ENGR 40M Project 3c: Making your LED cube awesome (overview)

Prelab due 24 hours before your section, August 7–10
Lab due before your section, August 15–18

You've built your LED cube hardware, and you've programmed an abstraction layer so that you can easily control it by setting an array of values for the LEDs. Now it's time to use those to do something cool.

This lab is somewhat more free-form than the past labs. We will provide handouts to walk you through some of the options, but we encourage you to be creative and try your own ideas, too.

Important note on what to submit: You must complete and submit one of the additional handouts for this lab. If your project is unique, extravagant, and does not pertain to any of the additional lab handouts, your TA may grant you an exemption from the additional handouts, and instead you will submit a report detailing your project's construction (both hardware and software) and functionality. If you have any questions about what to submit, ask your TA!

1 Requirements

To complete this lab, you need to make your LED cube respond to some input from the outside world. There are therefore two design decisions for you to make:

(1) **What type of input you want to use to control your cube.**

You'll probably need to add some hardware to feed an input (digital or analog) to your Arduino. Alternatively, you could use the serial port to interface with your computer.

(2) **How your LED cube will respond to those inputs.**

Your cube should respond in some interesting way: lighting up in various patterns, displaying letters or numbers, playing an animation, or something interactive.

Be creative! There are endless patterns you could do with your LEDs.

Recognizing that some input options are harder than others, we score your submission on complexity. Table 1 below lays out the options you can choose to earn these points. The number of points you must fulfill depends on what you built in lab 3a:

- 6×6 planes need 35 points.
- Planes larger than 6×6 need 30 points.
- All other projects (e.g. the $4 \times 4 \times 4$ cube) need 25 points.

Note: We'll love you if you do both complex hardware and software, but there are no bonus points.

If you propose your own input option, your TA will tell you which row of the above table it goes into, taking into account the fact that you need to figure out how to do it yourself.

You're also welcome (and encouraged) to combine input options. In that case, you will be scored on your *highest* input option, but the fact that you have used multiple types of input will factor into your software complexity score.

As with all labs in this class, it pays to be prepared. Therefore, your prelab for this week is to submit a plan to your TA, with what you intend to do and how you intend to go about it.

> **P1:** Decide what you project you want to implement on your cube and describe it in one or two sentences. Your section leader will give you feedback on whether this meets our complexity requirements.

Points	Hardware option	Software option
5	No additional hardware • Serial monitor input • No input	Simple response • Cube just has to react in some way (on/off)
10	Minor additional hardware • Potentiometer (variable resistor) • Switch	Minor complexity • 2D pattern (plane by plane) • Memory/state • Non-trivial randomness • Raindrop pattern *(see additional handout)*
15	Moderate additional hardware • Audio raw signal *(see additional handout)* • Pushbuttons *(see additional handout)*	Moderate complexity • 3D pattern (*e.g.* expand from corner) • Next pattern is a function of last pattern *e.g. two-player reaction game*
20	Complex additional hardware • Capacitive touch sensing • Audio raw signal and frequency response *(see additional handout)*	Impressive complexity • External Python/C++/Java interface • Complex state space • Extensive decomposition *e.g. space invaders, displaying the weather pulled from online*

Table 1: Points earned for complexity

2 Hardware options

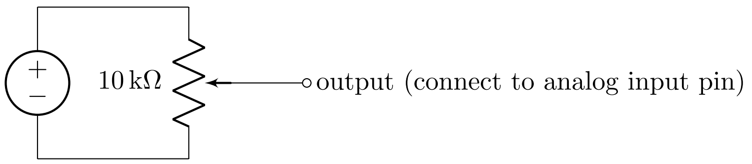
2.1 Audio input

Your lab kit contains an audio socket and an audio cable, with which you can take audio from your computer or phone and feed it into your Arduino.

Audio input uses the *analog input* capabilities of thee Arduino, available on I/O pins A0 through A5. The `analogRead()` function converts the voltage on the pin to a number between 0 and 255. Using this digital data, you can make the cube respond to the signal directly, or respond to a frequency representation of the music. There is a handout that walks you through how to do this.

2.2 Potentiometer

A potentiometer ("pot" in EE-slang) is a variable resistor, controlled by a knob or slider. It works by having a piece of resistive material and a metal contact (known as the wiper) that can slide back and forth across the material. It usually has three terminals: one on each end of the resistor, and one on the wiper. This makes it very easy to build a voltage divider to get a variable output voltage. Connecting the one side of the pot to ground and the other side to V_{DD} makes the voltage at the wiper pin vary from V_{DD} to 0 V. The wiper pin can then be connected to an analog input pin on the Arduino (pins A0 to A5) and read using `analogInput()`.



You can use a potentiometer to control games or other things, similar to how you might use buttons. You can use it to control the position of the light and build a 3-D etch-a-sketch, or just use it to select between different light “shows” you have programmed on your cube.

2.3 Pushbuttons

We have momentary pushbuttons which you can use as input to your cube. You’ll need to write some code “debounce” the buttons, because the mechanical contacts actually bounce when they switch, and this can register as more than one push. There is a handout that walks you through this.

2.4 Capacitive touch sensor(s)

Your homework this week included a question about creating a capacitive sensor using two Arduino pins. Not only can it sense touch, it can also sense when a hand is nearby, and approximately how close it is. You could use this to control your cube by just waving your hands around. If you are going to use this input method, you should review homework problem, and submit your plan for how you are going to build this sensor (the electrical and mechanical design of the sense electrodes) and what resistor value you will use (and the touch range you are trying to get). Since this method of input will be less precise, use it in a situation which doesn’t require great precision.

This method of touch sensing is different than what is used on touch screens, and won’t give you (x, y) position.

2.5 Serial input

We only used the serial port for simple input/output, but connecting your computer to your cube gives you many more options—thanks to having a full keyboard and access to the internet. You can write to it with Python, C/C++, Java, or other language of your choice. With some code on your computer to fetch interesting information, and some code on your Arduino to display it, you can do all kinds of things.

3 Build Quality rubric

Your build quality grade in this project is based on the quality of your breadboarding and testing setup. Avoid stringing together jumper wires and alligator clips.

Plus

- Your project is unique and clever, or you took one of the suggested ideas and really made it shine.
- All additional hardware looks neat on your breadboard - wires are color-coded and about the right length, component leads are trimmed appropriately, and there is no risk of short-circuiting.
- Code is clean, well commented, and well-structured. It is easy to follow and understand.

Check

- Your project is a solid execution of one of the suggested ideas.
- Breadboard could be neater, but isn’t prone to things falling out or short-circuiting.
- Code follows a consistent set of style/indentation rules, but would benefit from more comments and/or better variable names.

Minus

- Your additional hardware is hard to follow - wires are not color-coded and components are at risk of short-circuiting.
- Code doesn't follow a consistent set of style/indentation rules and is poorly commented. Variables may be badly named and "magic numbers" appear in multiple places.