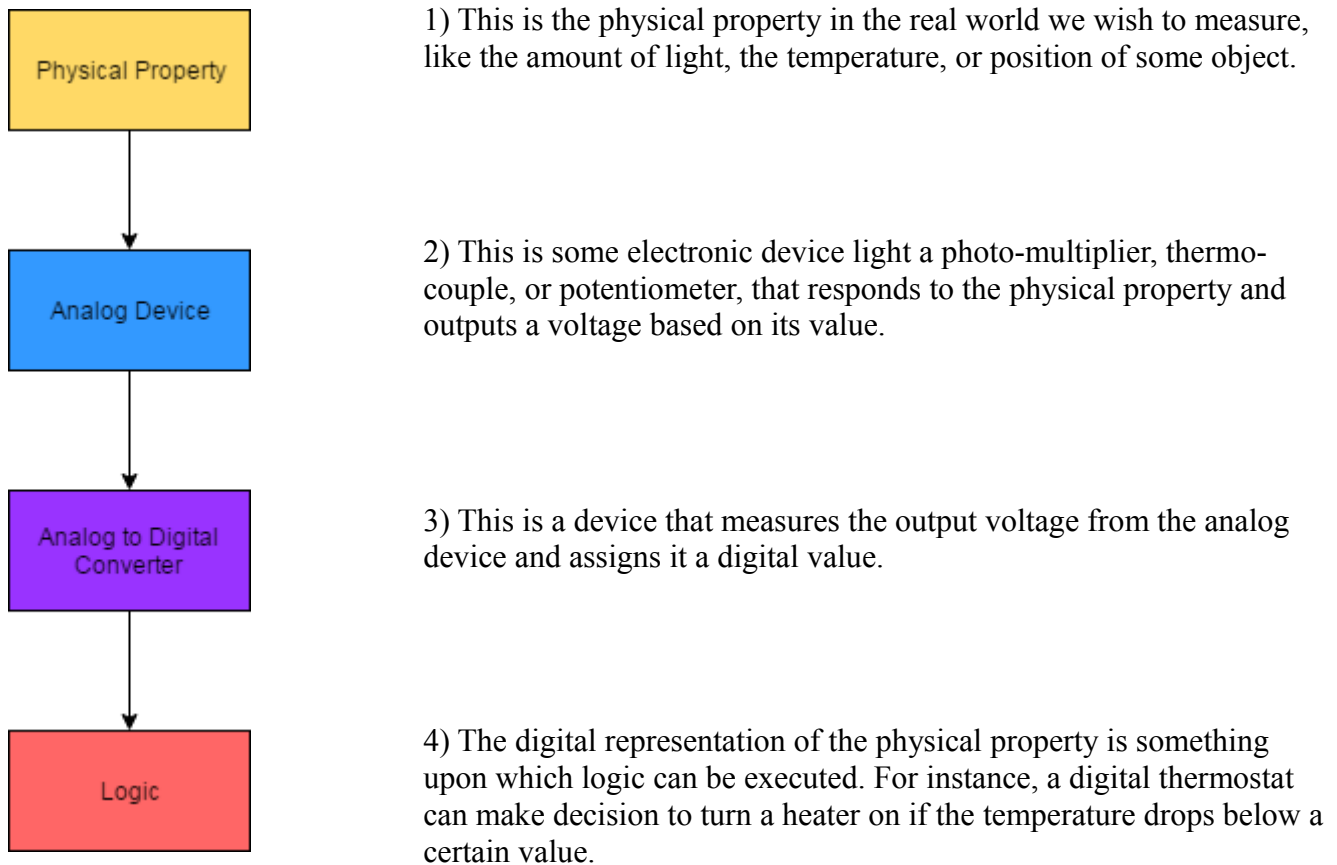


Analog Input Devices

Introduction

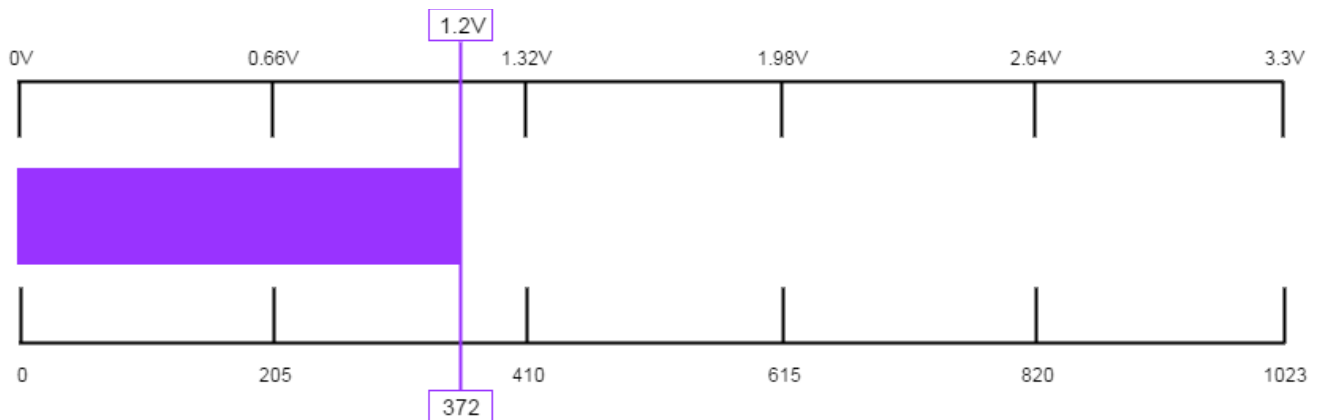
An analog sensor is the simplest possible electronic sensing device. It generates a voltage that serves as an *analogy* for the physical property which is being measured, hence the name. This class of devices includes things like light sensor, temperature sensors, and position sensors. For instance, with an analog temperature sensor, the device might generate an output voltage that varies from 0-3.3V as the temperature in the room varies from 0-100 °C. A microcontroller like the Arduino can measure this output voltage and perform calculations to determine the current temperature being measured. As you'll see illustrated below; the Analog layer provides an interface between the physical and digital worlds.



It's worth noting that even when we move on to digital sensors, this process is still accurate. A digital sensor integrates some number of analog devices and performs the conversion to digital for you with some kind of on-board processor.

Analog to Digital Conversion

Most microcontrollers will possess a number of analog pins. These connect to a device called an Analog to Digital Converter (ADC), which is often built right into the processor. The ADC outputs a digital number by comparing the input voltage against the reference voltage and mapping that ratio a range of possible digital values.



The ADC implements the following rule:

$$\frac{\text{input voltage}}{\text{reference voltage}} = \frac{\text{output}}{\text{resolution}}$$

Where the input voltage is the voltage being read from the analog pin; the reference voltage is the highest voltage the pin can read; the output is the digital number the device outputs; and the resolution is the highest number the device can output. We can re-organise this expression to give us the output.

$$\text{output} = \frac{\text{resolution} * \text{input voltage}}{\text{reference voltage}}$$

Arduino contains a 10-bit ADC, so it can output numbers that range from 0-1023. Any digital measurement it takes from any sensor will be expressed as some number in this range. The Arduino uses a reference voltage of 3.3V (by default). Input to the analog pins is compared to this voltage. If the input voltage was 1.20V:

$$\text{output} = \frac{1023 * 1.2}{3.3}$$

$$\text{output} = 372$$

ADU Conversion

The number we get from the ADC is expressed in units called Analog to Digital Units or ADU. In many cases you may want to express the reading from a sensor in some kind of standard units. For instance, you may want to express temperature in °C. This step requires information about how the change in the physical property being measured effects the output voltage of the device. Some times this relationship can be very simple, and the voltage goes up linearly as the physical property goes up linearly. Often this relationship is more complicated and voltage changes non-linearly as the physical property changes. The information about this relationship can often be found in the manufacturer documents for the device you are utilizing. Because of variances in manufacturing processes, this relationship often has to be determined empirically by testing the sensor.

Here is an example of a function that converts the output in ADU for the TMP-36 temperature sensor into a value in °C:

```
float calibrateSensor(int raw){  
    float y1 = 13; //cold deg C  
    float y2 = 55; //hot deg C  
    float x1 = 211; //cold adu  
    float x2 = 372; //hot adu  
  
    float m = (y2-y1)/(x2-x1);  
    float b = y1 - m* x1;  
    return raw*m+b;  
}
```

This approach uses linear interpolation with two known points (which I measured by submerging the sensor in hot and cold water at known temperatures) to calibrate the output of the sensor from ADU into °C for any other temperatures. Because this device is linear, I can use a simple linear interpolation as shown above. You can connect the TMP-36 similarly to how you connected the Light Sensor and run the included 'TempSensor' sketch to see this code in action.