3IE1 - ERU - Fall 2018

Student Course Syllabus
Online version of notes: https://goo.gl/WBdaiE

Class Schedule and Location:

Date	Time	Location
Friday, 02-Nov	1730 - 2100	Thode Makerspace (Basement, Thode Library)
Saturday, 03-Nov	0900 - 1600	Thode Makerspace (Basement, Thode Library)

Equipment

Each group will be given a kit containing:

- Arduino, USB connector cables
- Breadboards, jumper wires
- 1K and 10K potentiometer
- 2 pushbuttons
- red, green and blue LEDs, 1 (or maybe 2) RGB LEDs
- 220 ohm; 1K; 10K resistors
- 1 photocell, 1 thermistor, 1 piezo speaker

Course Preparation

- On your laptop, download the most current version (1.8.7 at the time of this document) of the Arduino IDE software (NOT the web editor) from the arduino download page
- Complete this short technology survey to help establish technology needs for the course.

Deliverables

The class will be graded on a pass/fail basis. There will be a total of four evaluation components; all components must be passed in order to pass the course.

- Under instructor guidance and working in pairs, students will learn the fundamentals by building a number of diverse and increasingly complex electronic devices.
- Working in pairs, students will apply their fundamental knowledge to build a working device of their own design. At the end of the course, groups will present their final device, and discuss the challenges they faced in its development. Outcomes of this activity will be evaluated according to the creativity of the device, its degree of difficulty, and the ingenuity applied to create it.
- Groups will be required to document their project by sharing their commented, working code to Github.
- Each group will complete a short summary and reflection in the form of a blog post (with at least one video of the working device included). With the students' permission, the blog post will be displayed publicly. The post will provide groups a means of documenting final products, and reflecting on their work and learning in the course. Blog posts will be evaluated according to completeness and depth of reflection.

Class Content

Friday (1730-2100)

Introductions (Slideshow link: https://goo.gl/qzBEKW)

- Introduction to instructors
- Student introductions
- Course introduction
    - Structure
    - Expectations
    - Deliverables / Assessment

Getting Started with Arduinos

- Open up and inspect the Arduinos
    - Explain the general types of connections and ports
    - Power (5V, 3.5V), Ground
    - Digital I/O ports,
    - Analog ports
- General introduction to the Arduino
    - What is the board?
    - What does it allow you to do?
    - How do you connect to it / run it?
    - How do you build things?
- Hardware and Software
    - Plug in the board to a laptop USB port
    - Download and install version 1.8.7 of the IDE (software) from the arduino download page
    - Open the Arduino IDE on the laptop, establish connection to the Arduino board.
    - Discuss blink as a pre-installed and running program [but no need to explain it right now]
- Arduino Components -- Go through kits and explain how they work.
    - Wires, resistors
    - LEDs
    - Trimpots (potentiometers)
    - Photoresistors & thermistor
    - Piezos, buttons
- Simple Examples (See table, below):

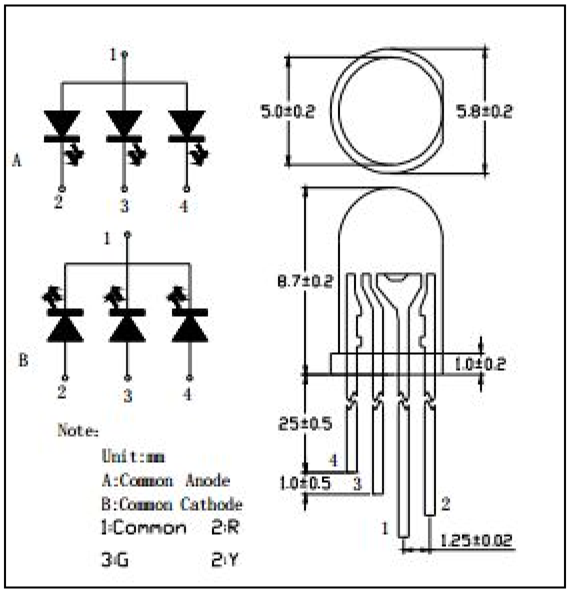
Friday - Introductory	
Blink (on board)	Explain that the program ‘blink’ is already running, and that the board itself has an on-board LED. Open up code in IDE. Explain that the code is what provides the instructions to the board. Briefly describe what the code is doing, block by block.

	Have the students change the delay in blink and send the new program to the Arduino
Blink LED by plugging LED into pin 13 & GND	Explain that “Pin 13” also corresponds to an I/O pin. Have students insert an LED into 13 & GND Explain that LEDs are unidirectional polarized devices -- - current enters through the anode and leaves through the cathode - The cathode is the flat side (cat is flat). Explain (really briefly) what a PWM pin is, how it works, how you identify one, and how it differs from a strictly digital or analog pin. Explain that we’re actually doing a bad thing here, in that we’re not putting the desired resistor in between the power source and the LED - will harm the LED. But how can we get a resistor in between?? CUE THE BREADBOARD
Break out to breadboard; Explore resistors; Include a resistor in their circuit http://arduino.cc/en/tutorial/blink	Explain how breadboards work Go carefully through wiring and breadboards Build Blink on the breadboard with no resistors Send students to find the documentation for blink on the arduino site (google it) should be placed between the source and the LED. Explain how resistors work, in that they resist the flow of charge. Also explain that higher value = more resistance. Introduce resistor colour-coding. - Introduce http://resisto.rs & www.csgnetwork.com/resistcolcalc.html as resources to figure out what resistors you need and have, respectively. Students asked to figure out the resistances for the 3 types of resistors that are included in their kit. They should use a pen to label the tape that holds each pack. Students will take turns plugging the three different resistors into their circuits, as shown in the diagram on the arduino page. - Explain that we don’t always do this (we want to use as close to the proper resistance as possible), but this is for teaching purposes only.
Schematic [If time permits; If not, give introduction on	Refer back to the wiring diagram on the arduino documentation for blink -- this is a type of schematic diagram.

Friday, after rebuilding Blink]	These are an incredibly useful resource when you want to document what you've done, share it, or replicate your work later. Can make by hand, or use assistive software such as fritzing (http://fritzing.org/home/) Ask students to draw a schematic diagram of their circuit -- they'll need it on Friday evening. Include some reference material -- common symbols

Friday - Intermediate	
Students recreate their wiring from the previous day for Blink + resistors on breadboard	They can use schematic if they've created one; otherwise, they can create it from the documentation webpage for the blink project
Schematic [If not done on Thursday]	If not covered previously, introduce schematics: Refer back to the wiring diagram on the arduino documentation for blink -- this is a type of schematic diagram. These are an incredibly useful resource when you want to document what you've done, share it, or replicate your work later. Can make by hand, or use assistive software such as fritzing (http://fritzing.org/home/)
Button: http://arduino.cc/en/Tutorial/Button	Follow the documentation online to complete the button example. We can keep our LED connection intact, since the example is set to output to pin 13 anyway. How the **button** works -- buttons have two legs, which run vertically through the left and right sides of the button -- in that the top and bottom sections of the same leg are always connected. The two legs are connected when the button is pushed down. A **pull-down resistor** holds the logic signal near zero volts when no other active device is connected. It works similarly to a pull-up resistor, except it attaches to ground (essentially, it makes sure that the #2 pin is experiencing a low reading (no voltage)

Moving to a more complex example: Potentiometer to control the brightness of the LED: http://arduino.cc/en/Tutorial/AnalogInOutSerial	Students will follow the documentation to work through this example. Prior to starting, introduce the **Serial Console**, and explain that the arduino can be set to provide serial output to the computer -- In this way, we can transmit meaningful information back to the user. They'll need a good deal of time with this likely. When they achieve it, lets go through the code and explain how it's working. At the end of this, we'll save our sketch under a new name -- reinforce the importance of saving as you go.
Replacing the trim pot with a photoresistor Merging of the following two tutorials: http://arduino.cc/en/Tutorial/AnalogInOutSerial http://arduino.cc/en/Tutorial/AnalogInput	In this example, we're going to work from the code and wiring from the previous example, and replace the trim pot with a photoresistor. The point of this is to demonstrate how both of these items can be used as control devices, and changing voltage (by providing resistance) is how we do this. Students will need to remove the potentiometer setup, and replace it with the photoresistor wiring from the AnalogInput example. Both devices connect through A0 with ground and 5V, so really we just need to add the voltage divider (via 10 kOhm resistor). Requires reading schematic and figuring out how to wire it (but it's not much work) When we first set it up, the light is not going to be changing very much. If we look at Serial Monitor, we'll notice that the sensor and output values don't change much We need to modify the code to remap the sensor value to outputValue = map(sensorValue, 400, 900, 0, 255);
Wiring up an RGB LED https://learn.adafruit.com/adafruit-arduino-lesson-3-rgb-leds/overview https://learn.adafruit.com/adafruit-arduino-lesson-3-rgb-leds/arduino-sketch	Follow the tutorial on the cell to the left. Sketch is included on the last page. A modified version can be found in the SCDS Github repository. **Common Cathode vs Common Anode:** There are two types of RGB LEDs: common cathode and common anode The difference between these two is that current flows through the circuit in opposite directions - From the I/O pins out to GND in common cathode - From the 5V pin out the I/O pins in common anode If we're using a common cathode LED, follow the directions listed in the tutorial If we're using common anode: ● Connect the longest leg (anode) to +5V instead of GND ● **Uncomment #define COMMON_ANODE** in the code.

	Wiring for the RGB LED: 2 1 3 4 2 = Red (nearest to the flat side) | | | | 1 = Common Anode (to 5V) or Common Cathode (to GND) | | | | | | 3 = Green | 4 = Blue **Schematic:**  1 A 2 3 4 1 B 2 3 4 Note: Unit:mm A:Common Anode B:Common Cathode 1:Common 2:R 3:G 2:Y 5.0±0.2 5.8±0.2 8.7±0.2 1.0±0.2 25±0.5 1.0±0.5 1.25±0.02 Wiring for the adafruit common anode RGB LED ([source]) **** Students should save this sketch after they are finished.****
Thermistor as a thermometer http://playground.arduino.cc/ComponentLib/Thermistor2 http://bildr.org/2012/11/thermistor-arduino/	Leave the RGB LED plugged in (except for power), but let's leave that alone and build a thermometer at another end of the board. Follow the documentation given (link to the left), and use the included code to run your thermometer. Let's modify it to output Celsius instead of F -- we're not barbarians, after all. - Save your sketch!
Controlling an RGB LED with a thermistor	Now, it's time to put these two together. We'll need to combine the code from the RGB LED example into the Thermistor example. What we need to do is use the **Temp** variable output from the thermistor to map values to the R, G, B channels of the LED. Students should work on this for a while.
http://playground.arduino.cc/Main/RGBLEDPWM	3 potentiometers to control individual LEDs in a RGB LED (Or perhaps a potentiometer, photoresistor and thermistor -- all controlling different LEDs (or piezos?)

Saturday (0900-1600)

On Saturday, students will put their skills to the test in the first annual ERU make-off challenge. During the allotted time, students will work in groups of 2 (or 3, if there are odd numbers) to build a device that meets (and exceeds) the specifications of a potential investor. They will then have 3 minutes to deliver a final product pitch. The winning group will receive prizes, glory and fame -- but more importantly, we'll all get pizza.

The Scenario:

You have been contacted by a potential investor to design a prototype household device that serves a number of purposes. Your potential investor has requested a device that does the following:

1. Your device must perform two of the following three functions:
    a. Is able to sense the air temperature in a room, and relays general information about room temperature by changing the colour of an RGB LED.
    b. Plays a warning sound when the room temperature either falls below (too cold) or goes above (too warm) a given temperature range.
    c. Senses the intensity of ambient light in a room (i.e. how much light is in a room), and turns on a one-colour LED when it becomes too dark.
2. The device should have additional, specialized functionality in **at least one** of the following categories:
    a. **Visual** - using advanced visual devices such as the 7-segment display (e.g. alarm clock numbers) or RGB LED strip.
    b. **Audio** - including either audio output (speakers) or able to sense sound and perform functions based upon this.
    c. **Mechanical** - providing physical movement (e.g. using a servo, stepper or dc motor)
    d. **Interactivity** - allows advanced interactivity with the user, through the incorporation of one or a number of input devices (e.g. push buttons, potentiometers, trim pots, flex sensors, motion sensors, etc.).

At the end of the day (approximately 15:15), you will be given 30 minutes to develop a 3-minute (max) presentation for your device. In your presentation, you'll want to outline and demonstrate your device's functionality to the potential investor, briefly explain how you built your device, and make your 'pitch' as to why your device is the best.

This project requires you to synthesize your learning over Friday and Saturday, and may also require you to search on the web for examples/demos, etc., which provide the information you need. Your instructors will be there to help you with your projects, and you are welcome (and encouraged) to use your peers as resources, as well.

Schedule:

0900 - 0950	Wrap up unfinished business from the previous day
0950 - 1000	Introduction to your task
1000 - 1230	Work time
1230 - 1300	Lunch (pizza!)
1300 - 1500	Work time II
1500 - 1530	Presentation development time
1530 - 1600	Final presentations; course wrap up

Submitting your final deliverables

Final deliverables to be submitted

Each group is required to submit the following deliverables:

- A fully-commented Arduino sketch (.ino file) for your final device. Your sketch should be commented to the point that an external reader can understand the tasks that your device and sketch are performing.
    - A description of your circuit's wiring should be included as either:
        - a) a textual description in the top (block) comment of your .ino file
        - b) a separate schematic diagram image, created using fritzing and uploaded with the sketch.
- A short summary and reflection, created by all members of the group. The document can be created and uploaded in any format (.txt, .docx, .pdf, etc.), but should include:
    - a description of the device that your group created, along with any challenges that were encountered and overcome
    - A short reflection of the course — What did you learn? Is it useful? How might you apply it in the future?
    - One or more videos should be included with the submission. This can be included as:
        - A URL to the video on a streaming service (e.g. YouTube)
        - A separate video file, uploaded with the summary.

Where and how to upload

All deliverables will be uploaded to the course's Github repository. Only one member of each group needs to submit to the repository. In order to submit:

- A Github account is required for one group member. Create one if no member currently has one.
- Once you create a Github account, email Jay with your username. He will add you to the repository.
- On your local computer, collect all deliverables into a single folder. Name the folder with either a) the last names of members in your group, or b) your self-designated team name.
- Once Jay has added you to the Github repository, navigate to the Submissions folder and click "Upload Files".
    - When prompted, drag the folder from your local PC into the repository.
    - Leave checked "Commit directly to the master branch."
    - Click "Commit changes"