

Re: Auto shutdown/restart software for FreeBSD?

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/questions/2007-05/msg01020.html>

- *From:* Ian Smith <smithi@xxxxxxxxxxxxxx>
 - *Date:* Thu, 17 May 2007 18:17:59 +1000 (EST)
-

On Tue, 15 May 2007 00:16:34 -0500 WizLayer <wizlayer@xxxxxxxxxx> wrote:

On Monday 14 May 2007 08:27:48 pm you wrote:

On May 13, 2007, at 7:13 PM, WizLayer wrote:

On Sunday 13 May 2007 07:17:14 pm Aftab Jahan Subedar wrote:

Would it recharge the battery fully after discharge? I dont think so.
So you got to recharge the external battery EXTERNALLY after power failure.

Indeed. UPSes are designed to start recharging their (usually 12V sealed lead-acid) batteries at around the 5-hour rate, which might be say 2.4A for a 12Ah battery, for a say 300VA UPS, tapering that off as voltage rises. Using (just) the UPS to recharge a say 100Ah vehicle battery at that rate would take maybe 2 days, assuming the UPS didn't overheat and perhaps blow up trying.

However a simple (regulated) charger on the external battery is fine, though possibly thoroughly confusing the UPS' brains about how long recharge should take, available capacity and such. For adding external batteries, simple 'dumb' UPSes are usually better than 'smarter' ones.

[..]

This is another approach that seems like it would be practical:
Use deep cycle car batteries, trickle charge with solar panels.

Sure, you can do that, assuming it's sunny during/after a power outage. Here at least, outages mostly tend to correlate with stormy weather.

Re: Auto shutdown/restart software for FreeBSD?

If a desktop computer can run on square wave generated by dc/ac converter, use that as a power backup system, It would have to have some kind of switching system to detect main power drop and switch to the backup system.

That's pretty much what a UPS is doing, though they usually provide a 'modified square wave' that somewhat more closely approximates a sine wave output. Note however that a UPS has to switch cleanly from mains to battery power supplying the last stage inverter within just a few milliseconds, ie a small fraction of a single mains 50 or 60Hz cycle.

Perhaps someone would be willing to, with engineering expertise put together servers that would work on laptop batteries, like a laptop. I do have one machine that has Yellow Dog linux (Mac Powerbook 3400c) that runs 24/7 as my backup DNS server.
JK

Laptops as servers is sure the way to go on solar-powered houses; hard to find or make a better UPS than a laptop p/s. For desktop boxes, there are available computer power supplies that run straight off 12V. I hunted for a mob called DC-PC, but they may be defunct. eg check:

<http://www.powerstream.com/mini-itx.htm>
<http://www.mini-box.com/s.nl/sc.8/category.13/f>

Why settle for a square wave? It's not hard to clean that up, and besides...

Very much easier said than done :) and besides ..

Wouldn't that bring mayhem and havoc on a scanner (ie, I'm pretty sure that you your screen would do very unhappy things)? LCD screen? don't know. (or a system's power supply over long term? hmmm)

Devices that aren't happy with non-sinewave power include such as laser printers, but those you mention here use switching supplies themselves, and are usually happy enough running on square-wave inverters, as long as the inverter can handle the peak startup currents often demanded.

Computer power supplies aren't fazed by square wave (or for that matter, high voltage DC) input, as they're chopping the input waveform anyway. Also, you probably don't want printers and such running off the UPS.

Re: Auto shutdown/restart software for FreeBSD?

As far as the type of batteries, deep cycle marine batteries, whatever. It doesn't really matter except to say that some types can be fully discharged and some would be ruined on a full discharge. The health and monitoring portion of the UPS would have to be designed with those limits in mind (and, hey... That could be part of the embedded mprocessors job, too... more options).

Well that's just what any decent UPS does. While I wouldn't want to discourage anyone from learning embedded design and programming, it's terrific fun, but you can expect to spend hundreds of hours and not a little cash doing so by trial and error, to save on a what, \$200 UPS?

Switching power from one source to another is something that I've not had a lot of luck with, esp with sensitive stuff like a computer's power supply (touchy). On the other hand clean, dc power in a parallel circuit is as simple as it gets. Edison had a good idea after all.

Look at the battery as your constant source, and work away from that. Your secondary source merely compliments the battery. So long as you use regulators for your "other sources", it will stay "Clean by default." :)

As far as switching power sources from regular charger to something like solar panels, same concept... Don't switch from-to anything. Keep it constantly hooked up in parallel with the battery.

For best efficiency, most UPSes use the mains as long as it's available, even in a degraded state such as under- or over-voltage, using 'boost' and 'buck' windings/circuits to maintain smooth power. It sounds good, but just using the mains to (float) charge the battery, then fulltime running an inverter off that adds another 15% or so inefficiency, and tailoring correct float voltages at different temperatures is another art again, essential to maintaining battery safety and longevity.

Some simple logics could perform circuits acts for the solar panels ("if sun is good and elec_co's bad , then close..." however you want to hack it.) Same for dis/associating the charger.

Ah Wiz, you make it sound so easy :)

I just need someone to point me in the right direction as far as embedded mprocessors. I've googled it, and found a few hobby kits, but I'd rather hear it from someone who deals with stuff like this and can suggest a "start here." I hate jumping into something only to have to unlearn in order to learn it right.

Re: Auto shutdown/restart software for FreeBSD?

I like the Atmel AVR processors, and am parttime working on a couple of things currently using AVR ATtiny45 8-pin parts. There's a gcc, tools and assemblers in the ports tree, and a plethora of free/cheap serial or parallel programmers out there. Others will advocate PICs, and others.

It's really a personal preference; I enjoy the relatively orthogonal AVR instruction set – despite being little-endian – and its memory and I/O layout, pin functionality and such. Try a few and see what feels good.

PS When I finally get this finished, I'm going to find some way of modifying the BSD license to apply it and release the whole thing to the public. One would only have to buy the parts, program, and assemble the thing... Why? Because UPS prices are a rip-off and some jerk told me I couldn't. :)

I'm amazed how cheap UPSes have become these days .. but do have fun!

Cheers, Ian

freebsd-questions@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-questions>

To unsubscribe, send any mail to "freebsd-questions-unsubscribe@xxxxxxxxxxx"