Keywords: PDA, multi-functional, gaming key, gaming control, key switch controller, add-on, smart phone, keypad control, auto-increment, key release detection, GPO

May 30, 2007

APPLICATION NOTE 4054

Using the MAX7359 for Multifunctional and Gaming Keys on PDAs/Smartphones: A Programming Guide

Abstract: The keypad control properties of cell phone chipsets are often stretched to their limits when an operating system or application needs to detect multiple keypresses (e.g. Ctrl-Alt-Del and gaming controls). The MAX7359 is an ideal add-on key-switch controller for implementing multifunctional and gaming keys on a PDA/smartphone QWERTY keypad. The MAX7359 can be connected to the cell phone chipset through an I²C interface, and simple program coding can be used to select a set of desired features.

Overview of the MAX7359

The MAX7359 is a low-power-consumption, special-purpose key-switch controller suitable for cell phones, printers, and other portable applications. Up to 64 keys can be implemented with this device. Separate keypress and release codes are assigned for every key. Multiple keys can be pressed simultaneously and/or held and released in different orders. Up to 16 keypress and release entries can be held in a FIFO register. Key-activity information collected by the MAX7359 is read through a simple I²C interface at one keypress or release entry per byte. To enhance the device's noise immunity, the detection of a keypress can be debounced. In other words, each pressed key is scanned twice within a very short time interval before it is detected.

To reduce power consumption, the MAX7359 consumes only 1.2µA in sleep mode while waiting for key activities. Upon a keypress, the controller wakes up in less than 200µs to collect the keypress/release information. After a specified key-activity idle time of between 0.256s and 8s, the device re-enters sleep mode. Note that key FIFO information can be accessed even in sleep mode. To relieve the host from permanent attention, an interrupt signal can be generated once a key is pressed or when the FIFO has reached a predefined amount of entries. The interrupt signal can be cleared by reading the device through the I²C interface or when the FIFO is emptied.

The MAX7359 is an enhanced version of the MAX7349 with a 1.8V to 3.3V supply-voltage range. **Figure 1** shows the typical connection of the MAX7359 to a host through the I²C interface. Because of the open-drain ports on the I²C interface, the MAX7359 is capable of interfacing to a host with a different supply voltage.

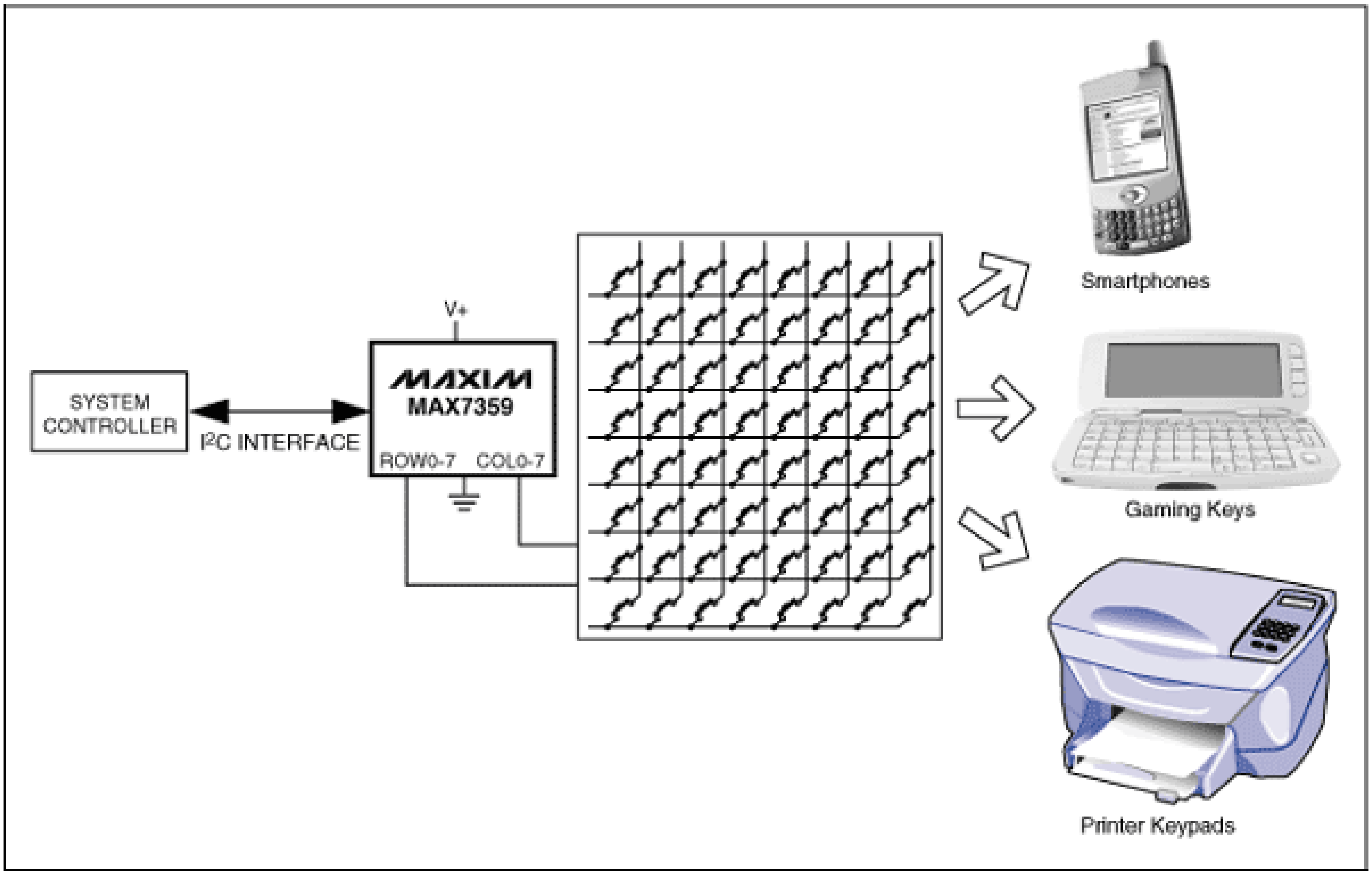

Figure 1. Connecting the MAX7359 through an I²C interface.

Programming the MAX7359

Key-activity information is contained in the FIFO, and the desired operational features are defined by other registers inside the controller. The register address of the FIFO is 0x00. The configuration register's address is 0x01, while the rest of the operational features are defined by registers with addresses from 0x02 through 0x06.

The contents of a MAX7359 register can be specified using an I²C write command and verified using an I²C read command. An I²C write command starts with the device address of the controller, which can correspond to 0x70, 0x74, 0x78, or 0x7C depending on the specific connection of the AD0 pin, followed by the register address. Following the register addresses, there may be some data bytes. If there is only one data byte present, it will be stored in the register as specified by the preceding byte. When there is more than just one data byte, the first byte is stored in the register as specified, and the next byte is stored in the register whose address is one number higher, and so on. This operation is based on the controller's register address autoincrement feature. In other words, a write command with 0x70, 0x01, 0x0A, and 0x00 will store the 0x0A byte in register 0x01 and the byte 0x00 in register 0x02. The register address autoincrement feature applies to all writable registers for both write and read commands, with the exception of the FIFO address 0x00. An I²C write command with no data bytes following the register address is normally used to set the register address for the next read command.

An I²C read command starts with the device address of the MAX7359, and is followed by one or more data bytes. When there is only one data byte, data is retrieved from the register specified by the preceding write command with no data byte. Otherwise, it is retrieved from the last register accessed by a write or read command. When there is more than one data byte, the first byte is retrieved from the register as specified, and the register address autoincrement mechanism applies to the rest of the data bytes, except for the FIFO register. In other words, repeated reading of FIFO register 0x00 does not require address resetting.

During power-on reset, the MAX7359 is readied for operation with key-release detection, keypress wake-up, and autoshutdown features enabled (default settings). Additionally, there are only two columns active for key-switch control and a total of 16 available keys. The rest of the six columns/GPO (general-purpose output) ports are in GPO mode at logic high. The following I²C command structure can be used to activate the six columns/GPO ports for key-switch control and a total of 64 available keys.

```
// A Write Command to disable GPO ports
0x70            // MAX7359 device address
0x02            // GPO enable and debounce register
0x00            // Disable GPO ports and 9ms debounce time
```

The following I²C commands can be used to read a FIFO entry:

```
// A write command to set the register address to 0x00 and a read command from the FIFO
// A write 0 data byte to address 0x00 command
0x70            // MAX7359 device address

0x00            // FIFO register
// A read one byte from FIFO command
0x71            // MAX7359 device address
0xXX            // A data byte from the FIFO. The value depends on what is there
```

The following I²C commands can be used to enable operations of the MAX7359 to monitor 64 keys and to send an interrupt signal when a key is pressed. The interrupt signal is cleared once the MAX7359 is read through the I²C interface.

```
// Initialization
More = 0x80                             // More keys in the FIFO mask
Key = 0x00                              // Key code variable
0x70, 0x02, 0x00                        // Disable GPO ports
0x70, 0x03, 0x02                        // Enable interrupt upon a keypress
0x70, 0x01, 0x2A                        // Enable interrupt cleared once read

// When an interrupt is received
0x70, 0x00                      // Set the register address to 0x00
Loop:   0x71, 0xXX              // Read the FIFO register
Key = 0xXX                      // Assign the key code to a variable
Save the key code               // Save the key code for application
If (Key | More) go to Loop      // If not the last entry, read more key codes
```

Table 1. Keypress Codes Last FIFO Entry

	Col. 0	Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6	Col. 7
Row 0	0x00	0x08	0x10	0x18	0x20	0x28	0x30	0x38
Row 1	0x01	0x09	0x11	0x19	0x21	0x29	0x31	0x39
Row 2	0x02	0x0A	0x12	0x1A	0x22	0x2A	0x32	0x3A
Row 3	0x03	0x0B	0x13	0x1B	0x23	0x2B	0x33	0x3B
Row 4	0x04	0x0C	0x14	0x1C	0x24	0x2C	0x34	0x3C
Row 5	0x05	0x0D	0x15	0x1D	0x25	0x2D	0x35	0x3D
Row 6	0x06	0x0E	0x16	0x1E	0x26	0x2E	0x36	0xBE*
Row 7	0x07	0x0F	0x17	0x1F	0x27	0x2F	0x37	0xBF*

* Read one more to see if the FIFO is empty.

Table 2. Key Release Codes Last FIFO Entry

	Col. 0	Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6	Col. 7
Row 0	0x40	0x48	0x50	0x58	0x60	0x68	0x70	0x78
Row 1	0x41	0x49	0x51	0x59	0x61	0x69	0x71	0x79
Row 2	0x42	0x4A	0x52	0x5A	0x62	0x6A	0x72	0x7A
Row 3	0x43	0x4B	0x53	0x5B	0x63	0x6B	0x73	0x7B
Row 4	0x44	0x4C	0x54	0x5C	0x64	0x6C	0x74	0x7C
Row 5	0x45	0x4D	0x55	0x5D	0x65	0x6D	0x75	0x7D
Row 6	0x46	0x4E	0x56	0x5E	0x66	0x6E	0x76	0xFE*
Row 7	0x47	0x4F	0x57	0x5F	0x67	0x6F	0x77	0xFF*

* Read one more to see if the FIFO is empty.

Table 3. Keypress Codes More in the FIFO

	Col. 0	Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6	Col. 7
Row 0	0x80	0x88	0x90	0x98	0xA0	0xA8	0xB0	0xB8
Row 1	0x81	0x89	0x91	0x99	0xA1	0xA9	0xB1	0xB9
Row 2	0x82	0x8A	0x92	0x9A	0xA2	0xAA	0xB2	0xBA
Row 3	0x83	0x8B	0x93	0x9B	0xA3	0xAB	0xB3	0xBB
Row 4	0x84	0x8C	0x94	0x9C	0xA4	0xAC	0xB4	0xBC
Row 5	0x85	0x8D	0x95	0x9D	0xA5	0xAD	0xB5	0xBD
Row 6	0x86	0x8E	0x96	0x9E	0xA6	0xAE	0xB6	0xBE*
Row 7	0x87	0x8F	0x97	0x9F	0xA7	0xAF	0xB7	0xBF*

* Read one more to see if the FIFO is empty.

Table 4. Key Release Codes More in the FIFO

	Col. 0	Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6	Col. 7
Row 0	0xC0	0xC8	0xD0	0xD8	0xE0	0xE8	0xF0	0xF8
Row 1	0xC1	0xC9	0xD1	0xD9	0xE1	0xE9	0xF1	0xF9
Row 2	0xC2	0xCA	0xD2	0xDA	0xE2	0xEA	0xF2	0xFA
Row 3	0xC3	0xCB	0xD3	0xDB	0xE3	0xEB	0xF3	0xFB
Row 4	0xC4	0xCC	0xD4	0xDC	0xE4	0xEC	0xF4	0xFC
Row 5	0xC5	0xCD	0xD5	0xDD	0xE5	0xED	0xF5	0xFD
Row 6	0xC6	0xCE	0xD6	0xDE	0xE6	0xEE	0xF6	0xFE*
Row 7	0xC7	0xCF	0xD7	0xDF	0xE7	0xEF	0xF7	0xFF*

* Read one more to see if the FIFO is empty.

The key code 0x3F is reserved for FIFO empty.
The key code 0x7F is reserved for FIFO overflow indication.
The key code 0x3E is reserved for a key repeat and it is the last FIFO entry.
The key code 0x7E is reserved for a key repeat with more in the FIFO.

Application Note 4054: http://www.maxim-ic.com/an4054

More Information

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

Related Parts

MAX7349: QuickView -- Full (PDF) Data Sheet -- Free Samples

MAX7359: QuickView