

# LM8323 Mobile I/O Companion Supporting Keypad, I/O Expansion, PWM, and ACCESS.bus Host Interface

Check for Samples: [LM8323](#)

## FEATURES

- **Key Features**
  - Supports Keypad Matrices of up to 8 × 12 Keys Plus 8 Special-function (SF) Keys for a Total of 104 Keys. SF Keys Pull Keypad Scan Inputs Directly to Ground, Rather than Connecting to a Keypad Scan Output.
  - Supports I<sup>2</sup>C-compatible ACCESS.bus Interface in Slave Mode up to 400 kHz (Fast-mode).
  - Three Host-programmable PWM Outputs Useful for Smooth LED Brightness Modulation.
  - Supports General-purpose I/O Expansion on Pins Not Otherwise Used for Keypad or Rotary Encoder Interface.
  - Key-scan Event Storage in a FIFO Buffer for up to 15 Events.
  - Key Events, Errors, and Dedicated Hardware Interrupts Request Host Service by Asserting the  $\overline{\text{IRQ}}$  Output.
  - The Correct Reception of a Command May be Assumed, if No Error is Reported from the LM8323 After Receiving it.
  - Wake-up from Halt Mode on any Matrix Key-scan Event, any Use of the SF Keys, or Any Activity on the ACCESS.bus Interface, or Any Change in the Rotary Encoder Counter Value (if Enabled).
- **Host-Controlled Functions**
  - Three PWM Outputs
  - Period of Inactivity that Triggers Entry into Halt Mode
  - Debounce Time for Reliable Key Event

## Polling

- Configuration of General-purpose I/O Ports
- Various Initialization Options (Keypad Size, etc.)
- **Key Device Characteristics**
  - 1.8V ± 180 mV Single-supply Operation
  - On-chip Power-on Reset (POR)
  - Watchdog Timer
  - Dedicated Slow Clock Input for 32 kHz
  - -40°C to +85°C Industrial Temperature Range
  - 36-pin csBGA Package

## APPLICATIONS

- Cordless Phones
- Smart Handheld Devices

## DESCRIPTION

The LM8323 key-scan controller is a dedicated device to unburden a host from scanning a matrix-addressed keypad. In addition, the LM8323 provides general-purpose I/O expansion, a rotary encoder interface and PWM outputs useful for dynamic LED brightness modulation.

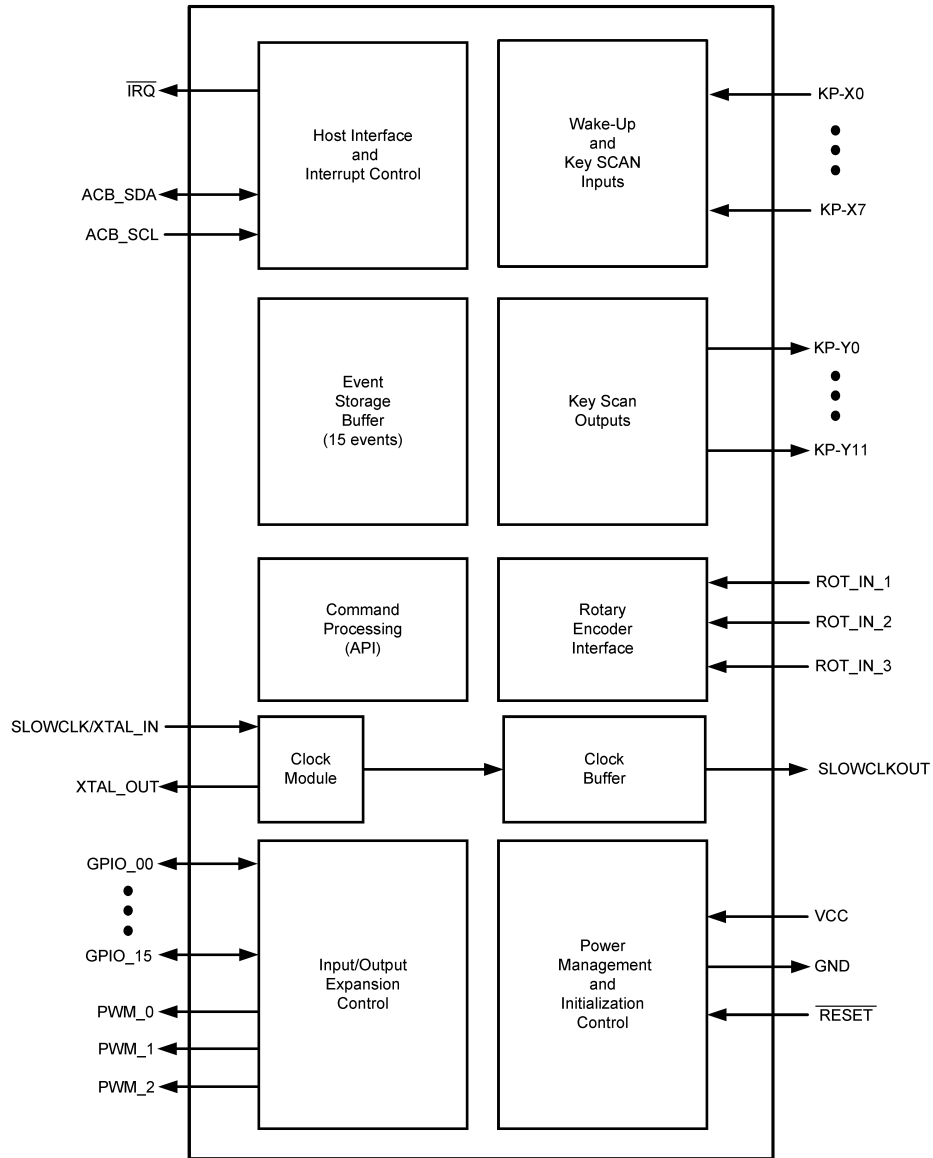
It communicates with the host through an I<sup>2</sup>C-compatible ACCESS.bus interface. An interrupt output is available for signaling key-press and key-release events. Communication frequencies up to 400 kHz (Fast-mode) bus speed are supported. The LM8323 supports a predefined set of commands. These commands enable a host device to keep control over all functions.

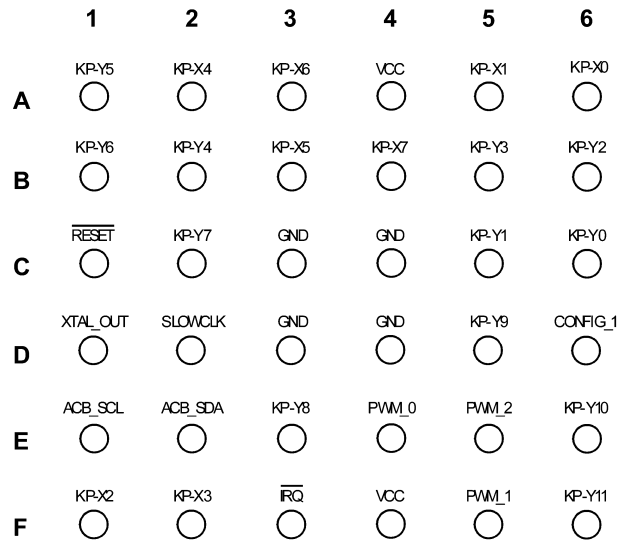


Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

All trademarks are the property of their respective owners.

**BLOCK DIAGRAM**



**PIN ASSIGNMENTS**


**Figure 1. Top View  
36-Pin csBGA Package  
See Package Number NYB0036A**

**SIGNAL DESCRIPTIONS**

| Pin | Function | I/O    | Description                             |
|-----|----------|--------|---|
| A6  | KP-X0    | Input  | Wake-up input/Keyboard scanning input 0 |
| A5  | KP-X1    | Input  | Wake-up input/Keyboard scanning input 1 |
| F1  | KP-X2    | Input  | Wake-up input/Keyboard scanning input 2 |
| F2  | KP-X3    | Input  | Wake-up input/Keyboard scanning input 3 |
|     | GPIO_13  | I/O    | General-purpose I/O port 13             |
| A2  | KP-X4    | Input  | Wake-up input/Keyboard scanning input 4 |
|     | GPIO_12  | I/O    | General-purpose I/O port 12             |
| B3  | KP-X5    | Input  | Wake-up input/Keyboard scanning input 5 |
|     | GPIO_11  | I/O    | General-purpose I/O port 11             |
| A3  | KP-X6    | Input  | Wake-up input/Keyboard scanning input 6 |
|     | GPIO_10  | I/O    | General-purpose I/O port 10             |
| B4  | KP-X7    | Input  | Wake-up input/Keyboard scanning input 7 |
|     | GPIO_09  | Input  | General-purpose I/O port 9              |
| C6  | KP_Y0    | Output | Keyboard scanning output 0              |
| C5  | KP-Y1    | Output | Keyboard scanning output 1              |
| B6  | KP-Y2    | Output | Keyboard scanning output 2              |
| B5  | KP-Y3    | Output | Keyboard scanning output 3              |
|     | GPIO_08  | I/O    | General-purpose I/O port 8              |
| B2  | KP-Y4    | Output | Keyboard scanning output 4              |
|     | GPIO_07  | I/O    | General-purpose I/O port 7              |
| A1  | KP-Y5    | Output | Keyboard scanning output 5              |
|     | GPIO_06  | I/O    | General-purpose I/O port 6              |
| B1  | KP-Y6    | Output | Keyboard scanning output 6              |
|     | GPIO_05  | I/O    | General-purpose I/O port 5              |
| C2  | KP-Y7    | Output | Keyboard scanning output 7              |
|     | GPIO_04  | I/O    | General-purpose I/O port 4              |

**SIGNAL DESCRIPTIONS (continued)**

| Pin               | Function                  | I/O    | Description                    |
|-------------------|---------------------------|--------|--------------------------------|
| E3                | KP-Y8                     | Output | Keyboard scanning output 8     |
|                   | SLOWCLKOUT                | Output | 32.768 kHz clock output        |
|                   | GPIO_03                   | I/O    | General-purpose I/O port 3     |
| D5                | KP-Y9                     | Output | Keyboard scanning output 9     |
|                   | MUX2_IN1                  | Input  | Multiplexer 2 input 1          |
|                   | GPIO_02                   | I/O    | General-purpose I/O port 2     |
| E6                | KP-Y10                    | Output | Keyboard scanning output 10    |
|                   | MUX2_IN2                  | Input  | Multiplexer 2 input 2          |
|                   | GPIO_01                   | I/O    | General-purpose I/O port 1     |
| F6                | KP-Y11                    | Output | Keyboard scanning output 11    |
|                   | MUX2_OUT                  | Output | Multiplexer 2 output           |
|                   | GPIO_00                   | I/O    | General-purpose I/O port 0     |
| E2                | ACB_SDA                   | I/O    | ACCESS.bus data signal         |
| E1                | ACB_SCL                   | I/O    | ACCESS.bus clock signal        |
| E4                | PWM_0                     | Output | Pulse-width modulated output 0 |
|                   | MUX_IN1                   | Input  | Multiplexer 1 input 1          |
| F5                | PWM_1                     | Output | Pulse-width modulated output 1 |
|                   | MUX_IN2                   | Input  | Multiplexer 1 input 2          |
| E5                | PWM_2                     | Output | Pulse-width modulated output 2 |
|                   | MUX1_OUT                  | Output | Multiplexer 1 output           |
|                   | CONFIG_2                  | Input  | Slave address select input 2   |
|                   | GPIO_15                   | I/O    | General-purpose I/O port 15    |
| D6                | CONFIG_1                  | Input  | Slave address select input 1   |
|                   | GPIO_14                   | I/O    | General-purpose I/O port 14    |
| D1                | XTAL_OUT                  | Output | 32.768 kHz crystal output      |
| D2                | SLOWCLK                   | Input  | 32.768 kHz clock               |
|                   | XTAL_IN                   | Input  | 32.768 kHz crystal input       |
| F3                | $\overline{\text{IRQ}}$   | Output | Interrupt request output       |
| C1                | $\overline{\text{RESET}}$ | Input  | Reset Input                    |
| A4, F4            | VCC                       | N/A    | V <sub>CC</sub>                |
| C3, C4,<br>D3, D4 | GND                       | N/A    | Ground                         |

## TERMINATION OF UNUSED SIGNALS

### TERMINATION OF UNUSED SIGNALS

| Signal                    | Termination  |
|---------------------------|--|
| $\overline{\text{RESET}}$ | Connect to VCC if not driven from an external Supervisory circuit.   |
| CONFIG_1                  | Connect to VCC or GND through a pullup or pulldown resistor because the slave address is selected by the level on this pin. This pin cannot be left unconnected.   |
| XTAL_IN                   | This pin is a high-impedance input and must be connected to VCC or GND if it is unused.  |
| XTAL_OUT                  | This pin has a weak pullup and can be left open-circuit if it is unused.   |
| KP-X[2:0]                 | These pins are dedicated keypad pins. In the minimum configuration, these pins are keypad inputs with weak pullups.  |
| KP-X[7:3]                 | These pins are in high-impedance mode after power-on initialization. There are two ways to handle these pins if unused:  |
|                           | <ul style="list-style-type: none"> <li>Connect to VCC or GND.</li> </ul>   |
|                           | <ul style="list-style-type: none"> <li>Program as inputs with weak pullups or outputs.</li> </ul>  |
|                           | Care must be taken when connecting to VCC or GND. Erroneous parameters sent with the WRITE_PORT_SEL or WRITE_PORT_STATE commands could cause excessive current consumption. A better approach is to leave unused keyboard inputs open-circuit and use the WRITE_PORT_SEL and WRITE_PORT_STATE commands to configure the pins as inputs with weak pullups or outputs. |
| KP-Y[2:0]                 | KP-X7 can only be an input. This pin should be programmed as an input with a weak pullup.  |
| KP-Y[11:3]                | These pins are dedicated keypad pins. In the minimum configuration, these pins are keypad outputs driven low.  |
| KP-Y[11:3]                | These pins are in high-impedance mode after power-on initialization. There are two ways to handle these pins if unused:  |
|                           | <ul style="list-style-type: none"> <li>Connect to VCC or GND.</li> </ul>   |
|                           | <ul style="list-style-type: none"> <li>Program as inputs with weak pullups or outputs</li> </ul>   |
|                           | Care must be taken when connecting to VCC or GND. Erroneous parameters sent with the WRITE_PORT_SEL or WRITE_PORT_STATE commands could cause excessive current consumption. A better approach is to leave unused keyboard inputs open-circuit and use the WRITE_PORT_SEL and WRITE_PORT_STATE commands to configure the pins as inputs with weak pullups or outputs. |
| PWM_0,<br>PWM_1           | These pins must be connected to VCC or GND if they are not used for any optional function described in the datasheet.  |
| PWM_2/<br>CONFIG_2        | Connect to VCC or GND through a pullup or pulldown resistor because the slave address is selected by the level on this pin. This pin cannot be left unconnected.   |
| $\overline{\text{IRQ}}$   | This pin must be connected.  |

## APPLICATION EXAMPLE

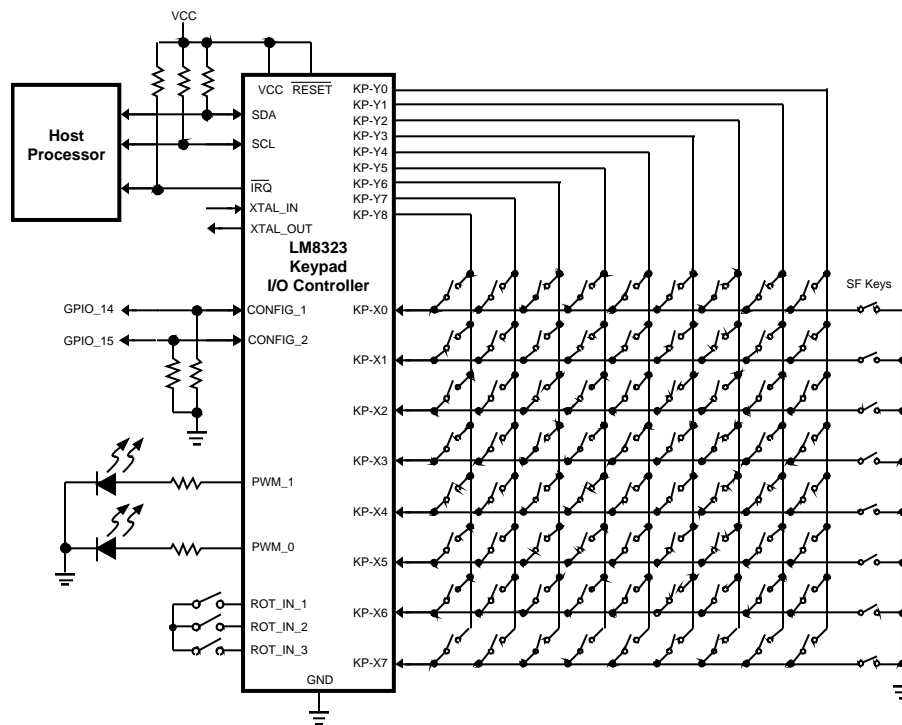


Figure 2. Typical Application

## FEATURES

The application example shown in [Figure 2](#) supports the following features:

- 8 x 9 standard keys.
- 8 special function keys (SF keys) with wake-up capability by forcing a WAKE\_INx pin to ground. Pressing a SF key overrides any other key in the same row.
- ACCESS.bus (I<sup>2</sup>C-compatible) interface for communication with the host.
- Hardware  $\overline{\text{IRQ}}$  interrupt to host to signal keypad, rotary encoder, error, and status events. By default, this is an open-drain output, so an external pullup resistor may be required to avoid false assertion. The host can program this output for push-pull mode, in which case the pullup might not be required, if the host can ignore a false assertion before the LM8323 has been programmed.
- Two LEDs driven by PWM outputs with programmable ramp-up and ramp-down. PWM\_2 (shared with GPIO\_15 and CONFIG\_2) could be used as an additional PWM driver port to control a third external LED.
- Rotary encoder interface shares pins with KP-Y9, KP-Y10, and KP-Y11. For larger keyboard configurations (such as QWERTY layouts), the rotary encoder interface is not available.
- ACCESS.bus address is selected by the CONFIG\_1 and CONFIG\_2 inputs. These pins may also be used as GPIO pins after reset initialization has occurred. If extra GPIO pins are not needed, CONFIG\_1 and CONFIG\_2 may be tied directly to VCC and GND.
- Crystal pins XTAL\_IN and XTAL\_OUT may be used to connect to an external 32.768 kHz crystal or receive an external 32.768 kHz clock input for running the PWM peripheral. By default, the PWM is clocked by an on-chip clock source.

## CLOCKS

- **System Clock (mclk)** — The system clock is in the range of about 21 MHz ( $\pm 7\%$ ) typical. This clock is used to drive the I<sup>2</sup>C-compatible serial ACCESS bus and is the input clock for other function blocks.
- **Processing and Command Execution Clock (tc)** — The internal processing is based on a 2MHz clock. This clock is derived from the System Clock.

- **Internal PWM Clock** — The internal PWM clock is a fixed scaled down clock ( $\div 64$ ) of the Processing and Command Execution Clock. This clock is close to 32 kHz which is in a good range to source the PWM function block as an alternative to an external clock source.
- **External 32.768 kHz Clock** — driven into the SLOWCLK input. May be used internally as the timebase for the PWM and driven on the SLOWCLKOUT output.
- **External 32.768 kHz Crystal** — connected across the XTAL\_IN and XTAL\_OUT pins (XTAL\_IN is an alternate function of the SLOWCLK pin). May be used internally as the timebase for the PWM and driven on the SLOWCLKOUT output.

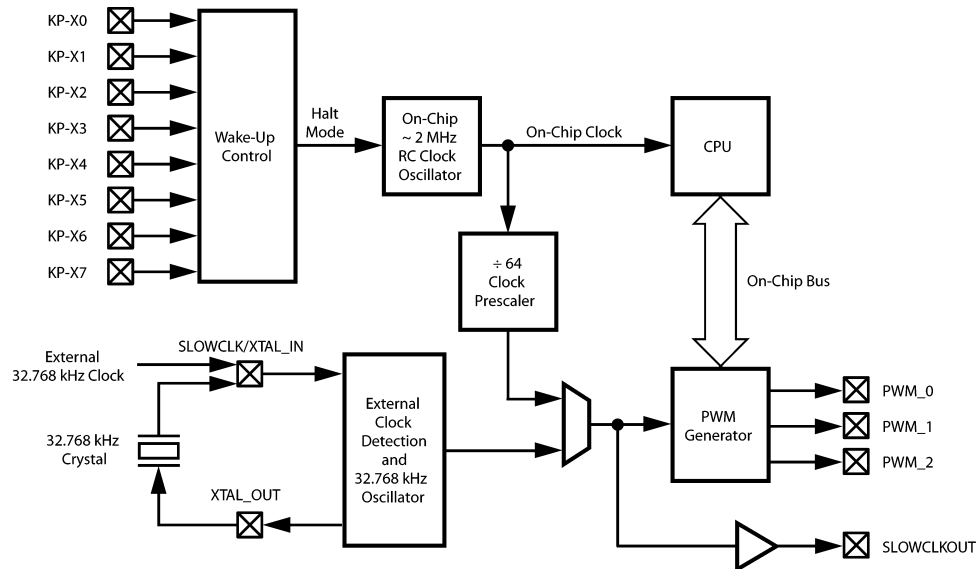


Figure 3. Clock Architecture

## INTERNAL EXECUTION CYCLE

The Processing - and Command - execution clock is about 2MHz. This clock is stopped in Halt mode, which only occurs under control of the LM8323. However, the host can set the period of inactivity which causes the device to enter Halt mode.

Exit from Halt mode can be triggered by any of these events:

- Occurrence of a key-press or key-release event.
- A Start condition driven by the host on the ACCESS.bus interface.
- Any change to the rotary encoder counter value (if the interface is enabled).
- Assertion of the RESET input.

After reset, the default timebase for the PWM outputs is the internal execution clock divided by 64.

## BUFFERED CLOCK

The timebase for the PWM comes from any of three sources:

- Prescaled internal Execution clock.
- External 32.768 kHz clock received on the SLOWCLK input.
- On-chip oscillator with an external crystal connected across XTAL\_IN and XTAL\_OUT.

Any of these sources may be buffered and driven on the SLOWCLKOUT output. The clock buffer is enabled with the WRITE\_CLOCK command.

If XTAL\_IN is not used it must be terminated to VCC or GND.

## CLOCK CONFIGURATION

Table 1 shows the clock configurations available by loading the clock configuration register with the WRITE\_CLOCK command. The WRITE\_CLOCK command must be issued only once during system initialization. This command is used to override the default settings.

**Table 1. Clock Configuration Register**

| 7 | 6          | 5 | 4 | 3         | 2 | 1     | 0 |
|---|------------|---|---|-----------|---|-------|---|
| 0 | SLOWCLKOUT | 0 | 0 | SLOWCLKEN | 0 | RCPWM |   |

| Bit        | Value | Description   |
|------------|-------|---|
| SLOWCLKOUT | 0     | Disable SLOWCLKOUT buffer.  |
|            | 1     | Enable SLOWCLKOUT buffer.   |
| SLOWCLKEN  | 0     | External 32.768 kHz crystal is installed between the XTAL_IN and XTAL_OUT pins.               |
|            | 1     | External 32.768 kHz clock is received on the SLOWCLK pin, or no 32.768 kHz clock is required. |
| RCPWM      | 00    | On-chip RC clock divided by 64 drives the PWM and clock buffer.                               |
|            | 01    | Reserved  |
|            | 10    | Reserved  |
|            | 11    | External 32.768 kHz clock or crystal drives the PWM and clock buffer.                         |

The SLOWCLKOUT signal is an alternate function of the pin used for the KP-Y8 scanning output and the GPIO\_03 port. If the SLOWCLKOUT function is enabled, these other functions of the pin are unavailable.

## RESET

The LM8323 may be reset by either an external reset,  $\overline{\text{RESET}}$  command, or an internally generated power-on reset (POR) signal. The  $\overline{\text{RESET}}$  input must not be allowed to float. If the external  $\overline{\text{RESET}}$  input is not used, it must be connected to VCC, either directly or through a pull-up resistor.

### EXTERNAL RESET

The device enters a reset state immediately when the  $\overline{\text{RESET}}$  input is driven low.  $\overline{\text{RESET}}$  must be held low for a minimum of 700 ns to ensure a valid reset. If  $\overline{\text{RESET}}$  is asserted at power-on, it must be held low until  $V_{\text{CC}}$  rises above the minimum operating voltage (1.62V). If an RC circuit is used to drive  $\overline{\text{RESET}}$ , it must have a time constant 5 times (5x) greater than the  $V_{\text{CC}}$  rise time to this level.

When  $\overline{\text{RESET}}$  goes low, the I/O ports are initialized immediately, any observed delay being only propagation delay. When the  $\overline{\text{RESET}}$  pin goes high, the LM8323 comes out of the reset state within about 1400 ns.

### POWER-ON RESET (POR)

The POR circuit is always enabled. When  $V_{\text{CC}}$  rises above the POR threshold voltage  $V_{\text{POR}}$  (about 1.2–1.5V), an on-chip reset signal is asserted. The  $V_{\text{CC}}$  rise time must be greater than 20  $\mu\text{s}$  and less than 10 ms, otherwise the on-chip reset signal may deassert before  $V_{\text{CC}}$  reaches the minimum operating voltage. While  $V_{\text{CC}}$  is below  $V_{\text{POR}}$ , the LM8323 is held in reset and a timer clocked by the on-chip RC clock is preset with 0xFF (256 clock cycles). When  $V_{\text{CC}}$  reaches a value greater than  $V_{\text{POR}}$ , the timer starts counting down. When it underflows, the on-chip reset signal is deasserted and the LM8323 begins operation.

### PIN CONFIGURATION AFTER RESET

Table 2 shows the pin configuration after reset.



**Table 2. Pin Configuration After Reset**

| Pins   | After Reset          | After LM8323 Initialization   |
|--|----------------------|---|
| KP-X00<br>KP-X01<br>KP-X02   | High-impedance mode. | Input mode with an on-chip pullup enabled.  |
| KP-X03<br>KP-X04<br>KP-X05<br>KP-X06<br>KP-X07   | High-impedance mode. | High-impedance mode, until host configures them as keypad inputs or GPIO.   |
| KP-Y00<br>KP-Y01<br>KP-Y02   | High-impedance mode. | Active drive low.   |
| KP-Y03<br>KP-Y04<br>KP-Y05<br>KP-Y06<br>KP-Y07<br>KP-Y08<br>KP-Y09<br>KP-Y10<br>KP-Y11 | High-impedance mode. | High-impedance mode, until host configures them as keypad outputs or GPIO.  |
| CONFIG_1<br>CONFIG_2   | High-impedance mode. | The ACCESS.bus slave address must be selected with external pullup or pulldown resistors or direct connections to VCC or GND. |
| $\overline{\text{IRQ}}$  | High-impedance mode. | Active drive low.   |
| PWM_0<br>PWM_1<br>PWM_2  | High-impedance mode. | High-impedance mode.  |
| ACB_SDA<br>ACB_SCL   | Open-drain mode.     | Open-drain mode.  |
| XTAL_IN  | High-impedance mode. | High-impedance mode. Terminate to VCC or GND if not used.   |
| XTAL_OUT   | Weak pullup device.  | Weak pullup device.   |
| $\overline{\text{RESET}}$  | High-impedance mode. | High-impedance mode.  |

## DEVICE CONFIGURATION AFTER RESET

After the LM8323 has completed its reset initialization, it will have the following internal configuration:

- **PWM Clock:** The PWM clock source is the on-chip clock divided by 64. This remains in effect until changed by a host command.
- **Keypad Size:** 3 × 3.
- **Rotary Encoder Interface:** disabled.
- **Digital Multiplexers:** disabled.
- $\overline{\text{IRQ}}$ : enabled, active low.
- **NOINIT Bit** : set.
- **Debounce Time:** 3 scan cycles (about 12 milliseconds).
- **Active Time:** 500 milliseconds.

**NOTE**

When FW6 version devices receive a  $\overline{\text{RESET}}$  command the  $\overline{\text{IRQ}}$  line is set high and held high for 60 ms and then pulled low to show the device was successfully reset and is ready to be used.

**CONFIGURATION INPUTS**

The states sampled from the CONFIG\_1 and CONFIG\_2 inputs during reset select the ACCESS.bus address used by the LM8323, as shown in Table 3. The address occupies the high seven bits of the first byte of a bus transaction, with the LSB (shown as X below) indicating the direction of transfer.

**Table 3. Bus Address Selection**

| CONFIG_1 | CONFIG_2 | Bus Address |
|----------|----------|-------------|
| 0        | 0        | 1000 010X   |
| 0        | 1        | 1000 011X   |
| 1        | 0        | 1000 100X   |
| 1        | 1        | 1000 101X   |

When these pins are used as GPIO ports, the design must ensure that they have the desired states during reset. For example, a 100-k $\Omega$  resistor to ground can impose a logic 0 during reset without interfering with normal operation as a GPIO port.

**INITIALIZATION**

The LM8323 waits for a WRITE\_CFG command from the host. During this time,  $\overline{\text{IRQ}}$  is asserted to request service from the host. Figure 4 describes the behavior of the LM8323 following reset.

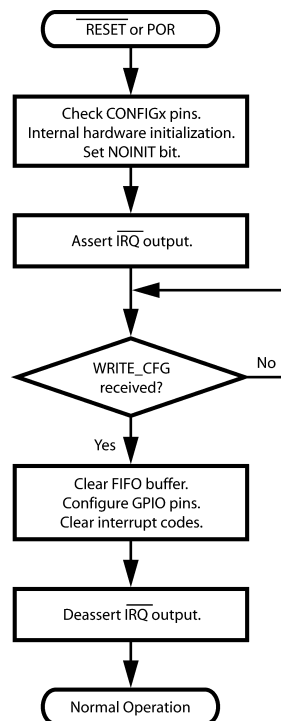
**Figure 4. LM8323 Initialization Behavior**

Figure 5 shows the timing of  $\overline{\text{IRQ}}$  relative to a  $\overline{\text{RESET}}$  or POR event and the WRITE\_CFG command. 100  $\mu\text{s}$  after a  $\overline{\text{RESET}}$  or POR event,  $\overline{\text{IRQ}}$  is asserted and any READ\_INT command will return an interrupt code with the NOINIT bit set. 90  $\mu\text{s}$  after a WRITE\_CFG command is received,  $\overline{\text{IRQ}}$  is deasserted.

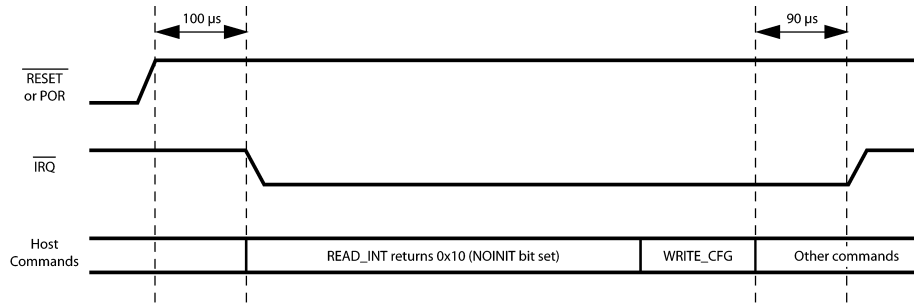


Figure 5.  $\overline{\text{IRQ}}$  Reset Timing

After sending the WRITE\_CFG command, the host must send a series of commands to configure the LM8323, as shown in Figure 6. (See left hand side.)

This Flow - diagram illustrates also the basic host code communication steps which the host must execute upon an  $\overline{\text{IRQ}}$  request received from the LM8323 during operation. Such requests will be made from the LM8323 as a result of key pressed events, the detection of an error, the termination of a PWM cycle and others.

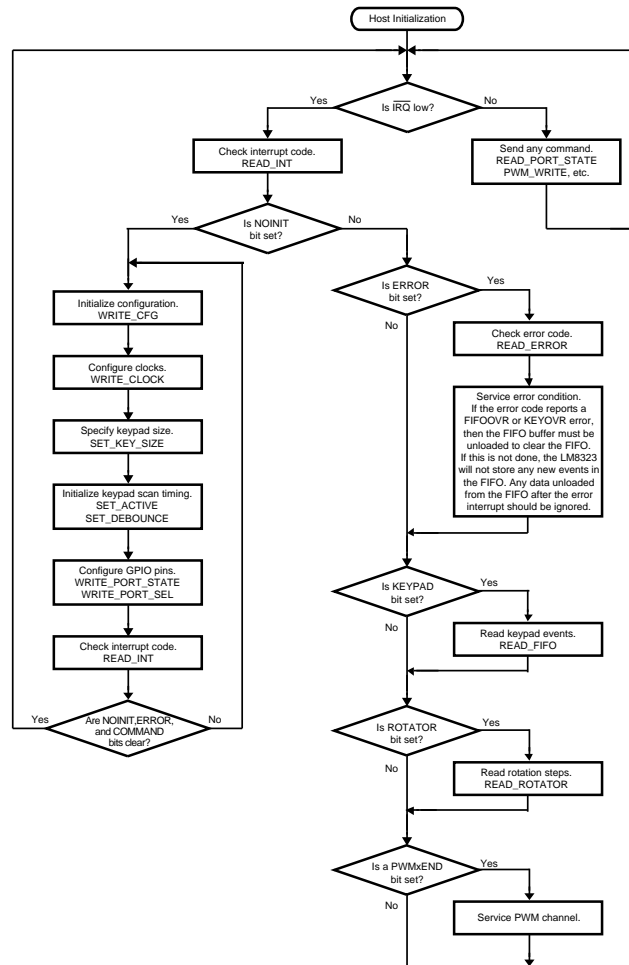


Figure 6. Host-Side LM8323 Initialization

## INITIALIZATION EXAMPLE

In the following example, the LM8323 is configured as:

- Keypad matrix configuration is 8 × 4.
- Rotary encoder interface enabled.
- GPIO\_03 through GPIO\_07 are available to use as GPIO pins.
- GPIO\_03 is an output driven low.
- GPIO\_4 and GPIO\_5 are outputs driven high.
- GPIO\_06 and GPIO\_07 are inputs with weak pulldowns.
- GPIO\_14 and GPIO\_15 are inputs with weak pullups.
- The PWM clock source is the internal execution clock divided by 64 (about 32 kHz).

Most of these settings can be verified by executing commands such as READ\_CONF, READ\_PORT\_SEL, READ\_CLOCK, etc.

ALL GPIO pin states can be read using the READ\_PORT\_STATE command, without regard to whether the pin is an input or an output.

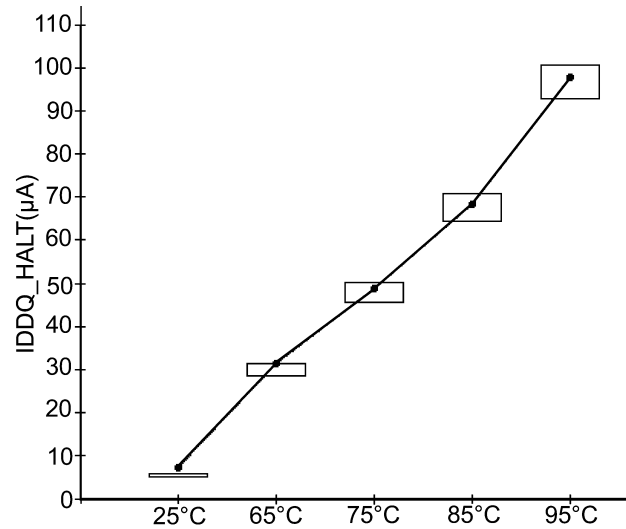
An open-drain signal can be created by alternating between input mode and driving the output low.

All GPIOs can sink and source 16 mA when configured as an output.

| Command          | Encoding | Parameter 1 | Parameter 2 | Description   |
|------------------|----------|-------------|-------------|---|
| WRITE_CFG        | 0x81     | 0x40        |             | Selects 36-pin package and disables the two digital multiplexers.   |
| WRITE_CLK        | 0x93     | 0x08        |             | SLOWCLKOUT disabled, no external 32.768 kHz clock required, PWM clock source is internal.   |
| SET_KEY_SIZE     | 0x90     | 0x84        |             | Selects a keypad matrix size of 8 × 4.  |
| SET_ACTIVE       | 0x8B     | 0x4B        |             | Sets the active time to about 300 milliseconds (75 × 4 milliseconds).   |
| SET_DEBOUNCE     | 0x8F     | 0x03        |             | Sets the key debouncing time to about 12 milliseconds (3 × 4 ms). This is actually the default and would not have to be performed.  |
| WRITE_PORT_SEL   | 0x85     | 0x00        | 0x38        | Configure GPIO_03, GPIO_04, and GPIO_05 as outputs. Configure GPIO_06, GPIO_07, GPIO_14, and GPIO_15 as inputs.   |
| WRITE_PULL_DOWN  | 0x84     | 0x00        | 0x3F        | Set the direction for the pullup/pulldown devices on GPIO_06 and GPIO_07 to pulldown. Set the direction for the pullup/pulldown devices on GPIO_14 and GPIO_15 to pullup. |
| WRITE_PORT_STATE | 0x86     | 0xC0        | 0xF0        | Set GPIO_04 and GPIO_05 to drive high. Enable the pullups on GPIO_06, GPIO_07, GPIO_14, and GPIO_15.  |

## HALT MODE

The fully static architecture of the LM8323 allows stopping the internal RC clock in Halt mode, which reduces power consumption to the minimum level. [Figure 7](#) shows the current in Halt mode at the maximum V<sub>CC</sub> (1.98V) from 25°C to +85°C.



**Figure 7. Halt Current vs. Temperature at 1.98V**

Halt mode is entered when no key-press event, key-release event, change in the rotary encoder counter value or ACCESS.bus activity is detected for a certain period of time (by default, 500 ms). The mechanism for entering Halt mode is always enabled in hardware, but the host can program the period of inactivity which triggers entry into Halt mode.

**NOTE**

When FW4 version devices enter the Halt mode there is approximately a 33% chance the device may miss key events during the period of 3ms before entering Halt mode until 3ms after entering Halt mode resulting in lost key events. This was corrected in FW6 devices so that 100% of all key events are captured, even as the device is entering Halt mode.

**ACCESS.bus ACTIVITY**

When the LM8323 is in Halt mode, any activity on the ACCESS.bus interface will cause the LM8323 to exit from Halt mode. However, the LM8323 will not be able to acknowledge the first bus cycle immediately following wake-up from Halt mode. It will respond with a negative acknowledgement, and the host should then repeat the cycle.

The LM8323 will be prevented from entering Halt mode if it shares the bus with peripherals that are continuously active. For lowest power consumption, the LM8323 should only share the bus with peripherals that require little or no bus activity after system initialization.

**KEYPAD INTERFACE**

**EVENT CODE ASSIGNMENT**

After power-on reset and host initialization, the LM8323 starts scanning the keypad. It stays active for a default time of about 500 ms after the last key is released, after which it enters Halt mode to minimize power consumption (typically <9 µA standby current).

Table 4 lists the codes assigned to the matrix positions encoded by the hardware. Key-press events are assigned the codes listed in Table 4, but with the MSB set. When a key is released, the MSB of the code is clear.

**Table 4. Keypad Matrix Code Assignments**

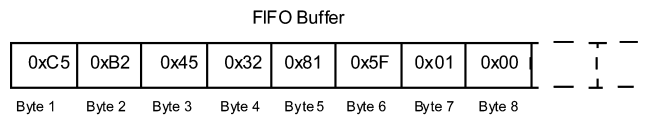
|              | KP-Y0 | KP-Y1 | KP-Y2 | KP-Y3 | KP-Y4 | KP-Y5 | KP-Y6 | KP-Y7 | KP-Y8 | KP-Y9       | KP-Y10      | KP-Y11      | SF Keys |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------------|-------------|-------------|---------|
| <b>KP-X0</b> | 0x01  | 0x02  | 0x03  | 0x04  | 0x05  | 0x06  | 0x07  | 0x08  | 0x09  | <b>0x0A</b> | <b>0x0B</b> | <b>0x0C</b> | 0x0F    |
| <b>KP-X1</b> | 0x11  | 0x12  | 0x13  | 0x14  | 0x15  | 0x16  | 0x17  | 0x18  | 0x19  | <b>0x1A</b> | <b>0x1B</b> | <b>0x1C</b> | 0x1F    |
| <b>KP-X2</b> | 0x21  | 0x22  | 0x23  | 0x24  | 0x25  | 0x26  | 0x27  | 0x28  | 0x29  | <b>0x2A</b> | <b>0x2B</b> | <b>0x2C</b> | 0x2F    |
| <b>KP-X3</b> | 0x31  | 0x32  | 0x33  | 0x34  | 0x35  | 0x36  | 0x37  | 0x38  | 0x39  | <b>0x3A</b> | <b>0x3B</b> | <b>0x3C</b> | 0x3F    |
| <b>KP-X4</b> | 0x41  | 0x42  | 0x43  | 0x44  | 0x45  | 0x46  | 0x47  | 0x48  | 0x49  | <b>0x4A</b> | <b>0x4B</b> | <b>0x4C</b> | 0x4F    |
| <b>KP-X5</b> | 0x51  | 0x52  | 0x53  | 0x54  | 0x55  | 0x56  | 0x57  | 0x58  | 0x59  | <b>0x5A</b> | <b>0x5B</b> | <b>0x5C</b> | 0x5F    |
| <b>KP-X6</b> | 0x61  | 0x62  | 0x63  | 0x64  | 0x65  | 0x66  | 0x67  | 0x68  | 0x69  | <b>0x6A</b> | <b>0x6B</b> | <b>0x6C</b> | 0x6F    |
| <b>KP-X7</b> | 0x71  | 0x72  | 0x73  | 0x74  | 0x75  | 0x76  | 0x77  | 0x78  | 0x79  | <b>0x7A</b> | <b>0x7B</b> | <b>0x7C</b> | 0x7F    |

When the rotary encoder interface is enabled, KP-Y9, KP-Y10, and KP-Y11 (**bolded** in Keypad Matrix Code Assignments) become unavailable for keypad scanning, which limits the keypad to a maximum size of  $8 \times 9 + 8$  SF keys.

The codes are loaded into the FIFO buffer in the order in which they occurred. [Table 5](#) shows an example sequence of events, and [Figure 8](#) shows the resulting sequence of event codes loaded into the FIFO buffer.

**Table 5. Example Sequence of Events**

| Event Number | Event Code | Event on Input | Driven Output | Description                    |
|--------------|------------|----------------|---------------|--------------------------------|
| 1            | 0xC5       | KP-X4          | KP-Y4         | Key is pressed                 |
| 2            | 0xB2       | KP-X3          | KP-Y1         | Key is pressed                 |
| 3            | 0x45       | KP-X4          | KP-Y4         | Key is released                |
| 4            | 0x32       | KP-X3          | KP-Y1         | Key is released                |
| 5            | 0x81       | KP-X0          | KP-Y0         | Key is pressed                 |
| 6            | 0x5F       | KP-X5          | N/A           | SF Key is released             |
| 7            | 0x01       | KP-X0          | KP-Y0         | Key is released                |
| 8            | 0x00       | N/A            | N/A           | Indicates end of stored events |

**Figure 8. Example Event Codes Loaded in FIFO Buffer**

## KEYPAD SCAN CYCLES

The LM8323 starts new scan cycles at fixed time intervals of about 4 milliseconds. If a change in the state of the keypad is detected, the keypad is rescanned after a debounce delay. When the state change has been reliably captured, it is encoded and written to the FIFO buffer.

[Figure 9](#) shows the relationship between a KP-Yx output and a KP-Xx input over multiple scan cycles during a key press event. Between scan cycles, the KP-Yx outputs that are specified by the SET\_KEY\_SIZE command (0x90) for keypad scanning are driven low.

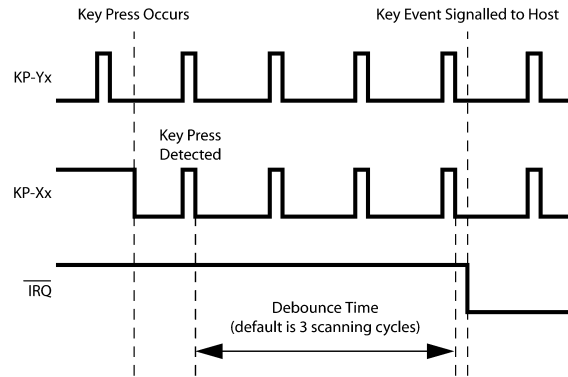


Figure 9. Keypad Scan Cycles

During a scan cycle, only one KP-Yx output pin will be driven low at any time, while the others are driven high or undriven. At the time scale used in Figure 9, the low phase of a KP-Yx output during a scan cycle is not visible. The KP-Xx input pins are pulled high by weak pullups.

There are capacitive loads on the KP-Xx inputs and KP-Yx outputs due to protection circuits, wiring, etc. The LM8323 inserts delays to allow complete charging or discharging of these loads before sampling the input levels on the KP-Xx inputs. The maximum parasitic load capacitance on the KP-Xx inputs is 5nF.

After detecting a key-press or key-release event, the debounce time specified by the SET\_DEBOUNCE command (0x8F) sets the minimum time for confirming the event before the  $\overline{\text{IRQ}}$  output is asserted.

If more than two keys are pressed simultaneously, the pattern of key closures may be ambiguous, in which case the interrupt code indicates an error and the  $\overline{\text{IRQ}}$  output is asserted (if enabled).

The SF keys connect KP-Xx inputs directly to ground. There can be up to eight SF keys. If any of these keys are pressed, other keys that use the same KP-Xx pin are ignored.

### Timing Parameters

Two timing parameters affect scanning of the keypad:

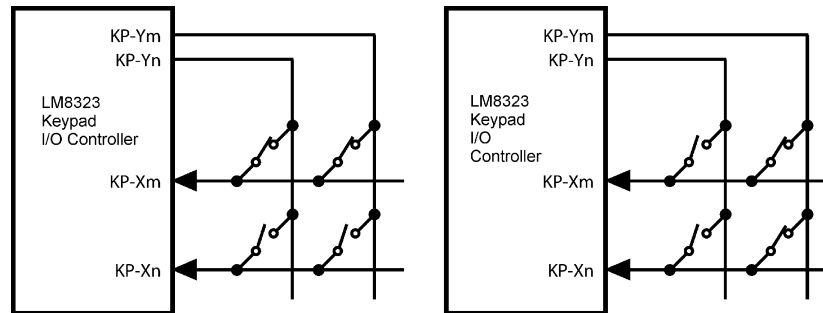
- **Debounce Time** — minimum delay between detecting a keypad event and confirming the event before asserting  $\overline{\text{IRQ}}$ . The default debounce time is 3 scan cycles (about 12 milliseconds), but the host can set values in the range 1–255 cycles (4–1020 milliseconds).
- **Active Time** — period without detecting a state change in the keypad or rotary encoder that triggers entry into Halt mode, during which keypad scanning is suspended. The default active time is 500 milliseconds, but the host can set it values in the range 4–1020 milliseconds. The active time must be greater than the debounce time.

### Multiple Key Pressings

If more than two keys are pressed at the same time, the LM8323 stores all key pressed and released events in the FIFO buffer in the sequence in which they were decoded.

For multiple key pressings the following circumstances have to be respected:

- A multiple key-press event is given if two or more key-press events are reported but no corresponding key-release event.
- With the activity time set between the minimum and maximum time (4 ms to 1 second) it is not safe to detect two simultaneous key pressings in one input row (see Figure 10 on the left hand side.)
- If all key pressings (two or more) are located in different input rows (see Figure 10 on the right hand side) then the key pressed events will be correctly found in the FIFO buffer without any restriction.



**Figure 10. Simultaneous Keys Pressed**

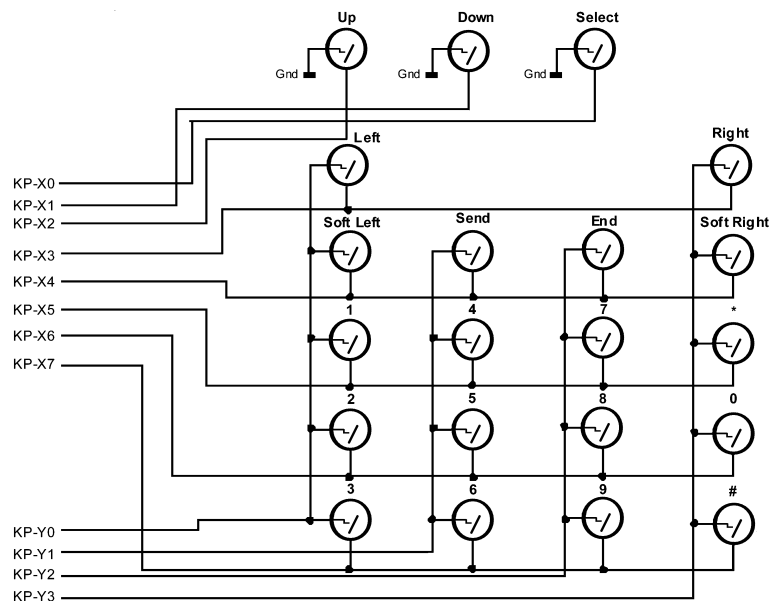
- In order to securely detect and store the key codes of simultaneous key pressings in the same input row the following precautions must be taken from the host side:

As soon as the host device has detected a key pressed event the host must send the SET\_ACTIVE Command with the parameter set to "00". This will prevent the LM8323 from entering HALT mode. If all keyboard events are resolved (no remaining key pressed status in the LM8323 anymore) then the host must send the SET\_ACTIVE Command again with the parameter setting the desired duration for the active time. This will enable the LM8323 to enter low power HALT mode once the activity time has passed without detecting any events.

- Once one or more key (pressed and/or released) events have been read from the host with the help of the READ FIFO command there are two conditions cleaning the FIFO buffer contents:
  - A second execution of the READ FIFO Command or,
  - A new key event detected from the LM8323.

### EXAMPLE KEYPAD CONFIGURATION

Figure 11 shows an 8 × 4 keypad matrix. This configuration occupies all scanning inputs (KP-X0 through KP-X7) and four scanning outputs (KP-Y0 through KP-Y3). The remaining scanning outputs KP-Y4 through KP-Y11 are available for use as GPIO pins. Enabling the rotary encoder interface reduces the number of available GPIO pins to KP-Y4 through KP-Y8.



**Figure 11. Keypad Interface Example**



In the example above, three keys (Up, Down, and Select) are connected as SF keys (connected directly to ground). Although they could have shared the KP-Xx inputs used with the scanned keys, the advantage of placing them on their own KP-Xx inputs is that it allows scanning the keypad while an SF key is pressed. If an SF key shares a KP-Xx input with any scanned keys, pressing the SF key prevents the LM8323 from reading the scanned keys.

The SET\_KEY\_SIZE command includes a data byte that specifies the keypad size. The upper 4 bits of the data byte specify the number of KP-Xx inputs, and the lower 4 bits specify the number of KP-Yx outputs. The minimum number of inputs and outputs is 3. Therefore, the minimum keypad configuration supports 3 × 3 + 3 SF keys (total of 12 keys). The maximum number of KP-Xx inputs is 8, and the maximum number of KP-Yx pins is 12. All KP-Xx and KP-Yx pins not used for the keyboard interface can be used for general-purpose I/O.

For the example shown in [Figure 11](#), the SET\_KEY\_SIZE command would specify 8 KP-Xx inputs and 4 KP-Yx outputs.

## GENERAL-PURPOSE I/O PORTS

Any unused KP-Xx and KP-Yx pins may be used as general-purpose I/O (GPIO) port pins. The WRITE\_PORT\_SEL (0x85) command selects the port direction, in which a clear bit in the parameter to the command selects the input direction and a set bit selects the output direction.

The WRITE\_PORT\_STATE (0x86) command selects either the port level when configured as output (by the WRITE\_PORT\_SEL command) or when configured as an input selects between a high-impedance input or an input with a pullup or pulldown device. The selection between pullup or pulldown devices is controlled by the parameter bytes to the WRITE\_PULL\_DOWN (0x84) command. Clear bits in the parameter bytes select pullup devices, while set bits select pulldown devices.

[Table 6](#) shows the GPIO port configurations selected by the bits in the WRITE\_PORT\_SEL, WRITE\_PORT\_STATE, and WRITE\_PULL\_DOWN command parameters.

**Table 6. GPIO Port Control Bits**

| WRITE_PORT_SEL | WRITE_PORT_STATE | WRITE_PULL_DOWN | Description                |
|----------------|------------------|-----------------|----------------------------|
| 0              | 0                | x               | High-Impedance Input       |
| 0              | 1                | 0               | Input with Pullup Device   |
| 0              | 1                | 1               | Input with Pulldown Device |
| 1              | 0                | x               | Output, Drive Low          |
| 1              | 1                | x               | Output, Drive High         |

Any pins used as GPIO ports must be configured after the peripheral configuration has been initialized with the WRITE\_CFG command (0x81) and the keypad configuration has been initialized with the SET\_KEY\_SIZE command (0x90). The default keypad configuration after reset is a 3 × 3 keyboard matrix. The default GPIO configuration is an input with the pullup disabled.

## USING THE CONFIG\_X PINS FOR GPIO

The CONFIG\_1 and CONFIG\_2 pins are available for use as GPIO pins after power-on or reset. However, stable states must be provided on these pins during power-on or reset to select the I<sup>2</sup>C-compatible ACCESS.bus address.

External pullup or pulldown resistors can be used to pull either CONFIG\_x pin low, while retaining the ability to drive it to another state when used as a GPIO pin.

CONFIG\_2 has two alternate functions, in addition to GPIO. It can be configured as a multiplexer output using the WRITE\_CFG command (0x81), in which case it will not be available as a GPIO pin. It can also be configured as a PWM output, which also would override its use as a GPIO pin.

## USING THE ROT\_IN\_X PINS FOR GPIO

The rotary encoder interface uses alternate functions of KP-Y9, KP-Y10, and KP-Y11. The maximum keypad size is automatically reduced to a 8 × 9 matrix if the rotary encoder interface is enabled.

## GPIO TIMING

When a WRITE\_PORT\_STATE command (0x86) is received, the GPIO outputs do not change to their new states immediately or simultaneously. The first one changes 54  $\mu$ s after the command is acknowledged, and the others change at intervals of 7.3  $\mu$ s, as shown in Figure 12.

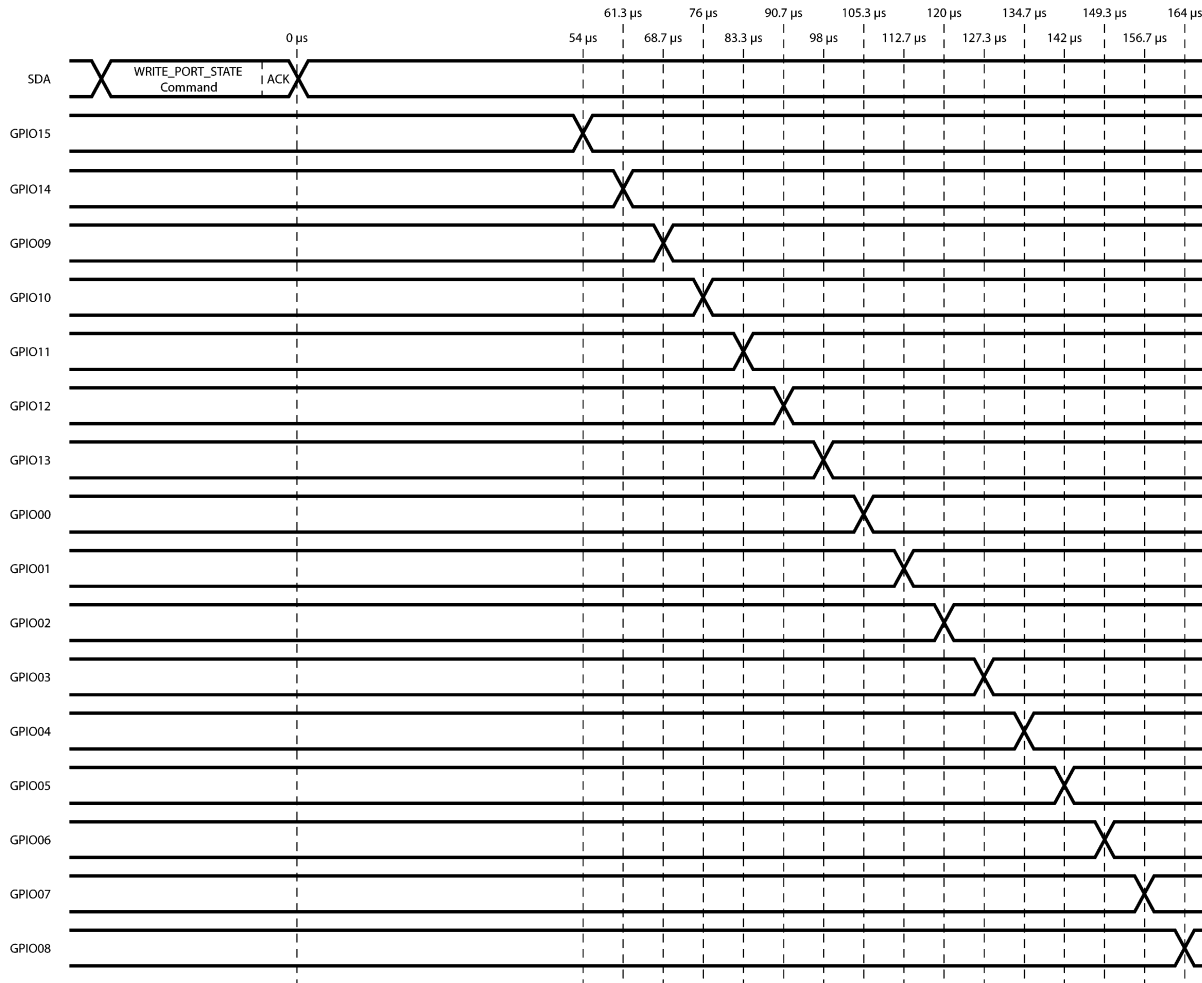


Figure 12. GPIO Port State Change Timing

## ROTARY ENCODER INTERFACE

A three-wire interface is provided for an external rotary encoder. Setting the ROTEN bit with the WRITE\_CFG command enables the interface and the ROT\_IN\_x inputs, which are alternate functions of certain keypad scanning pins. The ROT\_IN\_x inputs are bidirectional signals used to test the status of switches in an external rotary encoder, as shown in Figure 13.

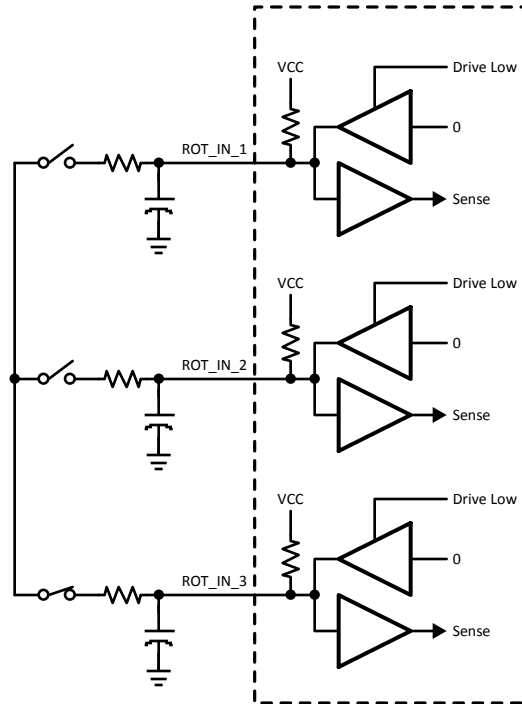


Figure 13. Rotary Encoder External Interface

The ROT\_IN\_x inputs are alternate functions of KP-Y9, KP-Y10, and KP-Y11. When the rotary encoder interface is enabled, these keypad scanning outputs are not available for keypad interface.

Steps which correspond to clockwise rotation increment a counter, while counterclockwise steps decrement the counter, as shown in the example sequence in Table 7. The READ\_ROTATOR command returns a data byte which indicates the accumulated count since the counter was last read.

Table 7. Rotary Encoder Example Sequence

| Switch 1-2 | Switch 2-3 | Switch 3-1 | Action    | Counter  |
|------------|------------|------------|-----------|----------|
| Closed     | Closed     | Open       | Increment | 00000000 |
| Open       | Closed     | Open       | No Change | 00000000 |
| Open       | Closed     | Closed     | Increment | 00000001 |
| Open       | Open       | Closed     | No Change | 00000001 |
| Closed     | Open       | Closed     | Increment | 00000010 |
| Closed     | Open       | Open       | No Change | 00000010 |
| Closed     | Closed     | Open       | Increment | 00000011 |
| Open       | Closed     | Open       | No Change | 00000011 |
| Open       | Closed     | Closed     | Increment | 00000100 |
| Open       | Open       | Closed     | No Change | 00000100 |
| Open       | Closed     | Closed     | Decrement | 00000011 |
| Open       | Closed     | Open       | No Change | 00000011 |
| Closed     | Closed     | Open       | Decrement | 00000010 |
| Closed     | Open       | Open       | No Change | 00000010 |
| Closed     | Open       | Closed     | Decrement | 00000001 |
| Open       | Open       | Closed     | No Change | 00000001 |
| Open       | Closed     | Closed     | Decrement | 00000000 |

**Table 7. Rotary Encoder Example Sequence (continued)**

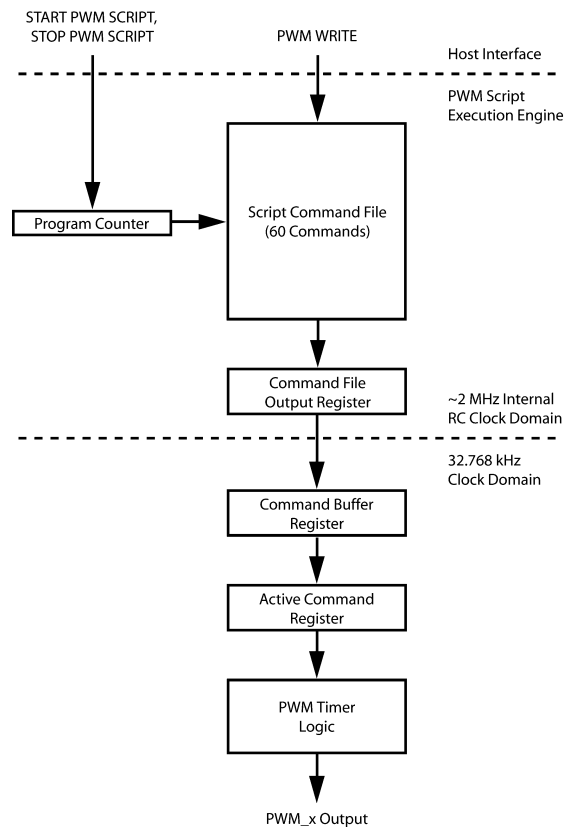
| Switch 1-2 | Switch 2-3 | Switch 3-1 | Action    | Counter  |
|------------|------------|------------|-----------|----------|
| Open       | Closed     | Open       | No Change | 00000000 |
| Closed     | Closed     | Open       | Decrement | 11111111 |
| Closed     | Open       | Open       | No Change | 11111111 |
| Closed     | Open       | Closed     | Decrement | 11111110 |
| Open       | Open       | Closed     | No Change | 11111110 |
| Open       | Closed     | Closed     | Decrement | 11111101 |

The value of the data byte is in two's complement form, in which positive values indicate clockwise rotation and negative values indicate counterclockwise rotation. This is shown in the example when the counter decrements below zero.

A rotary encoder event will only wake up the LM8323 from Halt mode if it changes the counter value.

## PWM OUTPUT GENERATION

Three pulse-width modulated (PWM) outputs are provided with advanced capabilities for ramp-up and ramp-down of the PWM duty cycle and execution of simple to complex command sequences. These capabilities are supported by three independent script-execution engines capable of autonomous operation after setup and launch by the host. [Figure 14](#) shows the architecture of a script-execution engine.

**Figure 14. PWM Script Execution Engine**

The host has three commands for interfacing to the script execution engine. The following commands are always associated with one particular PWM channel:

- **PWM\_WRITE** — load one word into the script command file at a specified address.
- **PWM\_START** — start execution of the script.
- **PWM\_STOP** — stop execution of the script.

**NOTE**

The PWM\_STOP command might not take immediate effect if the current command being executed is a command with long execution time. If a PWM\_STOP command is sent when the PWM engine is running a long RAMP command, the PWM will only stop after the RAMP is completed.

The script commands have their own fixed-length 16-bit format and encoding unrelated to the variable-length, byte-based format used for host commands. A script command is sent by the host to the LM8323 as a parameter to the PWM\_WRITE command. Another parameter to the PWM\_WRITE command specifies an address in the script command file for receiving the command.

**COMMAND QUEUE**

After the host issues a PWM\_START command, script commands are read from the script command file into a command queue which consists of a command file output register, command buffer, and active command register. This allows one command to be active while another command is queued in the command buffer, which allows seamless back-to-back command execution.

A command loaded into the command file output register is synchronized to the 32.768 kHz clock and stored in the command buffer. If no command is currently active, the command passes through to the active command register. In this case, another command can be read from the script command file, which is queued in the command buffer. On completion of the currently active command, the contents of the command buffer are transferred to the active command register, and the command buffer may then receive a new command.

The host does not have direct access to any of the registers in the command queue. The operations which read script commands from the script command file occur automatically after the host issues the PWM\_START command.

Script execution stops when the host sends a PWM\_STOP command or when the script engine executes an END command. Executing an END command asserts IRQ to the host.

**PWM TIMER OPERATION**

The timers implement a fixed 256-cycle period with a programmable duty cycle and programmable ramp-up/ramp-down of the duty cycle. Figure 15 shows the architecture of a PWM timer.

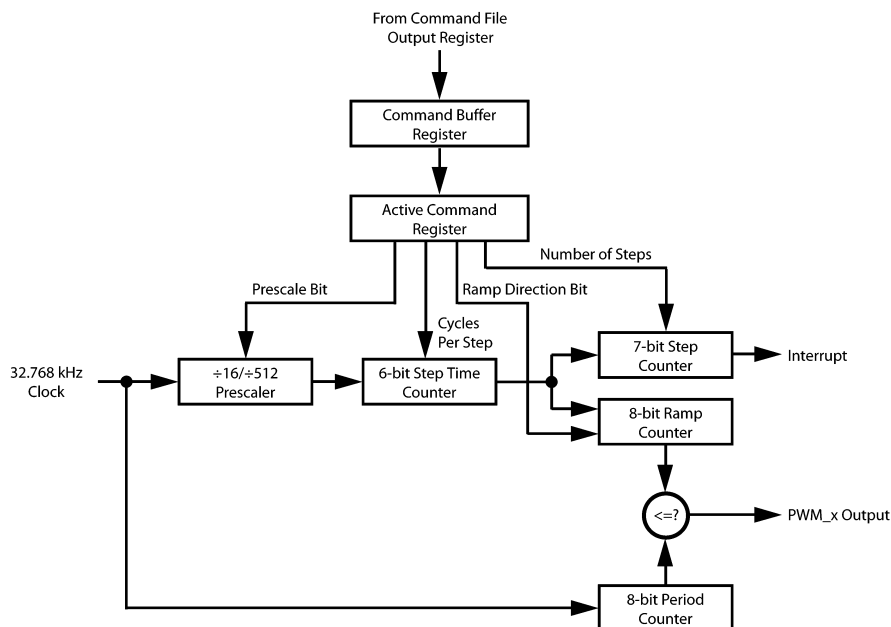


Figure 15. PWM Timer

The period counter is a free running 8-bit up-counter which starts counting when the script command file issues the first RAMP command. An END command stops the period counter.

The duty cycle of the PWM output is controlled by the ramp counter. If the PWM period counter is active, the PWM output signal is asserted while the period counter has a value less than or equal to the value of the ramp counter.

The ramp counter can increment or decrement at a rate controlled by the prescaler and step time counter. The prescaler selects a factor of 16 or 512 for dividing down the frequency of the 32.768 kHz clock. The ramp counter saturates at either 0x00 or 0xFF depending on the ramp direction.

The number of increment or decrement steps is specified by the INCREMENT field of the RAMP command, which is loaded into the step counter. Even if the ramp counter hits its saturation value, the requested number of steps will be performed. An option enables assertion of the  $\overline{\text{IRQ}}$  output to the host after the last step is performed.

## PWM SCRIPT COMMANDS

Table 8 summarizes the script commands.

**Table 8. PWM Script Commands**

| Command         | 15 | 14           | 13       | 12          | 11        | 10 | 9 | 8        | 7           | 6       | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------------|----|--------------|----------|-------------|-----------|----|---|----------|-------------|---------|---|---|---|---|---|---|
| RAMP            | 0  | PRES<br>CALE | STEPTIME |             |           |    |   | SIGN     | INCREMENT   |         |   |   |   |   |   |   |
| SET_PWM         | 0  | 1            | 0        |             |           |    |   | PWMVALUE |             |         |   |   |   |   |   |   |
| GO_TO_<br>START | 0  |              |          |             |           |    |   |          |             |         |   |   |   |   |   |   |
| BRANCH          | 1  | 0            | 1        | LOOPCOUNT   |           |    |   |          | 0           | ADDRESS |   |   |   |   |   |   |
| END             | 1  | 1            | 0        | 0           | RES<br>ET | 0  |   |          |             |         |   |   |   |   |   |   |
| TRIGGER         | 1  | 1            | 1        | WAITTRIGGER |           |    |   |          | SENDTRIGGER |         |   |   |   |   | 0 |   |

## RAMP COMMAND

The RAMP command generates a duty-cycle ramp starting from the current value. At each step, the ramp counter is incremented or decremented by one, unless it has reached its saturation value (0xFF for increment, or 0x00 for decrement). The time for one step is controlled by the PRESCALE bit and STEPTIME field. The minimum time for one step is 0.49 milliseconds, and the maximum time is about 1 second, which supports both very fast and very slow ramps. The INCREMENT field specifies the number of steps to be executed by the command. The maximum value is 126, which corresponds to half of full scale.

There are two special cases in the instruction encoding. If all bits and fields are 0, it is interpreted as the GO TO START command. If the STEPTIME field is 0 but any other bit or field is non-zero, it is interpreted as the SET\_PWM command.

| 15 | 14       | 13       | 12 | 11 | 10 | 9 | 8    | 7         | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----------|----------|----|----|----|---|------|-----------|---|---|---|---|---|---|---|
| 0  | PRESCALE | STEPTIME |    |    |    |   | SIGN | INCREMENT |   |   |   |   |   |   |   |

| Bit or Field | Value | Description                                  |
|--------------|-------|--|
| PRESCALE     | 0     | Divide the 32.768 kHz clock by 16            |
|              | 1     | Divide the 32.768 kHz clock by 512           |
| STEPTIME     | 1–63  | Number of prescaled clock cycles per step    |
| SIGN         | 0     | Increment ramp counter                       |
|              | 1     | Decrement ramp counter                       |
| INCREMENT    | 1–126 | Number of steps executed by this instruction |

## SET\_PWM COMMAND

The SET\_PWM command loads the ramp counter from the 8-bit DUTYCYCLE field in the instruction.

### NOTE

Only 0x00 and 0xFF are valid values for the duty cycle in SET\_PWM command. Other values can be established by initializing the duty cycle to either 100% or 0% followed by a RAMP command.

|    |    |    |    |    |    |   |   |           |   |   |   |   |   |   |   |
|----|----|----|----|----|----|---|---|-----------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7         | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0  | 1  | 0  | 0  | 0  | 0  | 0 | 0 | DUTYCYCLE |   |   |   |   |   |   |   |

| Bit or Field | Value | Description         |
|--------------|-------|---------------------|
| DUTYCYCLE    | 0     | Duty cycle is 0%.   |
|              | 255   | Duty cycle is 100%. |

## GO\_TO\_START COMMAND

The GO\_TO\_START command jumps to the first command in the script command file.

|    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

## BRANCH COMMAND

The BRANCH command jumps to the specified command in the script command file, with the option of looping for a specified number of repetitions. Nested loops are not allowed.

|    |    |    |           |    |    |   |   |   |   |         |   |   |   |   |   |  |
|----|----|----|-----------|----|----|---|---|---|---|---------|---|---|---|---|---|--|
| 15 | 14 | 13 | 12        | 11 | 10 | 9 | 8 | 7 | 6 | 5       | 4 | 3 | 2 | 1 | 0 |  |
| 1  | 0  | 1  | LOOPCOUNT |    |    |   |   |   | 0 | ADDRESS |   |   |   |   |   |  |

| Field     | Value | Description  |
|-----------|-------|--|
| LOOPCOUNT | 0     | Loop until a STOP PWM SCRIPT command is issued by the host.  |
|           | 1–63  | Number of repetitions to perform, biased by -1. The range is 0–62 repetitions.   |
| ADDRESS   | 0–59  | Branch destination address in the script command file. If this field is greater than 59, no looping will be performed. |

## END COMMAND

The END command terminates script execution and asserts an interrupt to the host if the RESET bit is set to “1” or “0”.

If the END command is executed with the RESET bit set to “1”, the PWM output will be disabled. If the RESET bit is “0” when executing the END command, the PWM channel remains active with the fixed duty cycle it was last set to.

### NOTE

If a PWM channel is waiting for the trigger (last executed command was "TRIGGER") and the script execution is halted then the "END" command can't be executed because the previous command is still pending. This is an exception - in this case the  $\overline{IRQ}$  signal will not be asserted.

|    |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
|----|----|----|----|-------|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11    | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1  | 1  | 0  | 0  | RESET | 0  |   |   |   |   |   |   |   |   |   |   |

| Bit   | Value | Description  |
|-------|-------|--|
| RESET | 0     | PWM_x output is active when script execution terminates.   |
|       | 1     | PWM_x output is Tristate when script execution terminates. |

## TRIGGER COMMAND

Triggers are used to synchronize operations between PWM channels. A TRIGGER command that sends a trigger takes sixteen 32.768 kHz clock cycles, and a command that waits for a trigger takes at least sixteen 32.768 kHz clock cycles.

A TRIGGER command that waits for a trigger (or triggers) will stall script execution until the trigger conditions are satisfied. Then, it will clear the trigger(s) and continue to the next command.

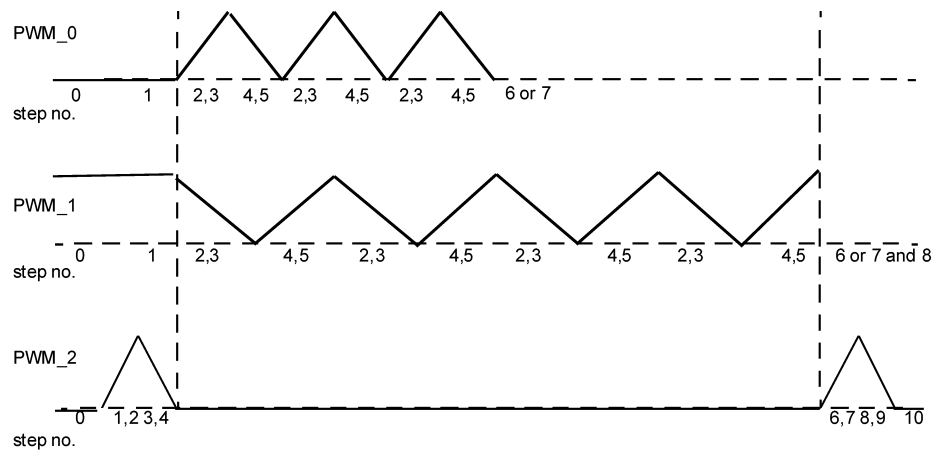
When a trigger is sent, it is stored by the receiving channel and can only be cleared when the receiving channel executes a TRIGGER command that waits for the trigger.

| 15 | 14 | 13 | 12          | 11 | 10 | 9 | 8 | 7 | 6           | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|-------------|----|----|---|---|---|-------------|---|---|---|---|---|---|
| 1  | 1  | 1  | WAITTRIGGER |    |    |   |   |   | SENDTRIGGER |   |   |   |   |   | 0 |

| Field       | Value  | Description                     |
|-------------|--------|---------------------------------|
| WAITTRIGGER | 000xx1 | Wait for trigger from channel 0 |
|             | 000x1x | Wait for trigger from channel 1 |
|             | 0001xx | Wait for trigger from channel 2 |
| SENDTRIGGER | 000xx1 | Send trigger to channel 0       |
|             | 000x1x | Send trigger to channel 1       |
|             | 0001xx | Send trigger to channel 2       |

## PWM SCRIPT EXAMPLE

This example shows a complex ramping sequence that uses triggers for synchronization. Three scripts implement the example. [Figure 16](#) shows the PWM outputs for this example.



**Figure 16. PWM Outputs**



**PWM Channel 0 Script**

| Script Command Address | PWM_WRITE Parameter 1 | PWM_WRITE Parameter 2 | PWM_WRITE Parameter 3 | Script Command | Description   |
|------------------------|-----------------------|-----------------------|-----------------------|----------------|---|
| 0x00                   | 0x01                  | 0x40                  | 0x00                  | SET_PWM        | Initialize channel for 0% duty cycle                        |
| 0x01                   | 0x05                  | 0xE2                  | 0x00                  | TRIGGER        | Wait for trigger from channel 2                             |
| 0x02                   | 0x09                  | 0x07                  | 0x7E                  | RAMP           | Ramp up by 126 steps  |
| 0x03                   | 0x0D                  | 0x07                  | 0x7E                  | RAMP           | Ramp up by 126 steps  |
| 0x04                   | 0x11                  | 0x07                  | 0xFE                  | RAMP           | Ramp down by 126 steps                                      |
| 0x05                   | 0x15                  | 0x07                  | 0xFE                  | RAMP           | Ramp down by 126 steps                                      |
| 0x06                   | 0x19                  | 0xA1                  | 0x82                  | BRANCH         | Loop 2 times starting at address 0x02                       |
| 0x07                   | 0x1D                  | 0xC8                  | 0x00                  | END            | Terminate script and assert $\overline{\text{IRQ}}$ to host |

**PWM Channel 1 Script**

| Script Command Address | PWM_WRITE Parameter 1 | PWM_WRITE Parameter 2 | PWM_WRITE Parameter 3 | Script Command | Description   |
|------------------------|-----------------------|-----------------------|-----------------------|----------------|---|
| 0x00                   | 0x02                  | 0x40                  | 0xFF                  | SET_PWM        | Initialize channel for 100% duty cycle                      |
| 0x01                   | 0x06                  | 0xE2                  | 0x00                  | TRIGGER        | Wait for trigger from channel 2                             |
| 0x02                   | 0x0A                  | 0x0F                  | 0xFE                  | RAMP           | Ramp down by 126 steps                                      |
| 0x03                   | 0x0E                  | 0x0F                  | 0xFE                  | RAMP           | Ramp down by 126 steps                                      |
| 0x04                   | 0x12                  | 0x0F                  | 0x7E                  | RAMP           | Ramp up by 126 steps  |
| 0x05                   | 0x16                  | 0x0F                  | 0x7E                  | RAMP           | Ramp up by 126 steps  |
| 0x06                   | 0x1A                  | 0xA2                  | 0x02                  | BRANCH         | Loop 3 times starting at address 0x02                       |
| 0x07                   | 0x1E                  | 0xE0                  | 0x08                  | TRIGGER        | Send trigger to channel 2                                   |
| 0x08                   | 0x22                  | 0xC8                  | 0x00                  | END            | Terminate script and assert $\overline{\text{IRQ}}$ to host |

**PWM Channel 2 Script**

| Script Command Address | PWM_WRITE Parameter 1 | PWM_WRITE Parameter 2 | PWM_WRITE Parameter 3 | Script Command | Description  |
|------------------------|-----------------------|-----------------------|-----------------------|----------------|--|
| 0x00                   | 0x03                  | 0x40                  | 0x00                  | SET_PWM        | Initialize channel for 0% duty cycle                               |
| 0x01                   | 0x07                  | 0x03                  | 0x7E                  | RAMP           | Ramp up by 126 steps   |
| 0x02                   | 0x0B                  | 0x03                  | 0x7E                  | RAMP           | Ramp up by 126 steps   |
| 0x03                   | 0x0F                  | 0x03                  | 0xFE                  | RAMP           | Ramp down by 126 steps   |
| 0x04                   | 0x13                  | 0x03                  | 0xFE                  | RAMP           | Ramp down by 126 steps   |
| 0x05                   | 0x17                  | 0xE1                  | 0x06                  | TRIGGER        | Send triggers to channels 0 and 1, wait for trigger from channel 1 |
| 0x06                   | 0x1B                  | 0x03                  | 0x7E                  | RAMP           | Ramp up by 126 steps   |
| 0x07                   | 0x1F                  | 0x03                  | 0x7E                  | RAMP           | Ramp up by 126 steps   |
| 0x08                   | 0x23                  | 0x03                  | 0xFE                  | RAMP           | Ramp down by 126 steps   |
| 0x09                   | 0x27                  | 0x03                  | 0xFE                  | RAMP           | Ramp down by 126 steps   |
| 0x0A                   | 0x2B                  | 0xC8                  | 0x00                  | END            | Terminate script and assert $\overline{\text{IRQ}}$ to host        |

## SELECTABLE SCRIPT EXAMPLE

Multiple scripts can be placed in a single buffer. The script which is executed is selected by the address in the parameter to the PWM\_START command (0x96).

| Script Command Address | PWM_WRITE Parameter 1                       | PWM_WRITE Parameter 2 | PWM_WRITE Parameter 3 | Script Command | Description   |
|------------------------|---|-----------------------|-----------------------|----------------|---|
| 0x00                   | Script 1                                    | 0x01                  | 0x40                  | 0x00           | Set PWM_0 to 0% duty cycle                            |
| 0x01                   |   | 0x05                  | 0x0F                  | 0x33           | Ramp up 51 steps                                      |
| 0x02                   |   | 0x09                  | 0xC0                  | 0x00           | Keep channel at 20% duty cycle                        |
| 0x03                   | Script 2                                    | 0x0D                  | 0x40                  | 0xFF           | Set PWM_0 to 100% duty cycle                          |
| 0x04                   |   | 0x11                  | 0x0F                  | 0xD5           | Ramp down 85 steps                                    |
| 0x05                   |   | 0x15                  | 0xC0                  | 0x00           | Keep channel at 66.6% duty cycle                      |
| 0x06                   | Script 3                                    | 0x19                  | 0x40                  | 0x00           | Set PWM_0 to 0% duty cycle                            |
| 0x07                   |   | 0x1D                  | 0x07                  | 0x7E           | Ramp up 126 steps                                     |
| 0x08                   |   | 0x21                  | 0x07                  | 0x7E           | Ramp up 126 steps                                     |
| 0x09                   |   | 0x25                  | 0x07                  | 0xFE           | Ramp down 126 steps                                   |
| 0x0A                   |   | 0x29                  | 0x07                  | 0xFE           | Ramp down 126 steps                                   |
| 0x0B                   |   | 0x2D                  | 0xA5                  | 0x07           | Loop ten times to script address 0x07                 |
| 0x0C                   | Script 4                                    | 0x31                  | 0xC8                  | 0x00           | Switch PWM_0 off (script 3 automatically enters here) |
| 0x0D                   | Script 5                                    | 0x35                  | 0x40                  | 0x00           | Set PWM_0 to 0% duty cycle                            |
| 0x0E                   |   | 0x39                  | 0x07                  | 0x25           | Ramp up 37 steps                                      |
| 0x0F                   |   | 0x3D                  | 0xC0                  | 0x00           | Keep channel at 14.5% duty cycle                      |
| 0x10                   | Script 6                                    | 0x41                  | 0x40                  | 0x00           | Set PWM_0 to 0% duty cycle                            |
| 0x11                   | (Alternates between 25% and 75% duty cycle) | 0x45                  | 0x01                  | 0x40           | Ramp up 64 steps                                      |
| 0x12                   |   | 0x49                  | 0x3F                  | 0x7E           | Ramp up 126 steps                                     |
| 0x13                   |   | 0x4D                  | 0x3F                  | 0xFE           | Ramp down 126 steps                                   |
| 0x14                   |   | 0x51                  | 0xA0                  | 0x12           | Always branch to script address 0x12                  |
| 0x15                   | Script 7                                    |                       |                       |                |   |
| .....                  |   |                       |                       |                |   |
| 0x3B                   |   |                       |                       |                |   |

To set a fixed duty cycle on a PWM channel requires 3 steps (see script 1 for duty cycles from 0% to 49% and script 2 for duty cycles from 51% to 100%).

To keep a PWM channel active providing a fixed duty cycle on its output, the script must terminate with the END command leaving the RESET bit clear. To switch this channel off, the host must send another PWM\_START command (0x96 followed by the parameter bytes) triggering the single command described in script 4. This END command will set the RESET bit and the dedicated PWM output will be disabled.

Script 3 will automatically enter into this command when the 10 loops of ramping up and down are executed.

Script 7 can be finished by two commands:

- PWM\_STOP command with parameter 0x01
- PWM\_START command with parameter 0x31 (start PWM\_0 from address 0x0C to run script 4)

The script address is the physical address to be used from BRANCH instructions inside the script file buffer. The parameter 1 byte contains the same address with the 2 channel bits appended and will be associated with the PWM\_START command.

## DIGITAL MULTIPLEXERS

Two 2:1 multiplexers are provided for host-controlled digital switching. Setting the MUX1EN or MUX2EN bits with the WRITE\_CFG command enables the corresponding multiplexer and its input and output signals, which overrides any other functions which may use these pins. The MUX1 signals are alternate functions of the PWM\_x outputs. The MUX2 signals are alternate functions of three KP-Yx pins shared with the rotary encoder interface, so MUX2 is unavailable when the interface is used.

The data select inputs for the multiplexers are controlled by the MUX1SEL and MUX2SEL bits, which are written by the WRITE\_CFG command. If it is important to avoid momentarily passing an incorrect input to the output, the select bit must be loaded with a first WRITE\_CFG command before sending a second WRITE\_CFG command to set the enable bit. The truth table for the multiplexers is shown in [Table 9](#).

**Table 9. Digital Multiplexer Function Table**

| MUXxEN Bit | MUXxSEL Bit | MUXx_IN2 Pin | MUXx_IN1 Pin | MUXx_OUT Pin         |
|------------|-------------|--------------|--------------|----------------------|
| 1          | 0           | X            | 0            | 0                    |
| 1          | 0           | X            | 1            | 1                    |
| 1          | 1           | 0            | X            | 0                    |
| 1          | 1           | 1            | X            | 1                    |
| 0          | X           | X            | X            | MUXx_OUT not enabled |

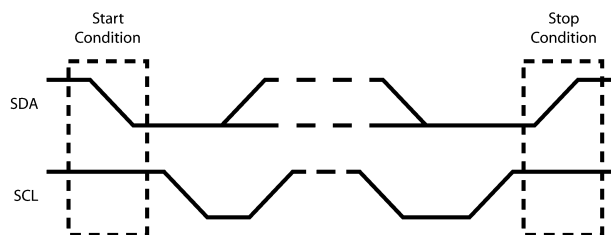
## HOST INTERFACE

The two-wire ACCESS.bus interface is used to communicate with a host. The ACCESS.bus interface is compatible with the I<sup>2</sup>C bus standard. The LM8323 operates as a bus slave at 400 kHz (Fast mode).

All communication with the LM8323 over the ACCESS.bus interface is initiated by the host, usually in response to an interrupt request (IRQ low) asserted by the LM8323. The LM8323 may request service from the host by asserting the IRQ interrupt output.

## START AND STOP CONDITIONS

Every transfer is preceded by a Start condition or a Repeated Start condition. The latter occurs when a command follows immediately upon another command without an intervening Stop condition. A Stop condition indicates the end of transmission. Every byte is acknowledged by the receiver.



**Figure 17. Start and Stop Conditions**

## CONTINUOUS COMMAND STRINGS

A host device may send a continuous string of commands using the Repeated Start condition, which would block another ACCESS.bus device from gaining control of the bus. After Power-On the host device must send multiple commands to initialize the LM8323 device. A minimal command string will include the commands shown in [Table 10](#).

**Table 10. Minimal Command String**

| Command          | Description                                     |
|------------------|---|
| READ_ID          | Read vendor ID and software version             |
| READ_INT         | Check if NOINT bit is set in interrupt register |
| WRITE_CFG        | Configure the LM8323                            |
| SET_KEY_SIZE     | Set the size of the keypad                      |
| WRITE_CLK        | Set the clock mode for the PWM unit             |
| WRITE_PORT_SEL   | Set port direction for GPIO pins                |
| WRITE_PORT_STATE | Set port states of GPIO pins                    |

A more comprehensive command string may include the additional commands shown in [Table 11](#).

**Table 11. Additional Commands**

| Command         | Description                                    |
|-----------------|--|
| SET_DEBOUNCE    | Set debounce time                              |
| SET_ACTIVE      | Set active time                                |
| READ_CLK        | Verify PWM clock settings                      |
| READ_CFG        | Verify configuration setting                   |
| READ_PORT_STATE | Read all port states (physical levels on pins) |

**NOTE**

Very long continuous command strings exceeding 30 milliseconds could overrun the ability of the LM8323 to process commands if the time from the last clock cycle of a command until the next Start condition or Repeated Start condition is always shorter than 60  $\mu$ s. A very long command chain could prevent the LM8323 from performing any watchdog service and consequently could trigger a physical RESET to the device.

To avoid overrunning the LM8323, the host should provide a 1ms break between long (>30 ms) command sequences for SCL frequencies > 100 kHz.

**DEVICE ADDRESS**

The device address is controlled by states sampled on the CONFIG\_1 and CONFIG\_2 pins, as shown in [Table 12](#). In the first byte of a bus transaction, a 7-bit address plus a direction bit are broadcast by the bus master to all bus slaves.

**Table 12. Device Address Selection**

| CONFIG_1 | CONFIG_2 | Device Address |
|----------|----------|----------------|
| 0        | 0        | 1000 010X      |
| 0        | 1        | 1000 011X      |
| 1        | 0        | 1000 100X      |
| 1        | 1        | 1000 101X      |

CONFIG\_1 and CONFIG\_2 pins should be connected to GND or  $V_{CC}$  using pulldown or pullup resistors. The pins cannot be left unconnected.

**HOST WRITE COMMANDS**

Some host commands include one or more data bytes written to the LM8323. [Figure 18](#) shows a SET\_KEY\_SIZE command, which consists of an address byte, a command byte, and one data byte.

The first byte is composed of a 7-bit slave address in bits 7:1 and a direction bit in bit 0. The state of the direction bit is 0 on writes from the host to the slave and 1 on reads from the slave to the host.

The second byte sends the command. The SET\_KEY\_SIZE command is 0x90.

The third byte sends the data, in this case specifying the number of rows and columns for the keypad.

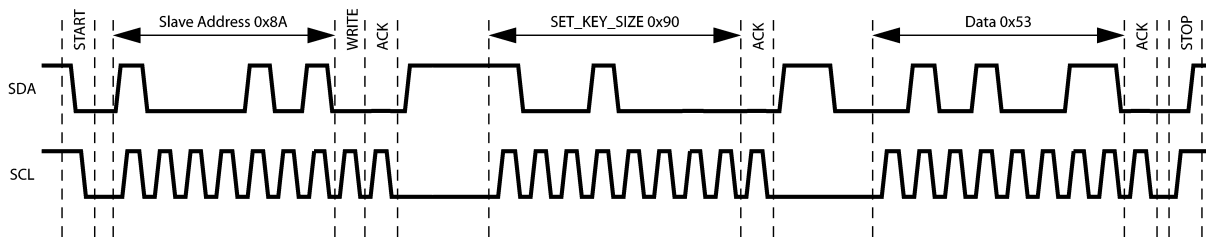


Figure 18. Host Write Command

## HOST READ COMMANDS

Some host commands include one or more data bytes read from the LM8323. Figure 19 shows a READ\_PORT\_SEL command which consists of an address byte, a command byte, a second address byte, and two data bytes.

The first address byte is sent with the direction bit driven low to indicate a write transaction of the command to the LM8323. The second address byte is sent with the direction bit undriven (pulled high) to indicate a read transaction of the data from the LM8323.

The Repeated Start condition must be repeated whenever the slave address or the direction bit is changed. In this case, the direction bit is changed.

The bus master can send any number of Repeated Start conditions without releasing control of the bus. This technique can be used to implement atomic transactions, in which the bus master sends a command and then reads a register without allowing any other device to get control of the bus between these events.

The data is sent from the slave to the host in the fourth and fifth bytes. The fifth byte ends with a negative acknowledgement (NACK) to indicate the end of the data.

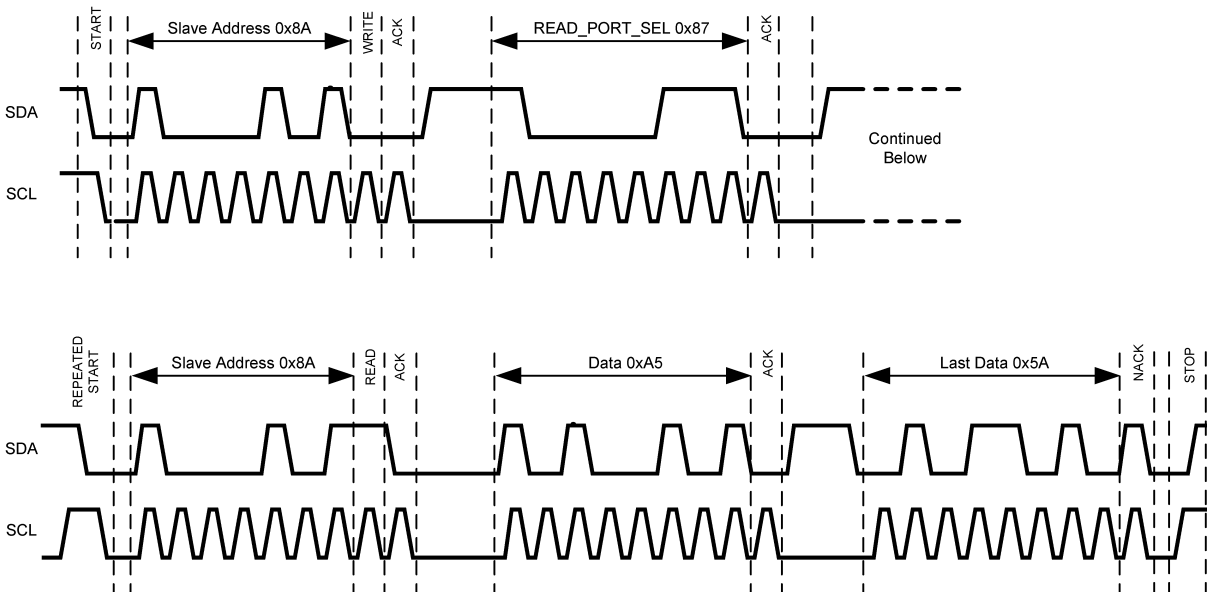


Figure 19. Host Read Command

## INTERRUPTS

The  $\overline{\text{IRQ}}$  output may be asserted on these conditions:

- Any new key-event after the last interrupt was asserted but not yet acknowledged by reading the interrupt code.
- Any change in the state of the rotary encoder inputs.

- Termination of a PWM script (END command).
- Any error condition, which is indicated by the error code.

## INTERRUPT CODE

The interrupt code is read and acknowledged with the READ\_INT command (0x82). This command clears the code and deasserts the IRQ output. [Table 13](#) shows the format of the interrupt code.

**Table 13. Interrupt Code**

| 7       | 6       | 5       | 4      | 3     | 2 | 1       | 0      |
|---------|---------|---------|--------|-------|---|---------|--------|
| PWM2END | PWM1END | PWM0END | NOINIT | ERROR | 0 | ROTATOR | KEYPAD |

| Bit     | Description   |
|---------|---|
| PWM2END | An END script command was executed by PWM channel 2.      |
| PWM1END | An END script command was executed by PWM channel 1.      |
| PWM0END | An END script command was executed by PWM channel 0.      |
| NOINIT  | The LM8323 is waiting for an initialization sequence.     |
| ERROR   | An error condition occurred.                              |
| ROTATOR | A state change was detected in the rotary encoder inputs. |
| KEYPAD  | A key-press or key-release event occurred.                |

## ERROR CODE

If the LM8323 reports an error, the READ\_ERROR command (0x8C) is used to read the error code. This command clears the error code. [Table 14](#) shows the format of the error code.

**Table 14. Error Code**

| 7 | 6      | 5 | 4 | 3 | 2      | 1      | 0      |
|---|--------|---|---|---|--------|--------|--------|
| 0 | FIFOVR | 0 | 0 | 0 | KEYOVR | CMDUNK | BADPAR |

| Bit    | Description                                     |
|--------|---|
| FIFOVR | Event occurred while the FIFO was full.         |
| KEYOVR | More than two keys were pressed simultaneously. |
| CMDUNK | Not a valid command.                            |
| BADPAR | Bad command parameter.                          |

## WAKE-UP FROM HALT MODE

Any bus transaction initiated by the host may encounter the LM8323 device in Halt mode or busy with processing data, such as controlling the FIFO buffer or executing interrupt service routines.

LM8323 shows the case in which the host sends a command while the LM8323 is in Halt mode (Internal execution clock is stopped). Any activity on the ACCESS.bus wakes up the LM8323, but it cannot acknowledge the first bus cycle immediately after wake-up.

The host drives a Start condition followed by seven address bits and a R/W bit. The host then releases SDA for one clock period, so that it can be driven by the LM8323.

If the LM8323 does not drive SDA low during the high phase of the clock period immediately after the R/W bit, the bus cycle terminates without being acknowledged (shown as NACK in [Figure 20](#)). The host then aborts the transaction by sending a Stop condition. After aborting the bus cycle, the host may then retry the bus cycle. On the second attempt, the LM8323 will be able to acknowledge the slave address, because it will be in Active mode.

Alternatively, the I<sup>2</sup>C specification allows sending a START byte (00000001), which will not be acknowledged by any device. This byte can be used to wake up the LM8323 from Halt mode.

The LM8323 may also stall the bus transaction by pulling the SCL low, which is a valid behavior defined by the I<sup>2</sup>C specification.

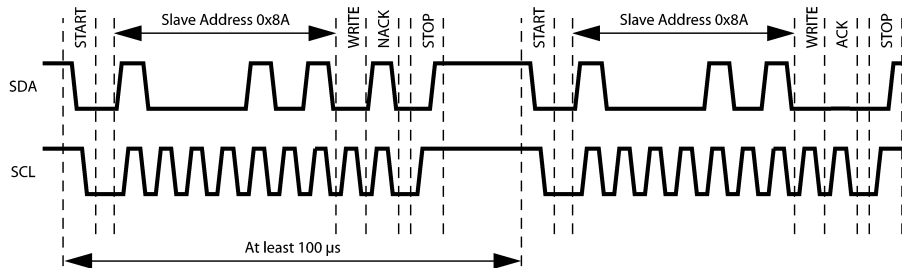


Figure 20. LM8323 Responds with NACK, Host Retries Command

HOST COMMANDS

| Function                  | Cmd  | Dir | Data Bytes           | Description   |
|---------------------------|------|-----|----------------------|---|
| READ_ID                   | 0x80 | R   | nnnn nnnn            | Read the manufacturer code (nnnn nnnn) and the device revision number (pppp pppp).  |
|                           |      |     | pppp pppp            |   |
| WRITE_CFG                 | 0x81 | W   | nnnn nnnn            | Write the hardware configuration register.  |
| READ_INT                  | 0x82 | R   | nnnn nnnn            | Read the interrupt code, deassert the $\overline{\text{IRQ}}$ output, and clear the code. (If the NOINIT bit is set, it remains set and $\overline{\text{IRQ}}$ remains asserted until a WRITE_CFG command is received.)                              |
| $\overline{\text{RESET}}$ | 0x83 | W   | nnnn nnnn            | Reset the LM8323. Error if nnnn nnnn is not 0xAA.   |
| WRITE_PULL_DOWN           | 0x84 | W   | nnnn nnnn            | Select pullup (0) or pulldown (1) direction for the corresponding general-purpose I/O (GPIO) port pins.   |
|                           |      |     | pppp pppp            |   |
| WRITE_PORT_SEL            | 0x85 | W   | nnnn nnnn            | Select input (0) or output (1) for the corresponding general-purpose I/O (GPIO) port pins.  |
|                           |      |     | pppp pppp            |   |
| WRITE_PORT_STATE          | 0x86 | W   | nnnn nnnn            | For pins configured as inputs, 0 selects high-impedance mode and 1 enables a weak pullup. For pins configured as outputs, each bit specifies the logic level driven on the pin.   |
|                           |      |     | pppp pppp            |   |
| READ_PORT_SEL             | 0x87 | R   | nnnn nnnn            | Read the direction of the corresponding GPIO port pins.   |
|                           |      |     | pppp pppp            |   |
| READ_PORT_STATE           | 0x88 | R   | nnnn nnnn            | Read the state on the corresponding GPIO port pins.   |
|                           |      |     | pppp pppp            |   |
| READ_FIFO                 | 0x89 | R   | Up to 15 event codes | Read an event from the FIFO. Maximum of 14 event codes stored in the FIFO.  |
| RPT_READ_FIFO             | 0x8A | R   | Up to 15 event codes | Repeats a FIFO read without advancing the FIFO pointer, for example to retry a read after an error.   |
| SET_ACTIVE                | 0x8B | W   | nnnn nnnn            | Set the time during which the LM8323 stays active before entering Halt mode. The active time must be greater than the debounce time. The default time is 500 milliseconds. The valid range is 1255. Active time = $n \times 4$ milliseconds.          |
| READ_ERROR                | 0x8C | R   | nnnn nnnn            | Read and clear the error code.  |
| READ_ROTATOR              | 0x8E | R   | nnnn nnnn            | Read accumulated rotation steps since previous read.  |
| SET_DEBOUNCE              | 0x8F | W   | nnnn nnnn            | Set the time for rescanning the keypad after detecting a key-press or key-release event to verify the event. The default time is 12 milliseconds. The valid range is 1255. Debounce time = $n \times 4$ milliseconds and must not exceed active time. |
| SET_KEY_SIZE              | 0x90 | W   | nnnn pppp            | Set keypad size. nnnn = KP-Xx pins, pppp = KP-Yx pins   |
| READ_KEY_SIZE             | 0x91 | R   | nnnn pppp            | Read keypad size. nnnn = KP-Xx pins, pppp = KP-Yx pins  |
| READ_CFG                  | 0x92 | R   | nnnn nnnn            | Read the hardware configuration register.   |
| WRITE_CLOCK               | 0x93 | W   | nnnn nnnn            | Write the clock configuration register.   |
| READ_CLOCK                | 0x94 | R   | nnnn nnnn            | Read the clock configuration register.  |

| Function  | Cmd  | Dir | Data Bytes | Description  |
|-----------|------|-----|------------|--|
| PWM_WRITE | 0x95 | W   |            | Write a command to the PWM script command file.<br>nn = PWM channel number (01, 10, or 11) |
|           |      |     | aaaa aann  | aaaaaa = address in script command file (0-59)   |
|           |      |     | pppp pppp  | pppp pppp = high byte of script command  |
|           |      |     | qqqq qqqq  | qqqq qqqq = low byte of script command   |
| PWM_START | 0x96 | W   | aaaa aann  | Start script on channel nn (01, 10, or 11) at address aaaaaa.                              |
| PWM_STOP  | 0x97 | W   | 0000 00nn  | Stop script on channel nn (01, 10, or 11).   |

**NOTE**

The data bytes which follow the command can be reads (toward the host) or writes (toward the LM8323). In the case of the READ\_FIFO and RPT\_READ\_FIFO commands, the number of data bytes is variable, with the last transaction indicated by returning a negative acknowledgement (NACK).

**READ\_ID COMMAND**

The READ\_ID command consists of a command byte (0x80) from the host and two data bytes from the LM8323. The first data byte returns the manufacturer code, and the second byte returns the device revision level.

|   |   |   |   |   |   |   |   |              |   |   |   |   |   |          |   |  |  |  |  |
|---|---|---|---|---|---|---|---|--------------|---|---|---|---|---|----------|---|--|--|--|--|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7            | 6 | 5 | 4 | 3 | 2 | 1        | 0 |  |  |  |  |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | MANUFACTURER |   |   |   |   |   | REVISION |   |  |  |  |  |

**WRITE\_CFG COMMAND**

The WRITE\_CFG command consists of a command byte (0x81) and a data byte from the host. The data byte is loaded into the hardware configuration register. The default state of this register is 0x80.

|   |   |   |   |   |   |   |   |        |       |   |   |        |         |        |         |
|---|---|---|---|---|---|---|---|--------|-------|---|---|--------|---------|--------|---------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7      | 6     | 5 | 4 | 3      | 2       | 1      | 0       |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | IRQPST | ROTEN | 0 | 0 | MUX2EN | MUX2SEL | MUX1EN | MUX1SEL |

| Bit     | Value | Description   |
|---------|-------|---|
| IRQPST  | 0     | $\overline{\text{IRQ}}$ is an open-drain output.  |
|         | 1     | $\overline{\text{IRQ}}$ is a push-pull output.  |
| ROTEN   | 0     | Rotary encoder interface disabled.  |
|         | 1     | Rotary encoder interface enabled. This selection enables the ROT_IN_x inputs which are alternate functions of certain KP-Yx pins. |
| MUX2EN  | 0     | MUX2_OUT output disabled.   |
|         | 1     | MUX2_OUT output enabled. This overrides any other function available on this pin.   |
| MUX2SEL | 0     | If the MUX2 EN bit is 1, the MUX2_IN1 input drives the MUX2_OUT output.   |
|         | 1     | If the MUX2 EN bit is 1, the MUX2_IN2 input drives the MUX2_OUT output.   |
| MUX1EN  | 0     | MUX1_OUT output disabled.   |
|         | 1     | MUX1_OUT output enabled. This overrides any other function available on this pin.   |
| MUX1SEL | 0     | If the MUX1 EN bit is 1, the MUX1_IN1 input drives the MUX1_OUT output.   |
|         | 1     | If the MUX1 EN bit is 1, the MUX1_IN2 input drives the MUX1_OUT output.   |



## READ\_INT COMMAND

The READ\_INT command consists of a command byte (0x82) from the host and a data byte from the LM8323. The data byte is the interrupt code. Reading the interrupt code acknowledges the interrupt (which deasserts  $\overline{IRQ}$ ) and clears the interrupt code. An exception to this behavior occurs if the NOINIT bit is set, in which case  $\overline{IRQ}$  will not be deasserted and the interrupt code will not be cleared until a WRITE\_CFG command is received.

|          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>7</b> | <b>6</b> | <b>5</b> | <b>4</b> | <b>3</b> | <b>2</b> | <b>1</b> | <b>0</b> | <b>7</b> | <b>6</b> | <b>5</b> | <b>4</b> | <b>3</b> | <b>2</b> | <b>1</b> | <b>0</b> |
| 1        | 0        | 0        | 0        | 0        | 0        | 1        | 0        | PWM2END  | PWM1END  | PWM0END  | NOINIT   | ERROR    | 0        | ROTATOR  | KEYPAD   |

| Bit     | Value | Description  |
|---------|-------|--|
| PWM2END | 0     | No interrupt from PWM channel 2.                           |
|         | 1     | An END script command was executed by PWM channel 2.       |
| PWM1END | 0     | No interrupt from PWM channel 1.                           |
|         | 1     | An END script command was executed by PWM channel 1.       |
| PWM0END | 0     | No interrupt from PWM channel 0.                           |
|         | 1     | An END script command was executed by PWM channel 0.       |
| NOINIT  | 0     | Normal operation.  |
|         | 1     | LM8323 is waiting for the initialization sequence.         |
| ERROR   | 0     | No error condition is indicated.                           |
|         | 1     | An error condition occurred.                               |
| ROTATOR | 0     | No state change in the rotary encoder inputs is indicated. |
|         | 1     | A state change was detected in the rotary encoder inputs.  |
| KEYPAD  | 0     | No key-press or key-release event is indicated.            |
|         | 1     | A key-press or key-release event occurred.                 |

## RESET COMMAND

The RESET command consists of a command byte (0x83) and one data byte from the host. The command causes a reset, identical to an external reset. The data byte must be 0xAA, otherwise no reset will occur and an error condition will be signalled.

### NOTE

When FW6 version devices receive a RESET command the  $\overline{IRQ}$  line is set high and held high for 60 ms, then pulled low to show the device was successfully reset and is ready to be used.

|          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>7</b> | <b>6</b> | <b>5</b> | <b>4</b> | <b>3</b> | <b>2</b> | <b>1</b> | <b>0</b> | <b>7</b> | <b>6</b> | <b>5</b> | <b>4</b> | <b>3</b> | <b>2</b> | <b>1</b> | <b>0</b> |
| 1        | 0        | 0        | 0        | 0        | 0        | 1        | 1        | 1        | 0        | 1        | 0        | 1        | 0        | 1        | 0        |

## WRITE\_PULL\_DOWN COMMAND

The WRITE\_PORT\_SEL command consists of a command byte (0x84) and two data bytes from the host. The data bytes configure the pullup/pulldown device (if enabled) for the corresponding general-purpose I/O ports as pullups (0) or pulldowns (1). The first data byte controls ports GPIO\_15 through GPIO\_08, and the second byte controls ports GPIO\_07 through GPIO\_00.

|          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |         |         |         |         |         |         |         |         |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|---------|---------|---------|---------|---------|---------|---------|---------|
| <b>7</b> | <b>6</b> | <b>5</b> | <b>4</b> | <b>3</b> | <b>2</b> | <b>1</b> | <b>0</b> | <b>7</b> | <b>6</b> | <b>5</b> | <b>4</b> | <b>3</b> | <b>2</b> | <b>1</b> | <b>0</b> |         |         |         |         |         |         |         |         |
| 1        | 0        | 0        | 0        | 0        | 1        | 0        | 0        | GPIO_15  | GPIO_14  | GPIO_13  | GPIO_12  | GPIO_11  | GPIO_10  | GPIO_09  | GPIO_08  | GPIO_07 | GPIO_06 | GPIO_05 | GPIO_04 | GPIO_03 | GPIO_02 | GPIO_01 | GPIO_00 |

| Bit     | Value | Description   |
|---------|-------|---|
| GPIO_xx | 0     | GPIO port pin pullup/pulldown device is a pullup.   |
|         | 1     | GPIO port pin pullup/pulldown device is a pulldown. |

### WRITE\_PORT\_SEL COMMAND

The WRITE\_PORT\_SEL command consists of a command byte (0x85) and two data bytes from the host. The data bytes configure the corresponding general-purpose I/O ports as inputs (0) or outputs (1). The first data byte controls ports GPIO\_15 through GPIO\_08, and the second byte controls ports GPIO\_07 through GPIO\_00.

|   |   |   |   |   |   |   |   |         |         |         |         |         |         |   |         |         |         |         |         |         |         |         |         |
|---|---|---|---|---|---|---|---|---------|---------|---------|---------|---------|---------|---|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7       | 6       | 5       | 4       | 3       | 2       | 1 | 0       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | GPIO_15 | GPIO_14 | GPIO_13 | GPIO_12 | GPIO_11 | GPIO_10 | 0 | GPIO_08 | GPIO_07 | GPIO_06 | GPIO_05 | GPIO_04 | GPIO_03 | GPIO_02 | GPIO_01 | GPIO_00 |

| Bit     | Value | Description                 |
|---------|-------|-----------------------------|
| GPIO_xx | 0     | GPIO port pin is an input.  |
|         | 1     | GPIO port pin is an output. |

The GPIO\_09 port pin can only be configured as an input with weak pullup/pulldown device.

### WRITE\_PORT\_STATE COMMAND

The WRITE\_PORT\_STATE command consists of a command byte (0x86) and two data bytes from the host. For general-purpose I/O ports configured as inputs, the data bytes select whether the inputs are high-impedance (0) or have a weak pullup (1). For ports configured as outputs, the data bytes control the state driven on the output. The first data byte controls ports GPIO\_15 through GPIO\_08, and the second byte controls ports GPIO\_07 through GPIO\_00.

|   |   |   |   |   |   |   |   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|---|---|---|---|---|---|---|---|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | GPIO_15 | GPIO_14 | GPIO_13 | GPIO_12 | GPIO_11 | GPIO_10 | GPIO_09 | GPIO_08 | GPIO_07 | GPIO_06 | GPIO_05 | GPIO_04 | GPIO_03 | GPIO_02 | GPIO_01 | GPIO_00 |

| Bit     | Value | Description   |
|---------|-------|---|
| GPIO_xx | 0     | If the GPIO port pin is an input, pullup device is disabled. If the GPIO port pin is an output, it is driven low. |
|         | 1     | If the GPIO port pin is an input, pullup device is enabled. If the GPIO port pin is an output, it is driven high. |

### READ\_PORT\_SEL COMMAND

The READ\_PORT\_SEL command consists of a command byte (0x87) from the host and two data bytes from the LM8323. The data bytes indicate the direction configured for the corresponding ports, either input (0) or output (1). The first data byte controls ports GPIO\_15 through GPIO\_08, and the second byte controls ports GPIO\_07 through GPIO\_00.

|   |   |   |   |   |   |   |   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|---|---|---|---|---|---|---|---|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | GPIO_15 | GPIO_14 | GPIO_13 | GPIO_12 | GPIO_11 | GPIO_10 | GPIO_09 | GPIO_08 | GPIO_07 | GPIO_06 | GPIO_05 | GPIO_04 | GPIO_03 | GPIO_02 | GPIO_01 | GPIO_00 |

| Bit     | Value | Description                 |
|---------|-------|-----------------------------|
| GPIO_xx | 0     | GPIO port pin is an input.  |
|         | 1     | GPIO port pin is an output. |

## READ\_PORT\_STATE COMMAND

The READ\_PORT\_STATE command consists of a command byte (0x88) from the host and two data bytes from the LM8323. The data bytes indicate the states on the corresponding ports. The first data byte controls ports GPIO\_15 through GPIO\_08, and the second byte controls ports GPIO\_07 through GPIO\_00.

|   |   |   |   |   |   |   |   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|---|---|---|---|---|---|---|---|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | GPIO_15 | GPIO_14 | GPIO_13 | GPIO_12 | GPIO_11 | GPIO_10 | GPIO_09 | GPIO_08 | GPIO_07 | GPIO_06 | GPIO_05 | GPIO_04 | GPIO_03 | GPIO_02 | GPIO_01 | GPIO_00 |

| Bit     | Value | Description  |
|---------|-------|--|
| GPIO_xx | 0     | If the GPIO port pin is an input, pullup is disabled. If the GPIO port pin is an output, it is driven low. |
|         | 1     | If the GPIO port pin is an input, pullup is enabled. If the GPIO port pin is an output, it is driven high. |

## READ\_FIFO COMMAND

The READ\_FIFO command consists of a command byte (0x89) sent from the host and a variable number of data bytes received from the LM8323. The LM8323 will provide data until the FIFO is empty. The last data byte is indicated by its value (0x00) and a negative acknowledgement (NACK) on the ACCESS.bus interface. The data bytes correspond to key-press and key-release events, as described in [Table 4](#).

|   |   |   |   |   |   |   |   |          |   |   |   |   |   |   |   |      |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|----------|---|---|---|---|---|---|---|------|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | FIFODATA |   |   |   |   |   |   |   | 0x00 |   |   |   |   |   |   |   |

| Field    | Value    | Description        |
|----------|----------|--------------------|
| FIFODATA | 0xxxxxxx | Key-release event. |
|          | 1xxxxxxx | Key-press event.   |

## RPT\_READ\_FIFO COMMAND

The RPT\_READ\_FIFO command consists of a command byte (0x8A) and from the host and a variable number of data bytes from the LM8323. This command provides the same data as a previous READ\_FIFO command, but without advancing the FIFO pointer. It may be used to recover from an error encountered during a READ\_FIFO command.

|   |   |   |   |   |   |   |   |          |   |   |   |   |   |   |   |      |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|----------|---|---|---|---|---|---|---|------|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | FIFODATA |   |   |   |   |   |   |   | 0x00 |   |   |   |   |   |   |   |

| Field    | Value    | Description        |
|----------|----------|--------------------|
| FIFODATA | 0xxxxxxx | Key-release event. |
|          | 1xxxxxxx | Key-press event.   |

## SET\_ACTIVE COMMAND

The SET\_ACTIVE command consists of a command byte (0x8B) and a data byte from the host. This command sets the time that the LM8323 stays active without detecting a key-press, key-release or rotary encoder event before entering Halt mode. The default active time is 500 milliseconds. The host can program ACTIVETIME from 4–1020 milliseconds with a granularity of 4 milliseconds.

|   |   |   |   |   |   |   |   |            |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|------------|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7          | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | ACTIVETIME |   |   |   |   |   |   |   |

| Field      | Value | Description                              |
|------------|-------|--|
| ACTIVETIME | 0     | Halt mode is disabled.                   |
|            | 1–255 | Active time = $n \times 4$ milliseconds. |

## READ\_ERROR COMMAND

The READ\_ERROR command consists of a command byte (0x8C) from the host and a data byte from the LM8323. After reading an interrupt code that indicates an error condition, this command is used to read an error code that indicates the cause of the error condition.

|   |   |   |   |   |   |   |   |   |        |   |   |   |        |        |        |
|---|---|---|---|---|---|---|---|---|--------|---|---|---|--------|--------|--------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6      | 5 | 4 | 3 | 2      | 1      | 0      |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | FIFOVR | 0 | 0 | 0 | KEYOVR | CMDUNK | BADPAR |

| Bit    | Value | Description                                     |
|--------|-------|---|
| FIFOVR | 0     | No FIFO overrun occurred.                       |
|        | 1     | Event occurred while the FIFO was full.         |
| KEYOVR | 0     | No keypad overrun occurred.                     |
|        | 1     | More than two keys were pressed simultaneously. |
| CMDUNK | 0     | No invalid command was encountered.             |
|        | 1     | Not a valid command.                            |
| BADPAR | 0     | No bad parameter was encountered.               |
|        | 1     | Bad command parameter.                          |

## READ\_ROTATOR COMMAND

The READ\_ROTATOR command consists of a command byte (0x8E) from the host and a data byte from the LM8323. The data byte is a signed two's complement value which indicates the accumulated number of rotation steps of an external rotary encoder since the last time the READ\_ROTATOR command was executed.

|   |   |   |   |   |   |   |   |          |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|----------|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | ROTATION |   |   |   |   |   |   |   |

| Field    | Value        | Description  |
|----------|--------------|--|
| ROTATION | –128 to +127 | Clockwise rotation is indicated by a positive value. Counterclockwise movement is indicated by a negative value. |

## SET\_DEBOUNCE COMMAND

The SET\_DEBOUNCE command consists of a command byte (0x8F) and a data byte from the host. This command sets the time that the LM8323 waits before rescanning the keypad to confirm a key-press or key-release event. The default debounce time is 12 milliseconds. The host can program DEBOUNCETIME from 4–1020 milliseconds with a granularity of 4 milliseconds. The DEBOUNCETIME must not exceed the active time set with the SET\_ACTIVE command.

|   |   |   |   |   |   |   |   |              |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|--------------|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | DEBOUNCETIME |   |   |   |   |   |   |   |

| Field        | Value | Description                       |
|--------------|-------|-----------------------------------|
| DEBOUNCETIME | 1–255 | Active time = n × 4 milliseconds. |

### SET\_KEY\_SIZE COMMAND

The SET\_KEY\_SIZE command consists of a command byte (0x90) and a data byte from the host. This command specifies the keypad size in terms of the number of KP-Xx inputs and KP-Yx outputs which are used. Any unused KP-Xx and KP-Yx pins may be used for general-purpose I/O. The minimum value for either field is 3, which corresponds to a keypad configuration that supports 3 × 3 + 3 SF keys (total of 12 keys).

The maximum number of KP-Xx inputs is 8, and the maximum number of KP-Yx outputs is 12. If the digital multiplexer MUX2 or the rotary encoder interface is used, the maximum number of KP-Yx outputs is 9. If the SLOWCLKOUT pin is used, the maximum number is 8.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7    | 6 | 5 | 4 | 3    | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|------|---|---|---|------|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | KP-X |   |   |   | KP-Y |   |   |   |

| Field | Value | Description              |
|-------|-------|--------------------------|
| KP-X  | 3–8   | Number of KP-Xx inputs.  |
| KP-Y  | 3–12  | Number of KP-Yx outputs. |

### READ\_KEY\_SIZE COMMAND

The READ\_KEY\_SIZE command consists of a command byte (0x91) from the host and a data byte from the LM8323. The host can issue the command at any time to read the configuration of the keypad.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7    | 6 | 5 | 4 | 3    | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|------|---|---|---|------|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | KP-X |   |   |   | KP-Y |   |   |   |

| Field | Value | Description              |
|-------|-------|--------------------------|
| KP-X  | 3–8   | Number of KP-Xx inputs.  |
| KP-Y  | 3–12  | Number of KP-Yx outputs. |

### READ\_CFG COMMAND

The READ\_CFG command consists of a command byte (0x92) from the host and a data byte from the LM8323. The data byte returns the settings in the hardware configuration register. The default state of this register is 0x80.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3      | 2       | 1      | 0       |
|---|---|---|---|---|---|---|---|---|---|---|---|--------|---------|--------|---------|
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | MUX2EN | MUX2SEL | MUX1EN | MUX1SEL |

| Bit     | Value | Description  |
|---------|-------|--|
| ROTEN   | 0     | Rotary encoder interface disabled.   |
|         | 1     | Rotary encoder interface enabled. This selection enables the ROT_IN_x inputs, which are alternate functions of certain KP-Yx pins. |
| MUX2EN  | 0     | MUX2_OUT output disabled.  |
|         | 1     | MUX2_OUT output enabled. This overrides any other function available on this pin.  |
| MUX2SEL | 0     | If the MUX2 EN bit is 1, the MUX2_IN1 input drives the MUX2_OUT output.  |
|         | 1     | If the MUX2 EN bit is 1, the MUX2_IN2 input drives the MUX2_OUT output.  |
| MUX1EN  | 0     | MUX1_OUT output disabled.  |
|         | 1     | MUX1_OUT output enabled. This overrides any other function available on this pin.  |
| MUX1SEL | 0     | If the MUX1 EN bit is 1, the MUX1_IN1 input drives the MUX1_OUT output.  |
|         | 1     | If the MUX1 EN bit is 1, the MUX1_IN2 input drives the MUX1_OUT output.  |

### WRITE\_CLOCK COMMAND

The WRITE\_CLOCK command consists of a command byte (0x93) and a data byte from the host. This command sets the clock configuration, as described in [Table 1](#).

|   |   |   |   |   |   |   |   |               |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---------------|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7             | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | CONFIGURATION |   |   |   |   |   |   |   |

### READ\_CLOCK COMMAND

The READ\_CLOCK command consists of a command byte (0x94) from the host and a data byte from the LM8323. This command reads bits 7:2 of the clock configuration, as described in [Table 1](#).

|   |   |   |   |   |   |   |   |               |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---------------|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7             | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | CONFIGURATION |   |   |   |   |   | 1 | 0 |

### PWM\_WRITE COMMAND

The PWM\_WRITE command consists of a command byte (0x95) and three data bytes from the host. The command writes a 16-bit script command into a specified address in the script command file of the specified PWM channel.

|   |   |   |   |   |   |   |   |         |   |   |   |    |         |   |   |   |   |   |   |   |   |   |   |  |
|---|---|---|---|---|---|---|---|---------|---|---|---|----|---------|---|---|---|---|---|---|---|---|---|---|--|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7       | 6 | 5 | 4 | 3  | 2       | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | ADDRESS |   |   |   | CH | COMMAND |   |   |   |   |   |   |   |   |   |   |  |

| Bit     | Value | Description                              |
|---------|-------|--|
| ADDRESS | 0–59  | Location in the PWM script command file. |
| CH      | 01    | PWM channel 0.                           |
|         | 10    | PWM channel 1.                           |
|         | 11    | PWM channel 2.                           |

### PWM\_START COMMAND

The PWM\_START command consists of a command byte (0x96) and a data byte from the host. This command starts execution of the script command file at the specified address for the specified channel.

|   |   |   |   |   |   |   |   |         |   |   |   |   |   |    |   |
|---|---|---|---|---|---|---|---|---------|---|---|---|---|---|----|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7       | 6 | 5 | 4 | 3 | 2 | 1  | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | ADDRESS |   |   |   |   |   | CH |   |

| Bit     | Value | Description                                   |
|---------|-------|---|
| ADDRESS | 0–59  | Start address in the PWM script command file. |
| CH      | 01    | PWM channel 0.                                |
|         | 10    | PWM channel 1.                                |
|         | 11    | PWM channel 2.                                |

### PWM\_STOP COMMAND

The PWM\_STOP command consists of a command byte (0x97) and a data byte from the host. This command stops execution of the script command file for the specified channel.

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0  |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | CH |

| Bit | Value | Description    |
|-----|-------|----------------|
| CH  | 01    | PWM channel 0. |
|     | 10    | PWM channel 1. |
|     | 11    | PWM channel 2. |



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

## ABSOLUTE MAXIMUM RATINGS<sup>(1)(2)</sup>

|  |                       |                          |
|--|-----------------------|--------------------------|
| Supply Voltage ( $V_{CC}$ )              |                       | 2V                       |
| Voltage at Any Pin                       |                       | -0.3V to $V_{CC} + 0.3V$ |
| Maximum Input Current Without Latchup    |                       | $\pm 100$ mA             |
| ESD Protection Level                     | (Human Body Model)    | 2 kV                     |
|  | (Machine Model)       | 200V                     |
|  | (Charge Device Model) | 750V                     |
| Total Current into $V_{CC}$ Pin (Source) |                       | 100 mA                   |
| Total Current out of GND Pin (Sink)      |                       | 100 mA                   |
| Storage Temperature Range                |                       | -65°C to +140°C          |

- (1) Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but specific performance is not ensured. For ensured specifications and test conditions, see the [DC ELECTRICAL CHARACTERISTICS](#) and [AC ELECTRICAL CHARACTERISTICS](#) tables.
- (2) If Military/Aerospace specified devices are required, please contact the TI Sales Office/ Distributors for availability and specifications.

## DC ELECTRICAL CHARACTERISTICS

(Temperature:  $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ , unless otherwise specified)

Data sheet specification limits are specified by design, test, or statistical analysis.

| Symbol     | Parameter  | Conditions  | Min                 | Typ | Max                 | Units         |
|------------|--|---|---------------------|-----|---------------------|---------------|
| $V_{CC}$   | Operating Voltage  |   | 1.62                |     | 1.98                | V             |
| $I_{DD}$   | Supply Current <sup>(1)</sup>                            | Internal Clock,   |                     |     |                     |               |
|            |  | No loads on pins,   |                     | 1.9 | 3.0                 | mA            |
|            |  | $V_{CC} = 1.9V$ , $T_C = 0.5 \mu\text{s}$ <sup>(2)</sup>        |                     |     |                     |               |
| $I_{HALT}$ | Standby Mode Current <sup>(3)</sup>                      | <b>Typical:</b><br>$V_{CC} = 1.9V$ , $T_A = 25^{\circ}\text{C}$ |                     | <9  | 40                  | $\mu\text{A}$ |
| $V_{IL}$   | Logical 0 Input Voltage <sup>(4)</sup>                   |   | $0.3 \times V_{CC}$ |     |                     | V             |
| $V_{IH}$   | Logical 1 Input Voltage <sup>(4)</sup>                   |   |                     |     | $0.7 \times V_{CC}$ | V             |
|            | Hi-Z Input Leakage (TRI-STATE Output)                    | $V_{CC} = 1.8V$   | -2                  |     | 2                   | $\mu\text{A}$ |
|            | Port Input Hysteresis <sup>(4) (5)</sup>                 |   | 100                 | 400 |                     | mV            |
|            | Weak Pull-Up/Pull-Down Current                           | $1.6V < V_{CC} < 2.0V$  |                     |     | 150                 | $\mu\text{A}$ |
|            | Output Current Source (Push-Pull Mode)                   | $V_{CC} = 1.62V$ , $V_{OH} = 0.7 \times V_{CC}$                 |                     |     | -16                 | mA            |
|            | Output Current Sink (Push-Pull Mode)                     | $V_{CC} = 1.62V$ , $V_{OL} = 0.3 \times V_{CC}$                 | 16                  |     |                     | mA            |
|            | Allowable Sink and Source Current per Pin <sup>(6)</sup> |   |                     |     | 16                  | mA            |
| $C_{PAD}$  | Input/Output Capacitance <sup>(5)</sup>                  |   |                     |     | 5                   | pF            |

- (1) Supply current is measured with inputs connected to  $V_{CC}$  and outputs driven low but not connected to a load.
- (2) Command execution cycle =  $0.5 \mu\text{s}$ .
- (3) In standby mode, the internal clock is switched off. Supply current in standby mode is measured with inputs connected to  $V_{CC}$  and outputs driven low but not connected to a load.
- (4) Applied to all digital pins (including RESET) except for SLOWCLK when configured for an external clock.
- (5) Specified by design, not tested.
- (6) The sum of all I/O sink/source current must not exceed the maximum total current into  $V_{CC}$  and out of GND as specified in the absolute maximum ratings.

## AC ELECTRICAL CHARACTERISTICS

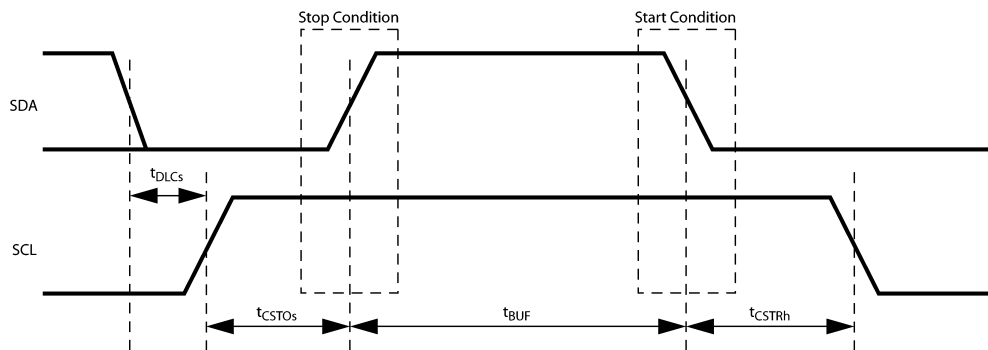
(Temperature:  $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ )

Data sheet specification limits are specified by design, test, or statistical analysis.

| Parameter  | Conditions                                   | Min | Typ | Max | Units         |
|--|--|-----|-----|-----|---------------|
| System Clock Frequency   | Internal RC                                  |     | 21  |     | MHz           |
| System Clock Period (mclk)   | $1.62\text{V} \leq V_{CC} \leq 1.98\text{V}$ |     | 48  |     | ns            |
| Processing and Command Execution Cycle ( $t_C$ )                             | $1.62\text{V} \leq V_{CC} \leq 1.98\text{V}$ |     | 0.5 |     | $\mu\text{s}$ |
| System Clock, Processing and Command Execution Cycle Variation               |  |     |     | 7   | %             |
| General-Purpose I/O (GPIO)   |  |     |     |     |               |
| Output Rise Time <sup>(1)</sup>  | $C_{LOAD} = 50\text{ pF}$                    |     |     | 15  | ns            |
| Output Fall Time <sup>(1)</sup>  |  |     |     | 15  | ns            |
| ACCESS.bus Input Signals   |  |     |     |     |               |
| Bus Free Time Between Stop and Start Condition ( $t_{BUFi}$ ) <sup>(1)</sup> |  | 16  |     |     | mclk          |
| SCL Setup Time ( $t_{CSTOs}$ ) <sup>(1)</sup>                                | Before Stop Condition                        | 8   |     |     | mclk          |
| SCL Hold Time ( $t_{CSTRh}$ ) <sup>(1)</sup>                                 | After Start Condition                        | 8   |     |     | mclk          |
| SCL Setup Time ( $t_{CSTRs}$ ) <sup>(1)</sup>                                | Before Start Condition                       | 8   |     |     | mclk          |
| Data High Setup Time ( $t_{DHCs}$ ) <sup>(1) (2)</sup>                       | Before SCL Rising Edge (RE)                  | 2   |     |     | mclk          |
| Data Low Setup Time ( $t_{DLCs}$ ) <sup>(1) (2)</sup>                        | Before SCL RE                                | 2   |     |     | mclk          |
| SCL Low Time ( $t_{SCLlow}$ ) <sup>(1)</sup>                                 | After SCL Falling Edge (FE)                  | 12  |     |     | mclk          |
| SCL High Time ( $t_{SCLhigh}$ ) <sup>(1) (2)</sup>                           | After SCL RE                                 | 12  |     |     | mclk          |
| SDA Hold Time ( $t_{SDAh}$ ) <sup>(1)</sup>                                  | After SCL FE                                 | 0   |     |     | mclk          |
| SDA Setup Time ( $t_{SDAs}$ ) <sup>(1) (2)</sup>                             | Before SCL RE                                | 2   |     |     | mclk          |
| ACCESS.bus Output Signals  |  |     |     |     |               |
| SCL Hold Time ( $t_{SDAho}$ ) <sup>(1)</sup>                                 | After SCL FE                                 | 7   |     |     | mclk          |

(1) Specified by design, not tested.

(2) The ACCESS.bus interface implements and meets the timing necessary for interface to the I<sup>2</sup>C and SMBus protocol at logic levels. The bus drivers are designed with open-drain output for bidirectional operation. Due to System Clock (mclk) Variation, this specification may not meet the AC timing and current/voltage drive requirements of the full bus specifications.



Note: In the timing tables the parameter name is added with an "o" for output signal timing and "i" for input signal timing.

**Figure 21. ACB Start and Stop Condition Timing**



---

**REVISION HISTORY**

| <b>Changes from Original (March 2013) to Revision A</b>  | <b>Page</b>           |
|--|-----------------------|
| <hr/> <ul style="list-style-type: none"><li>• Changed layout of National Data Sheet to TI format .....</li></ul> <hr/> | <hr/> <b>40</b> <hr/> |

**PACKAGING INFORMATION**

| Orderable Device    | Status<br>(1) | Package Type | Package<br>Drawing | Pins | Package<br>Qty | Eco Plan<br>(2) | Lead finish/<br>Ball material<br>(6) | MSL Peak Temp<br>(3) | Op Temp (°C) | Device Marking<br>(4/5) | Samples                 |
|---------------------|---------------|--------------|--------------------|------|----------------|-----------------|--------------------------------------|----------------------|--------------|-------------------------|-------------------------|
| LM8323JGR8AXM/NOPB  | ACTIVE        | csBGA        | NYB                | 36   | 1000           | RoHS & Green    | SNAGCU                               | Level-1-260C-UNLIM   | -40 to 85    | LM8323                  | <a href="#">Samples</a> |
| LM8323JGR8AXMX/NOPB | ACTIVE        | csBGA        | NYB                | 36   | 3500           | RoHS & Green    | SNAGCU                               | Level-1-260C-UNLIM   | -40 to 85    | LM8323                  | <a href="#">Samples</a> |

(1) The marketing status values are defined as follows:

**ACTIVE:** Product device recommended for new designs.

**LIFEBUY:** TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

**NRND:** Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

**PREVIEW:** Device has been announced but is not in production. Samples may or may not be available.

**OBSELETE:** TI has discontinued the production of the device.

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".

**RoHS Exempt:** TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.

**Green:** TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) Lead finish/Ball material - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

**Important Information and Disclaimer:**The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.



**TAPE AND REEL INFORMATION**

**QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE**


\*All dimensions are nominal

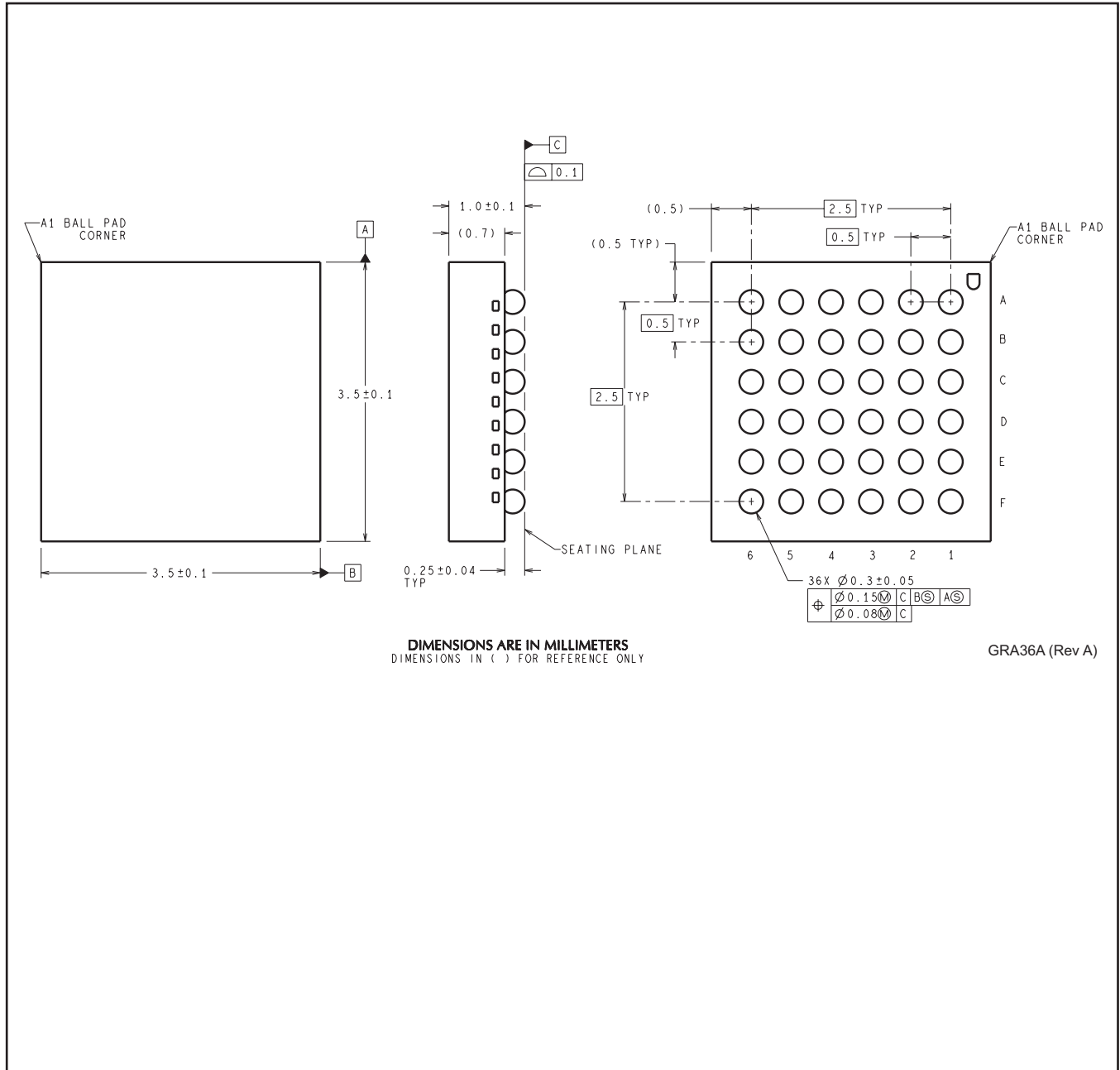
| Device              | Package Type | Package Drawing | Pins | SPQ  | Reel Diameter (mm) | Reel Width W1 (mm) | A0 (mm) | B0 (mm) | K0 (mm) | P1 (mm) | W (mm) | Pin1 Quadrant |
|---------------------|--------------|-----------------|------|------|--------------------|--------------------|---------|---------|---------|---------|--------|---------------|
| LM8323JGR8AXM/NOPB  | csBGA        | NYB             | 36   | 1000 | 178.0              | 12.4               | 3.8     | 3.8     | 1.5     | 8.0     | 12.0   | Q1            |
| LM8323JGR8AXMX/NOPB | csBGA        | NYB             | 36   | 3500 | 330.0              | 12.4               | 3.8     | 3.8     | 1.5     | 8.0     | 12.0   | Q1            |

**TAPE AND REEL BOX DIMENSIONS**


\*All dimensions are nominal

| Device              | Package Type | Package Drawing | Pins | SPQ  | Length (mm) | Width (mm) | Height (mm) |
|---------------------|--------------|-----------------|------|------|-------------|------------|-------------|
| LM8323JGR8AXM/NOPB  | csBGA        | NYB             | 36   | 1000 | 208.0       | 191.0      | 35.0        |
| LM8323JGR8AXMX/NOPB | csBGA        | NYB             | 36   | 3500 | 367.0       | 367.0      | 35.0        |

NYB0036A



DIMENSIONS ARE IN MILLIMETERS  
DIMENSIONS IN ( ) FOR REFERENCE ONLY

GRA36A (Rev A)

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](http://ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2021, Texas Instruments Incorporated