Re: High brightness white LEDs damaged by custom switcher

Source: http://sci.tech−archive.net/Archive/sci.electronics.design/2007−05/msg00685.html

- *From*: miso@xxxxxxxxx
- *Date*: 6 May 2007 01:23:10 −0700

On May 5, 3:44 pm, "Paul E. Schoen" <pst...@xxxxxxxxx> wrote:

> <m...@xxxxxxxxx> wrote in message
>
> news:1178400948.897551.99850@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
>
> > Just so I'm reading this correctly, you are not paralleling strings of leds. That is, you only drive one string of leds. If that is not the case, I would expect problems.
> >
> > When I think of the effort it takes to make a bullet−proof DC/DC chip, I just shake my head at the idea of doing it in software. In a chip, events take place simultaneously, while a uP is a step at a time using polling.
>
> The LEDs are in series, so all see the same current, and the voltage required is about 26 VDC for 7 and 49 VDC for 13. The PIC can respond to certain events within a few microseconds, by using interrupts. The difficulty is in generating the interrupt signal outside the PIC. That is why I plan to put a transistor on the current sense to detect an overcurrent. I could also add a similar circuit to detect output overvoltage, and generate the same interrupt.
>
> The first thing to do is disable the PWM output, which can be done in a few clock cycles. Less than 1 uSec for an 8 MHz clock. Then the A/Ds can be used to see what caused it, and act accordingly. All three analog inputs are now read within 1 mSec, but could be within 60 uSec. The A/D can make a reading in 18 uSec.
>
> As long as the external circuitry has a sufficiently slow response, I don't see any problem implementing a switcher with a PIC. The real advantage is that the hardware can be built in a simple, straightforward way, and then

changes can be implemented in PIC code. As requirements change, the same circuit can be used with little or no change, and the PIC can be reflashed to the new parameters.

I think the PIC is perfectly suited to this application. It may not be so for situations where the input voltage may change suddenly, or output loads are constantly changing. The main problem here seems to have been identified, and several possible fixes should eliminate it. Extensive testing should prove that.

Thanks,

Paul

It is one thing to get a circuit to work. It is another thing to turn it loose on the general public. This is where the controller chips shine over home brew designs. For instance, what happens if the user inserts weak batteries. That is, how good is the undervoltage lockout. What about an intermittent battery? Both at start up and during operation. There is quite a bit of engineering in a DC/DC chip that the user never sees, but it makes the design robust. Oh, and all this has to work over temperature.

The typical start−up circuit work like this. First, you have enough supply voltage to exceed a VT. One you have a VT, then you have trust worthy logic. Next up, you would wait for the voltage reference to exceed some simple reference, often just a N−fet fed with a current source. The bandgap can take microseconds to start up, to maybe hundreds if it is very low current. Once you trust the reference, you will measure the supply voltage to see if it is suitable. If the voltage is too low, the logic can be flaky. Once all conditions are met, you start a timer circuit because just maybe the voltage source is not steady (switch bounce, whatever). The you fire up the DC/DC, there are other safety circuits. For instance, a relay could fire and glitch the chip. [Probably not your situation.] A watch−dog timer will insure the logic gets reset if the pulse width is well out of spec. There are other safety features, typically over current protection on the power fet.

Basically, the off the shelf chip is (or perhaps should be) bullet proof. I just can't see doing this in a pic. The controller chip you buy has the history of a few in the field failures.

.