Function of sensors: convert physical quantity to information. Information will (usually) be read by computer via an interface. Ultimately, the information, in the desired form, will be stored in a memory location.

The following steps are typical:
1: A transducer converts process state to a raw electrical quantity.
2: A conditioning circuit converts the raw electrical quantity into a useful electrical quantity.
3: An analog-to-digital converter (ADC) converts the useful electrical quantity to information.
4: A buffer and interface store, format, and present the information to a computer.
5: An interface routine reads the information, converts it to the desired form, and stores it in the desired place.

The process state is the current condition of the process, down to infinitesimal detail. The process variable is a part or characterization of a process state, usually in terms of a common measure.

For example, consider a coffee maker.
- Process state: amount of water in carafe, water temperature, chemical description of water in carafe, type of coffee beans, etc.
- Process variable: temperature of water.
- Process variable value: 70°C.

Characteristics:
- It is impossible to know the complete process state (because of infinite detail).
- It is impossible to know the exact value of a process variable.
- A process variable value, however, can be determined to a high degree of precision.

A process variable’s value is usually expressed as the product of a number and a dimension. For example, let process variable \( T \) be the temperature of water in a coffee maker carafe.

Then a value for \( T \) might be 60°C.
An equivalent value might be \( T = 333.15 \text{ K} \).

Notation:
Dimensions will be written in Roman (upright) type. For example, mA, V, and m.
Symbols representing values (variables) will be written in italic type: \( T, x, \) and \( R \).
Thus, \( 3V \) means “three vee volts.”
Transducers

Transducer:
Device which converts a physical quantity from one form to another.

Usually from ... 
... a physical quantity which is a process variable to ... 
... some useful electrical quantity.
For example, a transducer might convert temperature to resistance.

Transducer Modeling
Mapping (function) from process variable to electrical quantity.
Symbol \( H_t \) denotes the function.
Let \( x \) be a process variable.
Then \( H_t(x) \) is the output of the transducer with function \( H_t \).

Example
A variable resistor can be used as a transducer.
Consider a variable resistor which consists of a slider that can move 15 mm while resistance varies linearly from 0 to 10 k\( \Omega \).
Process variable: position of slider, \( x \).
Mapping:
\[ H_t(x) = x \frac{10 \text{ k}\Omega}{15 \text{ mm}}. \]
Process variable value determined from transducer output.
Let \( y = H_t(x) \) where \( H_t \) and \( x \) are as above.
Quantity \( y \) is a resistance.
The position \( x \) is found by inverting \( H_t \):
\[ H_t^{-1}(y) = y \frac{15 \text{ mm}}{10 \text{ k}\Omega}. \]
The process of finding the inverse is equivalent to solving for \( x \) in the equation \( y = x \frac{10 \text{ k}\Omega}{15 \text{ mm}} \).

Conditioning Circuits

Purpose
The output of a transducer is a raw electrical quantity.
It might have to be amplified or otherwise processed.
This is done by conditioning circuits.
Conditioning circuits might have to do one or more of the following:
- Amplify a tiny voltage.
- Convert resistance to voltage.
- Detect tiny changes in resistance (e.g., 100.1 to 100.2\( \Omega \)).
- Add an offset to the transducer output.
- Correct for nonlinearities in the transducer function.
- Other functions.

Notation
The symbol \( H_t \) will be used for the conditioning circuit’s function.
An amplifier is a simple conditioning circuit: \( H_t(v) = Av \), where \( A \), the gain, is a dimensionless number.

For example, if \( x \) is a process variable, then \( H_t(x) \) is the transducer output and \( H_t(H_t(x)) \) is the conditioning circuit output.

Sensors
The combination of transducer and conditioning circuit is referred to as a sensor.

Analog to Digital Conversion

ADC Use
Conditioning-circuit output is usually fed to an ADC.
An analog-to-digital converter converts electrical quantities to information.
Input is usually a voltage, output is usually an integer.
Symbol \( H_{ADC}(v) \) will be used for an ADC function.

Standard ADC Function
Since most ADCs will convert voltage to integers a standard function will be used:
\[ H_{ADC}(h;b) = j v^b (2^b - 1), \]
where \( h \) is a voltage and \( b \) is an integer.
This ADC would convert voltages in the range 0 to \( h \) (inclusive) to a binary number from 0 to \( 2^b - 1 \).

For example, \( H_{ADC}(5 \text{ V};8) = 127 \) and \( H_{ADC}(1.3 \text{ V};8) = 5011 \).
Sampling, Buffering, and Interfacing

Buffering is the short-term storing information.

Two reasons for buffering the value of a process variable:

1. Value at a particular time is needed. (Value is buffered at that time.)
2. Value is only valid at certain times. (Value is buffered when valid.)

The buffer itself can be a simple flip-flop, a register, a RAM, etc.

Usually, buffer contents read by a computer through an interface. The interface presents the buffered data to the computer in some standard form.

The computer is running some RT (real time) program. The RT program has one or more interface routines.

An interface routine could read the buffer without being alerted by an external signal. For example, it might read the buffer every millisecond.

Sampling is the process of reading a process variable at regular intervals.

Interface Routine

The following is done to transfer a value from the interface to the memory location:

1: The interface routine makes a call to read the interface.
   `raw = readInterface();`
2: The interface routine, or some other code, applies a function, \( H \), to the value read.
3: The result is written into the memory location.
   `theMemoryLocation = HsubF(raw);`

The function \( H \) puts the value into the final form. It may perform one or more of the following operations:

- Convert the raw value to a floating-point quantity. (ADC output is usually an integer.)
- Correct for any nonlinearities in the transducer or conversion circuit.
- Convert the quantity to the desired dimensions. (E.g., meters, microns.)

In terms of the process variable, the final value written is:

\[
H_i(H_{ADC}(H_i(x)))
\]

Value is read by other parts of the RT program.

The Conditioning Problem

Archetypical Problem:

Design a system to write variable \( \text{procVar} \) with \( H(x) \), the value of \( \ldots \) in \( \ldots \), where process variable \( x \) is \( \ldots \), \( x \) can take values in the range \([x_{\min}; x_{\max}]\) and where \( H(x) = \ldots \).

Example Problem:

Design a system to write variable \( \text{waterLevel} \) with \( H(x) \), an integer giving the water level, where process variable \( x \) is the water level in room 2161 CEBA, \( x \) can take on values in the range \([0 \text{ m}; 1 \text{ m}]\) and where \( H(x) = x/5 \).

Solution Overview

1: Choose a transducer.
   A variable resistor connected to a float with cables.
2: Choose an ADC.
   Suppose an ADC with function \( H_{ADC}(5 \text{ V}; 8) \) is available.
3: Design a conversion circuit.
   This will convert resistance to voltage.
4: Design the buffer and interface.
   Details for this part will be skipped here.
5: Write the function for computing the final value.
   Easy, but tedious.

Example Problem Selection of Transducer

Use potentiometer: 1

Construction: shaft that can rotate 6 radians.

Three leads.

Resistance between center and lower lead, \( \theta \frac{100 \text{ k}\Omega}{6} \), where \( \theta \in [0, 6] \) is the shaft angle.

Resistance between center and upper lead, \( 100 \text{ k}\Omega - \frac{\theta}{6} 100 \text{ k}\Omega \).

Transfer function for center and lower lead, \( H_{ct}(\theta) = \frac{\theta}{6} 100 \text{ k}\Omega \).

Use of Potentiometer

Floats, guides, and cables will convert water level to shaft rotation.

These constructed so that \( \theta = x \frac{6}{\text{m}} \)

\[
H_i(x) = x \frac{100 \text{ k}\Omega}{\text{m}}
\]
The Conversion Circuit

The output of the conversion circuit will be:

\[ H_c(H_t(x)) = H_c \left( \frac{100 \, \text{k}\Omega}{x} \right) \]

The type of ADC chosen requires its input in the range of 0 to 5 V. A variety of conversion circuits could be used.

The simplest is a linear conversion from resistance to voltage.

\[ H_c(R) = \frac{5 \, \text{V}}{100 \, \text{k}\Omega} \]

\[ H_c(H_t(x)) = \frac{5 \, \text{V}}{x \cdot 100 \, \text{k}\Omega} \]

ADC, Buffering, and Final Processing

The ADC function is fixed at

\[ H_{ADC(5 \, \text{V} \cdot 8)}(v) = \left[ \frac{v(2^8 - 1)}{5 \, \text{V}} \right] = \left[ \frac{255}{5 \, \text{V}} \right] \]

\[ H_{ADC(5 \, \text{V} \cdot 8)}(H_c(H_t(x))) = \left[ \frac{5 \, \text{V}}{255 \, \text{m} \cdot 5 \, \text{V}} \right] = \left[ \frac{255}{5 \, \text{m}} \right] \]

The ADC output is clocked into a buffer and then transferred to the computer through an interface.
Details of these parts will be covered later in the semester.

Finally, the interface routine converts the raw form into the desired form: \( H(x) = \frac{5}{x} \text{m} \).

\[ H_t(H_{ADC(5 \, \text{V} \cdot 8)}(H_c(H_t(x)))) = H(x) \]

\[ H_t\left(\left[\frac{5 \, \text{V}}{255 \, \text{m} \cdot 5 \, \text{V}}\right]\right) = H(x) \]

Define \( y = g(x) = \left[\frac{x(255/\text{m})}{\text{m}}\right] \).

Then \( x = g^{-1}(y) = y \frac{\text{m}}{255} \) for \( x \in [0 \, \text{m}, 1 \, \text{m}] \).

Then

\[ H_t(g(x)) = H(x) \]

\[ H_t(y) = H(g^{-1}(y)) = H\left(\frac{y}{\text{m}}\right) = \frac{5 \, \text{V}}{y \frac{\text{m}}{255}} = \frac{5 \, \text{V}}{y} \]

The code fragment in the RT program is then:

```c
int raw;
double waterLevel;
raw = readInterface();
waterLevel = raw/51.0;
... and we're done!
```