ASSIGNMENT - 1
INPUT / OUTPUT INTERFACE DESIGN

Exercise 1 (A) – Blink

1. To blink the LED with an Arduino for the amount of time equal to the sum of last two digits of your Roll number, leaving the amount of time the LED is 'OFF' to 2 second.
   *(For example, if your Roll No. is 1019***27, the LED must be kept on for 9 second).*

2. Hook up 8 LEDs to pins 2 through 9 (with resistors). Modify the code to turn on each one in order and then extinguish them in order.
   (HINT: hook them up one additional LED at a time and make sure the new one works before you add the next one.)

Hardware/Software Required

- Arduino Board
- Breadboard
- LED and Resistors

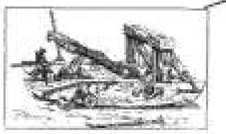
Circuit description

To build the circuit, attach a 100-ohm resistor to either leg of the diode. Attach the leg of the LED connected to the flat edge of the body (the negative leg, called the cathode) to ground. Connect the remaining leg (the positive leg, called the anode) to pin 13. Then plug your Arduino board into your computer, start the Arduino program, and upload the code.

Code

Schematic

Reflections (Conclusions):



Exercise 1 (B) – Push Button

On a breadboard, connect Pushbutton to turn on the built-in LED on pin 13 i.e., when you press the button LED should glow otherwise it should be in off state.

Hardware Required

- Arduino Board
- momentary button or switch
- 10K ohm resistor
- breadboard
- hook-up wire

Circuit description

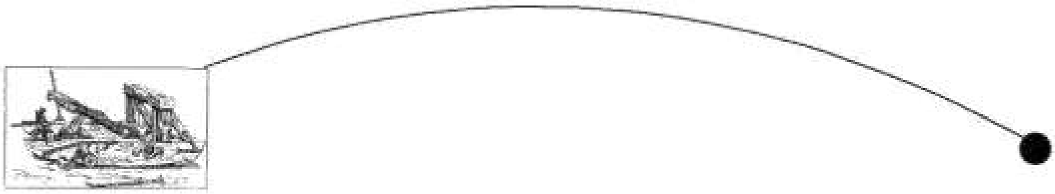
Connect three wires to the Arduino board. The first two, red and black, connect to the two long vertical rows on the side of the breadboard to provide access to the 5 volt supply and ground. The third wire goes from digital pin 10 to one leg of the pushbutton. The same leg of the button connects through a pull-down resistor (here 10 KOhms) to ground. The other leg of the button connects to the 5 volt supply.

When the pushbutton is open (unpressed) there is no connection between the two legs of the pushbutton, so the pin is connected to ground (through the pull-down resistor) and we read a LOW. When the button is closed (pressed), it makes a connection between its two legs, connecting the pin to 5 volts, so that we read a HIGH.

You can also wire this circuit the opposite way, with a pullup resistor keeping the input HIGH, and going LOW when the button is pressed. If so, the behaviour of the sketch will be reversed, with the LED normally on and turning off when you press the button.

If you disconnect the digital I/O pin from everything, the LED may blink erratically. This is because the input is "floating" - that is, it will randomly return either HIGH or LOW. That's why you need a pull-up or pull-down resistor in the circuit.

Code



Schematic

Reflections (Conclusions):

Exercise 1 (C) – Tinkercad

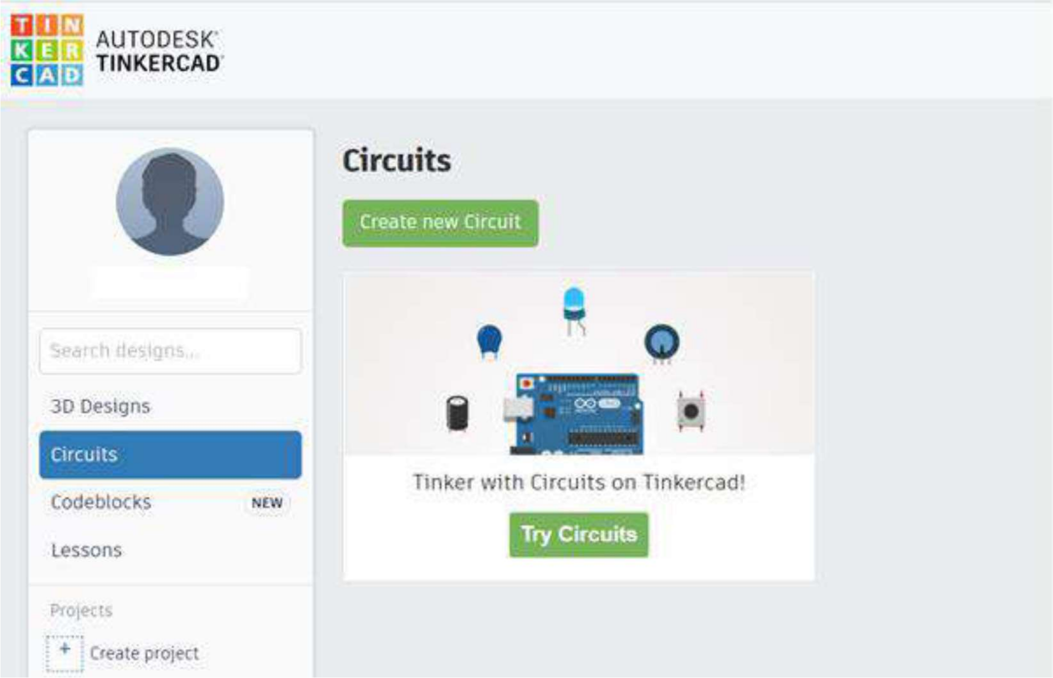
Tinkercad is a free, online collection of software tools for 3D design, electronics, and coding

Software Required

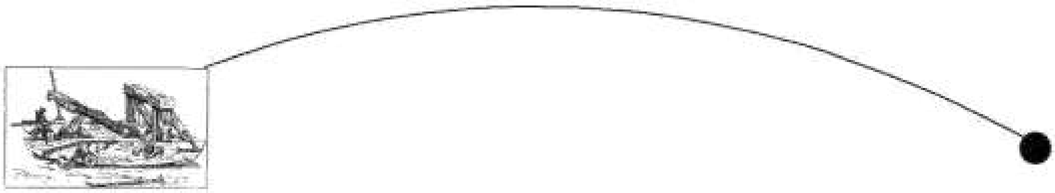
- Tinkercad Software tool (https://www.tinkercad.com/)

Getting Started:

1. Visit https://www.tinkercad.com/
2. Sign in through your google account (Thapar Email ID only)
3. On the Dashboard, select **Circuits** from the drop box and click on **Create new Circuit**



Tinker with Circuits on Tinkercad!



Assignment Tasks:

Note: Each student must attach separate sheets for submitting the below mentioned Assignment Task A-C.

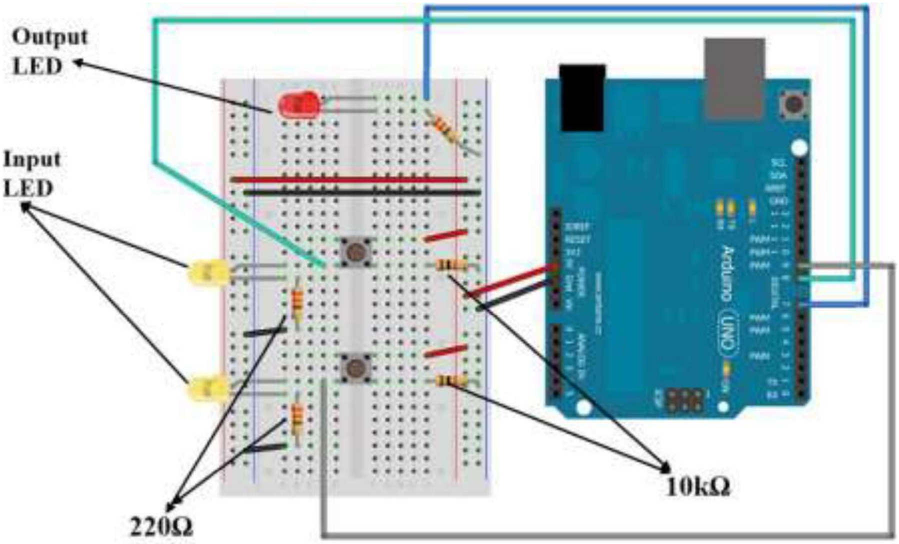
A. Using Tinkercad, hook up 5 LEDs to pins 2 through 6 (with resistors). Modify the code to turn on each one in order and then extinguish them in order.
(HINT: hook them up one additional LED at a time and make sure the new one works before you add the next one.)

B. With the help of Tinkercad, use push buttons to simulate the behaviour of listed logic gates (Without using ICs of logic gates)

a) Logic gates: AND, NAND, XOR for students with odd numbered Roll Number.

b) Logic gates: OR, NOR, XNOR for students with even numbered Roll Number.

Hint:


Output LED
Input LED
220Ω
10kΩ
Arduino
UNO