Microcontroller Applications in Solid-State Lighting

By Steve Bowling, Technical Staff Engineer, Microchip Technology, Inc.

Computing technology has come a long way. What once filled a room with power-hungry equipment and cost millions of dollars now fits on the tip of your finger, consumes microwatts of power and in many cases, costs less than a dollar. These advances in technology have enabled the practical integration of embedded microcontroller (MCU) technology into almost everything we use, including architectural lighting applications.

There are many types of architectural lighting applications including exterior, general interior, decorative and emergency lighting. You might not have considered designing a MCU into one of these applications, but there are many reasons to do so when solid-state lighting (SSL) technology is used. Each application will have unique requirements for intelligence. For example, emergency lighting should be able to perform self-diagnostics and communicate with fire-alarm systems. Decorative lighting may need variable color control. Interior lights may need to respond to an occupancy sensor or communicate with a central lighting controller. SSL technology usually requires an electronic driver and a MCU is well suited for this purpose, provided that it has the correct combination of cost, size, memory, performance and peripherals for the application.

A small MCU can add a tremendous amount of flexibility and value to a SSL application. Some of the ways that a MCU can help include regulating quality of light, extending LED lifetime and reliability, providing communication, sensing and control and improving energy efficiency.

With MCU intelligence in the design, you can easily change how the system responds to different events. Product features and functions can be changed with simple code modifications. For example, a soft-start profile for a light fixture can be programmed to have any rate desired. As another example, an over-voltage, over-current or over-temperature fault can be responded to in any way that is desired. You might want the light fixture to attempt to restart five times. After this, the fixture can shutdown to avoid damage to driver components. Later on, you may decide that the fixture should only attempt to restart twice. No hardware modifications are necessary to make this change.

There are three ways that a MCU can be integrated into a SSL system. First, a separate MCU and driver solution can be used. Secondly, a mixed-signal solution can be used. (The mixed-signal solution integrates a MCU along with the analog peripherals required to implement the driver into the same package.) Finally, a fully digital solution can be used, where a software control loop regulates the driver instead of an analog control loop.

I mentioned that a MCU can help regulate the quality of light and you may wonder how this is possible. A MCU can help LED light quality in two ways-lumen maintenance and color adjustment. It is well known that the light output of an LED varies from unit-to-unit with age and temperature. Most LED suppliers offer binning services to help reduce unit-to-unit variations. However, the light output of an LED will still decay with age and temperature. Furthermore, the aging process will be accelerated if the operating temperature is above a certain level.

In order to get the long lifetimes associated with power LEDs, the temperature of the die must be kept below the manufacturer's specified limits. LEDs don't radiate heat in the form of infrared radiation, like other sources of light do. This is an advantage to the end application. However, the downside is that the LED die and package will absorb heat that has to be conducted out of the light fixture.

If the LED light fixture and the LED driver are designed as a system, the designer can integrate thermal paths into the lighting fixture's mechanical features. This thermal protection can take the form of additional heatsink material or proper ventilation of the light enclosure. However, a LED-driver designer may not know what kind of fixture into

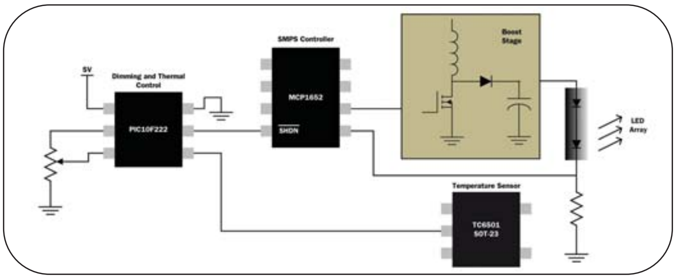

Figure 1. The 6-pin PIC10F222 MCU adds dimming control and thermal protection to a switch-mode LED driver.

which the driver will be installed. In fact, the driver could be installed in a variety of different fixtures to save cost. In this situation, active thermal protection can be built into the driver system.

A MCU can actively regulate the light output and operating temperature of the LED. The LED brightness can be controlled using one of two methods. First, the LED drive current can be varied to achieve the desired brightness. Secondly, the LED driver output can be modulated using a pulse-width-modulation (PWM) control signal. The resulting brightness will be proportional to the average amount of time that the LED is turned on by the PWM signal. The frequency of the PWM signal is set high enough so that the human eye will not detect any amount of flickering. The PWM control method is preferred, because it allows the LED to always operate at its most efficient operating current.

The addition of a MCU to a SSL circuit does not have to be complex or expensive. As shown in Figure 1, a 6-pin MCU has been connected to a LED driver circuit to provide dimming and thermal control. The MCU has an 8-bit analog-to-digital converter (ADC) and four I/O pins. The ADC reads the potentiometer value and then generates a PWM signal to control the LED driver's dimming input pin. A temperature sensor has been included in the system, to monitor the temperature of the LED heatsink. The MCU software can reduce the light output level, if required, to protect the LEDs.

You may notice that the potentiometer receives power from an I/O pin in this application. This is a power-saving technique that allows a very simple and inexpensive bias-supply circuit to be used with the MCU. Current is only supplied to the potentiometer for a short duration, when a voltage sample is taken. Therefore, the average current consumed by the MCU and the potentiometer bias can be quite low.

The MCU circuit shown in Figure 1 can be used to provide many different kinds of sensing functions. You may want to sense ambient light levels with a light sensor, for daylight harvesting applications. A photo-cell or a photo-diode circuit can be connected to the MCU's ADC input and measured for this purpose. A passive infrared (PIR) detector can also be added for occupancy sensing.

The PWM dimming technique can also be applied to variable color applications. In color applications, 3 PWM signals are used to control separate red, green and blue LED drive circuits. If high color resolution is not required, you can use a very basic MCU without PWM peripherals. The three PWM signals have a relatively low frequency and can be generated using software.

The color-control application shown in Figure 2A uses a low-cost 8-pin MCU with an internal voltage regulator and oscillator. This type of system might be used in an automotive-interior or a decorative-lighting application. The MCU reads the voltage on a potentiometer to get the color requested by the user. Then, the duty cycles of the three software

PWM signals are set to produce the appropriate color. Each PWM signal can have 64 duty-cycle steps, which provides a reasonable range of color combinations.

For applications that require more precise control of color, a MCU with hardware PWM peripherals can be used, to provide high-resolution dimming control of the red, green and blue channels. The system shown in Figure 2B provides 16,384 adjustment steps on each color channel. An optical color sensor can be added to provide closed-loop color control. The color sensor keeps the color constant over a broad range of dimming levels, and can be used to correct for LED aging characteristics.

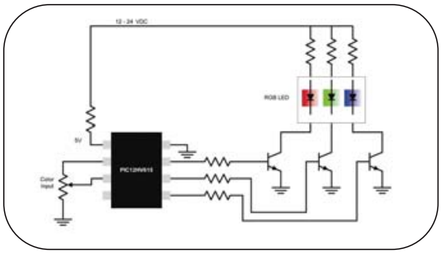

Figure 2A. A PIC12HV615 8-pin MCU is used to provide simple color control, using a RGB LED.

Many digital communication protocols can be applied to lighting applications. One of these protocols is DMX-512. The DMX-512 protocol was designed to serve theatrical lighting control applications, but it is gaining popularity in architectural lighting applications because of its simplicity and low cost. The protocol allows uni-directional, high-speed control of up to 512 lighting channels. A MCU with a UART peripheral is required to implement the DMX-512 protocol. The UART is configured for 250 kbps communication. RS-485 transceivers are used to physically connect the nodes together, which allows long connection distances using twisted-pair wiring.

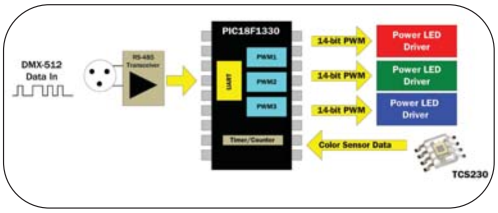

Figure 2B. A PIC18F1330 MCU is used with a TCS230 color sensor to provide accurate color control of a RGB light source.

Large buildings may utilize the Digitally Addressable Lighting Interface (DALI), which provides a way to network groups of lighting fixtures. DALI uses a 1,200 bps Manchester-encoded serial data stream, which is easy to implement on almost any MCU. Other protocols commonly used in building-management systems include Ethernet and RS-485. IEEE 802.15.4 wireless networking protocol stacks, such as the ZigBee and MiWi wireless protocols, show promise for use in lighting-control networks. These wireless protocols are designed to be run on a small MCU with limited memory resources.

Power LEDs are best driven with a source of constant current, and most power LED applications will require a constant-current driver. Depending upon the power level, this driver might be a linear or a switch-mode solution. Where energy efficiency is a concern, this driver is almost always of the switch-mode variety.

The same switch-mode topologies used for voltage power supplies can be used for constant-current LED drivers. At the specified drive current, a power LED will have a forward voltage drop (VF) across the anode and cathode. The VF of the LED, isolation requirements and the available power-supply voltage will determine what kind of switch-mode topology should be used.

As mentioned earlier, a MCU can be used to provide full digital control of the power-supply function. In the case of a LED driver, the LED current is measured using a MCU's on-chip ADC. An error signal is calculated, which is simply the difference between the power supply's desired and measured output currents. A control loop is then calculated in software, and the result is used to set a PWM duty cycle that controls the power-supply output. A proportional-integral (PI) controller algorithm is commonly used for this purpose, and allows the user to tweak the coefficients of the controller to get the desired closed-loop response.

Implementing full digital control of a typical LED power supply does not require a lot of MCU processing power. This means that a low-cost, low-power MCU can do the job. Some power supplies need very fast response to changes in load or input voltage. Fast ADC sampling rates and fast PID loop calculations are required in these cases, requiring a more expensive MCU or Digital Signal Controller (DSC). However, the LED will present a relatively constant load to the power supply, and fast control response is usually not required.

Figure 3 shows a constant-current LED driver. The MCU has an internal voltage regulator and oscillator, so no external components are required. The MCU also has a PWM module, which is used to drive a buck circuit at 125 kHz. The LED current is measured when the MOSFET is turned on, and an Op Amp circuit amplifies and filters the signal before it is sampled by the ADC.

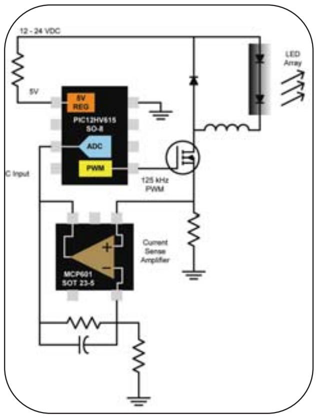

Figure 3. A PIC12HV615 MCU and a MCP601 Op Amp are used to implement a step-down LED constant-current driver. LED current regulation is performed in software, using a PI controller.

The current is sampled and the PI controller is executed at a 1 kHz rate. The current-controller calculations use approximately 45 percent of the available CPU processing time, leaving plenty of time for other functions. If the measured current is within specified limits, the PI controller is not executed. This provides even more CPU processing time for other tasks. Less than 5 percent of the CPU processing time is used, after the LED current has stabilized to the target value.

The PI controller is relatively simple to implement and can be written in a few lines of C code. It responds to the current error signal in two ways. First, the error signal is multiplied by a gain value to generate a proportional response. If the error is large, the controller makes a big change to increase or decrease the power-supply output. If the error is small, the controller makes a small output change.

A controller that has only a proportional response will never decrease the error all the way to zero. In other words, the LED drive current will never reach the target value. The integral part of the controller adds the calculated error to a running sum, each time the output is measured. This sum of all prior errors is multiplied by a gain value and then added to the proportional output of the controller. The integral term provides a way to correct small, steady-state errors in the power-supply output, since a small error value will be added over time to produce a larger output.

The same PI-controller software can be used to drive a variety of LED loads and power-conversion stages. For each different application, the controller's software gain values would be adjusted to provide the best system performance. Other software values can be used to specify operating limits for the specific driver application.

This article has shown a few ways in which a MCU can add value and flexibility to SSL. Many architectural lighting applications can benefit from the intelligence that a MCU brings including general interior lighting, emergency lighting and decorative accent lighting. The MCU can serve a variety of functions in the application, including user-interface functions, system protection and compensation for LED deficiencies. The MCU can even control the power stage directly, providing regulation of the LED drive current.