

### General Description

The MIC74 is a fully programmable serial-to-parallel I/O expander compatible with the SMBus™ (system management bus) protocol. It acts as a “slave” on the bus, providing eight independent I/O lines.

Each I/O bit can be individually programmed as an input or output. If programmed as an output; each I/O bit can be programmed as an open-drain or complementary push-pull output. If desired, the four most significant I/O bits can be programmed to implement fan speed control. An internal clock generator and state machine eliminate the overhead generally associated with “bit-banging” fan speed control.

Programming the device and reading/writing the I/O bits is accomplished using seven internal registers. All registers can be read by the host. Output bits are capable of directly driving high-current loads such as LEDs. A separate interrupt output can notify the host of state changes on the input bits without requiring the MIC74 to perform a transaction on the serial bus or be polled by the host. Three address selection inputs are provided, allowing up to eight devices to share the same bus and provide a total of 64 bits of I/O.

The MIC74 is available in an ultra-small-footprint 16-pin QSOP. Low quiescent current, small footprint, and low package height make the MIC74 ideal for portable and desktop applications.

Data sheets and support documentation can be found on Micrel’s web site at [www.micrel.com](http://www.micrel.com).

### Features

- Provides eight bits of general purpose I/O
- Built in fan speed control logic (optional)
- 2-wire SMBus™/I<sup>2</sup>C™ compatible serial interface plus interrupt output
- 2.7V to 3.6V operating voltage range
- 5V-tolerant I/O
- Low quiescent current: 2µA (typical)
- Bit-programmable I/O options:
  - input or output
  - push-pull or open-drain output
  - interrupt on input changes
- Outputs can directly drive LEDs (10mA I<sub>OL</sub>)
- Up to 8 devices per bus

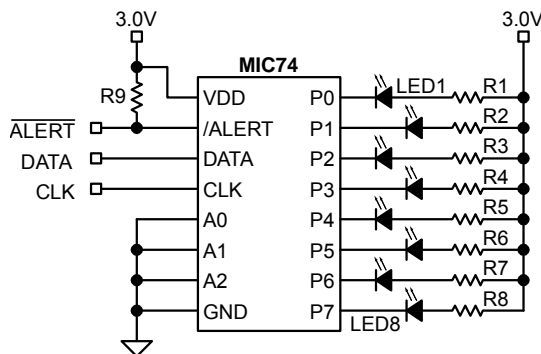
### Applications

- General purpose I/O expansion via serial bus
- Personal computer system management
- Distributed sensing and control
- Microcontroller I/O expansion
- Fan Control

### Ordering Information

Part Number		Temperature Range	Package
Standard	Pb-Free		
MIC74BQS	MIC74YQS	-40° to +85°C	16-Pin QSOP

### Typical Application

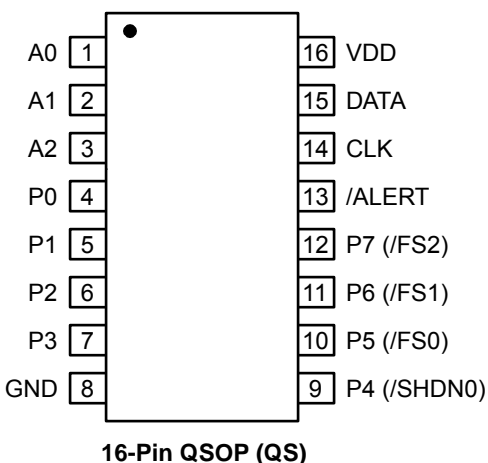


Serial-Bus-Controlled LED Annunciator

SMBus is a trademark of Intel Corporation. I<sup>2</sup>C is a trademark of Phillips Electronics N.V.

Micrel Inc. • 2180 Fortune Drive • San Jose, CA 95131 • USA • tel +1 (408) 944-0800 • fax + 1 (408) 474-1000 • <http://www.micrel.com>

## Pin Configuration



## Pin Description

Pin Number	Pin Name	Pin Function
1 – 3	A0 – A2	Address (Input): Slave address selection inputs; sets the three least significant bits of the MIC74's slave address.
4 – 7	P0 – P3	Parallel I/O (Input/Output): General-purpose I/O pin. Direction and output type are user programmable.
8	GND	Ground
9 – 12	P4 – P7 (/SHDN, /FS0 – FS2)	Parallel I/O (Input/output): P4–P7 are general-purpose I/O pins. Direction and output type are user programmable. Shutdown (Output): When the FAN bit is set, pin 9 becomes SHDN. Fan Speed (Output): When the FAN bit is set, pins 10 through 12 become /FS0–/FS2 respectively, controlled by the FAN_SPEED register.
13	/ALERT	Interrupt (Output): Active-low, open-drain output signals input-change-interrupts to the host on this pin. Signal is cleared when the bus master (host) polls the ARA (alert response address = 0001 100) or reads status.
14	CLK	Serial Bus Clock (Input): The host provides the serial bit clock in this input.
15	DATA	Serial Data (Input/Output): Serial data input and open-drain serial data output.
16	VDD	Power Supply (Input).

**Absolute Maximum Ratings<sup>(1)</sup>**

Supply Voltage ( $V_{DD}$ ).....	+4.6V
Input Voltage [all pins except $V_{DD}$ and GND] ( $V_{IN}$ ).....	GND – 0.3V to 5.5V
Junction Temperature ( $T_J$ ) .....	+150°C
Lead Temperature (soldering, 10 sec.).....	+260°C
EDS Rating <sup>(3)</sup>	
$V_{DD}$ .....	1.5kV
A0, A1, A2.....	500V
Others.....	200V

**Operating Ratings<sup>(2)</sup>**

Supply Voltage ( $V_{DD}$ ).....	+2.7V to +3.6V
Ambient Temperature ( $T_A$ ) .....	–40°C to +85°C
Package Thermal Resistance .....	163°C/W

**Electrical Characteristics**

2.7V  $\leq$   $V_{DD}$   $\leq$  3.6V;  $T_A$  = 25°C, **bold** values indicate –40°C  $\leq$   $T_A$   $\leq$  +85°C, unless noted.

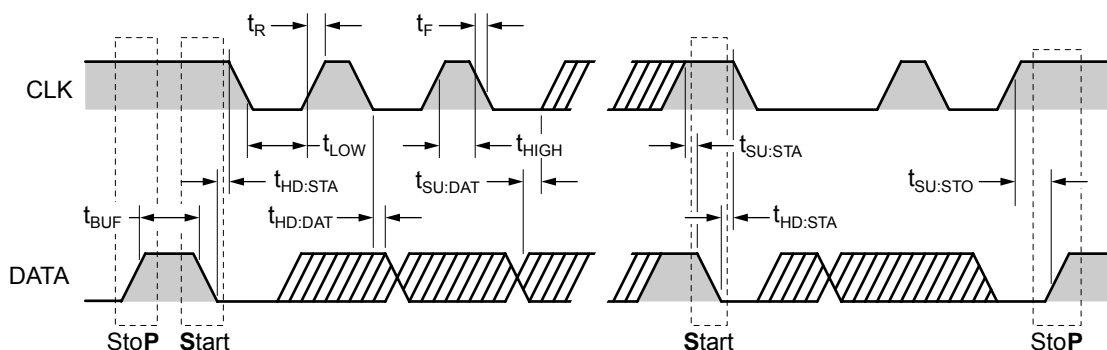
Symbol	Parameter	Condition	Min	Typ	Max	Units
$V_{IN}$	Input Voltage, any pin except $V_{DD}$ and GND		GND–0.3		5.5	V
$I_{DD}$	Operating Supply Current	P[7:0] inputs; P[7:0] = $V_{DD}$ or GND /ALERT open; $f_{CLK}$ = 100kHz		2	6	$\mu$ A
$I_{START}$	Fan Startup Supply Current (Fan Mode Only)	during $t_{START}$ ; /ALERT, /SHDN, /FS2[2:0] = open; $V_{SMBCLK}$ = $V_{SMBDATA}$ = $V_{DD}$ ; P[3:0] = inputs			1.75	mA
$I_{STBY}$	Standby Supply Current	/ALERT = open, $V_{SMBCLK}$ = $V_{SMBDATA}$ = $V_{DD}$ ; P[3:0] = inputs		1	3	$\mu$ A
<b>Serial I/O (DATA, CLK)</b>						
$V_{IL}$	Input Low Voltage		<b>–0.3</b>		<b>0.8</b>	V
$V_{IH}$	Input High Voltage		<b>2</b>		<b>5.5</b>	V
$V_{OL}$	Output Low Voltage	$I_{OL}$ = 3mA			<b>0.4</b>	V
$I_{LEAK}$	Leakage Current	$V_{IN}$ = 5.5V or GND	–1		+1	$\mu$ A
$C_{IN}$	Input Capacitance			10		pF
<b>Parallel I/O [P0–P3, P4(/SHDN), P5(/FS0)–P7(/FS2)]</b>						
$V_{IL}$	Input Low Voltage		<b>–0.5</b>		<b>0.8</b>	V
$V_{IH}$	Input High Voltage		<b>2</b>		<b>5.5</b>	V
$I_{OL}$	Output Low Current	$V_{OL}$ = 0.4V, $V_{DD}$ = 2.7V	<b>7</b>			mA
		$V_{OL}$ = 1V, $V_{DD}$ = 3.3V	<b>10</b>			mA
$I_{OH}$	Output High Current	$V_{OH}$ = 2.4V	<b>7</b>			mA
$I_{LEAK}$	Leakage Current	$V_{IN}$ = 5.5V or GND	–1		+1	$\mu$ A
$C_{IN}$	Input Capacitance			10		pF
$C_{OUT}$	Output Capacitance			10		pF
<b>Address Input (A0–A2)</b>						
$V_{IL}$	Input Low Voltage		<b>–0.3</b>		<b>0.3</b> $V_{DD}$	V
$V_{IH}$	Input High Voltage		<b>0.7</b> $V_{DD}$		<b><math>V_{DD}+0.3</math></b>	V
$I_{LEAK}$	Leakage Current	$V_{IN}$ = $V_{DD}$ or GND	–250		+250	nA
<b>/ALERT</b>						
$V_{OL}$	Output Low Voltage	$I_{OL}$ = 1mA			<b>0.4</b>	V
$I_{LEAK}$	Leakage Current	$V_{IN}$ = $V_{DD}$ or $V_{SS}$	–1	$\pm$ 250	+1	$\mu$ A

Symbol	Parameter	Condition	Min	Typ	Max	Units
<b>AC Characteristics</b>						
$t_{START}$	Fan Startup Interval	normal operation	<b>0.5</b>	1	<b>3.3</b>	sec
$t_{PULSE}$	Minimum Pulse-Width	minimum pulse-width on Pn to generate an interrupt, <b>Note 7</b>	<b>10</b>			ns
$t_{INT}$	Interrupt Delay	interrupt delay from state change on Pn to /ALERT $\leq V_{OL}$ , <b>Note 7</b>			<b>4</b>	$\mu$ s
$t_{IR}$	Delay from Status Read or ARA Response to /ALERT $\geq V_{OH}$				<b>4</b>	$\mu$ s
$t_{HD:STA}$	Hold Time, <b>Note 7</b>	hold time after repeated start condition, after this period, the first clock is generated	<b>4</b>			$\mu$ s
$t_{SU:STA}$	Setup Time, <b>Note 7</b>	repeated start condition setup time	4.7			$\mu$ s
$t_{SU:STO}$	Stop Condition Setup Time	<b>Note 7</b>	<b>4</b>			$\mu$ s
$t_{HD:DAT}$	Data Hold Time	<b>Note 7</b>	500			ns
$t_{SU:DAT}$	Data Setup Time	<b>Note 7</b>	<b>0</b>			ns
$t_{TIMEOUT}$	Clock Low Time-Out	<b>Note 4, 7</b>	25	35		ms
$t_{LOW}$	Clock Low Period	<b>Note 5, 7</b>	4.7			$\mu$ s
$t_{HIGH}$	Clock High Period	<b>Note 5, 7</b>	<b>4</b>	50		$\mu$ s
$t_F$	Clock/Data Fall Time	<b>Note 6, 7</b>		300		ns
$t_R$	Clock/Data Rise Time	<b>Note 6, 7</b>		1000		ns
$t_{BUF}$	Bus free time between stop and Start condition	<b>Note 7</b>	4.7			$\mu$ s

**Notes:**

1. Exceeding the absolute maximum rating may damage the device.
2. The device is not guaranteed to function outside its operating rating.
3. Devices are ESD sensitive. Handling precautions recommended. Human body model, 1.5k in series with 100pF.
4. Devices participating in a transfer will timeout when any clock low exceeds the value of  $t_{TIMEOUT(min)}$  of 25ms. Devices that have detected a timeout condition must reset the communication no later than  $t_{TIMEOUT(max)}$  of 35ms. The maximum value specified must be adhered to by both a master and a slave as it incorporates the cumulative stretch limit for both a master (10ms) and a slave (25ms).
5.  $t_{HIGH(max)}$  provides a simple guaranteed method for devices to detect bus idle conditions.
6. Rise and fall time is defined as follows:  $t_R = V_{IL(max)} - 0.15V$  to  $V_{IH(min)} + 0.15V$ ;  $t_F = 0.9V_{DD}$  to  $V_{IL(max)} - 0.15V$ .
7. Guaranteed by design.

**Timing Definitions**



## Register Descriptions

### Device Configuration Register

DEV_CFG							
D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
Always write as zero						FAN	IE

Power-On Default Value: 0000 0000<sub>b</sub>, 00<sub>h</sub>  
 Interrupts disabled  
 Not in Fan Mode

Command\_byte address: 0000 0000<sub>b</sub>, 00<sub>h</sub>

Type: 8-bits, read/write

Bit Name: IE

Function: Global interrupt enable

Operation: 1 = enabled  
 0 = disabled

Bit Name: FAN

Function: Selects Fan Mode  
 (P[7:4] vs. /FS[2:0], /SHDN)

Operation: 1 = Fan Mode  
 0 = I/O Mode

Bit Name: D[2] through D[6]

Function: Reserved

Operation: Reserved—always write as zero

### Data Direction Register

DIR							
D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0

Power-On Default Value: 0000 0000<sub>b</sub>, 00<sub>h</sub>  
 all P<sub>n</sub>'s configured as inputs

Command\_byte address: 0000 0001<sub>b</sub>, 01<sub>h</sub>

Type: 8-bits, read/write

Bit Name: DIR<sub>n</sub>

Function: Selects data direction, input or output, of P<sub>n</sub>

Operation: 1 = output  
 0 = input

Notes: If Fan Mode is selected, that is, the FAN bit of the DEV\_CFG register is set to one, P[7:4] are automatically configured as open-drain outputs. They are then referred to as /FS[2:0] and /SHDN. The DIR register has no effect on these I/O bits while in Fan Mode.

### Output Configuration Register

OUT_CFG							
D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0

Power-On Default Value: 0000 0000<sub>b</sub>, 00<sub>h</sub>  
 all outputs open-drain

Command\_byte address: 0000 0010<sub>b</sub>, 02<sub>h</sub>

Type: 8-bits, read/write

Bit Name: OUT<sub>n</sub>

Function: Selects output driver configuration of P<sub>n</sub> when P<sub>n</sub> is configured as an output.

Operation: 1 = push-pull  
 0 = open-drain

Notes: If Fan Mode is selected, that is, the FAN bit of the DEV\_CFG register is set to one, P[7:4] are automatically configured as open-drain outputs. They are then referred to as /FS[2:0] and /SHDN. The OUT\_CFG register has no effect on these I/O bits while in Fan Mode.

### Status Register

STATUS							
D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
S7	S6	S5	S4	S3	S2	S1	S0

Power-On Default Value: 0000 0000<sub>b</sub>, 00<sub>h</sub>  
 no interrupts pending

Command\_byte address: 0000 0011<sub>b</sub>, 03<sub>h</sub>

Type: 8-bits, read only

Bit Name: S<sub>n</sub>

Function: Flag for P<sub>n</sub> input-change event when P<sub>n</sub> is configured as an input; S<sub>n</sub> is set when the corresponding input changes state.

Operation: 1 = change occurred  
 0 = no change occurred

Notes: If Fan Mode is selected, that is, the FAN bit of the DEV\_CFG register is set to one, P[7:4] are automatically configured as open-drain outputs. They are then referred to as /FS[2:0] and /SHDN. No interrupts of any kind are generated by these pins while in Fan Mode. All status bits are cleared after any read operation is performed on STATUS.

**Interrupt Mask Register**

INT_MASK							
D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0

Power-On Default Value: 0000 0000<sub>b</sub>, 00<sub>h</sub>

Command\_byte address: 0000 0100<sub>b</sub>, 04<sub>h</sub>

Type: 8-bits, read/write

Bit Name: IM<sub>n</sub>

Function: Interrupt enable bit for P<sub>n</sub> when P<sub>n</sub> is configured as an input

Operation: 1 = enabled  
0 = disabled

Notes: If Fan Mode is selected, that is, the FAN bit of the DEV\_CFG register is set to one, P[7:4] are automatically configured as open-drain outputs. They are then referred to as /FS[2:0] and /SHDN. No interrupts of any kind are generated by these pins while in Fan Mode.

**Data Register**

DATA							
D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
P7	P6	P5	P4	P3	P2	P1	P0

Power-On Default Value: 1111 1111<sub>b</sub>, FF<sub>h</sub>

Command\_byte address: 0000 0101<sub>b</sub>, 05<sub>h</sub>

Type: 8-bits, read/write

Bit Name: P<sub>n</sub>

Function: Returns the current state of any P<sub>n</sub> configured as an input and the last value written to P<sub>n</sub>'s configured as outputs; Writing the DATA register sets the output state of any P<sub>n</sub>'s configured as outputs; writes to I/O bits configured as inputs are ignored

Read Operation: 1 = P<sub>n</sub> is high  
0 = P<sub>n</sub> is low

Write Operation: 1 = P<sub>n</sub> is set to one  
0 = P<sub>n</sub> is cleared

Notes: If Fan Mode is selected, that is, the FAN bit of the DEV\_CFG register is set to one, P[7:4] are automatically configured as open-drain outputs. They are then referred to as /FS[2:0] and /SHDN. The state of these pins is determined by the FAN\_SPEED register. While in Fan Mode, D[7:4] of the DATA registers have no effect.

**Fan Speed Register**

FAN_SPEED							
D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
Always write as zero					Fan Speed		

Power-On Default Value: 0000 0000<sub>b</sub>, 00<sub>h</sub>  
fan off

Command\_byte address: 0000 0110<sub>b</sub>, 06<sub>h</sub>

Type: 8-bits, read/write

Bit Name: D[0] through D[2]

Function: Determines bit-pattern on FS[2:0]

Operation:

D[2:0] Value	Output State		Fan Speed
	/FS[2:0]	/SHDN	
000	111	0	off
001	110	1	speed 1 (slowest)
010	101	1	speed 2
011	100	1	speed 3
100	011	1	speed 4
101	010	1	speed 5
110	001	1	speed 6
111	000	1	speed 7 (fastest)

**Fan Speed Settings**

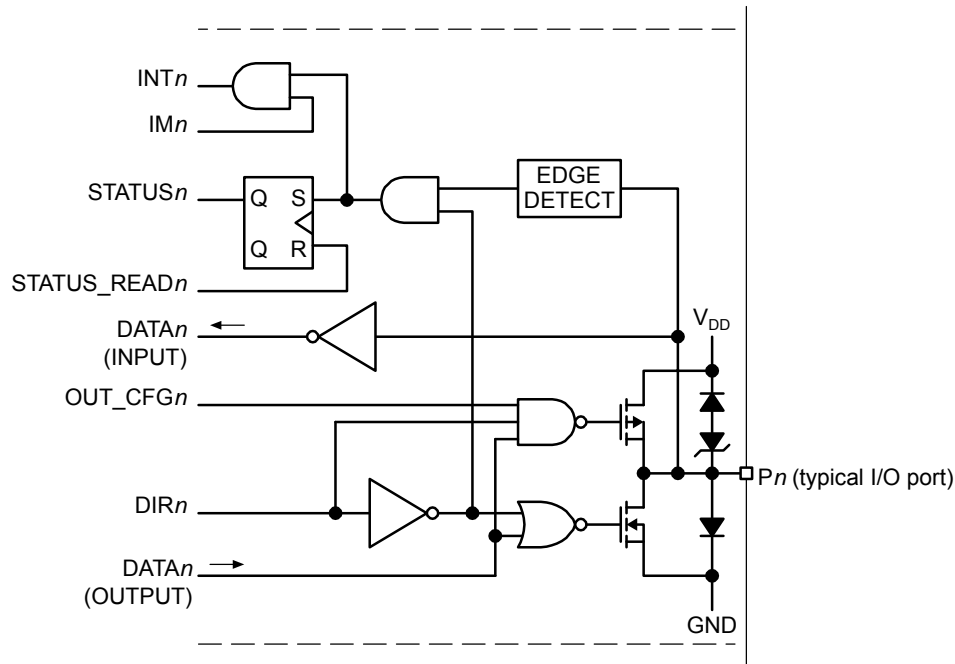
Notes: Any time the fan speed register contains zero, that is, the fan is shut down, and a non-zero value is written into the fan speed register, the /FS[2:0] and /SHDN outputs will assume the highest fan speed state for approximately one second (t<sub>START</sub>). Following this interval, the state of the fan speed control outputs will assume the value indicated by the contents of FAN\_SPEED. This insures that the fan will start reliably when low speed operation is desired.

Bit Name: D[3] through D[7]

Function: Reserved

Operation: Always write as zero

## Functional Diagram



Typical I/O Port (Fan Speed Control Logic Not Shown)

## Functional Description

### Pin Descriptions

#### VDD

Power supply input connection. See “Operating Ratings.”

#### GND

Ground or return connection for all MIC74 functions.

#### CLK

An CLK signal is provided by the host (master) and is common to all devices on the bus. The CLK signal controls all transactions in both directions on the bus and is applied to each MIC74 at the CLK pin.

#### DATA

Serial data is bidirectional and is common to all devices on the bus. The MIC74’s DATA output is open-drain.

The DATA line requires one external pull-up resistor or current source per system that can be located anywhere along the line.

#### A2, A1, A0

An MIC74 responds to its own unique address which is assigned using the A0–A2 pins. A0–A2 set the three LSBs (least significant bits) of the MIC74’s 7-bit slave address. The three address pins allow eight unique MIC74 addresses in a system. When the MIC74’s address matches an address received in the serial bit stream, communication is initiated.

A2, A1 and A0 should be connected to GND or VDD. The state of these pins is sampled only once at device power-on. New slave addresses are not accepted unless the MIC74 is powered off then on.

Inputs			MIC74 Slave Address	
A2	A1	A0	Binary	Hex
0	0	0	010 0000 <sub>b</sub>	20 <sub>h</sub>
0	0	1	010 0001 <sub>b</sub>	21 <sub>h</sub>
0	1	0	010 0010 <sub>b</sub>	22 <sub>h</sub>
0	1	1	010 0011 <sub>b</sub>	23 <sub>h</sub>
1	0	0	010 0100 <sub>b</sub>	24 <sub>h</sub>
1	0	1	010 0101 <sub>b</sub>	25 <sub>h</sub>
1	1	0	010 0110 <sub>b</sub>	26 <sub>h</sub>
1	1	1	010 0111 <sub>b</sub>	27 <sub>h</sub>

Table 1. MIC74 Address Configuration

### Alert Response Address

The MIC74 also responds to the ARA (Alert Response Address). The ARA is used by the master (host) to request the address of a slave that has provided an interrupt to the master via the /ALERT line.

The ARA is a single address (0001 100) common to all slaves and is described in more detail under “Interrupt Generation” with related information under “/ALERT.” Also see Figure 7.

***P<sub>n</sub>, /SHDN, and /FS0–/FS2***

P0 through P7 are general-purpose input/output bits. Each bit is independently programmable as an input or an output. If programmed as an output, each bit is further programmable as either a complementary push-pull or open-drain output.

If properly enabled, any *P<sub>n</sub>* programmed as an input will generate an interrupt to the host using the /ALERT output when the input changes state. In this way, the MIC74 can notify the host of an input change without requiring periodic polling by the host or a message transaction on the bus.

Regardless of whether interrupts are enabled or disabled, each input-change event also sets the corresponding bit in the status register. I/O configuration is performed using the output configuration (OUT\_CFG), I/O direction (DIR), and interrupt mask (INT\_MASK) registers.

If the FAN bit in the device configuration register is set, the states of P[7:4] are controlled by the FAN\_SPEED register. The bits in the OUT\_CFG, DIR, and INT\_MASK registers corresponding to P[7:4] are ignored. When in Fan Mode, P[7:4] are referred to as /FS2, /FS1, /FS0, and /SHDN. While in this mode, no interrupts of any kind will be generated by these pins.

***/ALERT***

The alert signal is an open-drain, active-low output. The operation of the /ALERT output is controlled by the IM<sub>*n*</sub> bits in the INT\_MASK register and the global interrupt enable bit (IE) in the DEV\_CFG register.

If the IE bit is set to zero, or if the corresponding interrupt enable bit, IM<sub>*n*</sub>, is set to zero, no input-change interrupts will be generated. (Regardless of the IE bit setting, the change will be reflected in the status register.)

If the IE bit is set to one, IM<sub>*n*</sub> is set to one, and *P<sub>n</sub>* is an input, then /ALERT is driven active whenever *P<sub>n</sub>* changes state, (goes from a high-to-low or low-to-high state). Once triggered, /ALERT is unconditionally reset to its inactive state once the MIC74 successfully responds to the alert response addressor STATUS is read.

***Serial Port Operation***

The MIC74 uses standard SMBus Read\_Byte and Write\_Byte operations to communicate with its host.

The Read\_Byte operation is a composite read-write operation consisting of first sending the MIC74's slave address followed by a command byte (a write) and then resending the slave address and clocking out the data byte (a read). The command byte is the address of the target register. See Table 2. An example of a Read\_Byte operation is shown in Figure 8.

Similarly, the write-byte operation consists of sending the device's slave address followed by a command byte and the byte to be written to the target register. Again, in the case of the MIC74, the command byte is the address of the target register. See Table 2.

In addition, to the read byte and write byte protocols, the MIC74 adheres to the SMBus protocol for response to the ARA (alert response address). An MIC74 expects to be interrogated using the ARA when it has asserted its /ALERT output. /ALERT interrupts can be enabled or disabled using the IE bit in the DEV\_CFG register.

***Power-On***

When power is initially applied, the MIC74's internal registers will assume their power-up default state and the state of the address inputs, A2, A1 and A0, will be read to establish the device's slave address. See the individual register descriptions for each registers default state. Also see Table 2.

***I/O Ports***

Each I/O bit, P0 through P7, may be individually programmed as an input or output using the corresponding bit in the I/O direction register, DIR.

If programmed as an output, each is further programmable as either a complementary push-pull or open-drain output using the output configuration register, OUT\_CFG.

If enabled by the corresponding bit, IM<sub>*n*</sub>, in the interrupt mask register INT\_MASK, each *P<sub>n</sub>* programmed as an input will generate an interrupt to the host on /ALERT if the input changes state. In this way, the MIC74 can notify the host of an input change without requiring periodic polling by the host or a transaction on the bus.

Each input-change event also sets the corresponding bit in the status register, STATUS.

See "Functional Diagram" for the logic arrangement of atypical MIC74 I/O port.

***Fan Speed Control***

If the FAN bit in the device configuration register is set, the state of P[7:4] is controlled by the FAN\_SPEED register. The bits in the OUT\_CFG, DIR, and INT\_MASK registers corresponding to P[7:4] are ignored. When in Fan Control Mode, P[7:4] are referred to as /FS2, /FS1, /FS0, and /SHDN. While in this mode, no interrupts of any kind will be generated by these pins. See "Applications Information" for typical fan speed control applications.



Register Name	Register Description	Address		Available Operations	Power-On Default	
		Binary	Hex		Binary	Hex
DEV_CONFIG	Device Configuration	0000 0000 <sub>b</sub>	00 <sub>h</sub>	8-bit read/write	0000 0000 <sub>b</sub>	00 <sub>h</sub>
DIR	I/O Direction	0000 0001 <sub>b</sub>	01 <sub>h</sub>	8-bit read/write	0000 0000 <sub>b</sub>	00 <sub>h</sub>
OUT_CFG	Output Configuration	0000 0010 <sub>b</sub>	02 <sub>h</sub>	8-bit read/write	0000 0000 <sub>b</sub>	00 <sub>h</sub>
STATUS	Interrupt Status	0000 0011 <sub>b</sub>	03 <sub>h</sub>	8-bit read	0000 0000 <sub>b</sub>	00 <sub>h</sub>
INT_MASK	Interrupt Mask	0000 0100 <sub>b</sub>	04 <sub>h</sub>	8-bit read/write	0000 0000 <sub>b</sub>	00 <sub>h</sub>
DATA	General-Purpose I/O	0000 0101 <sub>b</sub>	05 <sub>h</sub>	8-bit read/write	1111 1111 <sub>b</sub>	FF <sub>h</sub>
FAN_SPEED	Fan Speed	0000 0110 <sub>b</sub>	06 <sub>h</sub>	8-bit read/write	0000 0000 <sub>b</sub>	00 <sub>h</sub>

Table 2. Register Summary

**Fan Start-Up**

Any time the fan speed register contains zero (fan is off) and then a nonzero value is written to FAN\_SPEED, the /FS[2:0] and /SHDN outputs will assume the highest fan speed state for approximately one second ( $t_{START}$ ). Following this interval, the state of the fan speed control outputs will assume the value indicated by the contents of FAN\_SPEED. This insures that the fan will start reliably when low speed operation is desired. The  $t_{START}$  interval is generated by an internal oscillator and counters. At the end of  $t_{START}$ , this oscillator is powered down to reduce overall power consumption.

half of  $t_{START}$ , the /SHDN pin is deasserted. Conversely, when the fan is shutdown (zero is written to FAN\_SPEED), the /SHDN pin is deasserted first. The /FS[2:0] lines are subsequently deasserted after a delay of  $1/2t_{START}$ . The internal oscillator is also powered down following the  $t_{START}/2$  interval at fan shut-down. These timing relationships are illustrated in Figure 2.

**Interrupt Generation**

Assuming that any or all of the I/O's are configured as inputs, the MIC74 will reflect the occurrence of an input change in the corresponding bit in the status register, STATUS. This action cannot be masked. An input change will only generate an interrupt to the host if interrupts are properly configured and enabled.

The MIC74 can operate in either polled mode or interrupt mode. In the case of polled operation, the host periodically reads the contents of STATUS to determine the device state. The act of reading STATUS clears its contents. Repeating events which have occurred since the last read from STATUS will not be discernable to the host.

Interrupts are only generated if the global interrupt enable bit, IE, in the DEV\_CFG register is set. The /ALERT signal will be asserted (driven low) when an interrupt is generated. The MIC74 expects to be interrogated using the ARA when it has generated an interrupt output. Once it has successfully responded to the ARA (Alert Response Address), the /ALERT output will be deasserted. The contents of the status register will not be cleared until it is read using a read byte operation.

If a given system does not wish to use the SMBus ARA protocol for reporting interrupts, the system may simply poll the contents of the status register after detecting an interrupt on /ALERT. This action will clear the contents of STATUS and cause /ALERT to be deasserted. Reading the status register is an acceptable substitute for using the ARA protocol. Presumably, however, it will involve higher system overhead since all the devices on the bus must be polled to determine which one generated the interrupt.

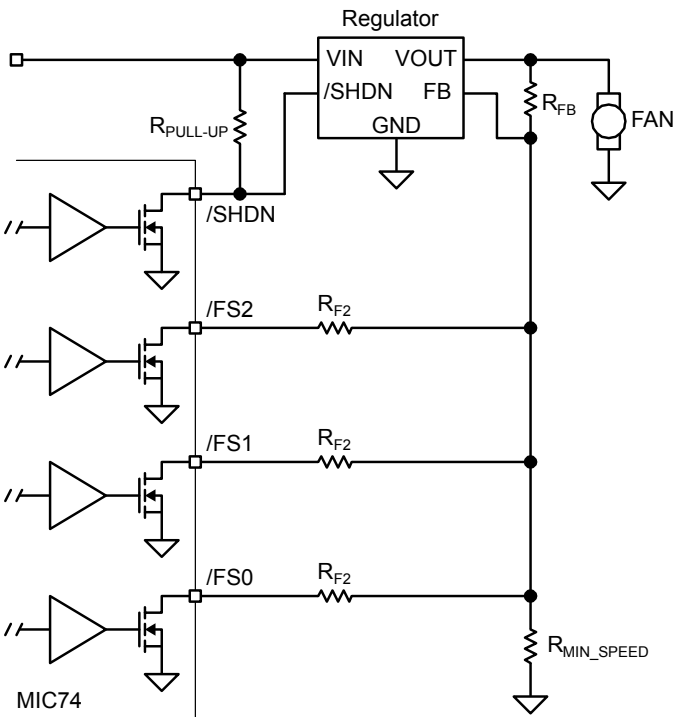


Figure 1. Fan Speed Control Application

Proper sequencing of the /FS[2:0] and /SHDN signals is performed by the MIC74's internal logic state machine. When activating the fan from the off state, the /FS[2:0] lines change state first, then, after a delay equal to one-

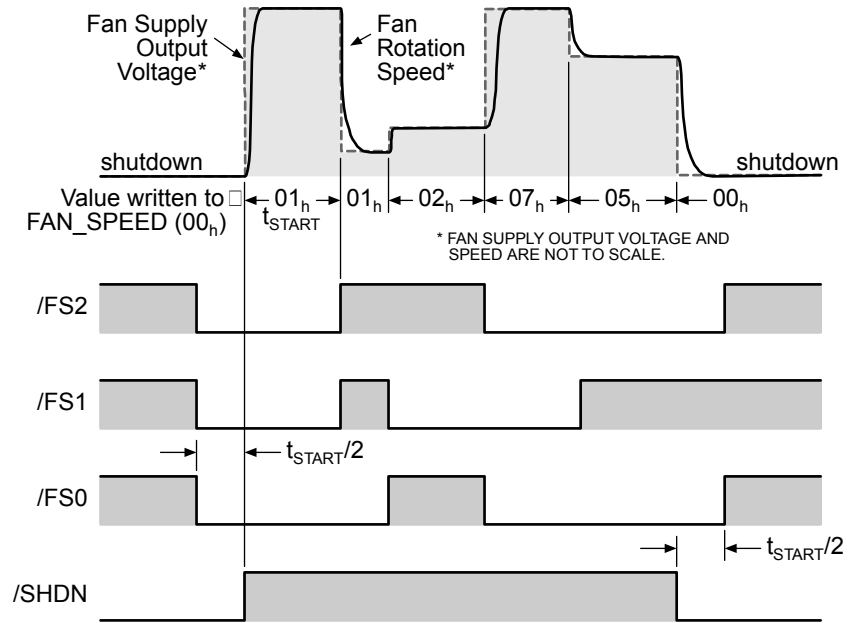
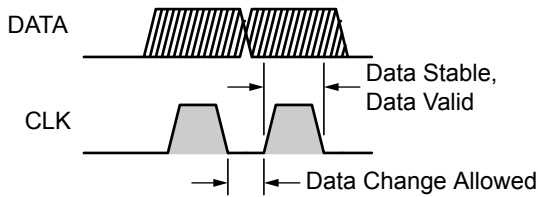


Figure 2. Typical MIC74 Fan-Mode Timing and System Behavior

## Application Information

### Bit Transfer

The data received on the DATA pin must be stable during the high period of the clock.

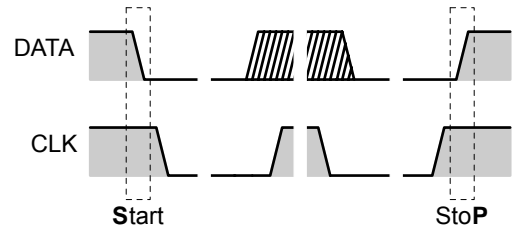


**Figure 3. Acceptable Bit Transfer Conditions**

Data can change state only when the CLK line is low. Refer to Figure 3.

### Start and Stop Conditions

Two unique bus situations define “start” and “stop” conditions. A high-to-low transition of the DATA line while CLK is high indicates a start condition. A low-to-high transition of the DATA line while CLK is high defines a stop condition. See Figure 4.

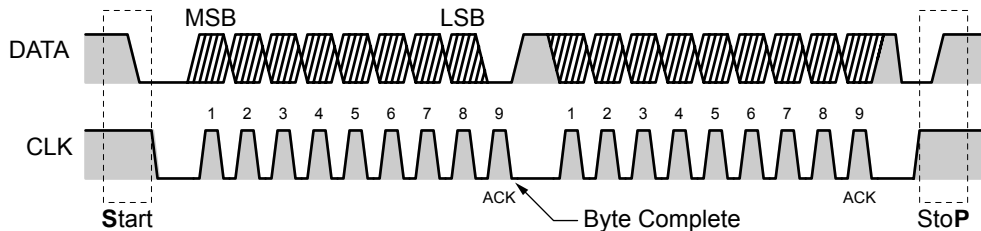


**Figure 4. Start and Stop Definitions**

Start (S) and stop (P) conditions are always generated by the bus master (host). After a start condition, the bus is considered to be busy. The bus becomes free again after a certain time following a stop condition or after both CLK and DATA lines remain high for more than 50µs.

### Serial Byte Format

Every byte consists of 8 bits. Each byte transferred on the bus must be followed by an acknowledge bit. Bytes are transferred with the MSB (most significant bit) first. See Figure 5.

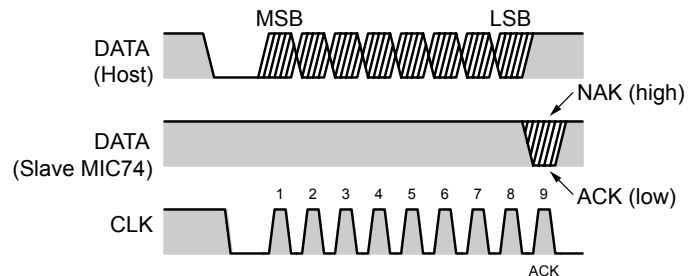


**Figure 5. Serial Byte Format**

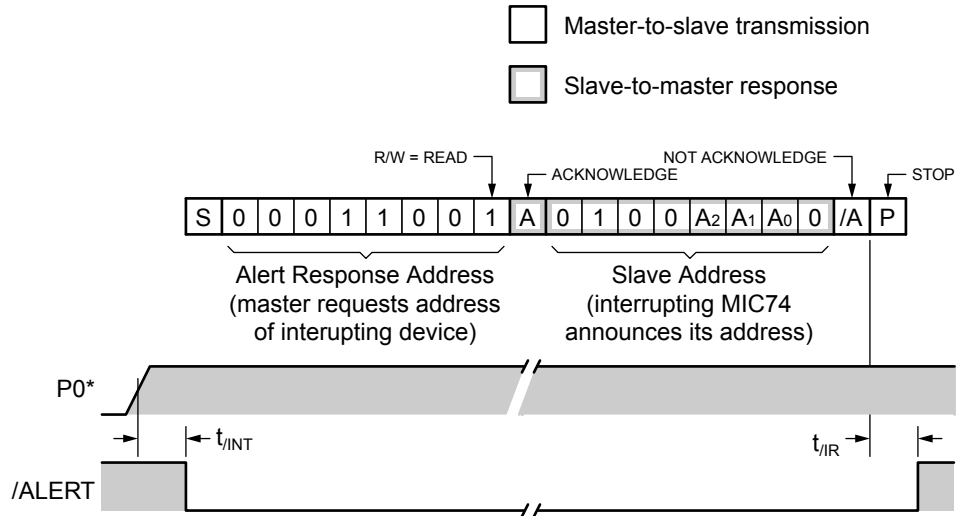
### Acknowledge and Not Acknowledge

The acknowledge related clock pulse is generated by the master. The transmitter releases the DATA line (high) during the acknowledge clock cycle.

In order to acknowledge (ACK) a byte, the receiver must pull the DATA line low during the high period of the clock pulse according the bus timing specifications. A slave device that wishes to not acknowledge a byte must let the DATA line remain high during the acknowledge clock pulse. See Figure 6.

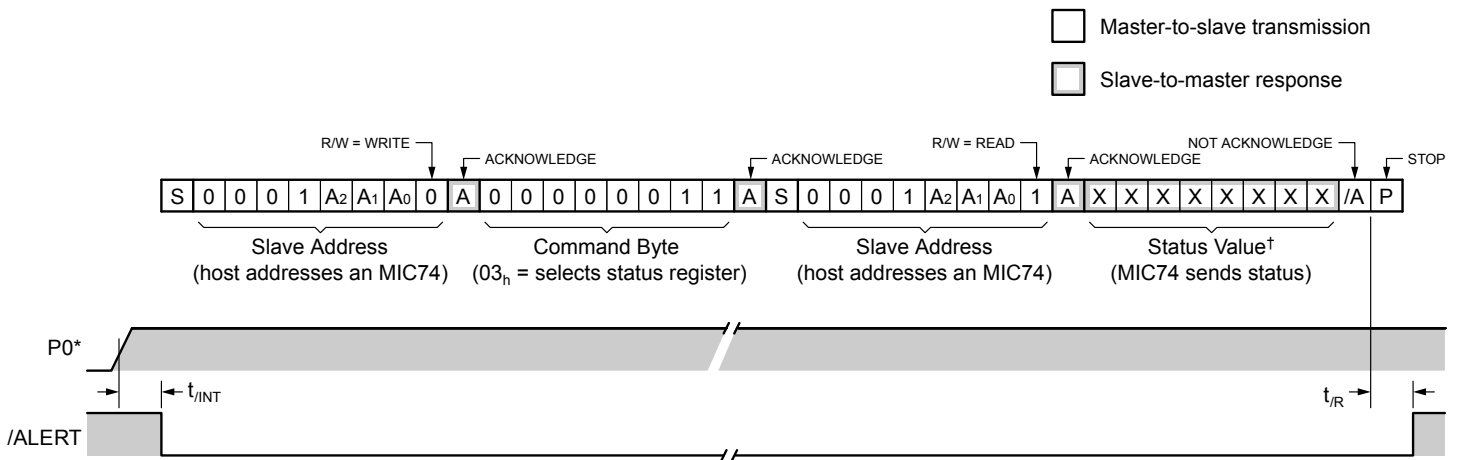


**Figure 6. Acknowledge and Not Acknowledge**



\* Assumes P0 interrupts properly configured and enabled. P0 used as an example. Timing for P1 to P7 is identical.

**Figure 7. Interrupt Handling Using the Alert Response Address**



\* Assumes P0 interrupts properly configured and enabled. P0 used as an example. Timing for P1 to P7 is identical.

<sup>†</sup> STATUS register is cleared to zero following this operation.

**Figure 8. Interrupt Handling Without the Alert Response Address**

**Initializing the MIC74**

The MIC74's internal registers are reset to their default state at power-on. The MIC74's default state can be summarized as follows:

- All I/O's configured as inputs (DIR = 00h)
- Output configuration set to open-drain(OUT\_CFG = 00h)
- All outputs high/floating (DATA = FFh)
- Fan functions disabled (FAN\_SPEED = 00h, FAN bit of DEV\_CFG = 0)
- All interrupts masked (IE bit of DEV\_CFG = 0)

The result of this configuration is that all I/O pins will essentially float unless driven by external circuitry. Any system using the MIC74 will need to initialize the internal registers to the state required for proper system operation. The recommended order for initializing the MIC74's registers is as follows:

1. Write DATA
2. Write OUT\_CFG
3. Write DIR
4. Write FAN\_SPEED (if using fan mode)
5. Write INT\_MASK (if using interrupts)
6. Read STATUS to clear it.
7. Write DEV\_CFG to enable fan mode and/or interrupts, if using

At the conclusion of step three, any I/O's configured as outputs in step two will be driven to the levels programmed into the data register in step one. The order of step 1 through step 3 is important to insure that spurious data does not appear at the I/O's during configuration. Following step 7, programming the device configuration register, the MIC74 will begin generating interrupts if they are enabled, and the fan will be started if FAN\_SPEED contains a nonzero value. The corresponding interrupt service routines (if any) must be initialized and enabled prior to step seven. STATUS should be cleared (step 6) in both polled and interrupt driven systems.

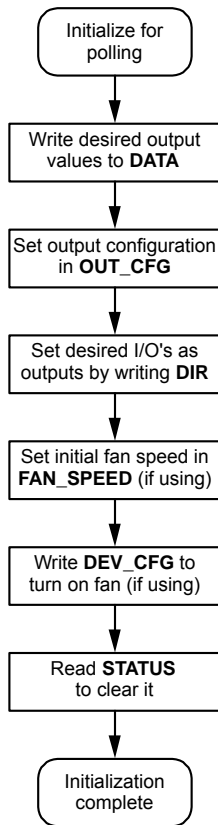


Figure 9a. Initializing the MIC74 for Polled Operation

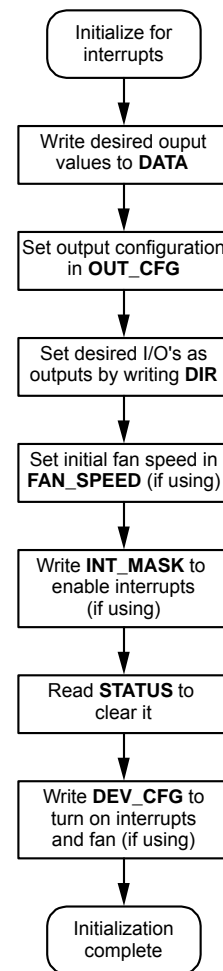


Figure 9b. Initializing the MIC74 for Interrupts

**Polled Mode**

Input state changes on I/O's configured as inputs will be reflected in the status register regardless of the state of the global interrupt enable bit (IE) and the individual interrupt mask bits in INT\_MASK. In a system utilizing polling to monitor for input changes, the status register is periodically read to check for input events. The act of reading STATUS clears it in preparation for detecting future events. The status bits corresponding to I/O's configured as outputs or corresponding to P[7:4] when in fan mode will not be set by state changes on these pins. It is always good practice, however, to mask the value obtained when reading STATUS to eliminate any bits, output or otherwise, that are not of immediate concern. This will help avoid problems if software changes are made in the future.

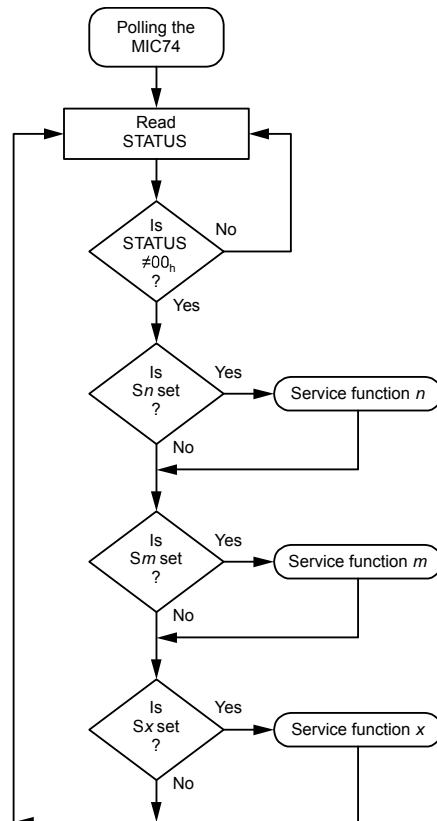
The flowchart shown in Figure 9a illustrates the steps involved in initializing the MIC74 for polled operation. The flowchart in Figure 10 illustrates the corresponding polling routine. The process for writing output data is straight-forward—simply write the desired bit pattern to DATA. (Special precautions may be required when changing output data in an interrupt driven system, however. See the discussion below under “Writing to the Data Register.”)

**Interrupt Mode**

Input state changes on I/Os configured as inputs will be reflected in the status register regardless of the state of the global interrupt enable bit (IE) and the individual interrupt mask bits in INT\_MASK. In a system utilizing interrupts to detect input changes, one or more of the bits in the interrupt mask register, INT\_MASK, are set to allow interrupts on/ALERT to be generated by input events. The global interrupt enable bit, IE, in the device configuration register must also be set to enable interrupts.

The flowchart shown in Figure 9b illustrates the steps involved in initializing the MIC74 for interrupt-driven operation. The flowchart in Figure 11 illustrates the corresponding interrupt service routine using the SMBus ARA (alert response address). The corresponding timing diagram is shown in Figure 7. The flowchart in Figure 12 illustrates the corresponding interrupt service routine using polling to determine the interrupt source. Figure 8 illustrates the timing. Utilizing the ARA greatly speeds identification of the interrupting slave device and lowers latency, as only a single transaction on the bus is necessary to identify the interrupt source.

Using either method, STATUS must be read to determine the exact source of the interrupt within the MIC74.



**Figure 10. Polling the MIC74**

The act of reading STATUS clears it in preparation for detecting future events. The status bits corresponding to I/O's configured as outputs or corresponding to P[7:4] when in fan mode will not be set by state changes on these pins. It is always good practice, however, for the interrupt service routine to mask the value obtained when reading STATUS to eliminate any bits, output or otherwise, that are not of immediate concern. This will help avoid problems if software changes are made in the future.

The process for writing output data is straight-forward—simply write the desired bit pattern to DATA. Special precautions may be required, however, when changing output data in an interrupt driven system. See the discussion below under “Writing to the Data Register.”

**Writing To The Data Register**

Multiple software routines may use the various output bits available on the MIC74 to control individual functions such as power switches, LED's, etc. These various functions may be handled by independent software routines which must manipulate individual output bits without regard for other bits. Care must be taken to insure that these various software routines do not interfere with each other when modifying output data. The recommended procedure for changing isolated output bits is as follows:

1. Read DATA
2. Set desired bits by ORing the value read from DATA with an appropriate mask value
3. Clear desired bits by ANDing the value read from DATA with an appropriate mask value
4. Write the result back to DATA

A functionally equivalent alternative to this procedure is to keep an image of the data register in software. Any independent routines would make changes to this image using the procedure above and then call a routine that actually writes the new image to DATA. Interrupts would be disabled briefly while DATA is being modified.

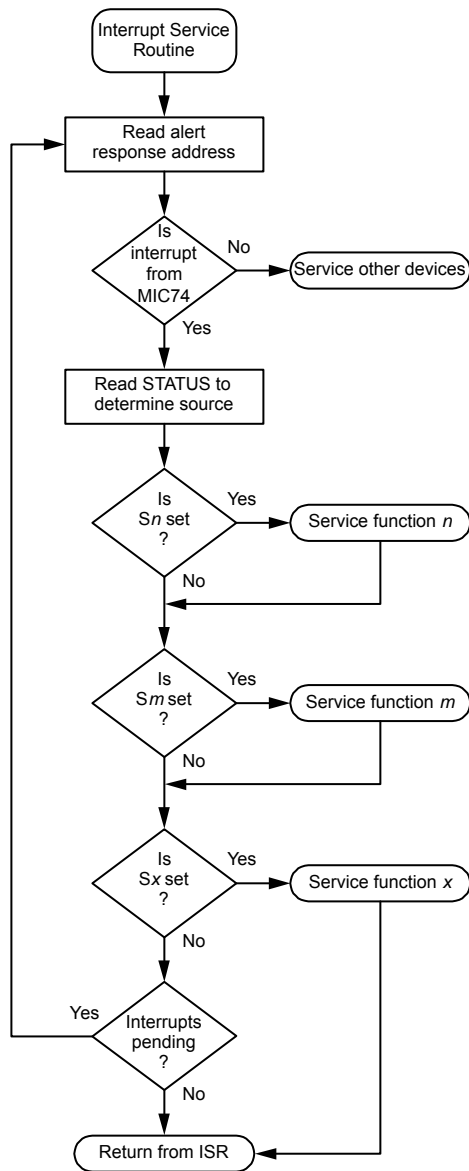


Figure 11: Interrupt Service Routine Using the ARA

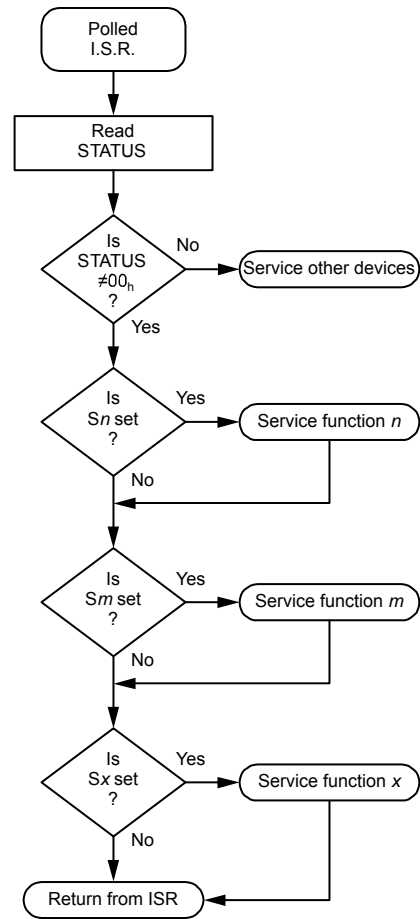


Figure 12: Interrupt Service Routine Without ARA

Regardless of which procedure is used, it is important that only one software routine at a time attempts to make changes to the output data. In a system where polling is the exclusive method for servicing inputs, this is usually not a problem. If interrupts are employed to any degree in dealing with MIC74 inputs, care must be taken to insure that a software routine in the midst of making changes to outputs is not interrupted by another routine that proceeds to make its own changes. The risk is that the value in DATA will be changed by an interrupting routine after it is read by a different routine in the process of making its own changes. If this occurs, the value written to DATA by the first routine may be incorrect. The most straight-forward solution to this potential problem is to disable system interrupts while the data register is actually being modified.

**Application Circuits**

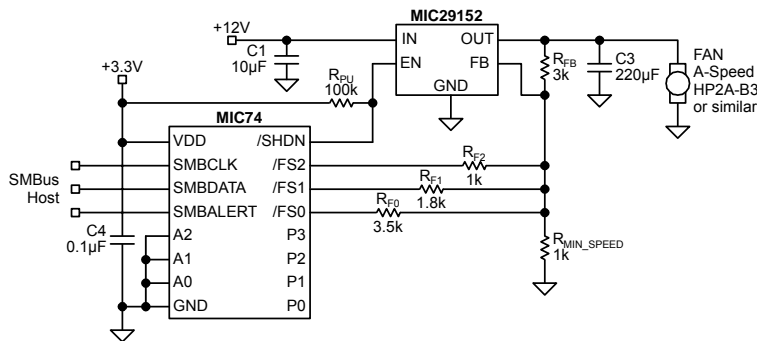
The MIC74, in conjunction with a linear low-dropout or switching regulator, can be configured as a fan speed controller. Most adjustable regulators have a feedback pin and use an external resistor divider to adjust the output voltage. The MIC74 is designed to take advantage of this configuration with its ability to manipulate multiple feedback resistors connected to the P4–P7 outputs. Individual open-drain output bits are selectively grounded or allowed to float under the control of the internal state machine. This action raises or lowers the

equivalent resistance seen in the regulator’s feedback path, thus changing the output voltage.

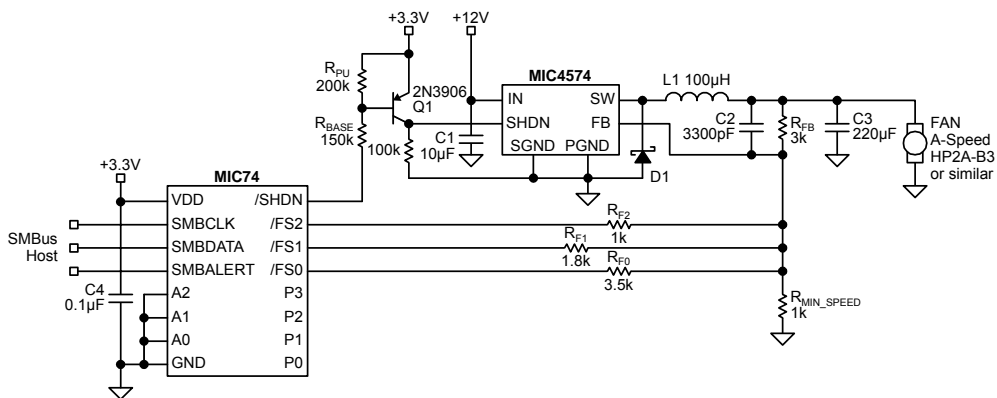
Any conventional adjustable regulator is usually suitable for use with the MIC74. The output voltage corresponding to each value to be programmed into the fan speed register can be determined by selecting the resistors in the circuit. The regulator itself can be chosen to meet the needs of the application, such as input voltage, output voltage, current handling capability, maximum power dissipation, and physical space constraints. Two circuit examples are shown below.

The circuit of Figure 13 illustrates use of a typical LDO linear regulator such as the MIC29152. A switching regulator-based fan control circuit using the MIC4574 200kHz Simple 0.5A Buck Regulator is shown in Figure 14. Both circuits assume a 12V fan power supply but will accommodate much higher input voltages if required (MIC4574: 24V, MIC29152:26V). *Care must be taken, however, to insure that the maximum power dissipation of the regulator is not exceeded. If the regulator overheats, its internal thermal shutdown circuitry will deactivate it. (See MIC29152 or MIC4574 datasheet.)*

Since the MIC74 powers up with all its I/O’s inputs (floating), both circuits will power-up with the fan running at a minimum speed determined by the value of RMIN\_SPEED. Once the MIC74’s fan mode is activated by setting the appropriate bit in the configuration register,



**Figure 13. Fan Speed Control Using an Adjustable Low-Dropout Regulator**



**Figure 14. Fan Speed Control Using a Buck Converter**



the fan will be shutdown by the assertion of the /SHDN output if FAN\_SPEED is zero. If FAN\_SPEED is programmed with any nonzero value, the fan will be driven to its maximum speed for the duration of  $t_{START}$  (about 1 second) and then assume the programmed speed. Note that the circuit in Figure 14 contains an additional transistor, Q1, as an inverter because the regulator in this example has an active-high shutdown input rather than an enable input. Otherwise the circuits function identically.

Table 3 lists the output voltages corresponding to all the fan speeds and system states possible with these circuits. The following equations are used to calculate the resistor values used in MIC74 fan speed control circuits. It is assumed here that the regulator's internal reference voltage is 1.24V. If the regulator uses a different reference voltage, that value should be used instead.

The following equations show how to calculate the resistor values for the fan controllers. For example, when the fan speed register contains  $011_b$ , which is the 3rd lowest speed,  $R_{F1}$  and  $R_{F0}$  are parallel to  $R_{MIN}$  to give the equivalence resistor ( $R_{EQ}$ ) value of 545 $\Omega$ .

$$R_{EQ} = R_{F1} \parallel R_{F0} \parallel R_{MIN}$$

$$R_{EQ} = 1.8k \parallel 3.6k \parallel 1k$$

$$R_{EQ} = 545\Omega$$

The output voltage is calculated by using:

$$V_{OUT} = 1.24V \left( 1 + \frac{R_{FB}}{R_{EQ}} \right)$$

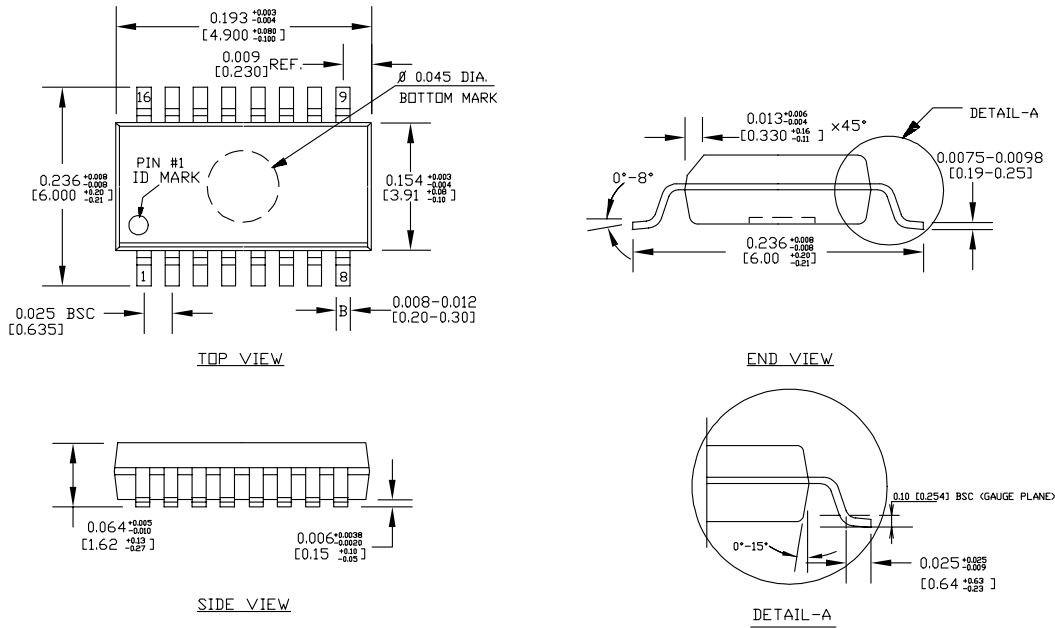
$$V_{OUT} = 1.24V \left( 1 + \frac{3k}{545\Omega} \right)$$

$$V_{OUT} = 8.06V$$

FAN_SPEED Value	Fan Speed Selected	R <sub>FB</sub>	R <sub>MIN</sub>	R <sub>F2</sub>	R <sub>F1</sub>	R <sub>F0</sub>	R <sub>EQ</sub>	V <sub>OUT</sub>
0000 0000b	power-up	3k	1k	open	open	open	1k	4.96V
0000 0000b	fan off	3k	1k	open	open	open	1k	0V
0000 0001b	lowest	3k	1k	open	open	3.6k	783	5.99V
0000 0010b	2 <sup>nd</sup> lowest	3k	1k	open	1.8k	open	643	7.03V
0000 0011b	3 <sup>rd</sup> lowest	3k	1k	open	1.8k	3.6k	545	8.06V
0000 0100b	medium	3k	1k	1k	open	open	500	8.68V
0000 0101b	3 <sup>rd</sup> highest	3k	1k	1k	open	3.6k	439	9.71V
0000 0110b	2 <sup>nd</sup> highest	3k	1k	1k	1.8k	open	391	10.75V
0000 0111b	highest	3k	1k	1k	1.8k	3.6k	353	11.78V

Table 3. Fan Speed Selection

# Package Information



**NOTE:**

1. ALL DIMENSIONS ARE IN INCHES [MM].
2. LEAD COPLANARITY SHOULD BE 0.004" [0.10 mm] MAX.
3. MAX MISALIGNMENT BETWEEN TOP AND BOTTOM CENTER OF PACKAGE TO BE 0.004" [0.10 mm].
4. THE LEAD WIDTH, B TO BE DETERMINED AT .0075 [0.19 mm] FROM THE LEAD TIP.
5. BOTTOM MARK IS OPTIONAL, IT MAY NOT APPEAR ON THE ACTUAL UNITS.

## 16-Pin QSOP (QS)

**MICREL, INC. 2180 FORTUNE DRIVE SAN JOSE, CA 95131 USA**  
 TEL +1 (408) 944-0800 FAX +1 (408) 474-1000 WEB <http://www.micrel.com>

The information furnished by Micrel in this data sheet is believed to be accurate and reliable. However, no responsibility is assumed by Micrel for its use. Micrel reserves the right to change circuitry and specifications at any time without notification to the customer.

Micrel Products are not designed or authorized for use as components in life support appliances, devices or systems where malfunction of a product can reasonably be expected to result in personal injury. Life support devices or systems are devices or systems that (a) are intended for surgical implant into the body or (b) support or sustain life, and whose failure to perform can be reasonably expected to result in a significant injury to the user. A Purchaser's use or sale of Micrel Products for use in life support appliances, devices or systems is a Purchaser's own risk and Purchaser agrees to fully indemnify Micrel for any damages resulting from such use or sale.

© 2000 Micrel, Incorporated.