Monitoring Systems

A monitoring system can be used to create a record of the condition of a system over a period of time. A monitoring system is used more often to detect when a particular physical property of a system goes outside a desired range; for example, if the CPU is too hot.

Let's consider temperature as an example. If this was being monitored under human control, the measurement could be made with a standard mercury thermometer. However, in this chapter we are interested in systems where a computer or microprocessor is being used. These systems require a measuring device that records a value which can be transmitted to the computer. Such a measuring device is a called a sensor. An example of a sensor for measuring temperature is a thermocouple, which outputs an electrical voltage that changes with temperature.

It is important to understand that in a monitoring system, a sensor does not have any built-in intelligence, so it cannot take any action if there is a problem. If the temperature measured becomes dangerously high it is the computer that sounds an alarm.

There are a wide variety of sensors available. For some the name indicates the property being measured such as pressure, humidity, carbon monoxide, pH or sound. For others such as an infrared sensor there are different methods of use. A passive infrared sensor just measures the level of infrared light received. In other cases, there is transmission of infrared light with the sensor possibly measuring the level of the light that is reflected back. Other sensors are given a generic name such as a motion sensor, for which different examples will be measuring different physical properties.

Control Systems

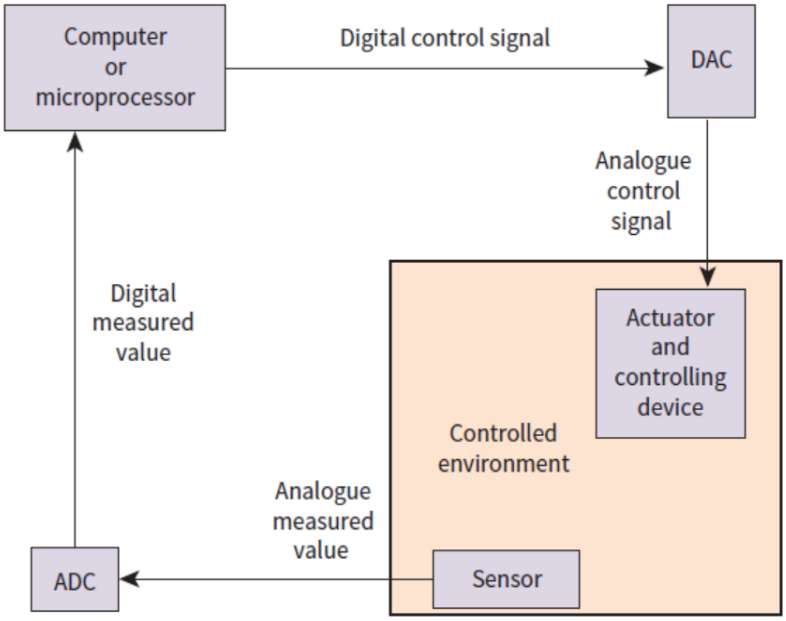
A control system has the monitoring activity plus the capability to control a system. The control element of a monitoring and control system needs a device called an actuator. An actuator is an electric motor that is connected to a controlling device. It might be used for switching on or off or for adjusting a setting.

Sensor: A hardware device that measures a property and transmits a value to a controlling computer.
Actuator: A hardware device that receives a signal from a computer and adjusts the setting of a controlling device

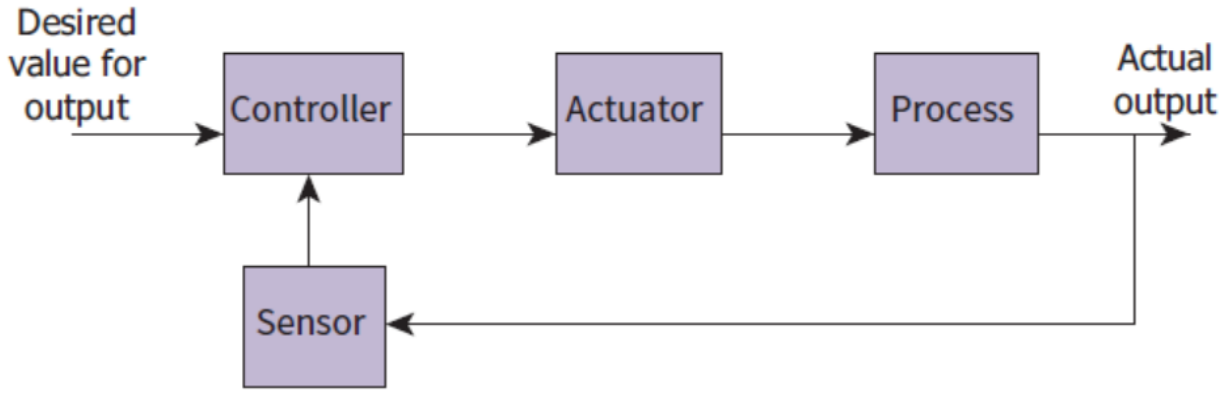
Note that below figure includes an analogue-to-digital converter (ADC) and a digital-to-analogue converter (DAC) as separate components. In a real system they are likely to be integral to the sensor or actuator device.

For the system below figure there is a continuing process where the computer at regularly timed intervals signals the sensor to provide a measurement. If the measurement value received by the

computer is not in the desired range the computer initiates a control action. The next timed measurement will happen after this control action has taken place. In effect this next measurement provides feedback to the computer on the effect of the control action. Feedback is essential in a control system.



A closed-loop feedback control system is a special type of monitoring and control system where the feedback directly controls the operation. Below figure shows a schematic diagram of such a system. A microprocessor functions as the controller. This compares the value for the actual output, as read by the sensor, with the desired output. It then transmits a value to the actuator which depends on the difference calculated.



Bit Manipulation to Control Devices

The controlling computer or microprocessor has to have a real-time program running continuously. The program can set values for Boolean variables subject to what the sensors detect. For instance, if a controlled environment had two properties to be monitored and controlled, four Boolean variables could be used. Values could be set by assignment statements such as:

IF SensorDifference1 > 0 THEN Sensor1HighFlag ← TRUE
IF SensorDifference1 < 0 THEN Sensor1LowFlag ← TRUE
IF SensorDifference2 > 0 THEN Sensor2HighFlag ← TRUE
IF SensorDifference2 < 0 THEN Sensor2LowFlag ← TRUE

Another part of the monitoring and control program would then be checking whether any of the four flags were set. The machine code for running such a program could use individual bits to represent each flag. The way that flags could be set and read are illustrated by the following assembly language code fragments. In these code fragments the three least significant bits (positions 0, 1 and 2) of the byte are used as flags.

LDD 0034	Loads a byte into the accumulator from an address.
AND #B00000000	Uses a bitwise AND operation of the contents of the accumulator with the operand to convert each bit to 0.
STO 0034	Stores the altered byte in the original address.
The following illustrates the toggling of the value for one bit. This changes the value of the flag it represents. It might be needed because a problem has been encountered or alternatively because a problem has been solved.	
LDD 0034	Loads a byte into the accumulator from an address.
XOR #B00000001	Uses a bitwise XOR operation of the contents of the accumulator with the operand to toggle the value of the bit stored in position 0.
STO 0034	Stores the altered byte in the original address.
The following illustrates the setting of a bit to have value 1 irrespective of its existing value. This would be a simple way of just reporting a condition repetitively.	
LDD 0034	Loads a byte into the accumulator from an address.
OR #B00000100	Uses a bitwise OR operation of the contents of the accumulator with the operand to set the flag represented by the bit in position 2. All other bit positions remain unchanged.
STO 0034	Stores the altered byte in the original address.

The following illustrates setting all bits to zero except one bit which is of interest. Following this operation a comparison can be made with a binary value to check if the bit is set. In this example the value would be compared to the binary equivalent of denary 2.	
LDD 0034	Loads a byte into the accumulator from an address.
AND #B00000010	Uses a bitwise AND operation of the contents of the accumulator with the operand to leave the value in position 1 unchanged but to convert every other bit to 0.
STO 0034	Stores the altered byte in the original address.