

3 x 3 LED Cube

Programming

Know the limit program within it!

Before programming your cube, it is VERY important to understand how much current draw your programming decisions will result in as too much current draw will damage your microcontroller!

With 5 volts supplied by the Arduino and 180 ohm resistors on each column the resultant current flow through any single LED when turned on is `16.4 mA (milliamps).

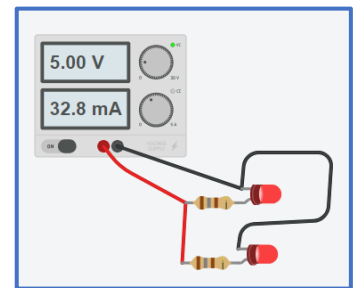
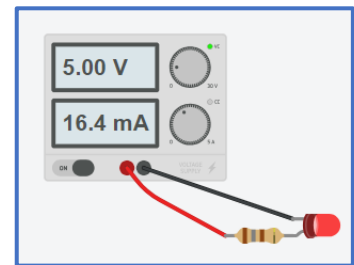
This is important to know because each pin of the Arduino has a current carrying capacity of $\sim 20\text{mA}$ and this means that a single LED with this resistor is safe for your Arduino to power.

As you would expect switching on more LEDs will increase the current flowing. If hooked in a 'parallel' configuration (shown), another 16.4 mA for each LED switched on will be added. The 9 LEDs that make up a 'level' of your cube are connected in parallel and if all nine were to be switched on at once the current draw would near 150 mA!

This is of concern, not so much for the positive feeds to the LEDs as each pin of the Arduino is only flowing 16.4 mA, but it is a concern on the negative side as this will all converge on the Arduino's GND pin and we are very much pushing the maximum allowable here.

Therefore, it is important, to protect the Arduino from the excessive heat that is the result of excessively high currents by employing one or both of the following strategies when programming:

1. Only ever switch on one layer at a time!
2. Avoid turning on LEDs continuously, rather make them appear to be solidly lit by turning them on and off at a rate that the human eye accepts as on. Yes they may flicker a bit, but this is better than not working at all due to microcontroller damage ☺



MAP YOUR CUBE

Ideally, your cube wiring is in keeping with the example and you will just now be confirming this. Do not worry however, if you have wired your cube differently this can likely be fixed in code.

Time to map your cube. Start by downloading and opening the Arduino Sketch called **'cubeMappingCode'** from the class folder on the school's ShareDrive (H:)

Use the program to one by one turn on each LED and record the Arduino pin associated with the LED on the map shown

C2	C5	C8
C3	C6	C9
C4	C7	C10

Now use the program to determine which Arduino pin is associated with each layer and record this here:

Level 1 (bottom) = Pin ____

Level 2 (middle) = Pin ____

Level 3 (top) = Pin ____

Final Function Test

Let's test to make sure everything is working properly. From the class folder on the school's ShareDrive (H:) download and open the Arduino sketch **'cube_LED_Sequence'**.

Compare your test results to the code in the **'cubeMappingCode'** program. If not the same, make the necessary adjustments by reassigning the pins by replacing the pin numbers with the ones you recorded on your Map above.

Now compile and run the program. You should see your cube stepping through each led in column sequence (C1-C9) one layer at a time and then repeating this. If desired, change the delay to make this easier to observe.

If LEDs are not lighting in the correct sequence you will need to troubleshoot the issue by re-checking your cube mapping and the pin assignments you have.

```
cubeMappingCode $
1
2 //establish level pins for
3 // if necessary, change the
4
5 int L1=11; //bottom level
6 int L2=12; //middle level
7 int L3=13; //top level
8
9 /*establish column pins and
10 * if necessary, change the
11
12     2 5 8
13     3 6 9
14     4 7 10
15 */
16
17 int C2=2;
18 int C3=3;
19 int C4=4;
20 int C5=5;
21 int C6=6;
22 int C7=7;
23 int C8=8;
24 int C9=9;
25 int C10=10;
```

Creating Patterns

Now it is time to create various patterns to display on the cube. We will be writing 'functions' for each pattern and then 'calling' the functions as desired in the 'Loop' portion of the program.

From the class folder on the school's ShareDrive (H:) download and open the Arduino sketch '**Cube_Final_Provided_Display**'.

Upload this code to your Nano. Open your serial monitor window. Observe the cube and study the program. By this point in the course the program should be looking very familiar. When ready, try to write an additional function for the cube to display.

When you have this working, **SHOW YOUR TEACHER**, before proceeding.

Extending Tasks

Now that everything is working, and you understand how to program the cube to display different patterns, do the following:

1. Create five completely different patterns from the ones that have been provided and add these to your cubes display loop so that the provided patterns and your new patterns are displayed in sequence.
2. Create code that will display the functions at random. If needed, look back at you final Useless Box code and borrow the appropriate portions of that pertain to randomizing.

Final Assembly

1. If not already completed, carefully assemble your project into its final presentation form.
2. **SHOW YOUR TEACHER YOUR COMPLETED PROJECT.**