LABORATORY 1

Introduction to the Arduino board and kit

OBJECTIVES

1. To learn basic functions and operation of the Arduino board.
2. Powering the board up, testing, loading an example sketch, using the serial monitor.
3. Update the Arduino library, connecting digital sensors

INFORMATION

- Arduino software is easy-to-use for beginners, yet flexible enough for advanced users.
- Cross-platform - runs on Mac, Windows, and Linux.
- Open source and extensible software. Its AVR C programming language can be expanded through C++ libraries.
- Open source and extensible hardware. A large variety of compatible sensors and actuators available, making different hardware designs achievable in a short time.

1. Arduino Uno board

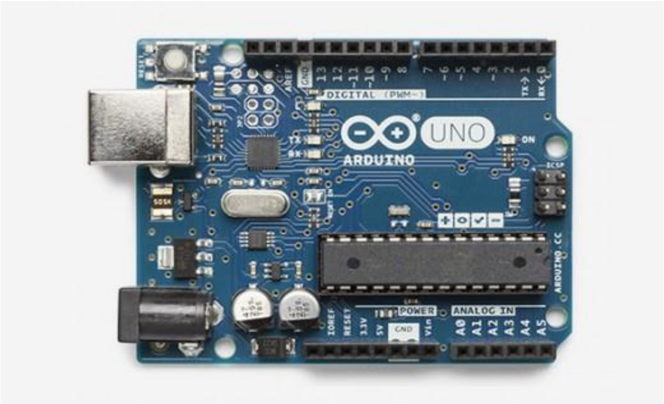


Figure 1.1. Arduino Uno board.

- **Arduino Uno** is a microcontroller board based on the Atmega 328P
- Provides 14 digital input/output pins (of which 6 can be used as PWM outputs)
- 6 analog inputs
- 16 MHz quartz crystal
- USB connection, a power jack, an ICSP header and a reset button.

The Arduino Uno board can be powered via a USB port of the laptop or from a separate 7 -12 VDC Centre positive power supply. External 9V battery power is also available.

1.1. Arduino Power Header

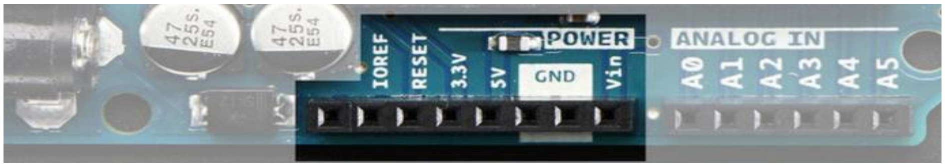


Figure 1.2. Power Header

- **Vin -** connected to the power input from the DC jack, so it is going to range from 7 V to 12 VDC, depending on what is plugged into the DC jack. If the DC jack is not powered, it will provide the 5V from the USB connection.
- **GND** - common *ground* connection for all power and data
- **5V** - This is the clean *regulated* 5V power. Provides up to about 500mA current draw.
- **3.3V** - This is a clean *regulated* 3.3V power. Provides up to about 100mA current draw.
- **Reset** - This is the same pin connected to the reset button

1.2. Arduino Digital Pin Header

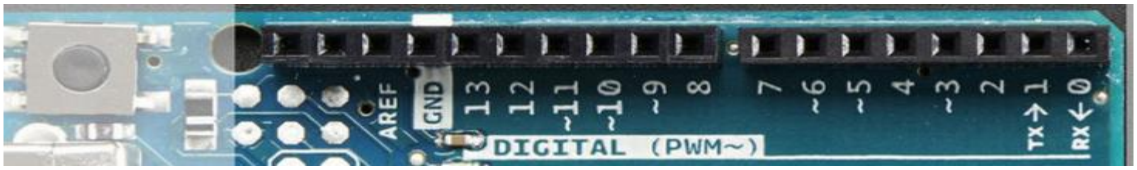


Figure 1.3. Digital Pin Header

- The pins labeled **0 (RX)** and **1 (TX)** are the two serial pins used for the USB communication.
- **Don't connect anything to Digital 0 or 1 because it will affect your Arduino's ability to communicate!**
- **Digital 2** through **Digital 12** are normal digital pins. Those that can output a PWM signal have a squiggly line next to them.
- **Digital 13** is a little special because it is also connected to the L LED , already installed on the board.. You can use this pin without affecting the Arduino just be aware that the **L** LED will also blink at the same time.
- A spare power **GND** Ground pin
- **AREF - A**nalog **Ref**erence pin.

1.3. Arduino Analog Pin Header

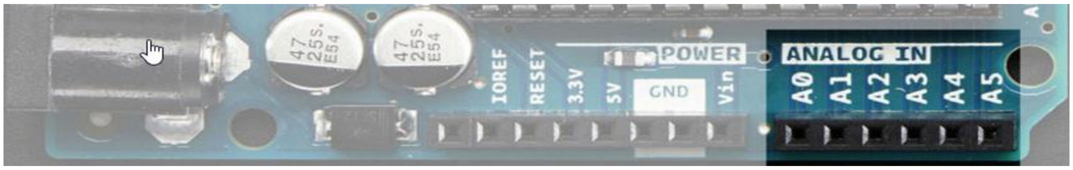


Figure 1.4. Analog Pin Header

- The Atmega controllers used for the Arduino contain an onboard 6 channel analog-to-digital (A/D) converter. The converter has 10 bit resolution, returning integers from 0 to 1023.
- The analog input pins are special pins that can read analog sensors.
- They can also be used as digital input/output pins (the same as digital pins 0 - 13).

1.4. Connecting the Arduino board

- Connect your Uno board with provided A B USB cable. This is also called a *USB printer cable*.
- The Uno automatically draws power from either the USB or an external power supply. Connect the board to your computer using the USB cable. The green power LED (labelled ON) should light up.
- Plug the USB cable into your computer.
- While powered, click the **Reset button** - you will see the **L** LED blink 3 times very rapidly.

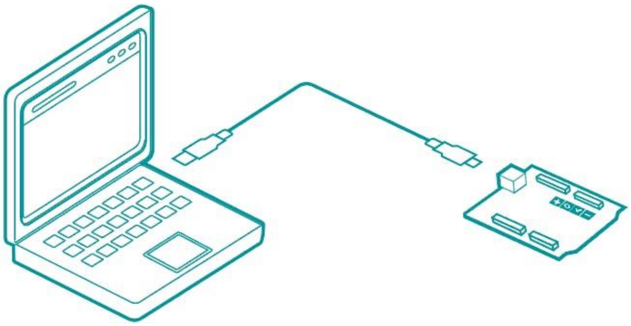


Figure 1.5. Connecting the Arduino board

2. Getting Started with Arduino

Main steps:

- Download the Arduino Software package https://www.arduino.cc/en/Main/Software
- Use the online IDE (web editor) to save your sketches (programs) in the cloud.
- The boards work out-of-the-box on the Web Editor, no need to install anything.
- Alternatively : Download the Arduino IDE

2.1. Download the Arduino Software (IDE)

- Choose the OS environment and download the IDE for Windows, MAC OS or Linux
- Follow the installation prompts and allow the driver installation process.
- Choose the components to install
- Choose the installation directory (we suggest to keep the default one)
- The process will extract and install all the required files to execute properly the Arduino Software (IDE)
- Go to the Guide web page and choose UNO: https://www.arduino.cc/en/Guide/ArduinoUno

Follow the instructions for Arduino WEB IDE OR Arduino Desktop IDE

- If you used the Installer, Windows - from XP up to 10 - will install drivers automatically as soon as you connect your board.

2.1.1. Select your board type

- You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino or Genuino Uno board.

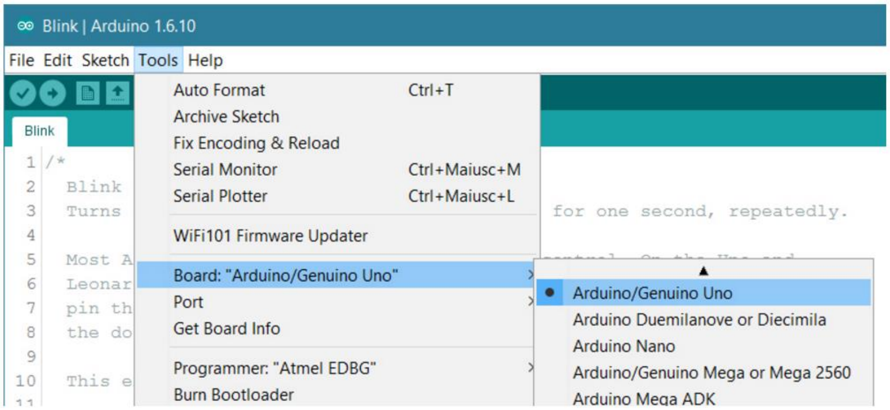


Figure 1.6. Select the Board type

2.1.2. Select the communication port

Select the serial device of the board from the ***Tools >Serial Port menu.*** This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports).

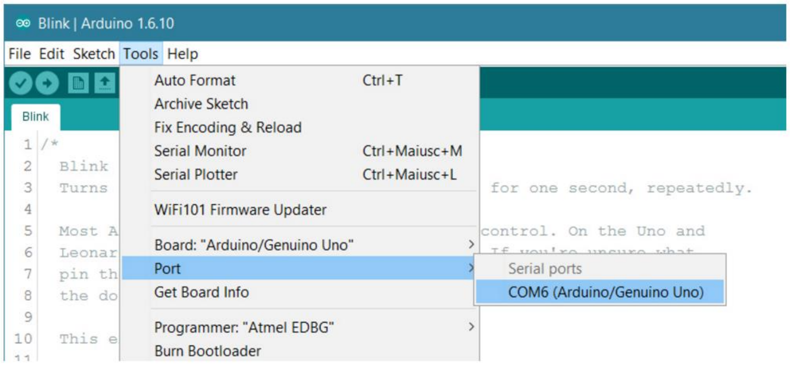


Figure 1.7. Select COM port

2.2. Open your first sketch from Examples menu:

Open the LED blink example sketch: ***File*** > ***Examples*** ***>01.Basics*** > ***Blink***

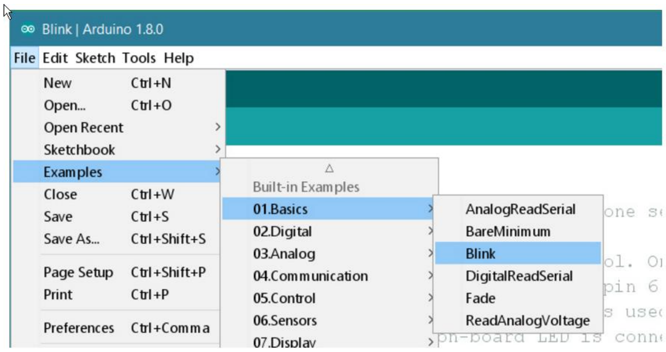


Figure 1.8. Opening Examples Menu and select "Blink"

2.2.1. Upload the program.

- Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX LEDs on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.

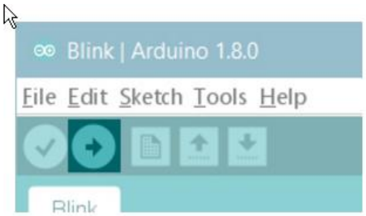
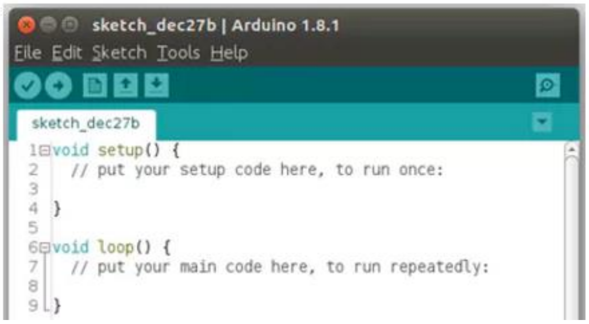


Figure 1.9. Upload button on the menu

- A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange). If it does, congratulations! You've gotten Arduino or Genuino up-and-running.

2.2.2. Run the IDE

- If this is the first time you run the Desktop IDE, you should see a tab (called sketch) filled with the two basic Arduino functions: the **setup()** and **loop()**.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

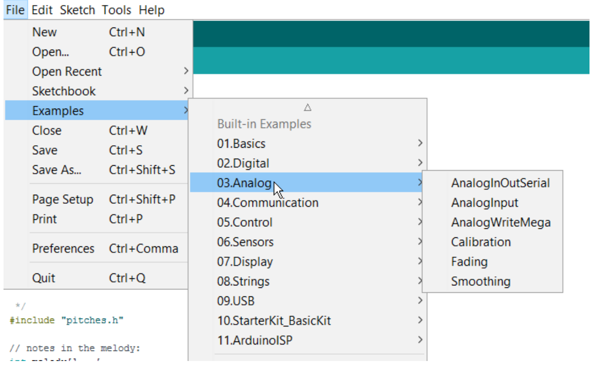
Figure 1.10. Basic sketch

2.2.3. Built-in Examples

There are number of built-in example sketches in Arduino IDE, which are divided in several sub-directories, as you see on the snapshot in Figure 1.11.

You should spend some time checking what programs are available in the Arduino IDE , which you can use in your Project work.

Since Arduino is an open source platform, a lot more information could be found in web forums, designated to users of this device.



```
*/
#include "pitches.h"

// notes in the melody:
```

Figure 1.11. Built in Examples

2.2.4. Arduino Library Manager

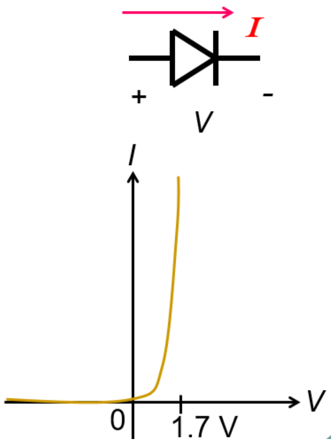
- An Arduino Library allows you to easily use modules, sensors and extra hardware without too much effort, since all the hard work to understand the code is already embedded in the library and only the functions are exposed to you. These functions are called methods or APIs.
- A library usually comes with a list of examples to practically show what you can do with this package.
- Step-by step instructions for uploading a sketch to the library and using it in your program can be found in available online tutorials.

3. Light Emitting Diode (LED)

An LED is a **p-n** junction diode that emits light when activated. When a suitable forward biasing voltage is applied to the p-n junction (typically larger than the 0.7V needed for a regular diode), electrons recombine with electron holes within the device, emitting energy in the form of photons. This effect is called electroluminescence.

Electrical characteristics diagram of the LED is shown in Figure 1.12.

- Current can only flow from anode to cathode when forward biased (i.e., V>1.7V)
- No current flows when reverse biased (V<0)



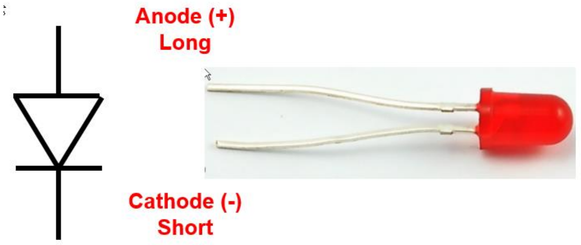


Figure 1.12. Electrical Characteristics

Figure 1.13. LED package

4. Digital Sensor – Push Button

The pushbutton in Figure 1.14 is a mechanical normally-open switch. When the pushbutton is open (not pressed) 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 DIGITAL LOW signal of 0V – Figure 1.15. 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 DIGITAL HIGH signal of +5V – Figure 1.16.

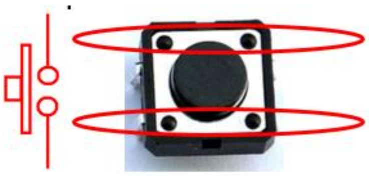


Figure 1.14

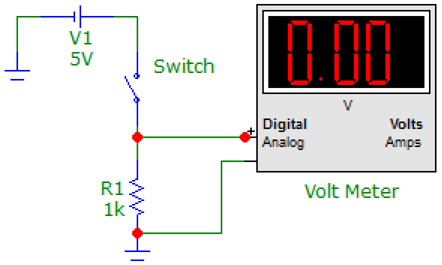


Figure 1.15

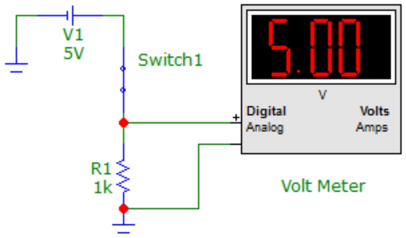


Figure 1.16

5. Serial Monitor on Arduino IDE

- Takes readings in real-time through the Arduino IDE's interface.
- Reads data directly from the Arduino as text and controls outputs.

The Serial Monitor connector is shown in Figure 1.17.

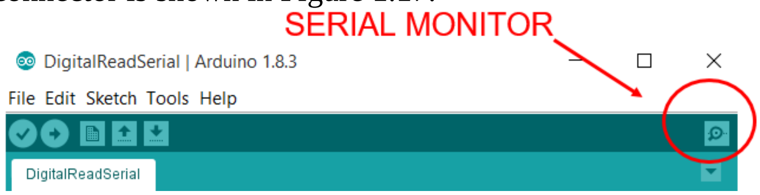

SERIAL MONITOR
DigitalReadSerial | Arduino 1.8.3
File Edit Sketch Tools Help
DigitalReadSerial

Figure 1.17. Turning on the Serial Monitor

TINKERCAD SIMULATION SOFTWARE

Create a free account on Tinkercad website. Then start the program at https://www.tinkercad.com/circuits and click on **Launch Tinkercad Circuits**.

Follow the explanations on Lab #1 PowerPoint presentation on now to build, code and simulate circuits on Tinkercad.
You can watch more tutorial videos on Tinkercad here:

LED : https://www.youtube.com/watch?v=3kDMYomFw5o
Push Button and LED: https://www.youtube.com/watch?v=PC15jBx2UxI

EQUIPMENT

1. Arduino Uno board
2. PROTO-BOARD (breadboard)
3. LED – 8
4. Push button – 1
5. Resistor R=220 Ω – 8
6. Resistor R=1 kΩ – 1

PRE-LABORATORY PREPARATION

Read the lecture material and search for other information sources to prepare for the lab.
The following lab needs a good understanding of the material and requires some modifications to Arduino example sketches. All these software modifications should be prepared as part of the pre-lab work,
You will need to take a video records of your experiment in order to demonstrate the working software when completing the Lab Procedure experiments.

1. Power up and Test the Arduino board

1.1. Follow the instructions in the lab manual and lectures to unpack, power-up and test the Arduino board BEFORE coming to the lab. Install Arduino IDE on YOUR laptop and bring it to the lab.
1.2. Make sure you know how to find and upload an example sketch.
1.3. Upload the "Blink" sketch and verify it works on your Arduino board.
1.4. Modify the "Blink" sketch to increase the speed to 2 times per second and store it under a different name.
1.5. Simulate "Fast Blink" experiment on Tinkercad and record a short video of working simulation on your computer screen. Upload the screenshot of the simulated circuit on Tinkercad and the simulation video to your OWL Dropbox.

2. LED experiments

2.1. Change the example "Blink" sketch to control pin D7 instead of pin D13 and save it under a different name, so you can use it later with the LED experiment.
2.2. Write a sketch to control the LEDs connected to the pins D7 and D13 simultaneously and save it under different name, so you can use it later with the LED experiment.

TIP: First define both pins 7 and 13 as "int" and OUTPUT.

2.3. Write a sketch to control the LEDs connected to the pins D7 and D13 in counter phase and save it under a different name, so you can use it later with the LED experiment.
2.4. Simulate the two LED experiment on Tinkercad and record a short video of working simulation on your computer screen. Upload the screenshot of the simulated circuit on Tinkercad, the generated Code and the simulation video to your OWL Dropbox.

3. Pushbutton as a Digital input sensor

3.1. Simulate on Tinkercad the circuit in Figure 1.15 using the available push button component, Power Supply and digital voltmeter.
3.2. Measure the voltage at the switch in both switch positions and print the Tinkercad simulations. Upload the simulations to your OWL Dropbox.
3.3. Enter the results in Table 1.1 of the LMS.
3.4. Upload the sketch ***File/ Examples/ Basics/ DigitalReadSerial*** and find out how it reads the pushbutton state.
3.5. Modify the sketch to turn the external LED ON when the button is pushed.
Tip: Use an ***if*** function in the sketch.
3.6. Simulate the experiment on Tinkercad and record a short video of working simulation on your computer screen. Upload the screenshot of the simulated circuit on Tinkercad, the generated Code and the simulation video to your OWL Dropbox.

4. Running Lights experiment

4.1. Upload the Running Lights sketch from the lecture notes to turn ON from LED1 to LED8 and repeat again from LED1.

4.2. Modify the program to make “Running Lights” from LED1 to LED8 and back from LED8 to LED1.

4.3. Simulate the experiment on Tinkercad and record a short video of working simulation on your computer screen. Upload the screenshot of the simulated circuit on Tinkercad, the generated Code and the simulation video to your OWL Dropbox.

PROCEDURE

1. Power up and Test the Arduino board

1.1. Power up the Arduino board and connect it to your laptop.

1.2. Upload and test the “Blink” sketch.

1.3. Modify the “Blink” sketch, to increase the speed to 2 times per second record a video of the working board and upload it to your OWL Dropbox.

2. Single LED experiment

2.1. Build the circuit in Figure 1.18. Upload the example “Blink” sketch to control D7 instead of D13. Make a video-record and post it in your OWL Dropbox.

2.2. Upload the modified “Blink” sketch to control the LED 13 and LED 7 simultaneously. Make a video-record and post it in your OWL Dropbox.

2.3. Upload the modified “Blink” sketch to control the LED 13 and LED 7 in counter phase. Make a video-record and post it in your OWL Dropbox.

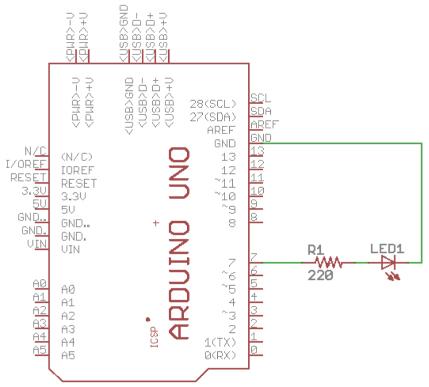


Figure 1.18.

3. Pushbutton as a Digital input sensor

3.1. Use continuity function of the DMM to determine how the push button switch is connected internally to its terminals and draw it in section 1 of LMS.

3.2. Build the circuit of Figure 1.15 and measure the voltage at the switch terminal using multimeter. Record the data in section 2 of the LMS.

3.3. Build the circuit in Figure 1.19. Upload the sketch ***File/ Examples/ Basics/ DigitalReadSerial*** and start the Serial Monitor.

Demonstrate reading the pushbutton state on Serial Monitor by video-recording the screen of your PC. Upload the video to your OWL Dropbox.

3.4. Upload the sketch to turn the external LED ON when the button is pushed.

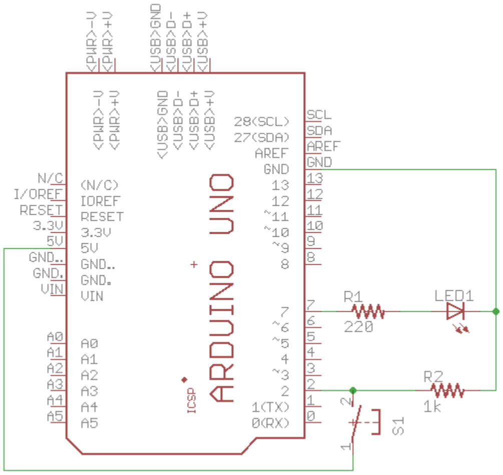


Figure 1.19.

Upload the video of the working circuit to your OWL Dropbox.

4. Running Lights experiment

4.1. Build the circuit in Figure 1.21 using 220Ω resistors and different color LEDs.
Upload the Running Lights sketch from the lecture notes to turn ON from LED1 to LED8 and repeat again from LED1 and video-record it. Upload the video of the working circuit to your OWL Dropbox.

4.2. Upload the modified sketch to make “Running Lights” from LED1 to LED8 and back from LED8 to LED1 and video-record it. Upload the video of the working circuit to your OWL Dropbox.

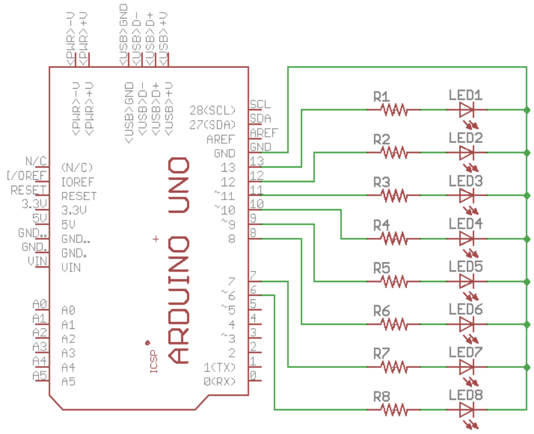


Figure 1.20.

REPORT

Your Lab report is due at the end of the current lab session. Please submit the LMS to your OWL Dropbox together with the Tinkercad simulations and videos. Please upload the video records of your working circuits to the OWL Dropbox.

LAB MEASUREMENTS SHEET – LAB #1

Name Arnav Goyal

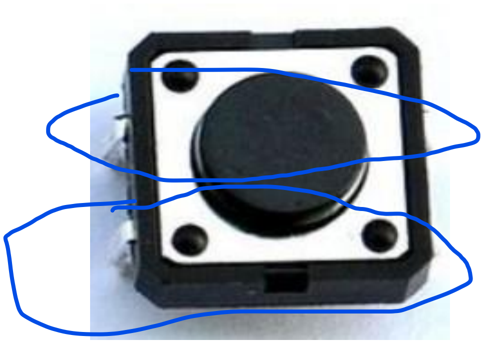
Student No 251244778

Workbench No #5

NOTE: Questions are related to observations, and must be answered as a part of the procedure of this experiment.

*Sections marked * are pre-lab preparation and must be completed BEFORE coming to the lab.*

1. Pushbutton switch pin connections



pin within each circle are connected, all pins are connected upon the pushing of button

2. Pushbutton Voltage measurements

Table 1.1.

Pushbutton Position	Tinkercad*	Measurements
Released	5.00 V	5.183 V
Depressed	5 mV	7.18 mV

MARKING SCHEME

TA name:_________________________

Task	Max. Marks	Granted Marks
Pre-lab: -***Power up the Arduino board***: Upload screenshot of the simulated circuit on Tinkercad and the simulation video to your OWL Dropbox. - ***LED experiments***: Upload the screenshot of the simulated circuit on Tinkercad, the generated Code and the simulation video to your OWL Dropbox. - ***Pushbutton***: 1. Print the Tinkercad simulations. Upload the simulations to your OWL Dropbox. 2. Simulate the experiment on Tinkercad and record a short video of working simulation on your computer screen. Upload the screenshot of the simulated circuit on Tinkercad, the generated Code and the simulation video to your OWL Dropbox. - ***Running Lights:***	5 5 5 5 5	

Simulate both experiments on Tinkercad and record a short video of working simulation on your computer screen. Upload the screenshot of the simulated circuit on Tinkercad, the generated Code and the simulation video to your OWL Dropbox.	5	
Power up the Arduino board and Single LED Build the circuit in Figure 1.18. Upload the example “Blink” sketch to control D7 instead of D13. Make a video-record and post it in your OWL Dropbox. Upload the modified “Blink” sketch to control the LED 13 and LED 7 simultaneously. Make a video-record and post it in your OWL Dropbox. Upload the modified “Blink” sketch to control the LED 13 and LED 7 in counter phase. Make a video-record and post it in your OWL Dropbox.	4 8 8	
Pushbutton experiment completed: Demonstrate reading the pushbutton state on Serial Monitor by video-recording the screen of your PC. Upload the video to your OWL Dropbox. Upload the sketch to turn the external LED ON when the button is pushed. Upload the video of the working circuit to your OWL Dropbox.	10 10	
“Running Lights” experiment completed: - Upload the Running Lights sketch from the lecture notes to turn ON from LED1 to LED8 and repeat again from LED1 and video-record it. Upload the video of the working circuit to your OWL Dropbox. - Upload the modified sketch to make “Running Lights” from LED1 to LED8 and back from LED8 to LED1 and video-record it. Upload the video of the working circuit to your OWL Dropbox.	15 15	
TOTAL MARKS	**100**	