Keywords: port expanders, GPIO, I2C, push-pull outputs, open-drain outputs, LED intensity control, constant-current drive, MAX6966, level translation, transition detection,

Apr 26, 2006

APPLICATION NOTE 3814

Port Expanders for Flip-Cell Phones Save Cost and Space

Abstract: This article discusses the features available in recent-design port expanders, which are particularly relevant for the design of space-limited, cost-sensitive flip phones. The MAX6966 and MAX6965 drivers illustrate the GPIO port-expander technology discussed in the note.

Introduction

The goal for general-purpose input output (GPIO) port expanders has always been to provide a modest number of I/O ports in a small, low-cost IC. Parts offering 8 or 16 ports have been around almost as long as the I²C and SPI serial busses to which they connect. Features offered by these early parts included simple open-drain or push-pull outputs with limited drive current, and logic inputs with non-latching transition detection. The smallest available package was a TSSOP.

This application notes discusses the features available in recent-design port expanders, which are particularly relevant for the design of space-limited, cost-sensitive flip phones.

The Flip-Phone Interconnection Problem

The case of a flip phone comprises two halves that fold together like a clamshell (**Figure 1**). Baseband and radio circuitry reside in the main section, usually the thicker half, together with the keyboard, battery, and antenna. A common layout for compact phones places a large display on the inside of the flip and a smaller one on the outside, as shown. The outside display, often a trans-reflective LCD that is readable without a backlight, operates at all times to show when the phone is idle and other status information. The flip enclosure holds the phone earpiece, and in some cases other audio and ring-tone circuitry. Many designs also include a camera module in the flip.

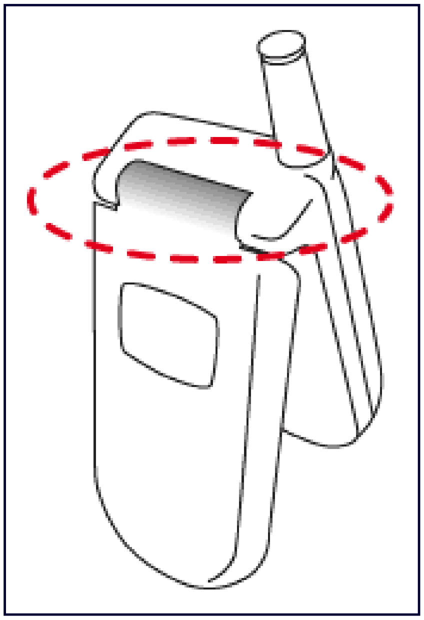


Figure 1. A flexible circuit connects the two cell-phone halves. A flip-phone's hinge is, however, an interconnection bottleneck.

In most flip enclosures, the display(s) and camera include separate, moderately fast (Mbytes/second) parallel-interface busses for updating the displays and downloading the camera pictures. Passing data from the flip to the main body through the hinge, however, creates a bottleneck. That hinge connection is typically a flexible circuit made of Mylar® with copper traces. Thus, to ensure a reliable circuit after repeated flexes, the trace density (and consequently the number of traces) must be limited. To exacerbate the problem, phone designers are always pressured to reduce the number of connections between the flip and the main body.

Advantage of GPIO Port Expanders in Flip-Phones

In general, a flip-phone design should minimize the number of copper traces on the flexible circuit (flexi) joining the flip to the main body. An industry trend serializes the high-speed parallel connections to the flip display(s) and camera. An easy way to reduce the other connections is to identify signals and controls that can be synthesized directly on the flip, rather than imported across the flexi. A small, low-cost port expander can control logic-input signals, outputs, LED drivers, or power-control switches. The port expander connects to the main board through an I²C or SPI interface, which may be already available on the flip.

Port expanders are also low-power devices. To be useful in a cell-phone architecture, a port expander must:

- Have a physically small package (2mm x 2mm or 3mm x 3mm thin QFN) to allow placement wherever needed.
- Have a standard serial-protocol interface such as I²C or SPI.
- Be interrupt driven to avoid power-wasting CPU polling.
- Operate main functions (PWM, input monitoring) without CPU intervention.
- Operate from a low supply voltage of 1.8V to 3V, and eventually below 1V.
- Draw supply currents in the 1µA range.

LED Driving

LEDs in cell phones are used for display and keypad backlights (2 to 6 LEDs), function and status indicators, RGB fun lights, and battery and signal-strength indicators. Port expanders save space and power while reducing unnecessary interaction within the system in several ways: PWM intensity control for individual LEDs; high-voltage and high-current drive without the need for space-consuming discrete transistors; and direct-from-battery LED drive, which saves the cost and EMI of charge-pump or inductive-boost power supplies.

Open-Drain Ports Provide High-Current Drive

An open-drain output port can easily drive an LED. The port operates as a hard output switch to ground, and a series resistor (often called the ballast resistor) sets the LED current. Port expanders suitable for driving LEDs have high-current ports rated for voltages higher than the supply voltage, and pulse-width modulation (PWM) to adjust the LED brightness. The MAX6965 LED driver, for example, is a 9-output device with intensity control and hot-insertion protection in a 3mm x 3mm QFN package. It provides nine 7V-rated open-drain GPIOs with 50mA current-sink capability and individual output PWM control.

Constant-Current Ports Drive LEDs Directly from the Battery

A better approach to LED driving is a constant-current sink, which replaces the more conventional hard output switch and current-limiting ballast resistor. Constant-current LED drivers offer two key advantages:

- LED current is independent of variations in the LED forward voltage or the LED supply voltage.
- LED supply voltage can be lower (closer to the LED forward voltage), which improves efficiency.

A constant-current driver allows a lower LED supply voltage because the voltage across a ballast resistor must be high enough to offset variations in the LED's supply voltage and forward voltage drop. As an example, consider a white LED driven from a 5V ±5% supply, with forward voltage specified at 3.1V ±0.25V. Voltage across the ballast resistor is 1.9V nominal, and varies from 1.4V to 2.4V. The worst-case variation in current is therefore ±26%. If the supply voltage is reduced to 4V ±3%, the ballast-resistor voltage is 0.9V nominal and varies from 0.53V to 1.27V. The worst-case current variation is now ±41%, despite a tighter tolerance on the supply.

A constant-current driver such as the MAX6966 (a 10-port - LED driver and I/O expander with PWM intensity control) correctly regulates its constant-current outputs, provided that the voltage drop across the port output is above the minimum specified (**Figure 2**). The port's output voltage is the difference between the load's supply voltage (typically for an LED) and the voltage across the load (the LED forward voltage). If a drop in the LED supply causes the port output voltage to fall below the minimum, the result is a brownout in the driver-output stages, which causes the load current to fall. Minimum port voltage for the MAX6966 is 0.5V at 10mA sink current, and 1V at 20mA sink current.

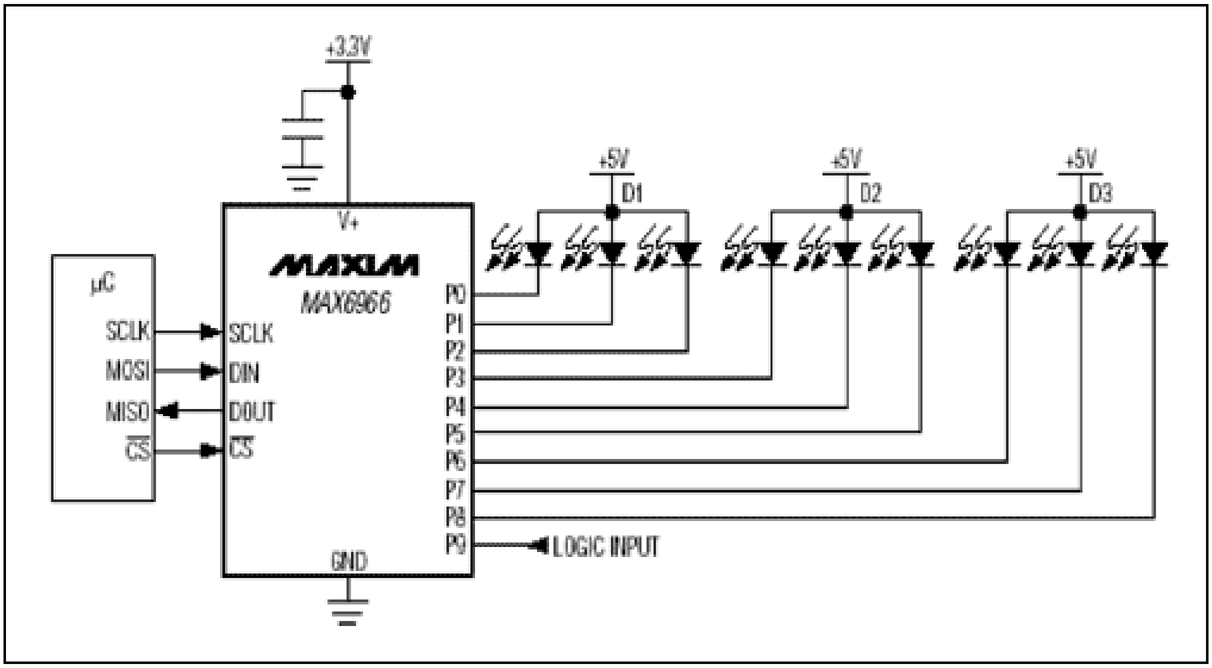


Figure 2. The MAX6966 GPIO IC contains LED drivers and logic I/Os.

Operating LEDs directly from a cell-phone battery saves the space and cost of a boost converter. The typical LED supply is, therefore, a rechargeable lithium battery with maximum terminal voltage of 4.2V when charged, 3.4V to 3.7V most of the time, and about 3V when discharged. When the battery approaches its end of life, the LED supply falls substantially below the brownout point.

Figure 3 shows the typical current sunk by a 3V blue LED (LITEON LTST-C170TBKT) as the LED supply voltage is varied from 2.5V to 7V. The LED currents shown are driven by ports programmed for 10mA and 20mA constant currents, and swept over a supply-voltage range of 2.5V to 7V. You can see that the LED forward voltage falls with current, allowing the LED current in brownout to fall gracefully rather than abruptly. The LED current drops to 6mA or 7mA for an LED supply voltage of 3V, which is acceptable performance for many backlight applications at the battery's end-of-life.

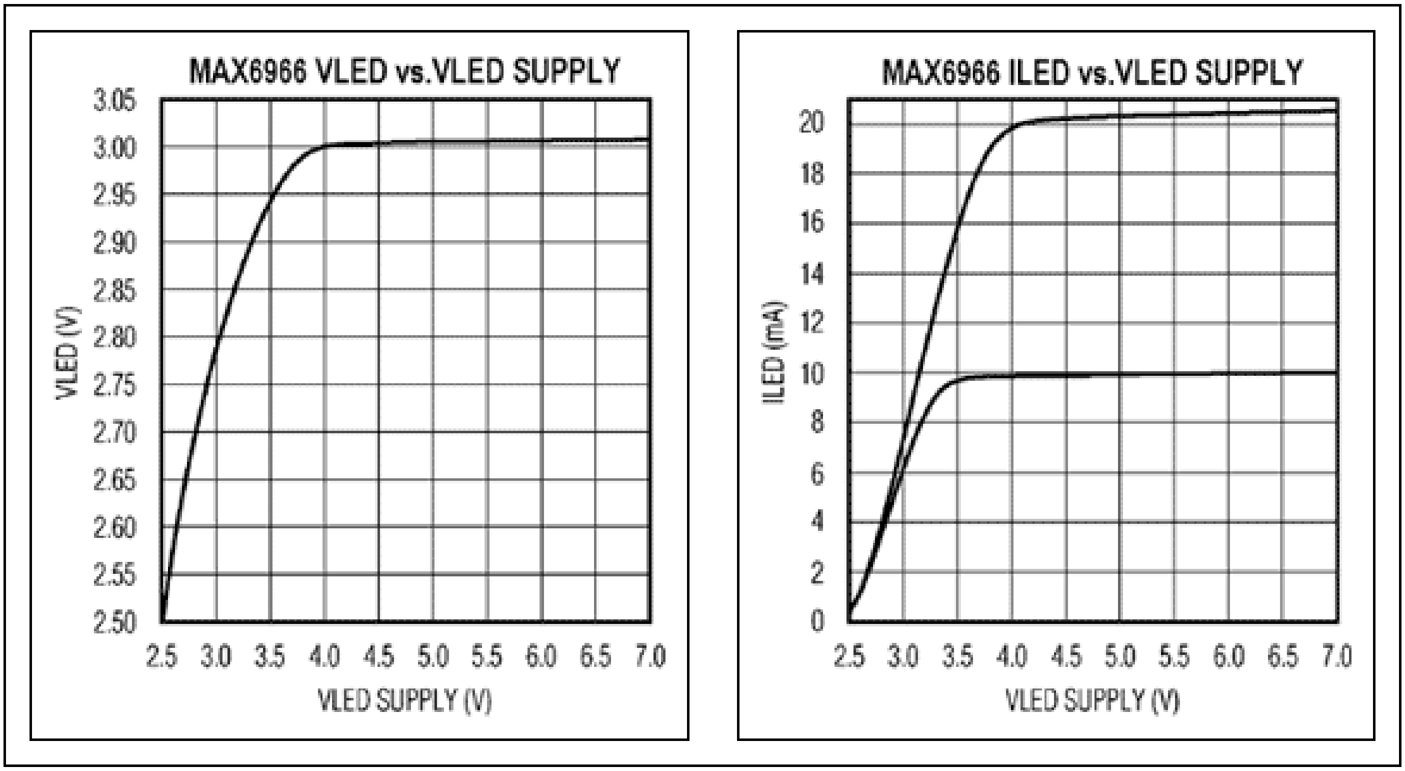


Figure 3. MAX6966 constant-current outputs drive white LEDs directly from a cell-phone battery.

Smoothing LED-Current Demand

Conventional PWM control for LEDs applies the same PWM period to all PWM outputs; all outputs switch on at the same time

(**Figure 4**). As a consequence, the LED driver outputs draw load currents at exactly the same time for a given PWM setting. If, for example, all outputs are set for a 50:50 duty cycle, the current draw is zero (all loads off) for half the time, and full scale (all loads on) for the other half.

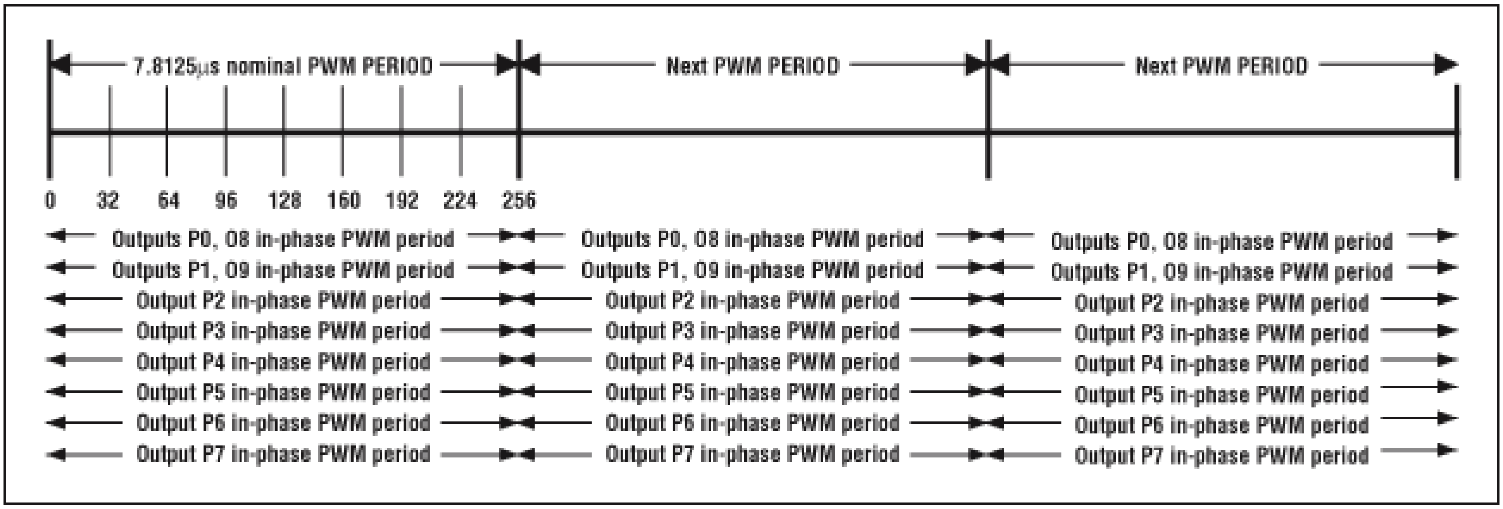

7.8125µs nominal PWM PERIOD
Next PWM PERIOD
Next PWM PERIOD
0
32
64
96
128
160
192
224
256
Outputs P0, O8 in-phase PWM period
Outputs P1, O9 in-phase PWM period
Output P2 in-phase PWM period
Output P3 in-phase PWM period
Output P4 in-phase PWM period
Output P5 in-phase PWM period
Output P6 in-phase PWM period
Output P7 in-phase PWM period

Figure 4. Conventional PWM causes all port outputs to switch at the same time.

The constant-current LED driver, the MAX6966, distributes port-output switching points across the PWM period by staggering the PWM timing of its port outputs by 1/8th increments of the PWM period (**Figure 5**). Staggering reduces the peak/mean current requirement by reducing the supply's di/dt output-switching transient. It also reduces EMI, and allows narrower PCB traces in the supply.

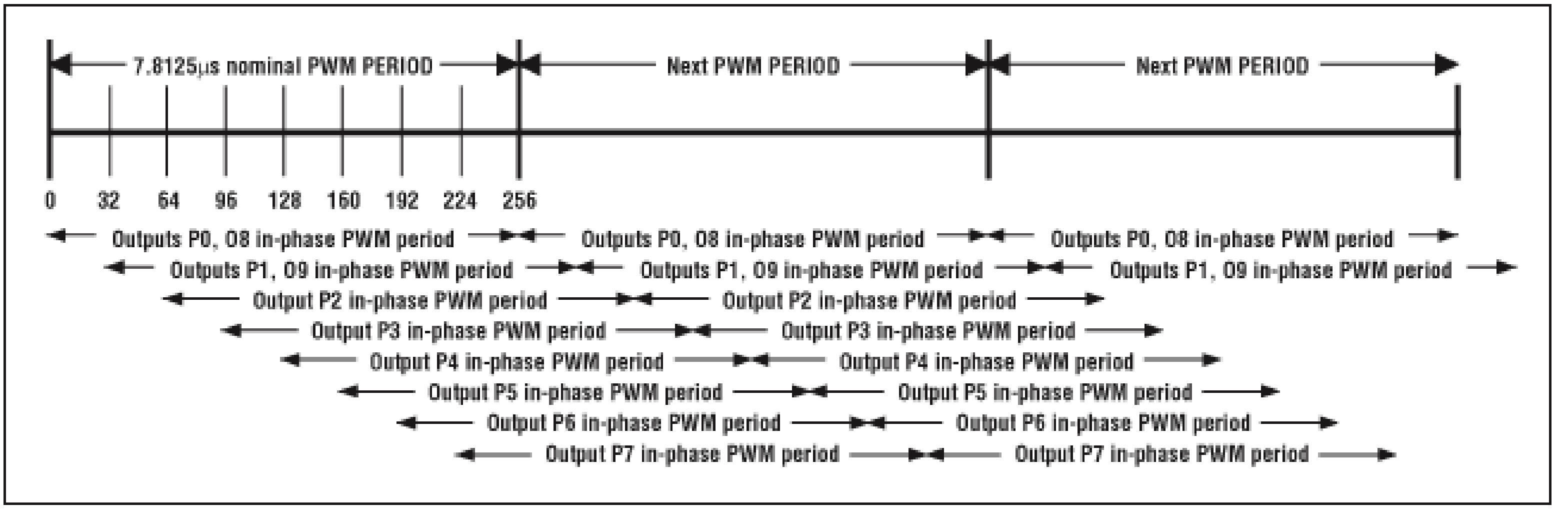
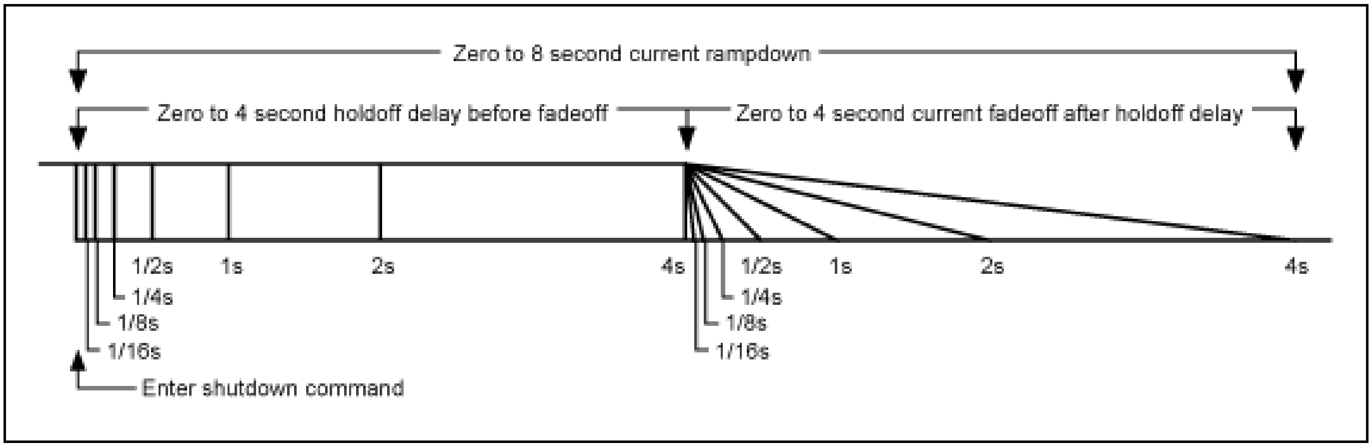

7.8125µs nominal PWM PERIOD
Next PWM PERIOD
Next PWM PERIOD
0
32
64
96
128
160
192
224
256
Outputs P0, O8 in-phase PWM period
Outputs P1, O9 in-phase PWM period
Output P2 in-phase PWM period
Output P3 in-phase PWM period
Output P4 in-phase PWM period
Output P5 in-phase PWM period
Output P6 in-phase PWM period
Output P7 in-phase PWM period

Figure 5. The MAX6966 flattens current demand by staggering the PWM start times.

Automating LED-Fade Effects

The MAX6966 provides automatic controls that ramp down the current outputs for automatic shutdown (ramp-down), and then ramp up again from shutdown without further interaction (**Figures 6** and **7**). Ramp-down includes a programmable hold-off delay that maintains the outputs at full current for an interval before the programmed fade-off interval, during which the current outputs are ramped down.



Zero to 8 second current rampdown
Zero to 4 second holdoff delay before fadeoff
Zero to 4 second current fadeoff after holdoff delay
1/2s
1s
2s
4s
1/4s
1/8s
1/16s
1/2s
1s
2s
4s
1/4s
1/8s
1/16s
Enter shutdown command

Figure 6. The MAX6966 automates LED intensity ramp-down, hold-off, and fade-off.

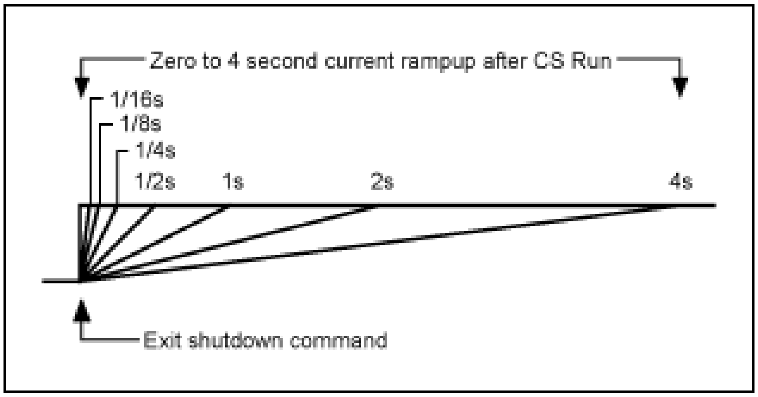


Figure 7. The MAX6966 can be hardware-triggered to control the rate at which the LED intensity ramps up, thereby allowing time to disconnect unwanted peripherals.

While in shutdown, the MAX6966 can be activated by a short pulse at the CS input. This hardware wakeup allows a power-management controller or equivalent ASIC to enable the MAX6966 with preconfigured LED-intensity settings. When coming out of shutdown, the LED outputs can then be ramped up to their programmed currents automatically (Figure 7). This design approach removes a difficult time-management problem from the system processor; it also allows the system to go into standby mode while the LED driver performs the timing functions itself.

With many GPIO devices providing high levels of source and sink current, those devices can turn many peripherals on and off through the direct application of power. There are some application tips to be learned from this realization:

- Wherever possible, control the peripheral's GND pin rather than the positive supply pin. Loads that can easily be ground-switched include vibrator motors, LEDs, and many IrDA interface modules. Ground switching is preferred because high-current, open-drain ports are commonly available.
- Use push-pull GPIOs (MAX7310, MAX7312, MAX7320) to control the positive supply pin of peripherals that demand less supply current. This approach is commonly used in powering down an IC.
- Use an external PFET to extend the high-side current drive of a push-pull GPIO. Simply connect the PFET gate to the push-pull port output, its source to the positive supply, and its drain to the load. Note that the control logic is now inverted: logic low at the port output turns the PFET on.
- Open-drain ports can be paralleled for higher current drive.
- Ensure that ports share the load by turning on and off together.

Port Expanders—Simple I/Os that Can Level-Translate

GPIOs have another important attribute: they translate inputs and outputs to and from higher or lower voltages. Some of these useful level-translation possibilities are listed below:

- Input ports can usually tolerate voltages higher than the GPIO's operating supply voltage. This capability allows, for example, the GPIO to operate from a conveniently low supply voltage while monitoring logic inputs at higher voltages.
- The I²C interface tolerates overvoltages to 5.5V, regardless of the GPIO's operating supply voltage. Thus, an I²C bus with pullup resistors terminating to 3.3V can communicate with a GPIO operating from 2.5V or 1.8V.
- Open-drain I/Os usually tolerate overvoltage to a level that is independent of the GPIO's operating supply voltage. For example, the I/O ports of the MAX6964/MAX6965 and MAX7313-MAX7316 families tolerate overvoltages to 5.5V or 7V. Using a pull-up resistor from the port to the desired rail, any port can therefore generate logic swings up to that limit.

- Open-drain I/Os and the I²C interface are usually hot-swappable, meaning that those connections withstand applied voltages without drawing parasitic currents when the GPIO's operating supply voltage is disconnected. Hot-swap capability is useful when connecting to accessories that can be powered independently of the phone. That is, the phone and accessory can have their power applied in either order.
- The default logic level of a push-pull I/O port at power-up can be set with a high-value pullup resistor (~1MΩ) to the supply voltage, or with a pulldown resistor to ground. I/O ports default as high-impedance inputs on power-up, so the resistor sets the initial logic level until the GPIO is programmed through its serial interface.

Port Expansion: Automatic Input Monitoring

As discussed earlier, peripherals should be interrupt-driven by events to save CPU polling. That is particularly true for GPIO inputs that monitor infrequent events such as the flip-shut switch in a cell phone or the power-fail warning. Many GPIOs include transition-detection circuits that monitor all logic inputs, and generate an interrupt when any input changes state. The MAX7319, an I²C port expander with eight open-drain I/Os, extends the traditional transition detection as described below:

- The interrupt output active-low INT is latched, so a transient change causes an interrupt assertion until the MAX7319 is read.
- A change flag register identifies any ports that have changed since the MAX7319 was last read, even if due to a transient.
- An interrupt-masks register allows only specific input ports to trigger the interrupt when they change.

The change flag of each port is set when the port's input changes; the flag remains set even if the input returns to its original state. Each port's interrupt mask determines whether a change on that input port causes an interrupt. When a response to input changes must be acted on quickly, you can use the interrupt masks to select a minimal set of priority inputs to be interrupt-enabled. Less important inputs can be polled. The change flags always show whether a permanent or transient change has occured on those inputs since they were last checked.

Mylar is a registered trademark of E. I. Du Pont De Nemours and Company Corp.

Application Note 3814: www.maxim-ic.com/an3814

More Information
For technical support: www.maxim-ic.com/support
For samples: www.maxim-ic.com/samples
Other questions and comments: www.maxim-ic.com/contact

Automatic Updates
Would you like to be automatically notified when new application notes are published in your areas of interest? Sign up for EE-Mail™.

Related Parts

MAX6965:	QuickView -- Full (PDF) Data Sheet -- Free Samples
MAX6966:	QuickView -- Full (PDF) Data Sheet -- Free Samples
MAX7310:	QuickView -- Full (PDF) Data Sheet
MAX7312:	QuickView -- Full (PDF) Data Sheet -- Free Samples
MAX7313:	QuickView -- Full (PDF) Data Sheet
MAX7314:	QuickView -- Full (PDF) Data Sheet
MAX7315:	QuickView -- Full (PDF) Data Sheet
MAX7316:	QuickView -- Full (PDF) Data Sheet
MAX7319:	QuickView -- Full (PDF) Data Sheet
MAX7320:	QuickView -- Full (PDF) Data Sheet

AN3814, AN 3814, APP3814, Appnote3814, Appnote 3814