This application note discusses what microcontroller supervisory devices are, why they are needed and some factors to consider when choosing one. Supervisory devices is a broad term that encompasses POR (power on reset) devices, BOD (brown-out detect) devices and watchdog timer devices. This application note will cover supervisor devices with POR and BOD functions only.

WHAT DOES A SUPERVISORY CIRCUIT DO?

A supervisory circuit can be used for several different applications, but there are two primary functions that a supervisor provides:

1. During a power up sequence, the device holds a microcontroller in reset until the system power has come up to the correct level and stabilized (the POR function), and
2. reset the controller immediately if the power drops below a nominal value either at power down or during a 'brown-out' condition.

Some supervisor devices also provide things like low battery warning, watchdog timer and other more elaborate functions that are beyond the scope of this application note.

WHY DO I NEED A SUPERVISORY CIRCUIT ANYWAY?

One question system designers may ask themselves is, “Why do I need one of these things anyway?” There are 3 situations that you must consider when answering this question:

3. What would happen to the microcontroller (or other devices in the system) if there was noise on the supply voltage as it powers up?
4. What would happen if there is a glitch on the power supply while the system is running?
5. What does the microcontroller do when the system power is turned off?

If you ponder these questions and have visions of phone calls from angry customers, then you might consider using a supervisor device.

FIGURE 1: POR FUNCTION
Brown-Out: A Dirty Little Problem

Brown-out (Figure 2) is a condition where the supply voltage dips or 'sags' down to a safe operating level before returning to a nominal level. This condition can be caused by many different things such as inadequate power regulation, system components turning on or off, system malfunctions, etc. Unfortunately, brown-out conditions often don’t show up in the system development stage, but wait until the production run begins with all the system components installed to show their ugly heads. It is often at this point that perplexing problems are discovered, and eventually tracked down to some kind of brown-out condition. These problems can manifest themselves in many different ways including logic levels being misinterpreted or high current situations by creating invalid CMOS input levels. It is also possible to cause a more insidious problem of corrupting RAM locations inside the microcontroller. This problem can lead to irrational behavior on the part of the microcontroller that does different things at different times and may not show itself at all when an emulator is used to track down the problem.

FIGURE 2: BROWN-OUT CONDITION

Problems at Power-Down

Most microcontrollers today do not have any on-board POR/BOD protection. Some of them do, but they may not offer adequate protection against some system failures. One system problem that is seen quite frequently is the “Microcontroller running amok” problem that occurs when the supply voltage is ramped down very slowly, such as when a bench power supply is turned down manually or during the decay of a battery supply. When this situation occurs, it is possible for many microcontrollers to begin running through its code in a somewhat random manner. There may not be enough voltage to sustain RAM locations, so the program counter as well as any other variable stored in RAM may not contain valid data. This provides the means for the micro to execute any or all portions of the code stored in program memory with indeterminate values in all RAM locations.

Obviously, the longer it takes for the supply to ramp down the greater the danger of this situation occurring and causing problems. See Figure 3. For some systems, this situation may not cause any problems more serious than some spurious data sent to a display as the system is powered down. However, if the system contains other components that work to a lower voltage such as EEPROM devices, the problem becomes potentially more serious. EEPROM devices are available on the market that work down to 1.8V and may respond to commands as low as 1.2V. If the microcontroller executes a portion of its code that controls writing to the EEPROM, then there is the distinct possibility that random data will be written to the EEPROM device, which may or may not be discovered when the system is powered up the next time. This problem very often does not show up in the system development phase because the system is not being powered up and down on a regular basis, or it is powered from a supply different from the one used in production. It often shows up when the system goes into production and the system is being tested at different stages of the production line with different power supplies. A typical scenario: Data is written into the EEPROM and the system is tested as good and then powered down. At the next station it is discovered that the EEPROM data has been corrupted. This often results in a call to the EEPROM vendor with complaints of data retention problems, when the actual problem was the microcontroller sending write commands to the EEPROM during power down.
FIGURE 3: MICROCONTROLLER LOSES CONTROL WITH SLOWLY DECAYING SUPPLY

SO HOW DO I CHOOSE THE RIGHT DEVICE?

For the standard POR/BOD type of supervisor device, there are really only a couple of factors that you need to consider when making your choice. The major factors to consider are: reset voltage, output driver type, and reset polarity. Most supervisor devices come in a variety of reset voltages to support both 5V and 3V systems. Table 1, below shows typical reset voltage ranges. Choosing the correct trip point depends mainly on the operating range of the controller you are using and the variation of your supply voltage. You want to choose the highest trip point you can that will not interfere with the normal variations of your supply voltage. For a typical microcontroller, it might operate at 5V ±10% or 4.5V - 5.5V. Choosing a device with a trip point range of 4.5V - 4.75V will ensure that the controller is reset before the low end of the operating range is reached.

TABLE 1: TYPICAL TRIP POINT VALUES

<table>
<thead>
<tr>
<th>Minimum Trip Point (V)</th>
<th>Typical Trip Point (V)</th>
<th>Maximum Trip Point (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.55</td>
<td>2.625</td>
<td>2.7</td>
</tr>
<tr>
<td>2.85</td>
<td>2.925</td>
<td>3.0</td>
</tr>
<tr>
<td>3.0</td>
<td>3.075</td>
<td>3.15</td>
</tr>
<tr>
<td>4.25</td>
<td>4.375</td>
<td>4.50</td>
</tr>
<tr>
<td>4.35</td>
<td>4.475</td>
<td>4.60</td>
</tr>
<tr>
<td>4.50</td>
<td>4.625</td>
<td>4.75</td>
</tr>
<tr>
<td>4.60</td>
<td>4.725</td>
<td>4.85</td>
</tr>
</tbody>
</table>

Many vendors also provide different output driver options for their devices. The usual choices are open drain, open drain with internal pull-up and standard push-pull output drivers. The open drain options allow more than one source to pull the reset line to the reset state, such as a pushbutton or some other component that has the ability to reset the controller such as an over-temperature safety switch.

Since some microcontrollers have a low active reset line and some are high active, you must also choose a reset device with the correct polarity. For reference, the MCP100/120/130 are all active low devices and the MCP101 is active high.

CONCLUSIONS

Using supervisory circuits can protect microcontroller based systems from a number of power-related problems. If you are experiencing problems in your system that are not making sense, it may be power related and if so, it may be beneficial to add a supervisory device to the system. This application note provides some guidelines that you can use in determining what the problem might be and what device should be chosen to solve the problem.