ME 220 – Introduction to Mechatronics

Arduino Assignment 01:

Assignment is composed of five parts, but all of them are written in a single file as follows:

When the program runs, it displays on the computer screen for a choice:

```
Enter:
1 for running lights
2 for running lights with pause and resume
3 for binary count up - on timer
4 for binary count up - on button
5 for binary display of numbers [1-255]
```

After this is displayed, the user is to make a choice.

If the user enters an invalid value, you can reply with a proper message and do whatever you want.

If the user enters a valid value, corresponding program should run forever.

Your code is expected to **respond immediately** without any delay to user input. If your implementation turns out to have delays, investigate the parts that you wait for user input and try to figure out how you can make the response of your code faster.

For this assignment, first interface 8 LEDs and one switch properly as shown in Figure 1.

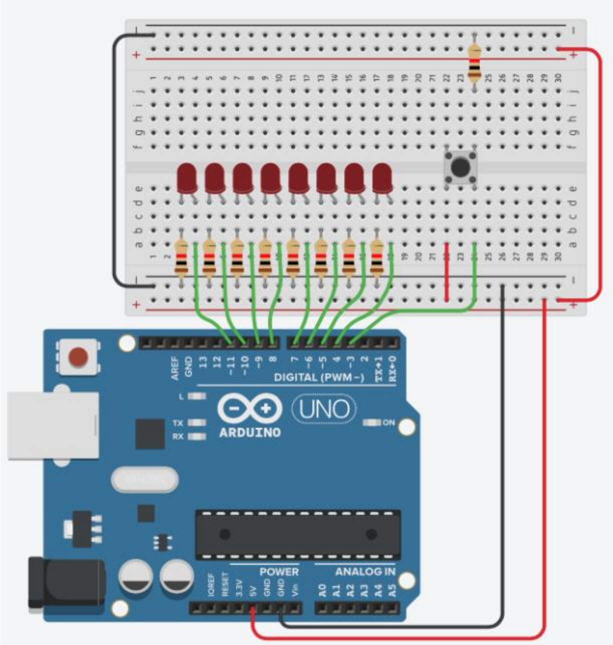

Figure 1: Interfacing of 8 LEDs and a pushbutton to Arduino

Running lights:

In this mode, lights are running (meaning, one out of 8 LEDs are on at a time, and this one moves) from left to right, and then right to left and so on. Check the first couple of seconds of this video out for motivation :)

Running lights with pause and resume:
In this mode, when the button is pressed, running stops, when the button is released, running resumes. Recall button bouncing!

Binary count up - on timer:
In this mode LEDs start from 0 and count up to 255 and then start over from zero.
In every 200 ms count is incremented by one, and "Current count is X" is printed on the Serial Monitor screen. Hence, 8 LEDs serve as a binary display of a byte.

Binary count up - on button:
In this mode LEDs start from 0 and count up to 255 after every time button is pressed and released. Once it reaches to 255 it rolls back to 0.
Every time the count changes and LEDs are updated and "Current count is X" is printed on the Serial Monitor screen.

Binary Display of Numbers [1-255]
In this mode numbers sent by the user in the [1-255] range are displayed, and until a new number is sent, old number is kept on the display.

Note that: Your **code** (excluding the comments**) is expected t**o be as **short** as possible. This means, you may need to write functions for repetitive tasks. Before starting coding, try to draw a flowchart and discover parts that seem to repeat themselves as patterns. Try to find the simplest pattern that solves the assignment.
Also note that, all hardware resources should be defined parametrically, nothing should be hardcoded as we have discussed in class.

HINT: Try to write the **shortest code** in Arduino (excluding comments, I expect **overwhelming amount of comments** and lack of comments will be reflected to grading). Consider using arrays :)
While writing your code use of *bitwise AND and bitwise OR* as well as *bitshift operators* might be helpful. Also, note that there will be 8 LEDs, spending 8 lines to initialize the pin variables and 8 more lines to set them as output is a waste of space. Consider using arrays and loops.