



Edited by Bill Travis

Circuit forms simple, low-cost, 1-kV driver

Tai-Shan Liao and Prasit Champa, National Science Council, Hsinchu, Taiwan

HIGH-VOLTAGE DRIVERS have recently received much attention, because they play an important role in driving piezoelectric and electro-optical components, for example. **Figure 1** shows a simple, low-cost, 1-kV driver. The circuit uses offline, current-mode-control techniques and a flyback switching-power-supply design. IC₁, a UC3844, is the major control component, using a switching frequency of 100 kHz. The IC provides frequency modulation to reduce the switching frequency under light- and no-load conditions. The feedback volt-

age, which you derive from the output of the error amplifier, serves as the indicator for load conditions. Once the feedback voltage becomes lower than the green-mode threshold voltage, the switching frequency starts to decrease.

All the power losses are in direct proportion to switching frequency. These losses include the switching losses of the transistor, core losses in the transformer and inductors, and the power loss of the snubber. The frequency modulation in the PWM-controller IC reduces the power consumption in the supply under

Circuit forms simple, low-cost, 1-kV driver.....87

Make a printer-port EEPROM programmer and dongle88

Circuit controls ratiometric or simultaneous power-up of multiple rails.....90

Scheme provides automatic power-off for batteries.....92

Publish your Design Idea in EDN. See the What's Up section at www.edn.com.

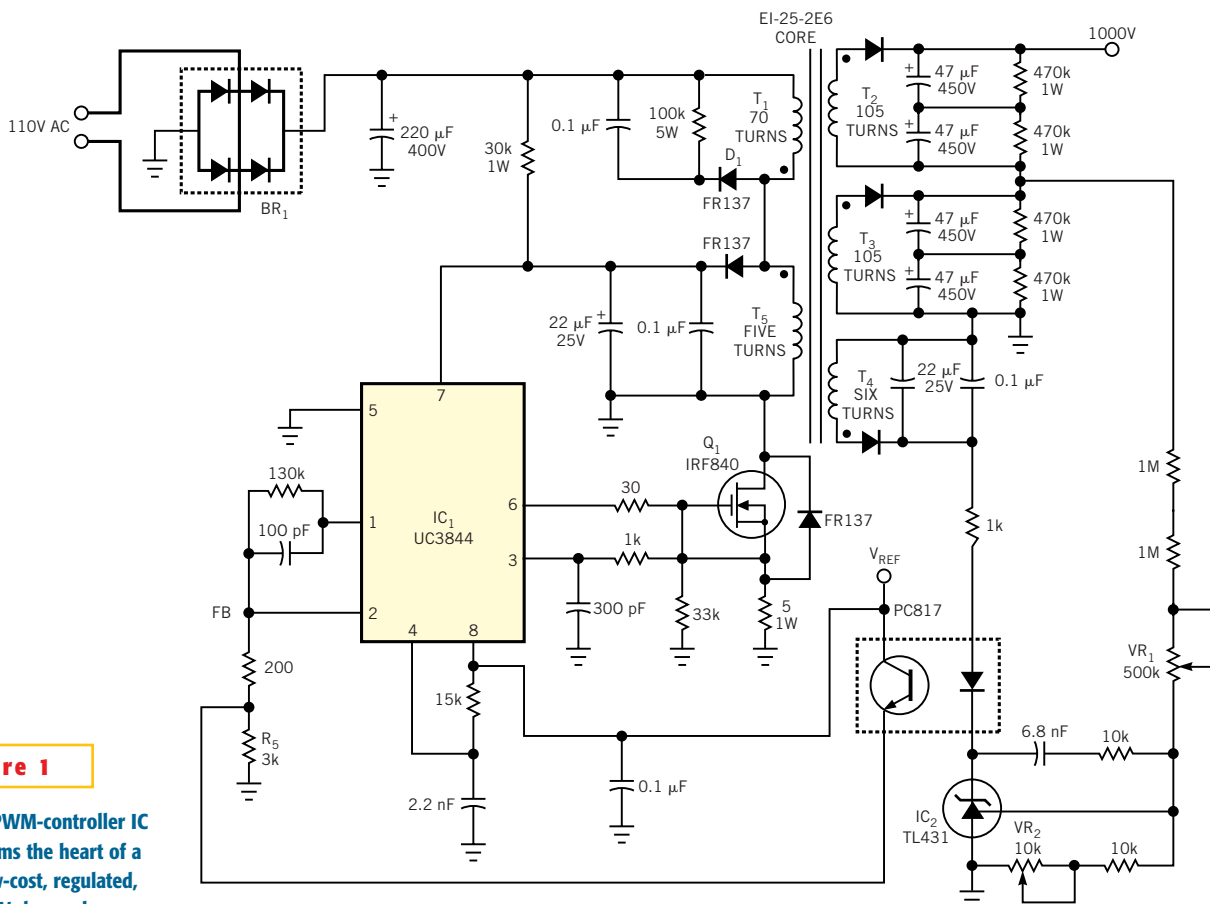


Figure 1

A PWM-controller IC forms the heart of a low-cost, regulated, 1-kV-dc supply.

light- and no-load conditions. But the frequency modulation has no effect on the PWM operation under normal- and high-load conditions.

Pin 2 (the feedback pin) of the UC3844 sums the current-sense signal, the output-voltage feedback signal, and any added slope compensation. The feedback-control circuit uses a TL431 adjustable shunt regulator to detect the output signal. A PC817 passes the signal to the feedback pin of the UC3844. The TL431 acts as an

open-loop error amplifier with a 2.5V temperature-compensated reference. When the output voltage is lower than the desired level, the feedback to the UC3844 automatically compensates the pulse-width modulation of the output triggering signal. Ceramic bypass capacitors (0.1 μ F) from V_{CC} and V_{REF} to ground provide low-impedance paths for high-frequency transients. This design uses a Tomita (www.tomita-electric.com) EI25-2E6 core set to fabricate the transformer. To

prevent core saturation, the gap is approximately 1 mm. The primary winding has 70 turns of 28-gauge wire. Both the secondary windings have 105 turns of 34-gauge wire. The primary and secondary auxiliary windings have five and six turns, respectively, of 34-gauge wire. The dc output voltage of the circuit in **Figure 1** is 1 kV (fixed). You can adjust the output voltage in a 50V range by adjusting VR_1 . Both load and line regulation are less than 1%, and power efficiency is 80% at full load. □

Make a printer-port EEPROM programmer and dongle

GY Xu, XuMicro, Houston, TX

YOU CAN EASILY USE a PC's printer port for serial-EEPROM programming. You can use a device-programmer circuit used to program the MicroWire serial EEPROM 93CXX (**Figure 1**). The circuit is so simple that any further simplification seems impossible. This programmer circuit contains no microcontroller, as most device programmers do. It needs neither a separate power supply, or "wall-wart," nor a cable. When in use, it directly plugs into the PC's printer port. However, you still can use a cable if convenient—for PC printer ports behind the PC, for example. The circuit also requires neither a resistor nor a decoupling capacitor.

These advantages come from the PC's printer-port resources and the architectural simplicity of the MicroWire serial EEPROM. The printer port comprises the 8-bit data, status, and control registers. Each register has its unique address. On the classic IBM PC, the data port serves solely for output, but the control port can serve as either input or output. The eight-pin, tiny, serial EEPROM consumes less than 1-mA current in the active state, and the printer port's data pin can supply a few milliamps, so this design uses D7 (Pin 9) as a power-supply pin. No decoupling capacitor is necessary in practice.

The MicroWire chip uses the CS (chip-select), SK (clock-signal), DI (data-input), and DO (data-output) pins to control its

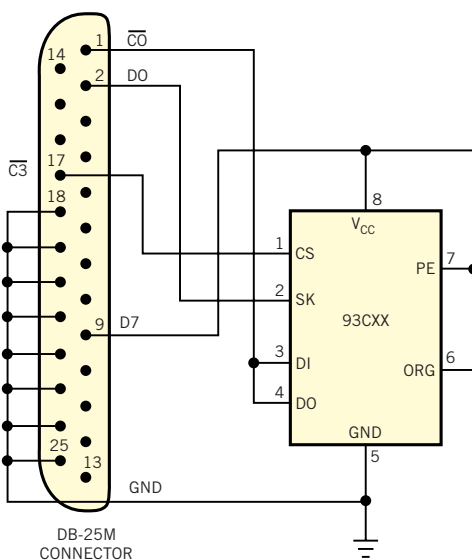


Figure 1 This printer-port serial EEPROM programmer can also act as a dongle once you program the device.

read/write operations. This design uses the chip-select signal from the reverse level of the control bit $\overline{C3}$ (Pin 17). It also ties together pins DI and DO and connects them to the Control bit $\overline{C0}$ (Pin 1), which can serve as input or output, thereby saving one pin. These selections caused no problems in practice. Because control Pin 1's logic is the reverse of the logic level on bit $\overline{C0}$, the software must take care of the inversion. The MicroWire interface normally requires a pullup resistor on the DO pin, but such a resistor is already inside the PC, so it's unnecessary.

Once you settle on the hardware design, the main task is to write software. This task is not difficult. For many embedded-system-software engineers, it's routine and interesting. A freeware executable program, Pseep2.exe, is available for this purpose. A sample demo program, secret.bin, allows you to practice the programming. You can download the software from the Web version of this Design Idea at www.edn.com. It handles only one MicroWire device—the popular 93C46's read/write operation as an example. Another important feature of this circuit is that, once you program the 93CXX device, the system becomes a primitive dongle. You can then use it as a hardware-protection device for your valuable software. Only you know whatever was programmed in the device.

When the protected software runs, it first checks whether the device is present at the printer port and whether the code matches what you programmed. If a match doesn't exist, the software refuses to continue and exits. The dongle is primitive, but it does illustrate the basic principle of dongle-protection technology. You can build the circuit using wire-wrapping or point-to-point soldering techniques on a solderless breadboard, in which case you'll need a cable, or with your own pc board. It's a one-evening project. □

Circuit controls ratiometric or simultaneous power-up of multiple rails

Dirk Gehrke, Texas Instruments, Freising, Germany

MANY APPLICATIONS use FPGAs, ASICs, or DSP chips, which usually require multiple voltage rails, typically two: the core voltage and the I/O voltage. The core voltage is usually lower than the I/O voltage. Guidelines for determining how to power up two or more voltage rails depend on the part and the manufacturer you use. The first implementation in **Figure 1** shows how to realize ratiometric sequencing, which means that both power-supply output rails simultaneously start and simultaneously reach their final regulated output voltage. This implementation uses resistor R_{15} connected to ground; the path and components in red are deleted. You can achieve the ratiometric function by stacking together multiple converters that share one soft-start capacitor. This connection ensures that both controllers ramp up their output voltage at the same time during power-up. Both the IC_1 and

IC_2 controllers share a soft-start capacitor, C_{14} . This example uses two buck converters with integrated synchronous-rectification FETs. From a 5V input-voltage rail, IC_1 generates the 3.3V I/O voltage. Buck converter IC_2 generates the 1.5V output voltage.

The soft-start pin, available on both controller ICs, serves two purposes. You can use it to enable the controller circuitry if required—an implementation you could realize by tying an open-collector or open-drain gate to the SS Pin. If the transistor or FET is active, it ties the SS Pin to ground potential, forcing both controllers to stay off. Once you release the SS Pin, both ICs start to charge C_{14} with their internal 5- μ A current sources. In total, 10- μ A current flows into C_{14} . Once C_{14} reaches the threshold voltage of 1.2V, both controllers start to operate. You can easily calculate the delay versus the capacitor's value: Delay

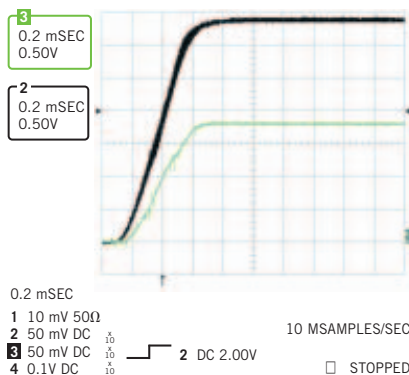


Figure 2

This graphic shows measurement results for the ratiometric implementation.

time = $C_{14}(1.2V/10\ \mu A)$. As the output activates, a brief ramp-up at the internal soft-start ramp may occur before the external soft-start rate takes control. The output then rises at a rate proportional

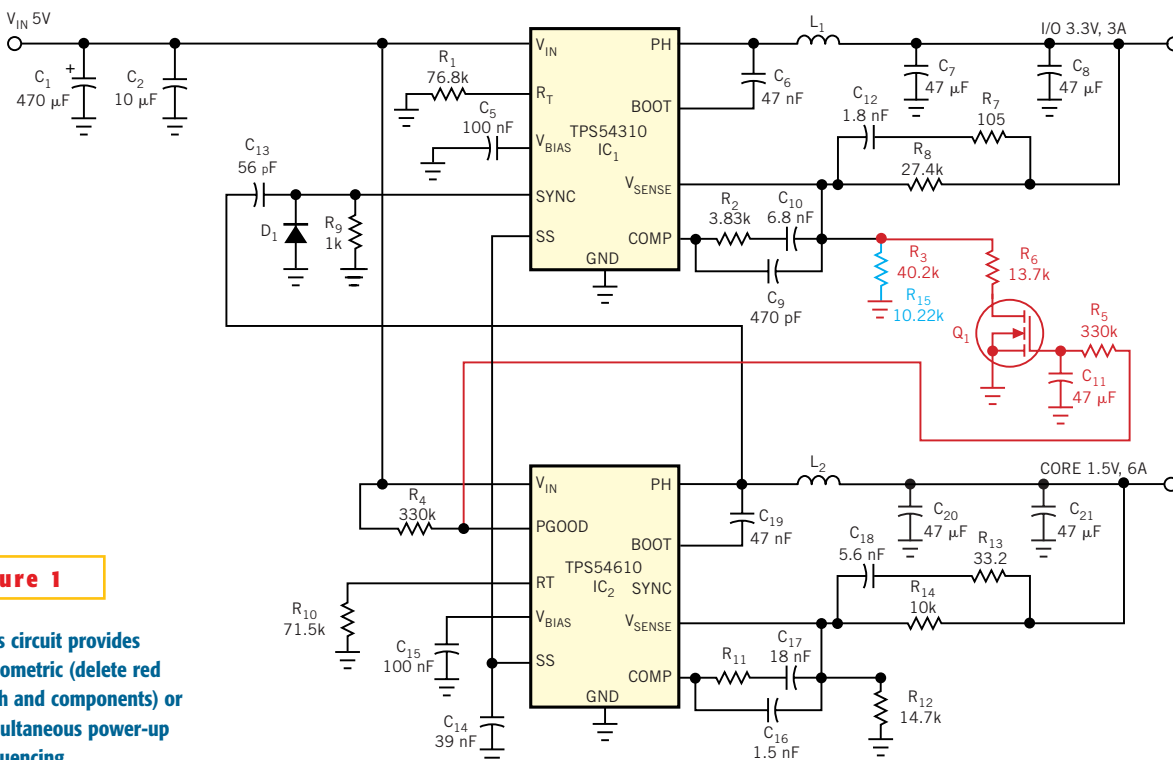


Figure 1

This circuit provides ratiometric (delete red path and components) or simultaneous power-up sequencing.

to the soft-start capacitor. You can program the soft-start time via C_{14} . The next equation represents the soft-start time calculation. The actual soft-start time is likely to be less than the calculated approximation because of the brief ramp-up at the internal rate. Soft-start time = $C_{14}(0.7V/10 \mu A)$. If you set IC_1 for

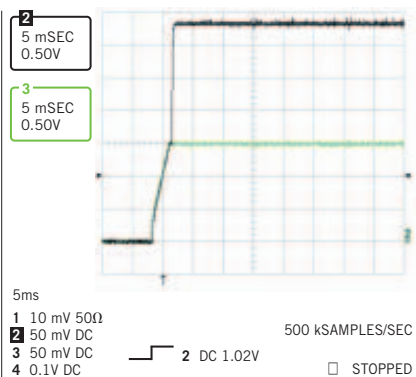


Figure 3

These curves, distinctly different from those in Figure 2, show simultaneous-sequencing results.

3.3V and IC_2 for 1.5V, they both reach their final voltage level at the same time. Figure 2 shows measured results of the ratio-metric sequencing.

In the simultaneous-sequencing scenario, IC_2 acts as the master controller. You program its output voltage via R_{14} and R_{12} to a value of 1.5V. R_8 and R_3 program the slave controller IC_1 's output voltage to a value of 1.5V. As the ratio-metric scenario describes, both voltages

start at the same time with the same ramp, reaching their final value at the same time. Once both rails reach 1.5V, you must increase IC_1 's output voltage to 3.3V, its final value. To make that increase happen, Q_1 places R_6 in parallel with R_3 . You can calculate the value of R_6 using the next three equations. The given parameters are: $V_{OUTCORE} = 1.5V$; $R_8 = 27.4 \text{ k}\Omega$; $V_{REF} = 0.891V$, the internal bandgap-reference voltage of IC_1 ; and $R_3 = 40.2 \text{ k}\Omega$. You can program $V_{OUTI/O}$ via R_8 and R_X . R_X represents the value of R_3 and R_6 in a parallel connection.

$$R_3 = \left[\frac{V_{REF}}{V_{OUTCORE} - V_{REF}} \right] \cdot R_8$$

$$R_X = \frac{V_{REF} \times R_8}{(V_{OUT I/O} - V_{REF})}$$

R_X must have a value of 10.22 k Ω to produce $V_{OUTI/O} = 3.3V$.

$$\frac{1}{R_X} = \frac{1}{R_3} + \frac{1}{R_6} \Rightarrow R_6 = \frac{1}{\frac{1}{R_3} - \frac{1}{R_X}}$$

In this example, R_6 needs a value of 13.7 k Ω . Applying 5V to the input-voltage rail activates both controllers at once, allowing them to start at the same time. Once the master controller, IC_2 , reaches an output voltage level equal to or greater than 90% of the initial value, the IC releases the power-good open-drain-com-

parator output pin. This action forces the pin to rise immediately to the output-voltage level because of resistor R_4 's pullup action. A lowpass filter consisting of R_5 and C_{11} forms a delay circuit, driving MOSFET transistor Q_1 's gate. This delay circuitry determines when Q_1 becomes active. Q_1 has a threshold voltage, V_{GSTH} , of 1.6V. Once the gate voltage reaches or exceeds the threshold voltage, V_{GSTH} , Q_1 starts to conduct, putting R_6 in parallel with R_3 . Because of the resistor-ratio change, IC_1 's output voltage ramps up to its final I/O-voltage value of 3.3V. The MOSFET this design uses has an on-resistance of roughly 10 Ω . This figure might sound high, but, because of the high-ohmic-resistive divider, this value does not affect performance. Figure 3 shows the results of the described implementation during power-up.

Significantly, in this implementation both converters run at the same switching frequency. IC_2 is the master controller, programmed to a 700-kHz switching frequency. IC_1 starts at a lower initial switching frequency of roughly 630 kHz, 10% below the switching frequency of IC_2 . Once IC_2 begins to operate, it synchronizes IC_1 via the Sync Pin. Diode D_1 limits negative voltage spikes at the Sync input. Placing a well-chosen Schottky diode between both output voltage rails can ensure that, even during power-down, both rails have a voltage difference of 400 to 600 mV for safety reasons. The cathode connects to the I/O-voltage rail, and the anode connects to the core rail. □

Scheme provides automatic power-off for batteries

Miguel Gimenez, Altair-Equipos Europeos Electronicos, Madrid, Spain

THE CIRCUIT IN Figure 1 provides a simple and inexpensive way to protect one of the most valuable components in portable applications: the battery. Applications include all portable equipment that requires a limited time of operation, such as test instruments, guitar tuners, and electronic toys. Pressing the on/off momentary switch starts the cycle, and the circuit provides power to the application circuit. If you again press the switch at any instant, the circuit switches off and "sleeps" until the next cycle. In case you forget to switch off the

circuit, the circuit incorporates an auto-power-off function with a time period that is a function of preprogrammed time constants.

IC_1 and related parts provide a bistable toggle function and also ensure protection against switch-contact bounce. IC_{1C} buffers the toggled signal and isolates the R_1 - C_1 charging current. This signal feeds the IC_2 timer, configured as a monostable multivibrator that remains activated until it times out, according to the expression $t = 1.1 \times R_1 C_1$. This figure is the auto-power-off time. In the example, this interval

is approximately six minutes. The timer's output feeds the Q_1 inverter that activates medium-power, pass-through Q_2 transistor. This circuit is configured as a pnp block to ensure low losses to the load. The loss comes only from $V_{CE(SAT)}$ —approximately 0.2V at 100 mA, or 20 mW. For applications demanding more current, you can choose a more suitable transistor. A MOSFET can be an efficient approach when you need either lower standby losses or a lower voltage drop between the battery and the application circuit.

Standby losses in the switched-off state

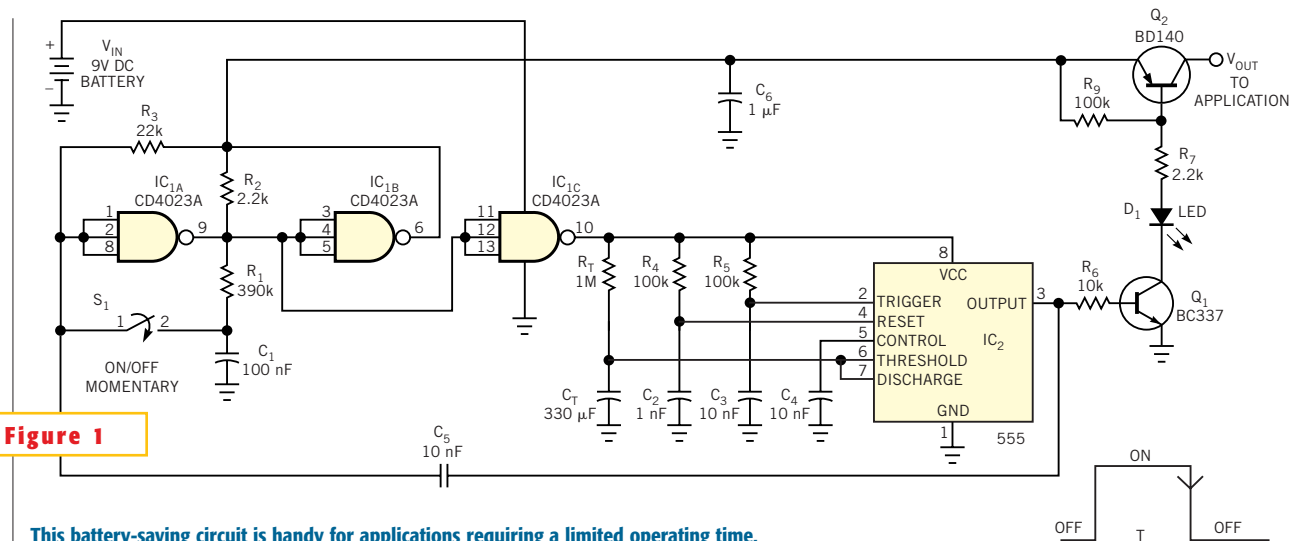


Figure 1

This battery-saving circuit is handy for applications requiring a limited operating time.

are negligible, because the circuit draws power only from the CMOS gate in the inactive off-state. LED D_1 indicates the on-off status of the circuit. No extra power comes from the battery to drive this

LED, because it is connected in the current-source leg of the driver transistor. The output transition to 0V during time-out ensures the timed power-off by means of the C_5 feedback loop that tog-

gles the bistable circuit to the off state, performing the same role you might have forgotten with the on/off switch. This simple circuit is useful when the application doesn't require a microcontroller. □