How do systems sense their environment?

All engineering systems have sensors. Without sensors, a system would not have any knowledge about its environment, and therefore would not be very useful.

Take an air-conditioning unit as an example. It requires a temperature sensor to know the temperature in the room. If it is much warmer than the desired temperature (also known as a *set-point*), the air-conditioner has to blow cold air to cool the room. If the temperature is cooler than the set-point, the air-conditioner should stop blowing cold air, and simply monitor the temperature. In addition to a temperature sensor, most air-conditioners will have other sensors. For example, they may have buttons or touch sensors to allow the user to change the set-point. Since buttons allow the system to sense when the user is pressing them, they are sensors too.

In this chapter, we will learn about various kinds of sensors, how they work, and how they are typically connected to an engineering system.

Student Preparation

In preparation for the studio sessions, students are expected to read and understand this chapter. Students are also expected to think through the questions posed in the margins (marked by the emoticon shown on the right), as we will discuss them and build on them in the studio sessions.

How sensors work?

Sensors come in various shapes and forms, ranging from simple circuit elements to integrated circuits (ICs) to sophisticated modules that can be directly interfaced using standard ports (e.g. USB port, serial port).

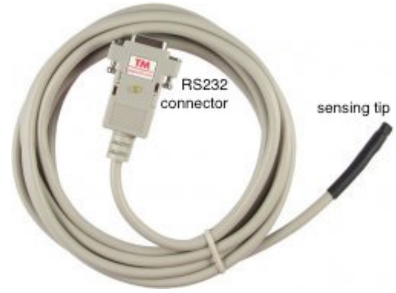


Figure 1: TM temperature sensor from

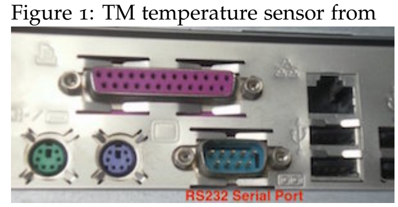


Figure 2: RS232 serial port on desktop PC.

A digital temperature sensor

Let us begin by exploring an off-the-shelf digital temperature sensor (TM) from papouch.com that can be connected to the RS232 serial port of a computer.

If you have a desktop computer or an older laptop, it'll probably have one or more RS232 serial ports (often called COM ports) that looks something like the one shown in Figure 2.

Newer laptops often only come with USB ports, but one can get a USB-to-RS232 dongle that will provide a RS232 serial port to connect to. In either case, once we have a RS232 serial port on a computer, we simply connect the sensor to that port and run a serial terminal software (e.g. termite) to obtain the data from the sensor:

```
+025.3C
+025.4C
+025.3C
```

This sensor automatically sends out a temperature measurement every 10 seconds, once it is connected to a RS232 serial port.

In practice, we may want to connect our sensor to an embedded computer (most embedded computers have serial ports), and read the temperature measurements from our own embedded software. While the details of the software would depend on the detailed of the embedded computer, it is not hard to write software that reads the sensor measurements. For example, if the embedded computer runs our software in Python on Linux, we may write something like this:

```
import serial
with serial.Serial('/dev/ttyS1', 9600) as port:  # open serial port @ 9600 bps
    s = port.readline()                          # read sensor output as string
    s = s[:-1]                                   # remove the trailing 'C'
    temperature = float(s)                       # convert string to float
```

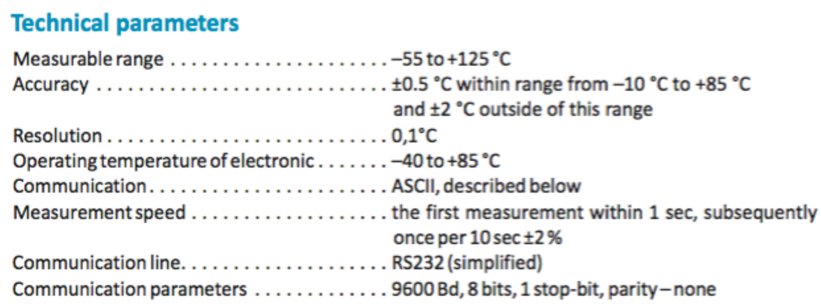
Okay, so that wasn't so hard!

Understanding sensor specifications

If we search for "temperature sensor" on the web, we will find many! So how do we select one? Are all of them the same, and we simply pick one? The answer is no! To choose one, we have to carefully look at the *technical specifications* of the sensor (typically in the *data sheet* for that sensor), and see which ones meet our application needs. Then other factors such as price, size, power consumption, etc may help narrow down the choice further.

Let us take a look at the technical specifications for the temperature sensor that we explored in the previous section (see Figure 3).

How do systems sense their environment?



Technical parameters

Measurable range	−55 to +125 °C
Accuracy	±0.5 °C within range from −10 °C to +85 °C and ±2 °C outside of this range
Resolution	0,1°C
Operating temperature of electronic	−40 to +85 °C
Communication	ASCII, described below
Measurement speed	the first measurement within 1 sec, subsequently once per 10 sec ±2 %
Communication line	RS232 (simplified)
Communication parameters	9600 Bd, 8 bits, 1 stop-bit, parity – none

Figure 3: Temperature sensor specifications from the data sheet.

- What is the difference between accuracy and resolution?
- How can it be that the operating temperature range of the electronics is narrower than the measurable range of the sensor? Wouldn't the electronics get damaged if we tried to measure a temperature of, say, 100°C?

The specifications tell us what is the temperature range that the sensor can measure, to what precision (resolution), to what accuracy, what interface the sensor reports its measurements over, and how often.

How does a digital sensor work?

We used a digital temperature sensor that gave us temperature data over a serial port. But how does it really measure temperature? And how does the measurement get converted to the ASCII text that is sent over the serial port?

Let us first assume that we have an *analog sensor* that produces a voltage that changes with the quantity we wish to measure. In the digital temperature sensor module above, we will perhaps find an analog temperature sensor that produces a voltage output that is dependent on the temperature:

$$V_o = f(T) \tag{1}$$

where V_o is the voltage output in Volts, T is the temperature in °C, and $f(\cdot)$ is some monotonic function.

- Why do we require that $f(\cdot)$ be monotonic? What would happen if it was not?

The output of this analog temperature sensor will be a voltage. If we have a way to measure this voltage, then we have a working sensor. Microcontrollers typically have analog-to-digital convertors (ADCs) that can measure voltage, and also have many standard ports. So our temperature sensor module would need a microcontroller with an ADC and a RS232 serial port:

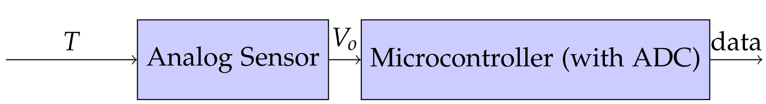


Figure 4: Functional breakdown of a typical digital sensor measuring T.

The microcontroller would run a simple program to convert the voltage into temperature:

$$T = g(V_o) \tag{2}$$

where $g \approx f^{-1}$. This value can then be sent as ASCII text over the RS232 serial port.

How does an analog sensor work?

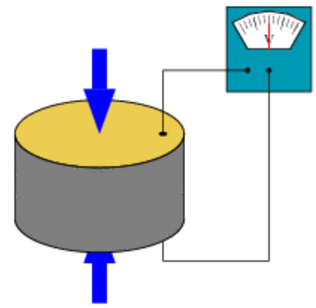


Figure 5: A piezoelectric material produces a voltage depending on the force applied on it.

In the previous section, we assumed that we had an analog sensor that produces a voltage depending on the parameter that we measure. In some cases, it is easy to find such a device or material. For example:

- *Photovoltaic materials* (used in solar cells) directly convert light into electricity at the atomic level. The atoms absorb photons of light and release electrons that can flow in an electric circuit.
- *Piezoelectric materials* naturally produces a voltage depending on the force applied on the material.

- Can you find a device whose output voltage changes in response to a magnetic field? What is it called?

However, quite often it is not possible to find such materials or devices directly. It is more common to find circuit elements whose electrical properties (e.g. resistance, capacitance, inductance) change with the quantity to be measured. We will call such circuit elements *sensing elements*.

For example, a *thermistor* is a type of resistor whose resistance R_s is strongly dependent[1] on temperature T:

$$\log R_s = \frac{\beta}{T} + \alpha \tag{3}$$

[1] Standard resistors exhibit a weak temperature dependence.

for some constants α, β. We can incorporate the thermistor (or other sensing elements) into an electrical circuit that generates a voltage that can be measured:

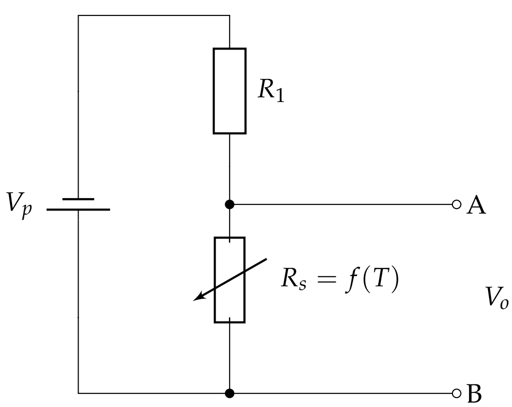


Figure 6: Simple electrical circuit to convert temperature T to voltage V_o that can be measured.

In this circuit, V_p is a voltage source of our choice (perhaps a battery), R_1 is a known resistance and R_s is our sensing element (thermistor). By analyzing the circuit, we easily see that:

$$V_o = \frac{R_s}{R_s + R_1} V_p. \tag{4}$$

How do systems sense their environment?

Since V_p and R_1 are known, we can calculate R_s once we measure V_o:

$$R_s = \frac{V_o}{V_p - V_o} R_1. \tag{5}$$

- Can you find a device whose resistance depends on the intensity of light incident on it? What is it called?
- In the circuit shown in Figure 6, how would you choose the values of V_p and R_1?

The temperature T is easily obtained once R_s is known:

$$T = \frac{\beta}{\log R_s - \alpha}. \tag{6}$$

If we find other sensing elements whose electrical properties depend on other quantities of interest, we can use the same technique to make other sensors.

Wheatstone bridge

Consider a sensing element R_s which changes by a small amount R_δ around a nominal value R_0, i.e., $R_s = R_0 + R_\delta(T)$, where T is the physical quantity being measured. If we use a potential divider circuit as shown in Figure 6, and assume R_δ to be small, we have:

$$\begin{aligned} V_o &= \frac{R_s}{R_1 + R_s} V_p \\ &= \frac{R_0 + R_\delta(T)}{R_1 + R_0 + R_\delta(T)} V_p \\ &\approx \frac{R_0 V_p}{R_0 + R_1} + \frac{R_1 V_p}{(R_0 + R_1)^2} R_\delta(T). \end{aligned} \tag{7}$$

Example:
If V_p = 10 V, $R_0 = R_1$ = 10 kΩ and $R_\delta = \pm 100\ \Omega$, then $V_o = 5 \pm 0.025$ V. To measure this with a 10 V Voltmeter, we require an accuracy of 0.25% or better.

The first term is usually large, and does not vary with the quantity T being measured, while the second term is small (zero-mean) and varies with T. The output voltage V_o therefore varies by a small amount around a large DC value. The small variation can be hard to measure accurately due to the large DC component.

Now consider the following Wheatstone bridge circuit instead: We

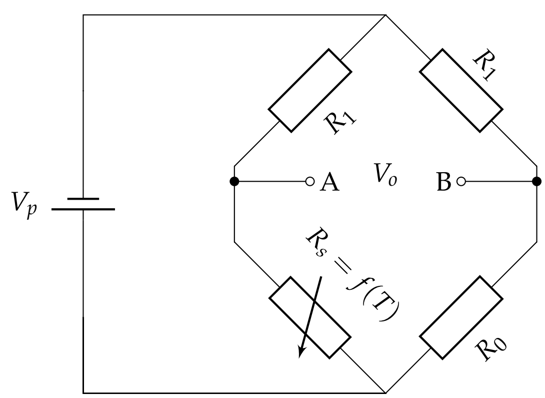


Figure 7: Wheatstone bridge circuit to convert temperature T to voltage V_o that can be measured.

now have:

$$V_o = V_B - V_A$$

$$
\begin{aligned}
&= \frac{R_0}{R_0+R_1}V_p - \frac{R_s}{R_1+R_s}V_p \\
&\approx \frac{R_0}{R_0+R_1}V_p - \left(\frac{R_0V_p}{R_0+R_1} + \frac{R_1V_p}{(R_0+R_1)^2}R_\delta(T)\right) \\
&= \frac{R_1V_p}{(R_0+R_1)^2}R_\delta(T). \qquad (8)
\end{aligned}
$$

The output voltage now has no DC component (is zero-mean) and can be easily measured more accurately. This illustrates one of the key benefits of using a Wheatstone bridge circuit for sensing, as compared to a simple potential divider.

Example:
If V_p = 10 V, $R_0 = R_1$ = 10 kΩ and $R_\delta = \pm 100\ \Omega$, then $V_o = \pm 0.025$ V. To measure this with a 100 mV Voltmeter, we require an accuracy of 25% or better only.

How do you measure the output voltage?

Once we know how to build an analog sensor, we can ignore the detailed circuit and simply think of it as a voltage source V_o with an output resistance R_o:

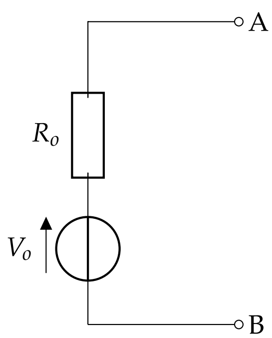


Figure 8: Equivalent circuit of an analog sensor.

- In modeling an analog sensor as a simple voltage source with an output resistance, are we making an approximation?

For the circuit in Figure 6, we can compute R_o using Thevenin's theorem:

$$R_o = \frac{R_1R_s}{R_1+R_s}. \qquad (9)$$

We can measure the output voltage between nodes A and B using a voltmeter or an oscilloscope, or in the case of the digital sensor, using an ADC. When we connect the leads of our measuring instrument to the nodes A and B, a tiny amount of current has to flow through the measuring instrument for it to sense the voltage. So the measuring instrument can be thought of as having an input resistance (aka *input impedance*) R_i. The instrument measures voltage V_{AB} between nodes A and B:

$$V_{AB} = \frac{R_i}{R_i+R_o}V_o. \qquad (10)$$

Clearly this is not the same as V_o, unless $R_o = 0$ or $R_i = \infty$. In order for our measurement to be as close to V_o as possible, we require that $R_o \ll R_i$.

- Should we choose R_1 to be small or large for $R_o \ll R_i$?
- What is the effect of this choice of R_1 on the power consumed by the sensor?

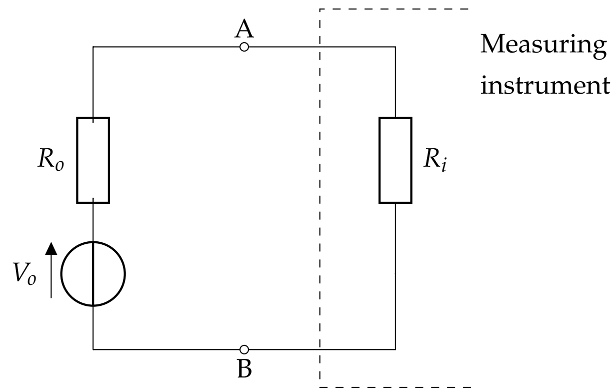


Figure 9: Equivalent circuit of an analog sensor and measuring instrument.

How do active sensors work?

So far, the sensors we have studied are *passive sensors*. They sense their environment without transmitting any energy. In some cases, sensors actively transmit energy in order to sense their environment. For example, a flash on a camera actively emits light. The light reflects off the subject and is received by the camera's CCD sensor to generate an image. The camera + flash system is therefore considered to be an *active sensor*. Typically, this means that active sensors require an energy source to operate, but passive sensors may operate with or without one.

Many active sensors are commonly found in engineering systems:

- Ultrasound ranging sensors, commonly used in reverse sensors in cars, emit a high frequency sound that echoes off obstacles. A measurement of the time taken for the sound to reach the obstacle and return back provides an estimate of the range to the obstacle.

- Drones commonly use ultrasound rangefinders to measure their altitude above the ground. An underwater version of ultrasound rangefinders are known as depth sounders, and are very commonly found on boats and ships.

- Laser rangefinders work on a similar principle, but use laser light instead of ultrasound to find range. Handheld laser rangefinders are commonly used in the construction and renovation industry, and by military.

- Active infrared proximity sensors are commonly used as touchless switches for taps and towel dispensers in public washrooms.

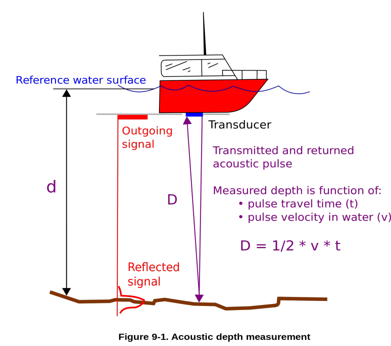


Figure 10: Principle of operation of a depth sounder.

Supplementary Reading

- How do photovoltaics work?
- Piezoelectric sensors
- Hall effect sensor
- Photoresistor
- Wheatstone bridge