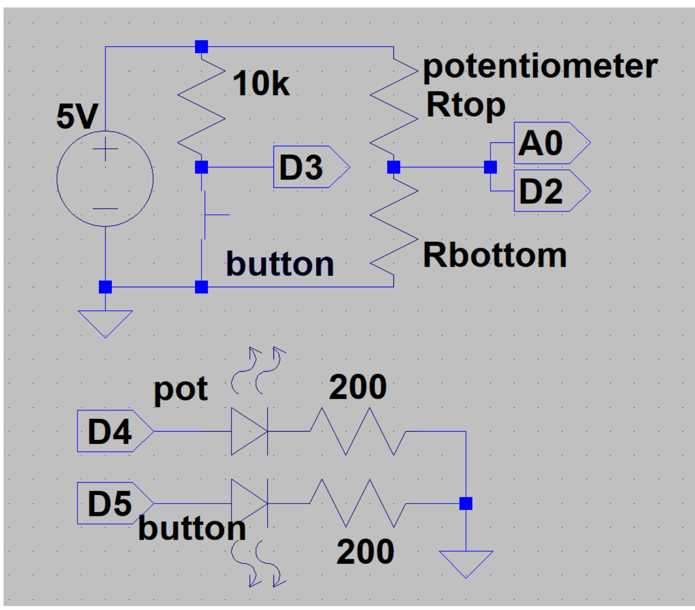
Practical led/button & TTL

Most computing devices that exist today are digital in variety. This means most information is used and encoded using an active or inactive state. Early computers had more primitive displays than today's high pixel density liquid crystal displays, some would have an interface that was as simple as simple switches and bulbs, and these are used to encode binary zeros and ones.

Today's simple equivalent is using simple buttons and leds to interact with the computer using digital logic. Transistor-Transistor-Logic refers to a particular type of conversion of a voltage to a zero state or a one state. In this exercise, we will examine at what particular voltage thresholds an arduino will interpret a logic 1 or 0. We will use its analog converter to display the analog value it is using for the conversion. In reality, the arduino never does any conversion from a voltage number in its memory to come up with a digital logic value, this instead occurs at the physics level in the transistors. We are simply using the analog input as a method to easily display the voltage alongside the resulting digital value in real-time.



Leds are simple outputs, and are current operated devices, or their brightness is proportional to the current passing through them while the voltage across them remains mostly steady. Most common 5mm leds have a maximum rated current of ~20 milliAmps but are visible at as low as 1mA, and the voltage they absorb will usually depend on their color. This means to light an led, we have to appropriately choose a resistor that will pass the appropriate current through the led such that its bright enough to see, but not so bright that it burns out. Since leds are diodes, they only conduct electricity in one direction.

To do this, remember ohms law, V=IR, which holds true for resistances. If the voltage the led absorbs is constant, the desired resistor value for a target current is $R=\frac{(Vsupply-Vled)}{Iled}$. To find the voltage an unknown led will drop, use the continuity mode of a digital multimeter, and when a test current is passed through the led, the meter will display the led's forward voltage. Sometimes the voltage is too high for the display to show, in which case, the led can simply be powered from a very high resistor value like 1 kiloOhm or 10kOhm and the voltage across the led can be measured with the multimeter as a volt meter. If using an arduino 5V digital output to power a yellow led with assumed 1.8Volt drop, and if I want specifically 5mA, then my resulting resistance is (5V – 1.8V) / (5*10^-3 Amps) = 640 Ohms. If instead I want 1mA or 20mA, then my resistance is 3200 Ohms and 160 Ohms respectively.

This means that the 100 Ohm resistors in our kit could result in currents of around 32mA, more than the 20mA maximum. To built the circuit above, we will need to connect 2 100 Ohm resistors together in “series” to get an equivalent 200Ohm resistance. To double check the resulting resistance, use a multimeter in Ohm meter mode. Construct the circuit above, and upload the ttlVisualizer program. Adjust the potentiometer knob and push the button while viewing the serial monitor and serial plotter to examine where the transition from 0 to 1 lies.

The program bringing this result is longer and slightly more complex, so it won’t be examined here. But code shows an example of different ways to declare variables, use of the digital inputs and outputs, and how to define and use functions that use arguments to return and output.