Designing a Robust Failure Indicator

By Chris Keeser, Cypress Semiconductor Corp.

The other day I was driving to the local dump to drop off a load of trash carefully stacked into the back of the family minivan. Suddenly I noticed that the “trunk open” light came on. I quickly pulled over just as the rear hatch popped all the way open and the van began to empty my garbage bags all over the road. I hadn’t closed the hatch all the way when loading up, and the jostling of driving had caused it to pop open. I was immediately thankful for that simple little light that had prevented me from getting a littering ticket. It also reminded me how a malfunctioning “idiot” light had caused my sister to blow the engine in my ‘66 mustang.

Vehicles are complex machines with many critical systems working together to safely transport us where we need to go. All of the components are carefully monitored by a computer checking and rechecking the health of the engine, drive train, tire pressure, coolant system, and so on. If something goes wrong, the computer informs the operator of the problem through the various lights in the instrument cluster. However, what if one of the failures is the status light itself?

A great deal of information is captured behind that little “check engine” light. Over the years, they have been dubbed “idiot lights” based on the idea that if you didn’t take action when the light went on, you were an idiot or because even an idiot could understand the glowing red light with a “Check Engine” written beneath it means something is wrong.

In the past, the idiot light was an incandescent bulb. These failed regularly as the constant vibration caused the filament to weaken and eventually break. In the 70’s with the arrival of inexpensive Light Emitting Diodes (LEDs), status lights became far more reliable, but they were still not perfect. Typically, incandescent bulbs fail in an open mode, but LEDs can fail in either mode (open or short). These types of failures can occur when the LED has a bad connection to the PCB (or falls off entirely), is stressed by too much heat, or simply breaks.

To troubleshoot a failed light, you would normally break out your multimeter and measure the resistance of the failed light. If the light was an open circuit, it was bad. If the resistance was very low, then the LED had probably failed shorted. With a programmable system on a chip, the CPU that controls the indicator light can also measure its resistance and determine if the light is good, bad, or even questionable. What is needed is a way to measure the resistance.

Measuring resistance requires either applying a voltage stimulus and measuring the current, or applying a current stimulus and measuring the voltage. A voltage stimulus can have the undesired side effect of generating too much current through the device if the resistance is unexpectedly low. On the other hand, a current

stimulus will try to apply the desired current but will be limited to the rails in an open condition. From a safety standpoint, a current stimulus is preferred (see Figure 1).

It is common practice to place a current limiting resistance in series with the LED. A standard red LED has a forward voltage drop of about 1.8V when turned on and for a desired current of 5 mA in a 5V system, this resistor would be 634 ohms. For the resistance measurement, you can apply 10uA and expect a voltage around 1.806V (1.8V + 10uA * 634 ohms = 1.806V). If the indicator is open, you will see a voltage close to the rail. If the indicator is shorted, you will see a voltage around 6.3 mV. This approach has the advantage of not requiring any extra connections to the LED (no impact on system robustness or cost) and allows dynamic checking of the indicator health. The pin driving the LED is configured as an open drain “drives high” I/O and an analog connection is made to the pin.

The GPIO can be driven high, turning the LED on, or it can be placed into a Hi-Z mode. The Hi-Z mode turns the LED off, and enables the optional test mode. By enabling a current DAC (IDAC), a precise current can be forced through the LED and the current limiting resistor. The voltage at the pin can be measured, and the LED status can be determined. This current should be small enough to not illuminate the LED.

With the IDAC, more intelligent indicator health monitoring can be done. A sweep of the current and a measurement of the resulting voltage can yield the I-V curve of the LED, providing much more detail on its health, plus an aesthetically pleasing increasing glow from the LED if the current is swept to a high enough value.

This technique can also be applied to a whole host of other indicators in, on, or around a vehicle. Imagine how helpful it would be if your car could tell you that the left rear brake light was out, rather than that pleasant policeman who just pulled you over. And while we’re checking the connection, we can also use the diode junction of the LED to determine its temperature. This allows us to measure the temperature of the instrument cluster. A diode can be modeled with the following equation:

With some manipulations the temperature can be derived as a forward voltage drop for an applied current, or as a function of the forward drop and a current ratio when two currents are applied. However, since LEDs are not ideal, the performance of a particular LED as a temperature sensor should be determined empirically.

Idiot lights can be an asset for complex systems to determine correct operation and to inform operators of critical failures, but only if they are functioning. With the addition of impedance measurement hardware, idiot lights can be turned into an idiot savant.

Chris Keeser is an Applications Engineer Sr. at Cypress Semiconductor Corp.

Source URL (retrieved on *12/28/2014 - 10:12am*):
http://www.wirelessdesignmag.com/articles/2010/12/designing-robust-failure-indicator?qt-most_popular=0&qt-blogs=0