

Keywords: 1-Wire switches, pushbutton switches, microcontrollers

APPLICATION NOTE 4849

Sense Multiple Pushbuttons Using Only Two Wires

By: Bernhard Linke, Principal Member Technical Staff

Feb 10, 2011

Abstract: For applications that require less than a keyboard or numeric keypad, this simple circuit monitors multiple pushbutton switches using a single multi-channel 1-Wire® addressable switch (DS2408), and a few external components.

A similar version of this article appeared in the September 9, 2010 issue of *EDN* magazine.

Keyboards and numeric keypads often provide the user interface for electronic equipment, but many applications require only a small number of pushbuttons. For those, you can monitor multiple pushbuttons over a single pair of wires (**Figure 1**).

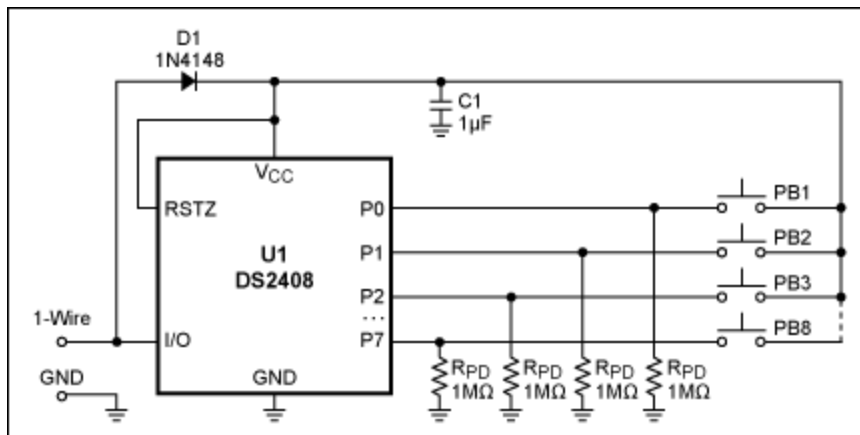


Figure 1. Connected to a microcontroller, this circuit can monitor 8 pushbuttons using only two wires.

The multichannel, 1-Wire addressable switch (U1, [DS2408](#)) provides input/output ports P0–P7, which in this application serve as inputs. The high-valued resistors (R_{PD}) connect these ports to ground to ensure a defined logic state (0), and D1–C1 forms a local power supply that steals energy from the 1-Wire communication line. When you press a pushbutton, it connects the corresponding port to the local supply voltage, which is equivalent to logic 1. This change of state sets the port's activity latch (see data sheet).

Conditional Search Polling

As a 1-Wire slave device, U1 does not initiate communications. Instead, the master (typically a microcontroller) polls the 1-Wire line. To minimize overhead, U1 supports a 1-Wire network function

called "conditional search." Before using that function, however, you must configure U1 according to the needs of the application. That includes defining the qualifying input ports (channel selection), specifying the polarity of the qualifying ports (channel polarity selection), choosing between pin or activity latch for the port, and specifying whether the device shall respond to activity at a single port (OR) or activity at all ports (AND).

Configuration Example

Consider, for example, that U1 is to perform a conditional search if activity is detected at any of the eight ports. This condition requires the following configuration settings: Channel Selection Mask (address 008Bh) = 11111111b (1 indicates that a port is selected); Channel Polarity Selection Register (address 008Ch) = 11111111b (1 indicates that the level must be high); Control/Status Register (address 008Dh) = 00000001b (selects the port's activity latch as source, and specifies OR as the conditional search term, i.e., activity at a single port).

After power-up, the configuration data must be loaded into U1 using the Write Conditional Search Register command. To function as inputs, the PIO output latches must be set to 1. The Channel Access Write Command (with FFh as PIO output data byte) defines the ports as inputs. Subsequently, the issue of a Reset Activity Latches command completes the configuration. U1 is now ready to handle pushbutton activity.

Software Outline

After U1 is configured, the application software on power-up enters an endless loop, in which a 1-Wire Reset is followed by a Conditional Search command. With no pushbutton activity there is no response from U1, as indicated by a "logic 1" for the two bits immediately after the conditional-search command code. In that case the microcontroller cancels the conditional search and starts over.

If U1 responds to the conditional search, the first two bits will be 1 and 0, representing the least significant bit of the device's family code (29h) in its true and inverted forms. In that case the microcontroller should complete the conditional search flow, which consists of a 192-bit sequence (see data sheet). Next, the microcontroller reads the register's data by issuing a Read PIO Registers command using the address of the PIO Activity Latch State Register (address 008Ah). The microcontroller then issues a 1-Wire reset, followed by the commands Resume and Reset Activity Latches. It then returns to the endless loop, polling for the next pushbutton event.

If U1 responds and no other 1-Wire slave is connected, the microcontroller could cancel the conditional search after reading the first 2 bits, issue a 1-Wire reset followed by a Skip ROM command, and then read the PIO Activity Latch State Register. Next, it must issue a 1-Wire reset followed by the commands Skip ROM and Reset Activity Latches, before returning to the endless loop.

Which Button was Pressed?

The code read from the PIO Activity Latch State Register contains the answer. If PB1 was pressed, the data is 00000001b, for PB2 it is 00000010b, etc. At least one of the 8 bits will be 1. If several buttons were pressed since the last Reset Activity Latches command, several bits will be 1. The application software must then decide whether such a condition is valid or not. (In the simplest case (1-of-8 code), the software considers all codes that have more than one bit set as invalid.)

More than 8 Pushbuttons

This concept can be expanded to more than 8 pushbuttons. Instead of associating one pushbutton with one port, you can associate additional pushbuttons with two ports that are activated simultaneously (2-of-8 code). These additional pushbuttons (PBn) are connected as shown in **Figure 2** ($x, y = 0$ to 7, and $x \neq y$). If Px or Py is activated by another pushbutton, the diodes prevent that activity from propagating to other ports. Again, the application software must check the code read from the PIO Activity Latch State Register to decide whether it is valid. The theoretical limit of this concept is 255 pushbuttons, which requires combinations of 2, 3, 4, 5, 6, 7, or 8 diodes per additional pushbutton. When the cost of diodes for each additional pushbutton begins to exceed the benefits, you will find that adding another DS2408 is more cost effective.

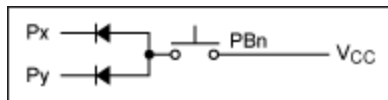


Figure 2. Additional pushbuttons (up to 28) can be monitored if they are connected to two ports using diodes.

Reference

DS2408 data sheet, Maxim Integrated Products Inc., Dallas, TX.

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

Related Parts

[DS2408](#)

1-Wire 8-Channel Addressable Switch

[Free Samples](#)

More Information

For Technical Support: <http://www.maximintegrated.com/support>

For Samples: <http://www.maximintegrated.com/samples>

Other Questions and Comments: <http://www.maximintegrated.com/contact>

Application Note 4849: <http://www.maximintegrated.com/an4849>

APPLICATION NOTE 4849, AN4849, AN 4849, APP4849, Appnote4849, Appnote 4849, DI #844

Copyright © by Maxim Integrated Products

Additional Legal Notices: <http://www.maximintegrated.com/legal>