exploration Projects

In-class projects that integrate students' developing MATLAB skills with more sophisticated microcontroller applications in the context of practicing stages of the engineering design process.

Teams given partially completed mechanism/structural designs and template MATLAB mfiles with code gaps they need to complete.

Water Level Indicator Project

- Application of microcontrollers and programming to Civil Engineering
- Student teams supplied with float arm consisting of:
  - Accelerometer
  - Limit switch
  - Float (3D printed in ABS plastic)
- Determine accelerometer from analog voltage readings in two axis directions
- Mechanical design connecting float arm to wall of water bucket
- Calibration of tilt angle measurement to water depth
- Design and program LED array depth indicator and limit switch triggered alarm

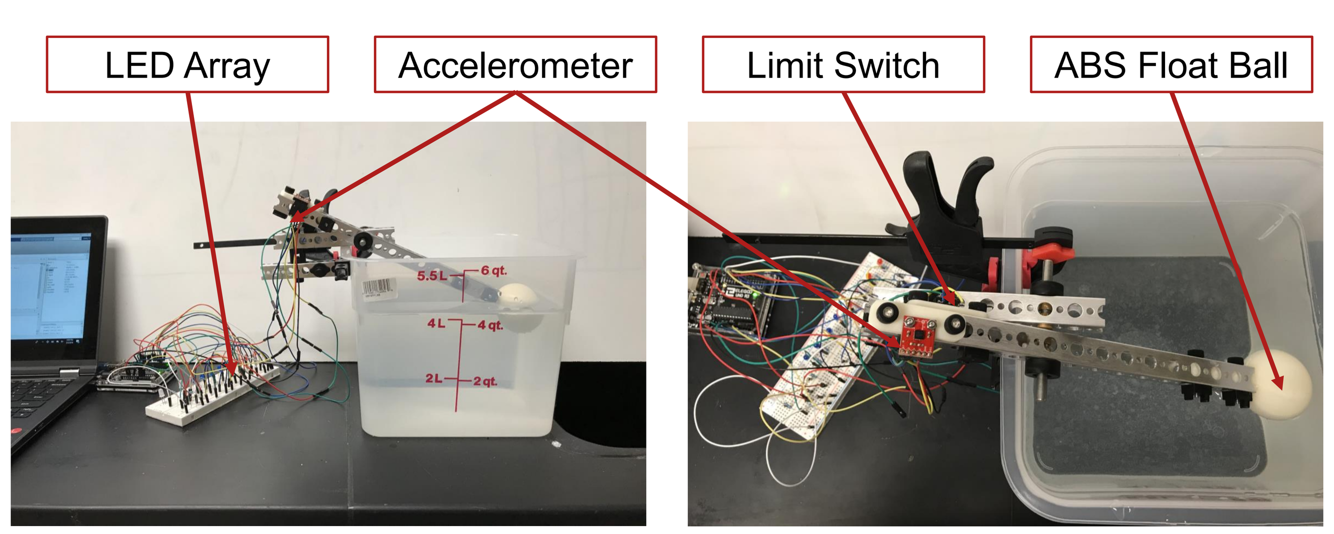


Figure 3. Water level indicator project example front view (left) and top view (right).

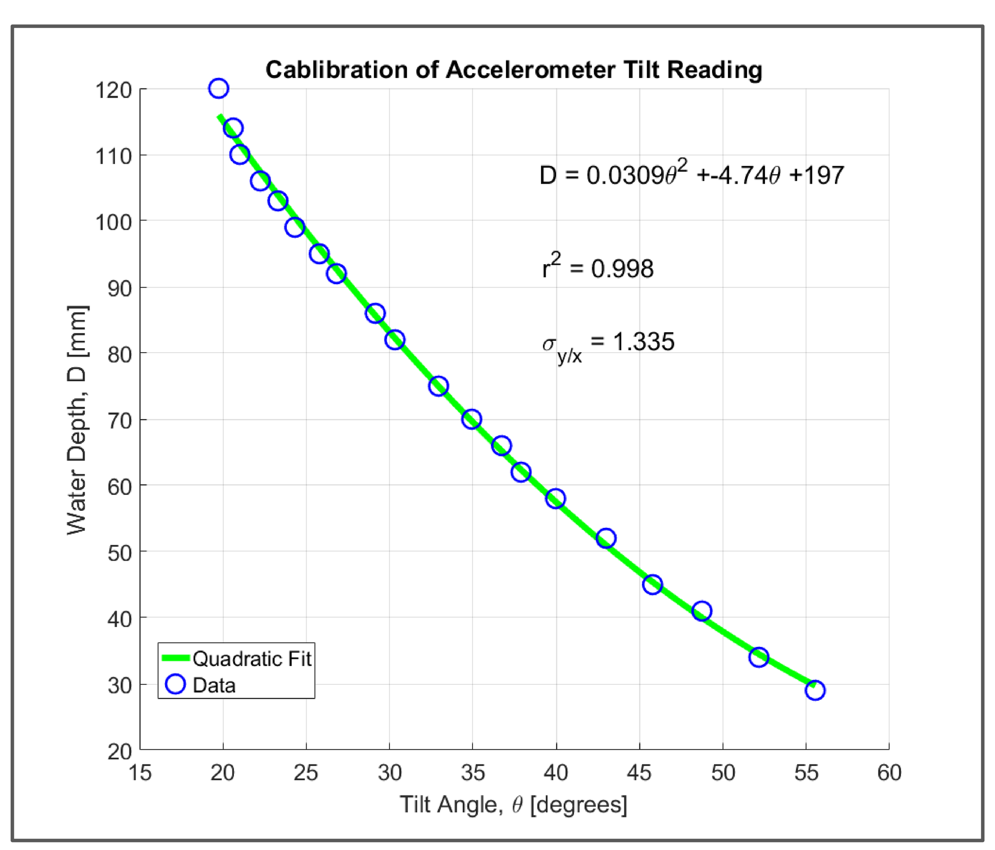


Figure 4. Students measure depths at series of tilt angles to generate a calibration curve for their design.

Course Description

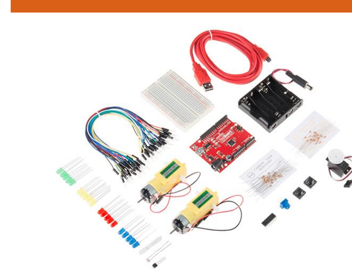
Project based experience with the engineering design process and technical computing. Explores the role of creativity, teamwork, and communication in promoting innovative design. Includes an introduction to computer programming, data analysis, sensors, and microcontrollers. Prerequisite: MATH& 152 (Integral Calculus) or both MATH& 142 (Pre-calculus) and ENGR 101 (Intro to Engineering); and placement in ENGL& 101.

Learning Outcomes

Upon successful completion of this course, each student should be able to…

1. Explain and apply the engineering design process.
2. Generate design ideas using strategies such as brainstorming, design heuristics and morphological analysis.
3. Formulate a detailed specification of constraints and performance objectives by analyzing a design problem.
4. Document engineering design work using reporting formats such as proposals, presentations, technical posters, and reports.
5. Develop a project team by applying strategies including peer feedback, team process reflection, and meeting documentation.
6. Perform data analysis, graphing, and microcontroller programming using scientific computing software.
7. Develop and debug simple computer programs that include data type control, variable assignments and arrays.
8. Develop and debug computer programs of increasing complexity by implementing loops, conditionals, and functions.

Equipment



SparkFun Tinker Kit - $49.95

- Students pay materials fee
- Issued to each student at second class

Team Projects

- PITSCO Tetrix® Prime Robotics Set
- Arduino MEGA 2560 R3
- SparkFun ADXL335 accelerometer
- SparkFun HC-SR04 ultrasonic sensor
- Assorted switches, jumpers, LEDs

Total cost approx. $500 per team

Course Resources

SparkFun TinkerKit for MATLAB, MathWorks File Exchange, https://www.mathworks.com/matlabcentral/fileexchange/62698-sparkfun-tinkerkit-for-matlab by E. Davishahl (2017)

MATLAB® OnRamp interactive tutorial by MathWorks® https://www.mathworks.com/training-schedule/matlab-onramp.html

MATLAB Basics lecture videos available at Davishahl Numerical Methods youtube channel https://www.youtube.com/channel/UCp3kUgiL-JY5Rt5er00WB6A

MathWorks® Cody™ Coursework™ https://coursework.mathworks.com/

CATME for team development https://www.catme.org

Acknowledgements

Special thanks to the Alcoa foundation for the grant that funded the purchase of most equipment for this course.

Works Cited

[1] R. d. Guzman, J. C. Vaccaro, A. H. Pesch and K. C. Craig, "Freshman Engineering Problem Solving with MATLAB for All Disciplines," *ASEE Annual Conference & Exposition*, New Orleans, LA, 2016.

[2] G. W. Recktenwald and D. E. Hall, "Using Arduino as a platform for programming, design and measurement in a freshman engineering course," *ASEE Annual Conference & Exposition*, Vancouver, BC, 2011.

[3] J. A. Riofrio and S. G. Northrup, "Teaching Undergraduate Introductory Course to Mechatronics in the Mechanical," *ASEE Annual Conference & Exposition*, Atlanta, GA, 2013.

[4] D. J. Mascaro and S. Mascaro, "An Integrated Project-Driven Course in Computer Programming for Mechanical," *ASEE Annual Conference & Exposition*, Seattle, WA, 2015.

[5] P. Wong and B. Pejcinovic, "Teaching MATLAB and C Programming in First Year Electrical Engineering," *ASEE Annual Conference & Exposition*, Seattle, WA, 2015.

[6] *Experiment Guide for the SparkFun Tinker Kit*, SparkFun Electronics. https://learn.sparkfun.com/tutorials/experiment-guide-for-the-sparkfun-tinker-kit/experiment-2-