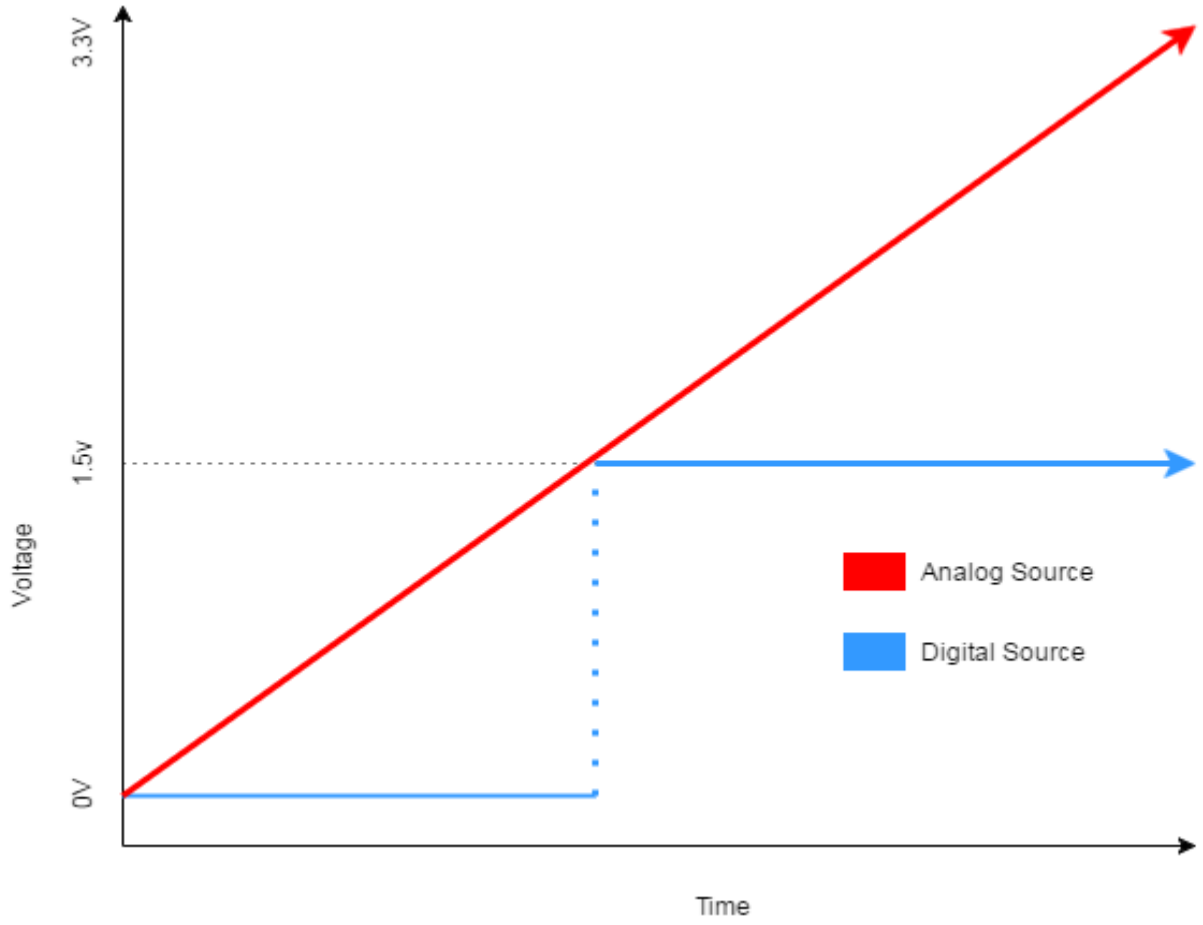
Digital Input Devices

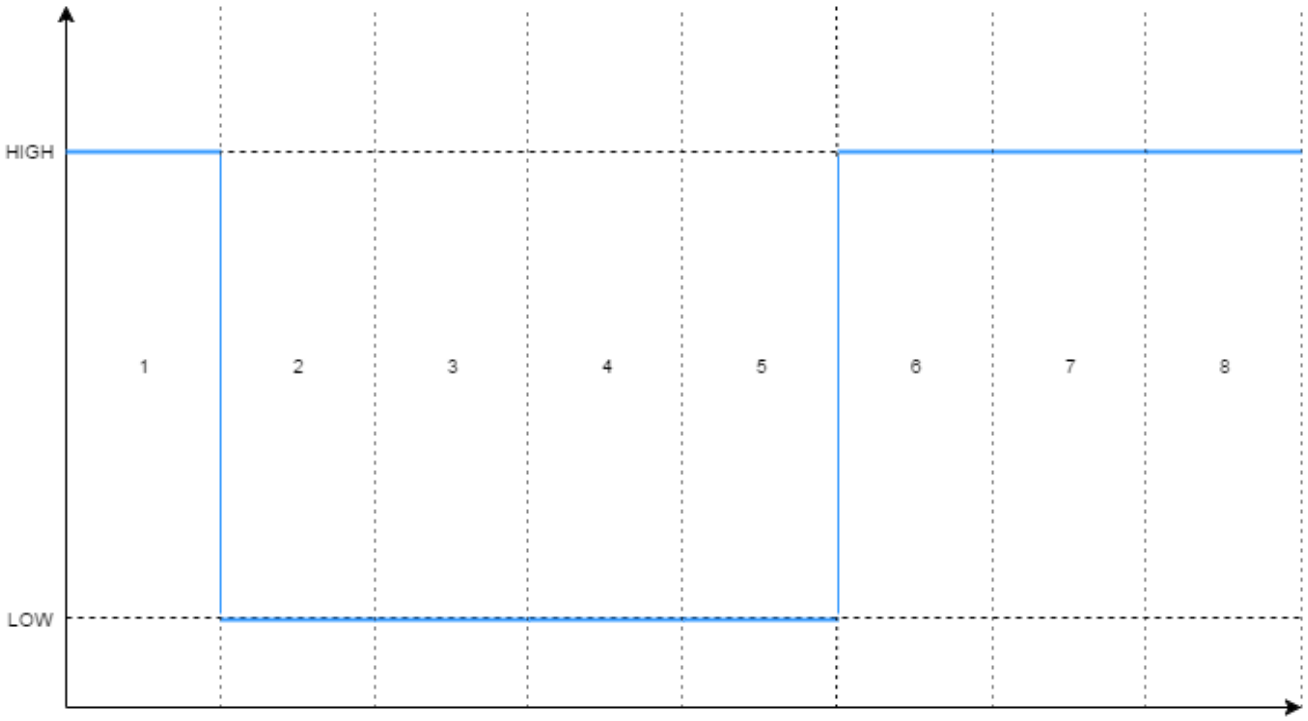
Introduction

In section two we introduced the concept of analog input devices. Here we will examine digital devices. While an analog pin has a voltage that can change continuously over the range of possible values. The number of possible states we can read from a single analog pin is bound only by the resolution of the ADC. A 10-bit ADC, like the one found on the Arduino, can specify a total of 1024 possible readings from the device. A digital pin is decidedly simpler. It has only two possible states: High and Low.



In the image above, we see a graph of the output of a pin over time. If we were to interpret the voltage of this pin as an analog value, we can see that it smoothly ramps up from 0v to 3.3v over time. If we were to read this as a digital value, we would see that the value is Low as long as the voltage remains lower than threshold voltage of 1.5V. Once we hit (or pass) the threshold voltage, the value is High.

Of course, for most digital input devices, we probably want to be able to express more than one possible state. We typically do this by some form of serial communication with the device, where we express the state of the device as a number that is sent bit by bit over the line. If, for instance, we wanted to express an 8-bit value of 135, we could express that with the following bits: 10000111. We could, equivalently express this pattern as: High, Low, Low, Low, Low, High, High, High. We could graph it as follows:



You should be able to imagine that we could write a larger number by just writing a number with more bits. We could express a 16-bit number, giving us a total of 65,536 possible states. All we would have to do is spend twice as long writing the extra bits.