

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.

For this assignment, first interface 8 LED's and one switch properly as shown in class.

Running lights:

In this mode, lights are running (meaning, one out of 8 LED's are on at a time, and this one moves) from left to right, and then right to left and so on.

Running lights with pause and resume:

In this mode, when the button is pressed, running stops

Binary count up - on timer:

In this mode LED's 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.

Binary count up - on button:

In this mode LED's 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 LED's 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.

Remember, everything should be written in a single file and your code (excluding the comments) should be as short as possible. This means, you may need to write functions for repetitive tasks.

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). While writing your code use of [bitwise AND and bitwise OR](#) as well as [bitshift operators](#) might be helpful.