National Semiconductor

May 2007

LM8322
Mobile I/O Companion Supporting Key-Scan, I/O Expansion, PWM, and ACCESS.bus Host Interface

1.0 General Description

The LM8322 Mobile I/O Companion is a dedicated device to unburden a host processor from scanning a matrix-addressed keypad. In addition, the LM8322 provides general-purpose I/O expansion, and PWM outputs useful for dynamic LED brightness modulation.

It communicates with the host through an I²C-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 LM8322 supports a predefined set of commands. These commands enable a host device to keep control over all functions.

2.0 Features

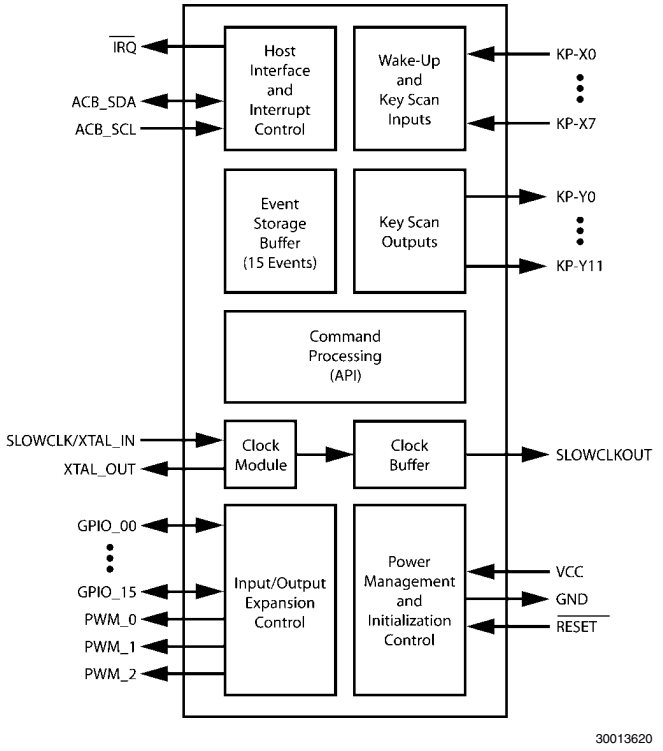
Key Features

- 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 I²C-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 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 LM8322 after receiving a command.
- 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.

3.0 Applications

- Mobile phones
- Personal Digital Assistants (PDAs)
- Smart handheld devices
- Personal media players

4.0 Block Diagram

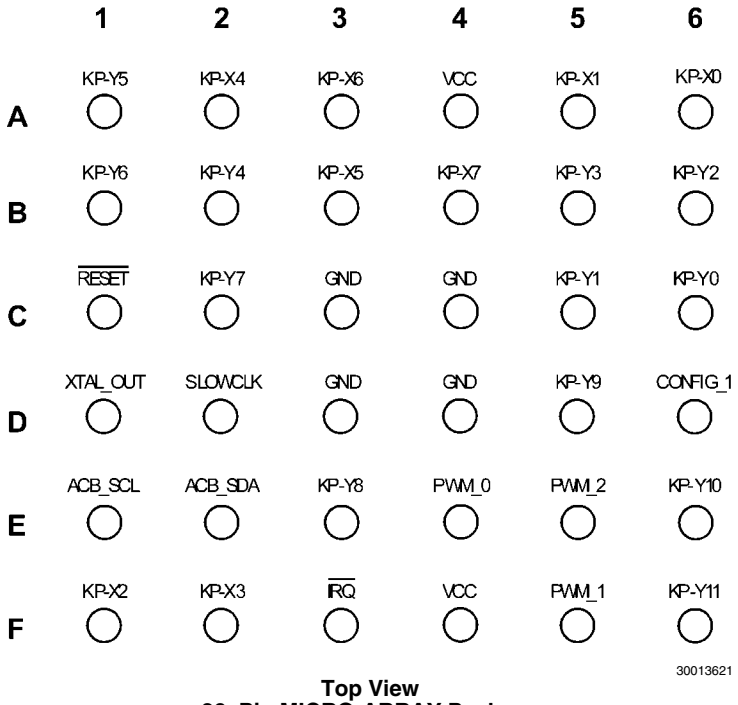


5.0 Ordering Information

NSID	Spec.	No. of Pins	Package Type	Temperature	Package Method
LM8322JGR8	NOPB	36	Micro-Array	−40 to + 85°C	1000 pcs Tape & Reel
LM8322JGR8X	NOPB	36	Micro-Array	−40 to + 85°C	3500 pcs Tape & Reel

NOPB = No PB (No Lead)

6.0 Pin Assignments



Top View
36–Pin MICRO-ARRAY Package
See NS Package Number GRA36A

Table of Contents

7.0 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
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

Pin	Function	I/O	Description
D6	CONFIG_1	Input	Slave address select input 1
	GPIO_14	I/O	General-purpose I/O port 14
D1	XTAL_OUT	Input	32.768 kHz crystal output
D2	SLOWCLK	Input	32.768 kHz clock
	XTAL_IN	Input	32.768 kHz crystal input
F3	$\overline{IRQ}$	Output	Interrupt request output
C1	$\overline{RESET}$	Input	Reset Input
A4, F4	V_{CC}	n.a.	V_{CC}
C3, C4, D3, D4	GND	n.a.	Ground

7.1 TERMINATION OF UNUSED SIGNALS

TABLE 1. Termination of Unused Signals

Signal	Termination
$\overline{RESET}$	Connect to V_{CC} if not driven from an external Supervisory circuit.
CONFIG_1	Connect to V_{CC} 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 V_{CC} 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: • Connect to V_{CC} or GND. • Program as inputs with weak pullups or outputs. Care must be taken when connecting to V_{CC} 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-X7 can only be an input. This pin should be programmed as an input with a weak pullup.
KP-Y[2:0]	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: • Connect to V_{CC} or GND. • Program as inputs with weak pullups or outputs Care must be taken when connecting to V_{CC} 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, PWM_1	These pins must be connected to V_{CC} or GND if they are not used for any optional function described in the datasheet.
PWM_2/ CONFIG_2	Connect to V_{CC} 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.
$\overline{IRQ}$	This pin must be connected.

8.0 Application Example

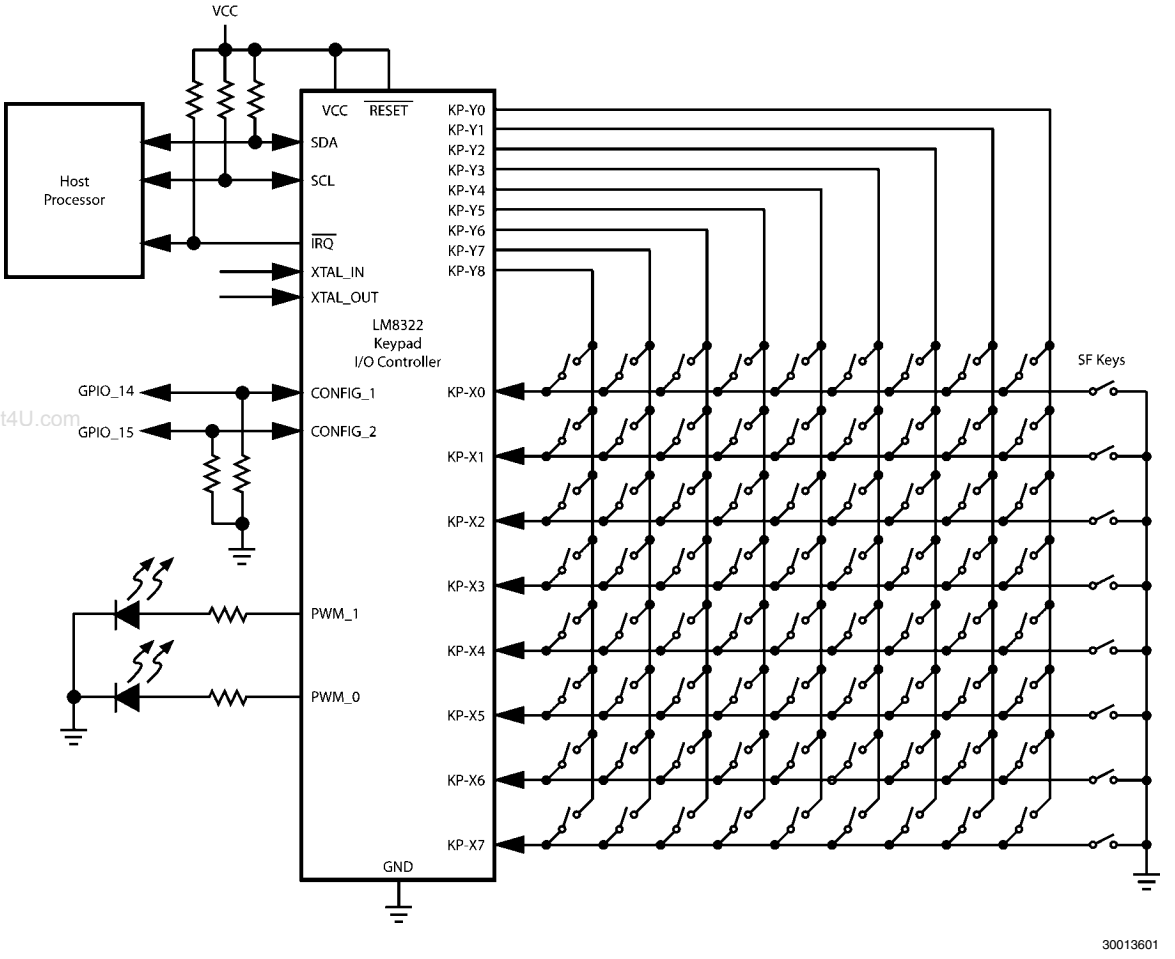


FIGURE 1. Typical Application

8.1 FEATURES

The application example shown in *Figure 1* 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^2C-compatible) interface for communication with the host.
- Hardware $\overline{IRQ}$ interrupt to host to signal keypad, 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 LM8322 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.
- 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 V_{CC} 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.

9.0 Clocks

- **System Clock (mclk)** — The system clock is in the range of about 21 MHz (± 7%) typical. This clock is used to drive the I²C compatible serial ACCESS bus and is the input clock for other function blocks.
- **Processing and Command Execution Clock (t_C)** — The internal processing is based on a 2 MHz clock. This clock is derived from the System Clock.
- **Internal PWM Clock** — The internal PWM clock is a fixed scaled down clock (÷ 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.

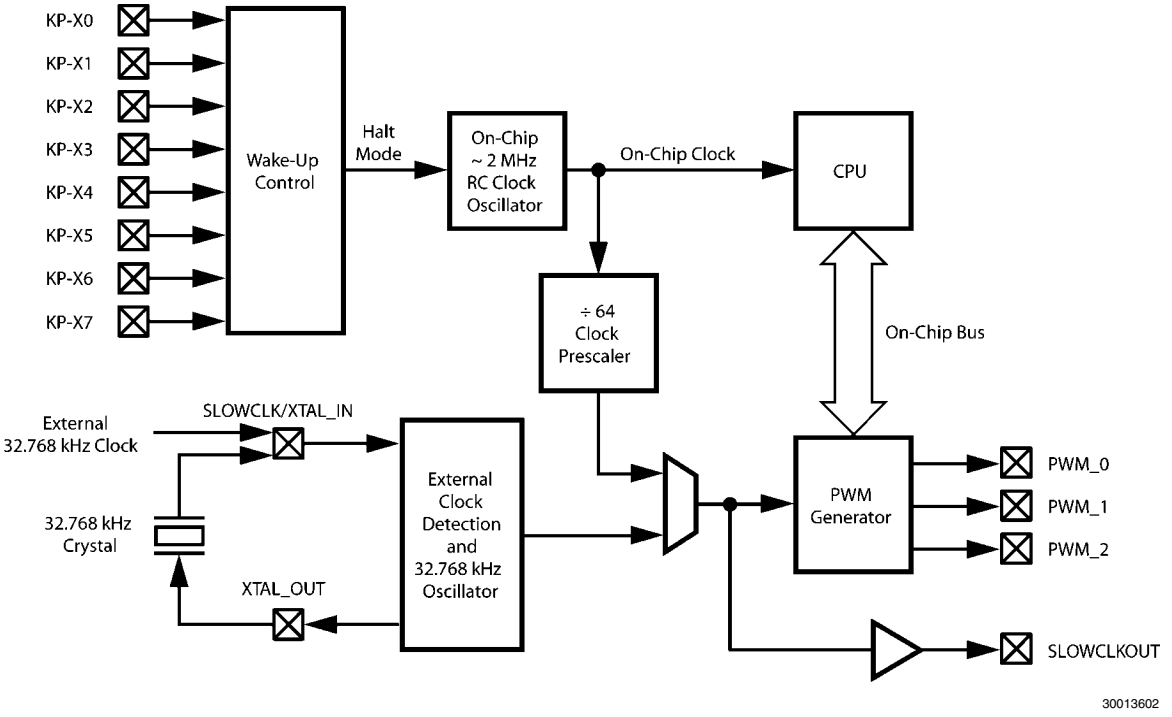


FIGURE 2. Clock Architecture

9.1 INTERNAL EXECUTION CYCLE

The Processing - and Command - execution clock is about 2 MHz. This clock is stopped in Halt mode, which only occurs under control of the LM8322. 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.
- Assertion of the $\overline{\text{RESET}}$ input.

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

9.2 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 V_{CC} or GND.

9.3 CLOCK CONFIGURATION

Table 2 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 2. 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.

10.0 Reset

The LM8322 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.

10.1 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 guarantee a valid reset. If $\overline{\text{RESET}}$ is asserted at power-on, it must be held low until V_{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 (5×) greater than the V_{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 LM8322 comes out of the reset state within about 1400 ns.

10.2 POWER-ON RESET (POR)

The POR circuit is always enabled. When V_{CC} rises above the POR threshold voltage VPOR (about 1.2–1.5V), an on-chip reset signal is asserted. The V_{CC} rise time must be greater than 20 µs and less than 10 ms, otherwise the on-chip reset signal may deassert before V_{CC} reaches the minimum operating voltage. While V_{CC} is below VPOR, the LM8322 is held in reset and a timer clocked by the on-chip RC clock is preset with 0xFF (256 clock cycles). When V_{CC} reaches a value greater than VPOR, the timer starts counting down. When it underflows, the on-chip reset signal is deasserted and the LM8322 begins operation.

10.3 PIN CONFIGURATION AFTER RESET

Table 2 shows the pin configuration after reset.

TABLE 3. Pin Configuration After Reset

Pins	After Reset	After LM8322 Initialization
KP-X00 KP-X01 KP-X02	High-impedance mode.	Input mode with an on-chip pullup enabled.
KP-X03 KP-X04 KP-X05 KP-X06 KP-X07	High-impedance mode.	High-impedance mode, until host configures them as keypad inputs or GPIO.
KP-Y00 KP-Y01 KP-Y02	High-impedance mode.	Active drive low.
KP-Y03 KP-Y04 KP-Y05 KP-Y06 KP-Y07 KP-Y08 KP-Y09 KP-Y10 KP-Y11	High-impedance mode.	High-impedance mode, until host configures them as keypad outputs or GPIO.
CONFIG_1 CONFIG_2	High-impedance mode.	The ACCESS.bus slave address must be selected with external pullup or pulldown resistors or direct connections to V_{CC} or GND.
$\overline{\text{IRQ}}$	High-impedance mode.	Active drive low.
PWM_0 PWM_1 PWM_2	High-impedance mode.	High-impedance mode.
ACB_SDA ACB_SCL	Open-drain mode.	Open-drain mode.
XTAL_IN	High-impedance mode.	High-impedance mode. Terminate to V_{CC} or GND if not used.
XTAL_OUT	Weak pullup device.	Weak pullup device.
$\overline{\text{RESET}}$	High-impedance mode.	High-impedance mode.

10.4 DEVICE CONFIGURATION AFTER RESET

After the LM8322 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.
- **Digital Multiplexers** — disabled.
- **$\overline{\text{IRQ}}$** — enabled, active low.
- **NOINIT Bit** — set.
- **Debounce Time** — 3 scan cycles (about 12 milliseconds).
- **Active Time** — 500 milliseconds.

10.5 CONFIGURATION INPUTS

The states sampled from the CONFIG_1 and CONFIG_2 inputs during reset select the ACCESS.bus address used by the LM8322, as shown in *Table 4*. 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 4. 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-kohm resistor to ground can impose a logic 0 during reset without interfering with normal operation as a GPIO port.

10.6 INITIALIZATION

The LM8322 waits for a WRITE_CFG command from the host. During this time, $\overline{\text{IRQ}}$ is asserted to request service from the host. *Figure 3* describes the behavior of the LM8322 following reset.

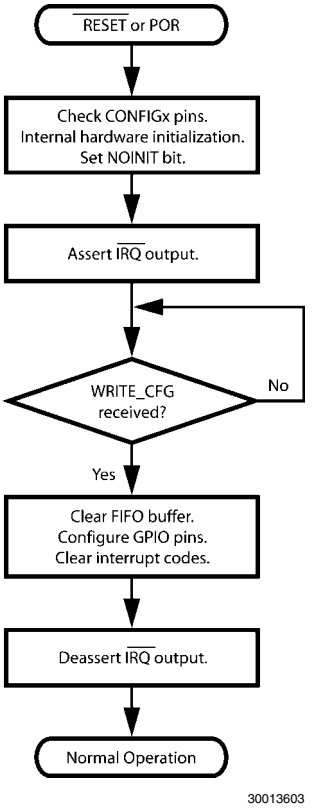


FIGURE 3. LM8322 Initialization Behavior

Figure 4 shows the timing of $\overline{\text{IRQ}}$ relative to a $\overline{\text{RESET}}$ or POR event and the WRITE_CFG command. 100 µs 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 µs after a WRITE_CFG command is received, $\overline{\text{IRQ}}$ is deasserted.

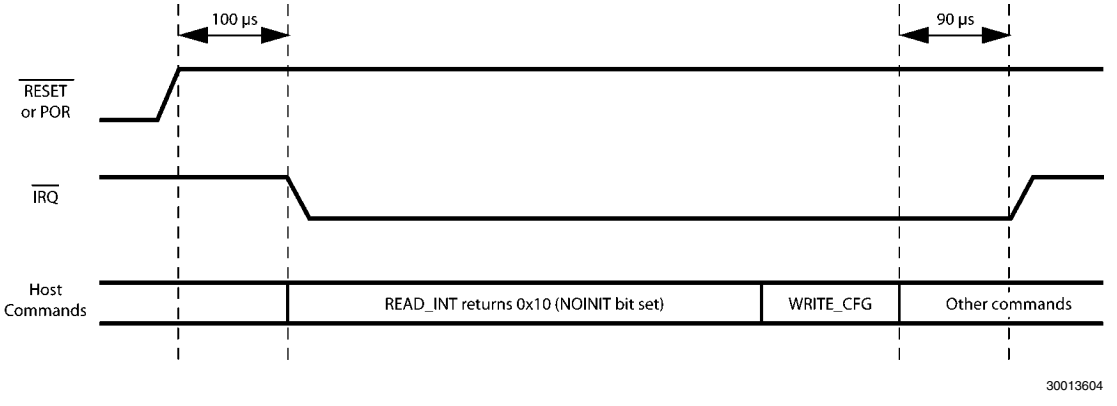


FIGURE 4. $\overline{\text{IRQ}}$ Reset Timing

After sending the WRITE_CFG command, the host must send a series of commands to configure the LM8322, as shown in *Figure 5* (see left hand side).

This Flow - diagram illustrates also the basic host communication steps which the host must execute upon an $\overline{IRQ}$ request received from the LM8322 during operation. Such requests will be made from the LM8322 as a result of key pressed events, the detection of an error, the termination of a PWM cycle and others.

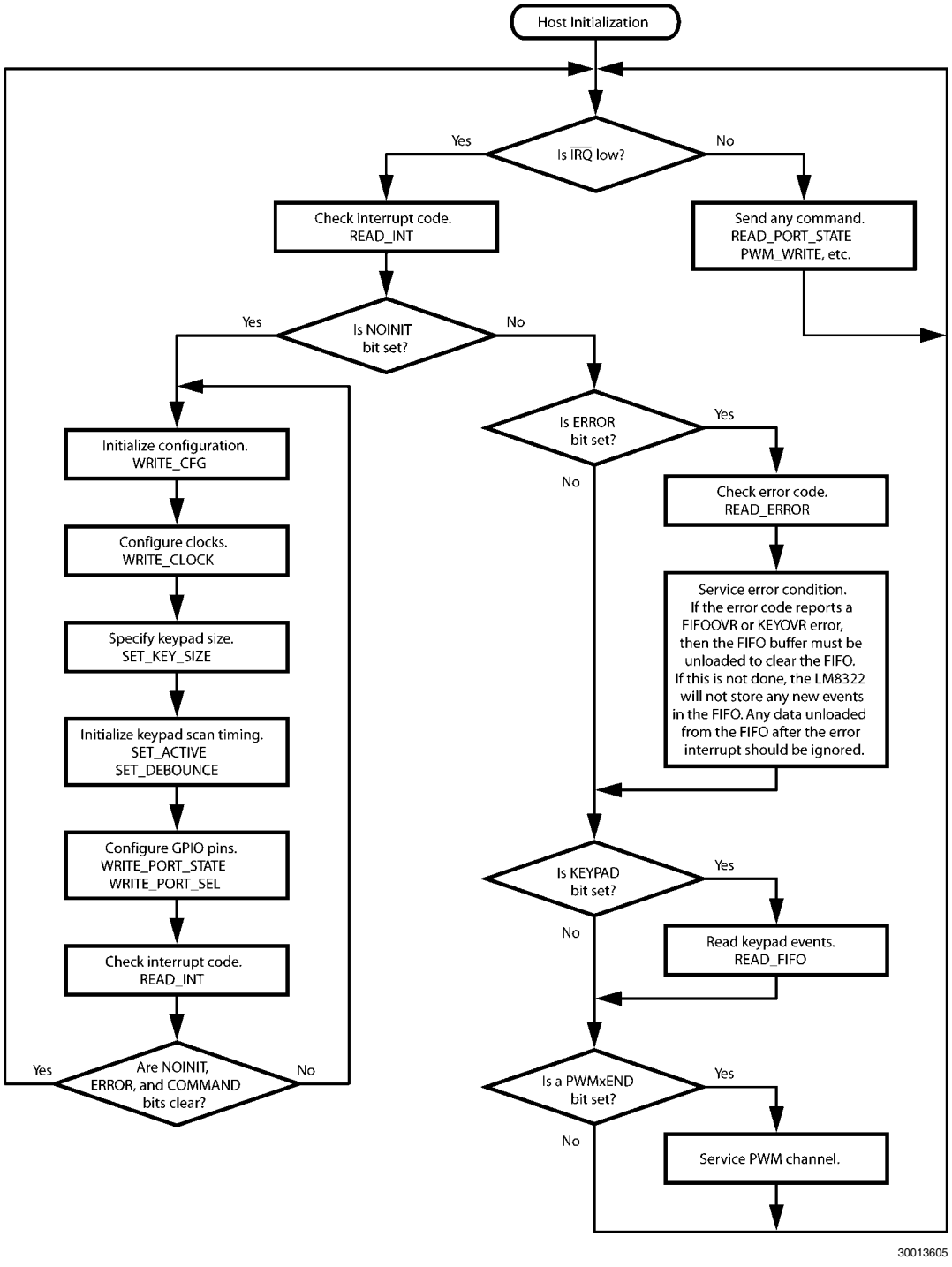


FIGURE 5. Host-Side LM8322 Initialization

10.7 INITIALIZATION EXAMPLE

In the following example, the LM8322 is configured as:

- Keypad matrix configuration is 8 × 4.
- 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 GPIO s 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.

11.0 Halt Mode

The fully static architecture of the LM8322 allows stopping the internal RC clock in Halt mode, which reduces power consumption to the minimum level. *Figure 6* shows the current in Halt mode at the maximum V_{CC} (1.98V) from 25°C to +85°C.

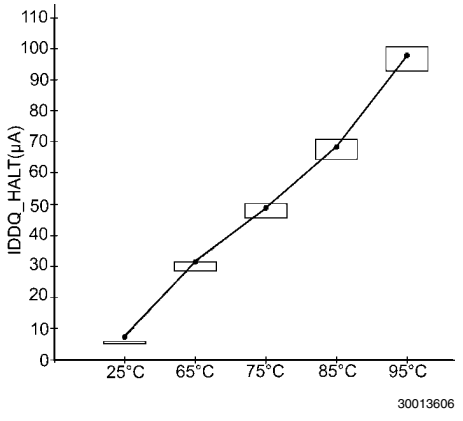


FIGURE 6. Halt Current vs. Temperature at 1.98V

Halt mode is entered when no key-press event, key-release event, or ACCESS.bus activity is detected for a certain period of time (by default, 500 milliseconds). 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.

11.1 ACCESS.bus ACTIVITY

When the LM8322 is in Halt mode, any activity on the ACCESS.bus interface will cause the LM8322 to exit from Halt mode. However, the LM8322 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 LM8322 will be prevented from entering Halt mode if it shares the bus with peripherals that are continuously active. For lowest power consumption, the LM8322 should only share the bus with peripherals that require little or no bus activity after system initialization.

12.0 Keypad Interface

12.1 EVENT CODE ASSIGNMENT

After power-on reset and host initialization, the LM8322 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 <5 µA standby current).

Table 5 lists the codes assigned to the matrix positions encoded by the hardware. Key-press events are assigned the codes listed in *Table 5* , but with the MSB set. When a key is released, the MSB of the code is clear.

TABLE 5. 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
KP-X0	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0F
KP-X1	0x11	0x12	0x13	0x14	0x15	0x16	0x17	0x18	0x19	0x1A	0x1B	0x1C	0x1F
KP-X2	0x21	0x22	0x23	0x24	0x25	0x26	0x27	0x28	0x29	0x2A	0x2B	0x2C	0x2F
KP-X3	0x31	0x32	0x33	0x34	0x35	0x36	0x37	0x38	0x39	0x3A	0x3B	0x3C	0x3F
KP-X4	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4F
KP-X5	0x51	0x52	0x53	0x54	0x55	0x56	0x57	0x58	0x59	0x5A	0x5B	0x5C	0x5F
KP-X6	0x61	0x62	0x63	0x64	0x65	0x66	0x67	0x68	0x69	0x6A	0x6B	0x6C	0x6F
KP-X7	0x71	0x72	0x73	0x74	0x75	0x76	0x77	0x78	0x79	0x7A	0x7B	0x7C	0x7F

The codes are loaded into the FIFO buffer in the order in which they occurred. *Table 6* shows an example sequence of events, and *Figure 7* shows the resulting sequence of event codes loaded into the FIFO buffer.

TABLE 6. 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

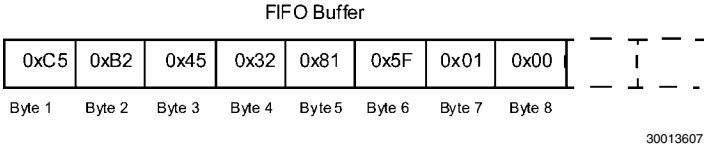


FIGURE 7. Example Event Codes Loaded in FIFO Buffer

12.2 KEYPAD SCAN CYCLES

The LM8322 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 8 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.

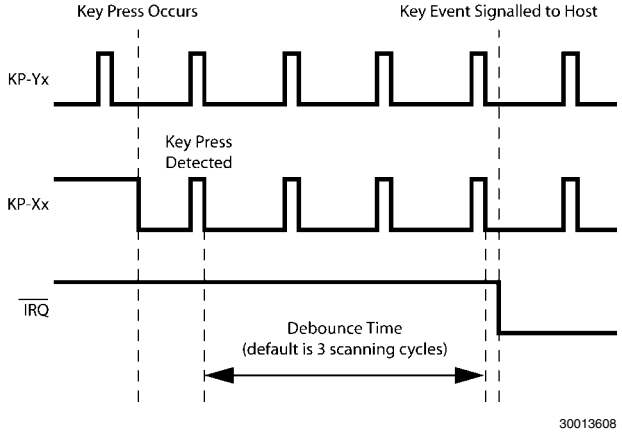


FIGURE 8. Keypad Scan Cycles

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 8*, 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 LM8322 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 5 nF.

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 $\overline{IRQ}$ output is asserted.

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

12.2.1 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 $\overline{IRQ}$. 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 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.

12.2.2 Multiple Key Pressings

If more than two keys are pressed at the same time, the LM8322 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 msec to 1 second) it is not safe to detect two simultaneous key pressings in one input row (see *Figure 9* on the left hand side.)
- If all key pressings (two or more) are located in different input rows (see *Figure 9* on the right hand side) then the key pressed events will be correctly found in the FIFO buffer without any restriction.

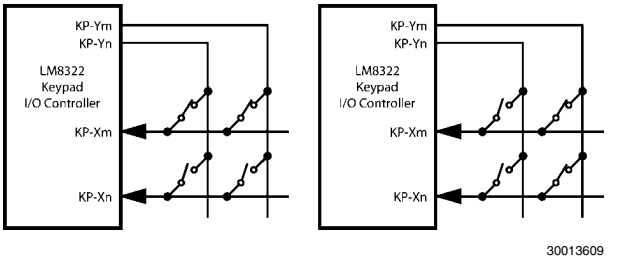


FIGURE 9. Simultaneous Keys Pressed

- 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 LM8322 from entering HALT mode. If all keyboard events are resolved (no remaining key pressed status in the LM8322 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 LM8322 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 LM8322.

12.3 EXAMPLE KEYPAD CONFIGURATION

Figure 10 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.

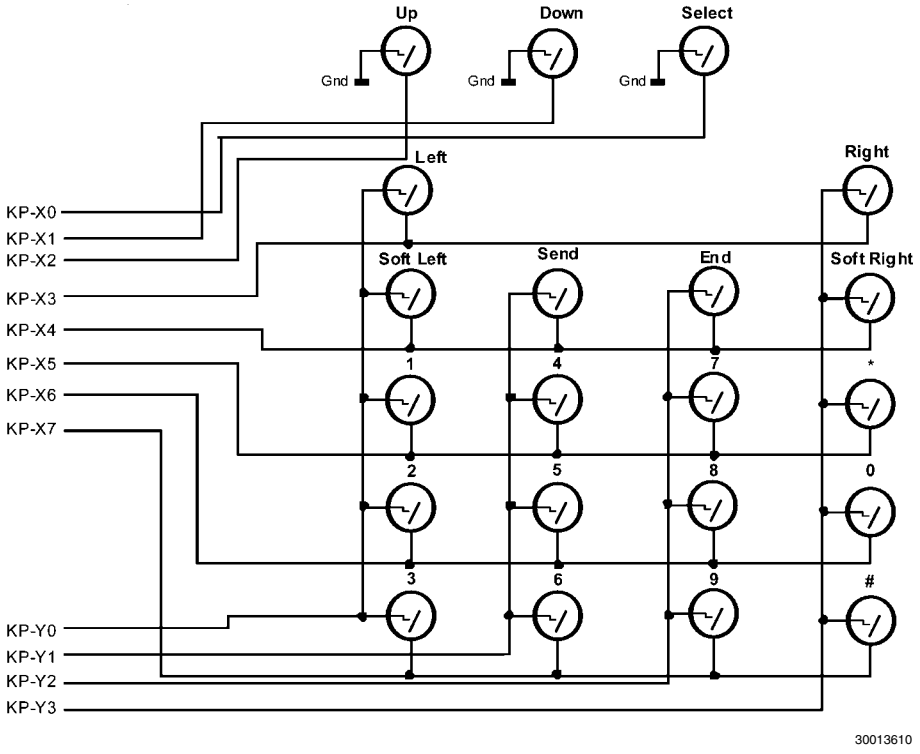


FIGURE 10. Keypad Interface Example

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 LM8322 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 10*, the SET_KEY_SIZE command would specify 8 KP-Xx inputs and 4 KP-Yx outputs.

13.0 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 7 shows the GPIO port configurations selected by the bits in the WRITE_PORT_SEL, WRITE_PORT_STATE, and WRITE_PULL_DOWN command parameters.

TABLE 7. 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.

13.1 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 ACCESS.bus (I²C) 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.

13.2 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 µs after the command is acknowledged, and the others change at intervals of 7.3 µs, as shown in *Figure 11*.

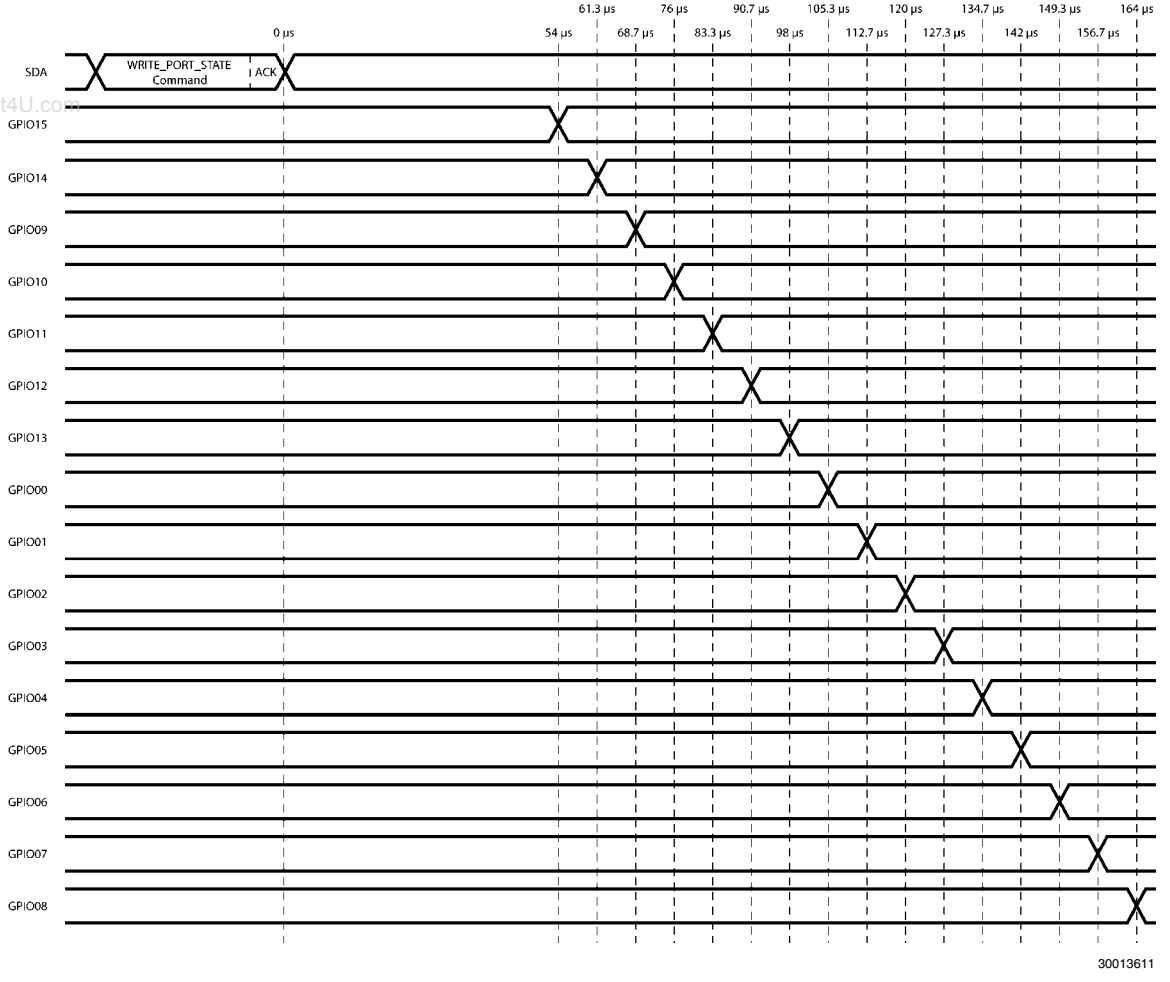


FIGURE 11. GPIO Port State Change Timing

14.0 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 12* shows the architecture of a script-execution engine.

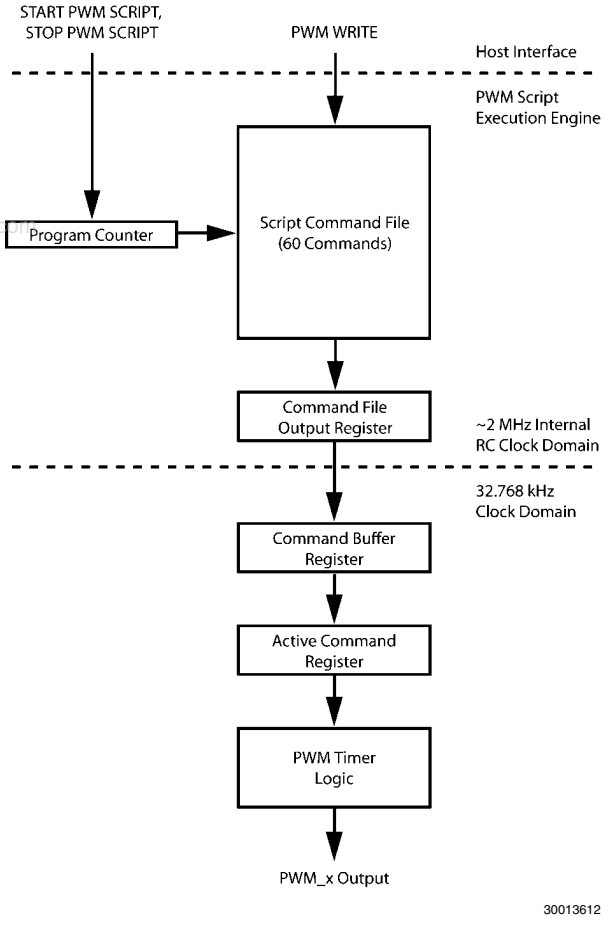


FIGURE 12. PWM Script Execution Engine

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.

Please 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 LM8322 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.

14.1 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 with the Reset bit set to 1. Executing an END command with the Reset bit set to “1” or the reception of a PWM_STOP command asserts $\overline{\text{IRQ}}$ to the host.

14.2 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 13* shows the architecture of a PWM timer.

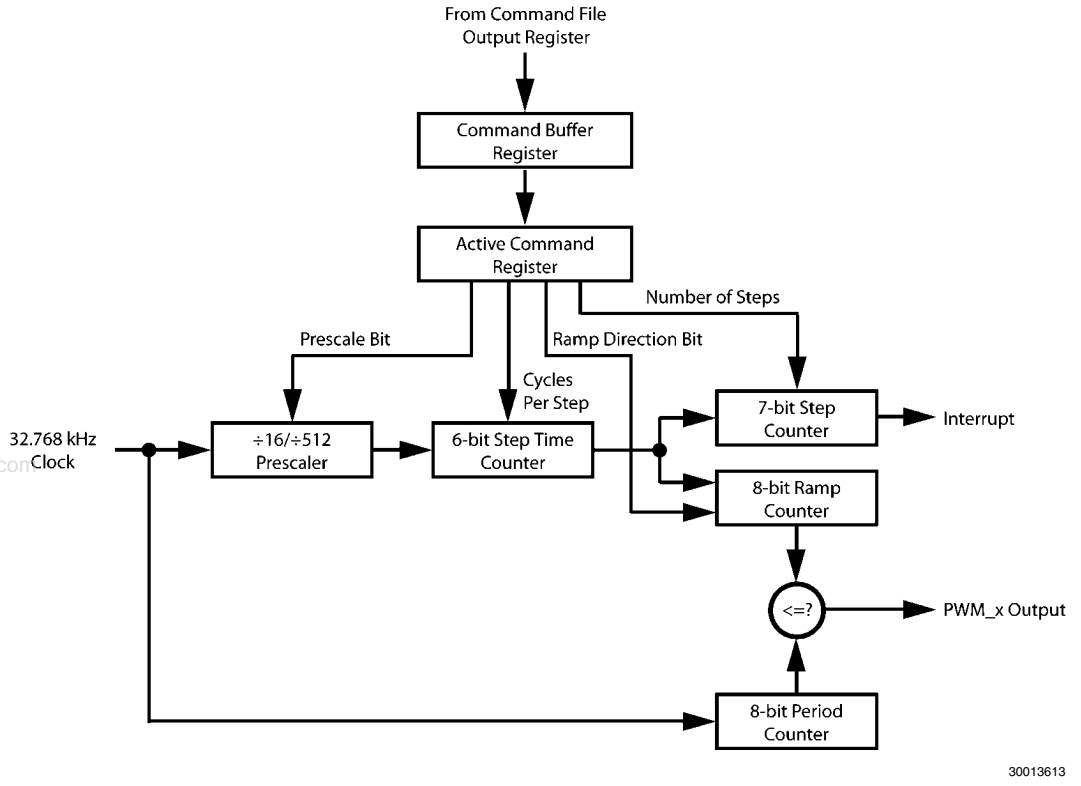


FIGURE 13. PWM Timer

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 $\overline{IRQ}$ output to the host after the last step is performed.

14.3 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 CALE	STEPTIME						SIGN	INCREMENT						
SET_PWM	0	1	0						PWMVALUE							
GO_TO_ START	0															
BRANCH	1	0	1	LOOPCOUNT						0	ADDRESS					
END	1	1	0	0	RESET	0										
TRIGGER	1	1	1	WAITTRIGGER						SENDTRIGGER						0

14.4 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 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

14.5 SET_PWM COMMAND

The SET_PWM command loads the ramp counter from the 8-bit DUTYCYCLE field in the instruction.

Please 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%.

14.6 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															

14.7 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.

14.8 END COMMAND

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

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

Please 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 $\overline{\text{IRQ}}$ 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
$\overline{\text{RESET}}$	0	PWM_x output is active when script execution terminates.
	1	PWM_x output is Tristate when script execution terminates.

14.9 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
	0001xx	Wait for trigger from channel 2
WAITTRIGGER	000xx1	Wait for trigger from channel 0
	000x1x	Wait for trigger from channel 1
SENDTRIGGER	000xx1	Send trigger to channel 0
	000x1x	Send trigger to channel 1
	0001xx	Send trigger to channel 2

14.10 PWM SCRIPT EXAMPLE

This example shows a complex ramping sequence that uses triggers for synchronization. Three scripts implement the example. *Figure 14* shows the PWM outputs for this example.

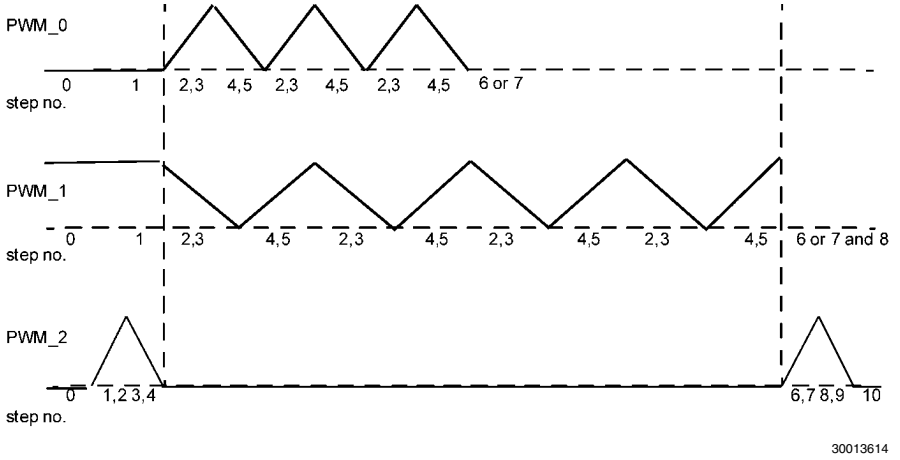


FIGURE 14. PWM Outputs

14.10.1 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 $\overline{IRQ}$ to host

14.10.2 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 $\overline{IRQ}$ to host

14.10.3 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 $\overline{IRQ}$ to host

14.11 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 $\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ 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.

15.0 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.

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

16.0 Host Interface

The two-wire ACCESS.bus interface is used to communicate with a host. The ACCESS.bus interface is fully compliant with the I²C bus standard. The LM8322 operates as a bus slave at 400 kHz (Fast mode).

All communication with the LM8322 over the ACCESS.bus interface is initiated by the host, usually in response to an interrupt request ($\overline{IRQ}$ low) asserted by the LM8322. The LM8322 may request service from the host by asserting the $\overline{IRQ}$ interrupt output.

16.1 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.

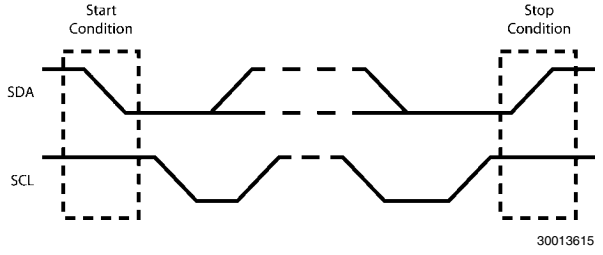


FIGURE 15. Start and Stop Conditions

16.2 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 LM8322 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 LM8322
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 LM8322 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 µs. A very long command chain could prevent the LM8322 from performing any watchdog service and consequently could trigger a physical $\overline{RESET}$ to the device.

To avoid overrunning the LM8322, the host should not send a Start condition or a Repeated Start condition less than 100 µs after the last Stop condition or the last clock of a preceding command.

16.3 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

If the CONFIG_1 and CONFIG_2 pins are left open, on-chip pullups will select 1000 101X by default.

16.4 HOST WRITE COMMANDS

Some host commands include one or more data bytes written to the LM8322. *Figure 16* 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 send the data, in this case specifying the number of rows and columns for the keypad.

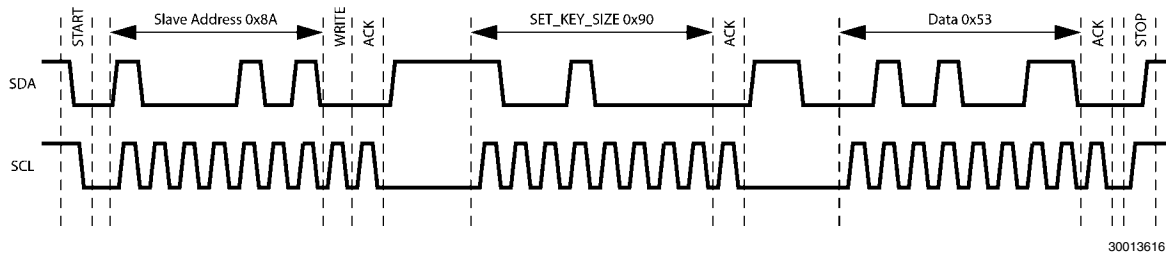


FIGURE 16. Host Write Command

16.5 HOST READ COMMANDS

Some host commands include one or more data bytes read from the LM8322. *Figure 17* 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 LM8322. The second address byte is sent with the direction bit undriven (pulled high) to indicate a read transaction of the data from the LM8322.

The Start (or 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.

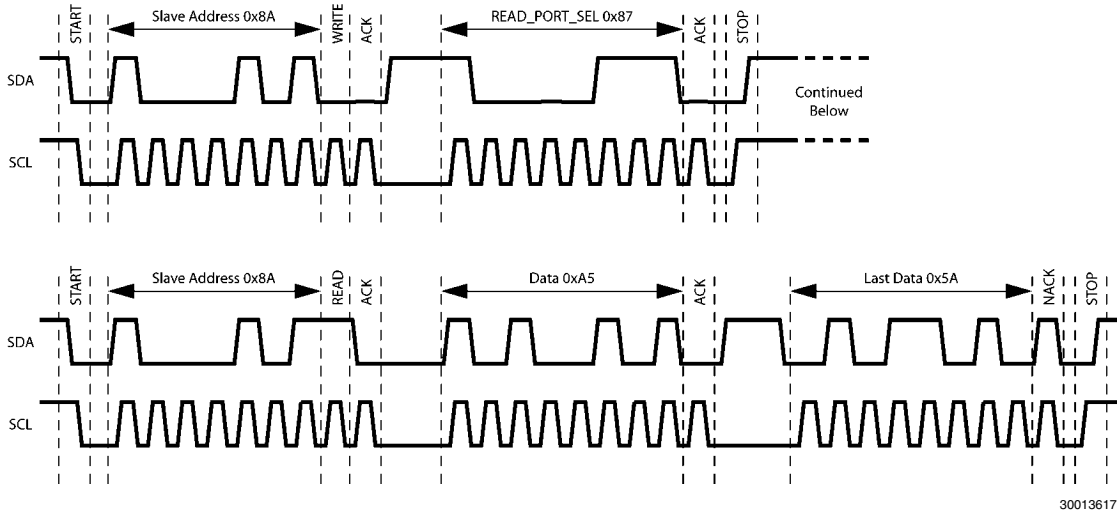


FIGURE 17. Host Read Command

16.6 INTERRUPTS

The $\overline{IRQ}$ 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.
- Termination of a PWM script (END command).
- Any error condition, which is indicated by the error code.

16.7 INTERRUPT CODE

The interrupt code is read and acknowledged with the READ_INT command (0x82). This command clears the code and deasserts the $\overline{IRQ}$ 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	0	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 LM8322 is waiting for an initialization sequence.
ERROR	An error condition occurred.
KEYPAD	A key-press or key-release event occurred.

16.8 ERROR CODE

If the LM8322 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	FIFOOVR	0	0	0	KEYOVR	CMDUNK	BADPAR

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

16.9 WAKE-UP FROM HALT MODE

Any bus transaction initiated by the host may encounter the LM8322 device in Halt mode or busy with processing data, such as controlling the FIFO buffer or executing interrupt service routines.

Figure 18 shows the case in which the host sends a command while the LM8322 is in Halt mode (Internal execution clock is stopped). Any activity on the ACCESS.bus wakes up the LM8322, 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 LM8322.

If the LM8322 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 18*). 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 LM8322 will be able to acknowledge the slave address, because it will be in Active mode.

Alternatively, the I²C specification allows sending a START byte (00000001), which will not be acknowledged by any device. This byte can be used to wake up the LM8322 from Halt mode. The LM8322 may also stall the bus transaction by pulling the SCL low, which is a valid behavior defined by the I²C specification.

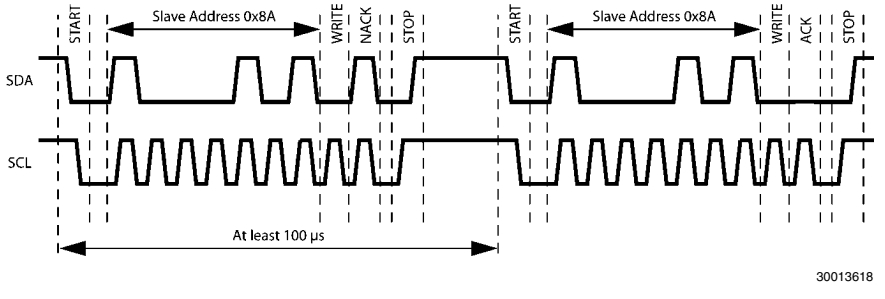


FIGURE 18. LM8322 Responds with NACK, Host Retries Command

17.0 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 $\overline{IRQ}$ output, and clear the code. (If the NOINIT bit is set, it remains set and $\overline{IRQ}$ remains asserted until a WRITE_CFG command is received.
$\overline{RESET}$	0x83	W	nnnn nnnn	Reset the LM8322. 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 LM8322 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 × 4 milliseconds.
READ_ERROR	0x8C	R	nnnn nnnn	Read and clear the error code.
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 × 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.
PWM_WRITE	0x95	W	aaaa aann	Write a command to the PWM script command file.
			pppp pppp	nn = PWM channel number (01, 10, or 11)
			qqqq qqqq	aaaaaa = address in script command file (059)
				pppp pppp = high byte of script command
				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).

Please note: The data bytes which follow the command can be reads (toward the host) or writes (toward the LM8322). 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).

17.1 READ_ID COMMAND

The READ_ID command consists of a command byte (0x80) from the host and two data bytes from the LM8322.

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

17.2 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	0	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.
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.

Please note: The WRITE_CFG COMMAND defines basic hardware operation characteristics. It should be placed at the beginning of the initialization sequence driven from the host device after power on. It is not recommended to change the configuration during run time. Anytime this command is used, it initializes important operating characteristics such as the the GPIO port. This means that the GPIO pins must be re-established via WRITE_PORT_SEL and WRITE_PORT_STATE commands in this case.

17.3 READ_INT COMMAND

The READ_INT command consists of a command byte (0x82) from the host and a data byte from the LM8322. 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.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	0	0	0	0	0	1	0	PWM2END	PWM1END	PWM0END	NOINIT	ERROR	0	0	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	LM8322 is waiting for the initialization sequence.
ERROR	0	No error condition is indicated.
	1	An error condition occurred.
KEYPAD	0	No key-press or key-release event is indicated.
	1	A key-press or key-release event occurred.

17.4 RESET COMMAND

The $\overline{RESET}$ 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.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	0	0	0	0	0	1	1	1	0	1	0	1	0	1	0

17.5 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.

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	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.

17.6 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.

17.7 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.

Bit	Value	Description
GPIO_xx	0	If the GPIO port pin is an input, pullup/pulldown 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/pulldown device is enabled. If the GPIO port pin is an output, it is driven high.

17.8 READ_PORT_SEL COMMAND

The READ_PORT_SEL command consists of a command byte (0x87) from the host and two data bytes from the LM8322. 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.

17.9 READ_PORT_STATE COMMAND

The READ_PORT_STATE command consists of a command byte (0x88) from the host and two data bytes from the LM8322. 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.

17.10 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 LM8322. The LM8322 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 5*.

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.

17.11 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 LM8322. 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.

17.12 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 LM8322 stays active without detecting a key-press or key-release 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 × 4 milliseconds.

17.13 READ_ERROR COMMAND

The READ_ERROR command consists of a command byte (0x8C) from the host and a data byte from the LM8322. 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	FIFOOVR	0	0	0	KEYOVR	CMDUNK	BADPAR

Bit	Value	Description
FIFOOVR	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.

17.14 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 LM8322 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.

17.15 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 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.

17.16 READ_KEY_SIZE COMMAND

The READ_KEY_SIZE command consists of a command byte (0x91) from the host and a data byte from the LM8322. 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.

17.17 READ_CFG COMMAND

The READ_CFG command consists of a command byte (0x92) from the host and a data byte from the LM8322. 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
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.

17.18 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 2*, *Section 9.3 CLOCK CONFIGURATION*.

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							

17.19 READ_CLOCK COMMAND

The READ_CLOCK command consists of a command byte (0x94) from the host and a data byte from the LM8322. This command reads bits 7:2 of the clock configuration, as described in *Table 2* , *Section 9.3 CLOCK CONFIGURATION*.

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

17.20 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	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.

17.21 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.

17.22 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	CH	

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

18.0 Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage (V_{CC})	2V
Voltage at Any Pin	-0.3V to V_{CC} +0.3V
Maximum Input Current Without Latchup	±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

19.0 DC Electrical Characteristics

(Temperature: -40°C ≤ T_A ≤ +85°C)

Data sheet specification limits are guaranteed 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 (Note 2)	Internal Clock, No loads on pins, V_{CC} = 1.9V, T_C = 0.5µs (Note 4)		1.9	3.0	mA
I_{HALT}	Standby Mode Current (Note 5)	**Typical**: V_{CC} = 1.9V, T_A = 25°C		<9	40	µA
V_{IL}	Logical 0 Input Voltage (Note 5)		0.3 x V_{CC}			V
V_{IH}	Logical 1 Input Voltage (Note 5)				0.7 x V_{CC}	V
	Hi-Z Input Leakage (TRI-STATE Output)	V_{CC} = 1.8V	-2		2	µA
	Port Input Hysteresis (Notes 5, 6)		100	400		mA
	Weak Pull-Up/Pull-Down Current	1.6V<V_{CC}< 2.0V			150	µA
	Output Current Source (Push-Pull Mode)	V_{CC} = 1.62V, V_{OH} = 0.7 x V_{CC}			-16	mA
	Output CurrentSink (Push-Pull Mode)	V_{CC} = 1.62V, V_{OL} = 0.3 x V_{CC}	16			mA
	Allowable Sink and Source Current per Pin (Note 7)				16	mA
C_{PAD}	Input Capacitance (Note 7)				5	pF

Note 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 guaranteed. For guaranteed specifications and test conditions, see the Electrical Characteristics tables.

Note 2: Supply current is measured with inputs connected to V_{CC} and outputs driven low but not connected to a load.

Note 3: T_C = instruction cycle time (min. 0.7 µs).

Note 4: 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.

Note 5: Applied to all digital pins (including $\overline{\text{RESET}}$) except for SLOWCLK when configured for an external clock..

Note 6: Guaranteed by design, not tested.

Note 7: The sum of all I/O sink/source current must not e4xceed the maximum total current into V_{CC} and out of GND as specified in the absolute maximem ratings.

20.0 AC Electrical Characteristics

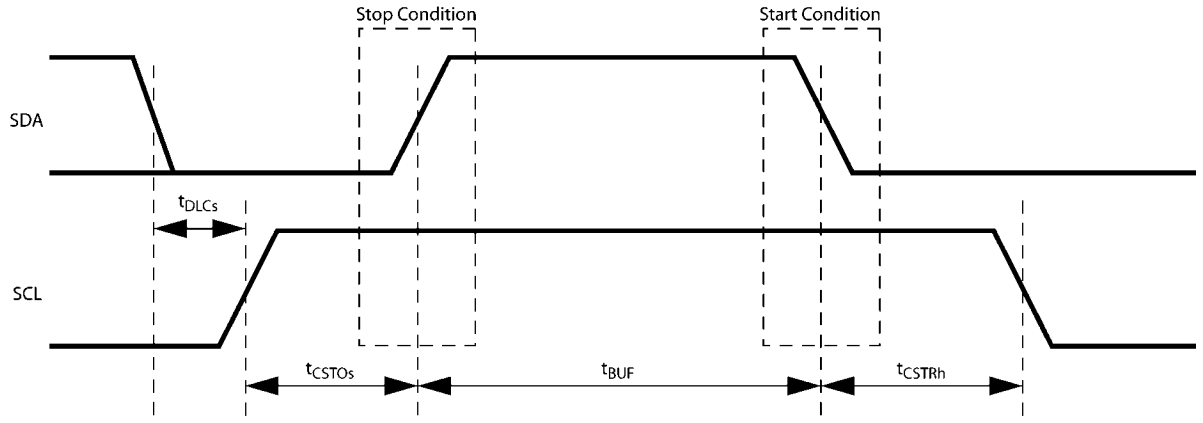
(Temperature: -40°C ≤ T_A ≤ +85°C)

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

Parameter	Conditions	Min	Typ	Max	Units
System Clock (mclk) (Note 8)	Internal RC 1.62V ≤ V_{CC} ≤ 1.98V		21		MHz
Processing and Command Execution Cycle (t_C) (Note 8)	1.62V ≤ V_{CC} ≤ 1.98V		0.5		µs
System Clock, Processing and Command Execution Cycle Variation(Note 8)				7	%
General-Purpose I/O (GPIO)					
Output Rise Time(Note 8)	C_{LOAD} = 50 pF			15	ns
Output Fall Time(Note 8)				15	ns
ACCESS.bus Input Signals(Note 9)					
Bus Free Time Between Stop and Start Condition (tBUFi) (Note 8)		$t_{SCLhigho}$			
SCL Setup Time (tCSTOsi) (Note 8)	Before Stop Condition k	8			mclk
SCL Hold Time (tCSTRhi) (Note 8)	After Start Condition	8			mclk
SCL Setup Time (tCSTRsi) (Note 8)	Before Start Condition	8			mclk
Data High Setup Time (tDHCsi) (Note 8)	Before SCL Rising Edge (RE)	2			mclk
Data Low Setup Time (tDLCsi) (Note 8)	Before SCL RE	2			mclk
SCL Low Time (tSCLlowi) (Note 8)	After SCL Falling Edge (FE)	12			mclk
SCL High Time (tSCLhighi) (Note 8)	After SCL RE	12			mclk
SDA Hold Time (tSDAhi) (Note 8)	After SCL FE	0			mclk
SDA Setup Time (tSDAsi) (Note 8)	Before SCL RE	2			mclk
ACCESS.bus Output Signals (Note 9)					
Bus Free Time Between Stop and Start Condition (tBUFo)(Note 9)		$t_{SCLhigho}$			
SCL Setup Time (tCSTOso) (Note 8)	Before Stop Condition	$t_{SCLhigho}$			
SCL Hold Time (tCSTRho) (Note 8)	After Start Condition	$t_{SCLhigho}$			
SCL Setup Time (tCSTRso) (Note 8)	Before Start Condition	$t_{SCLhigho}$			
Data High Setup Time (tDHCso) (Note 8)	Before SCL RE	$t_{SCLhigho}$			
Data Low Setup Time (tDLCso) (Note 8)	Before SCL RE	$t_{SCLhigho}$			
SCL Low Time (tSCLlowo) (Note 8)	After SCL FE	16			mclk
SCL High Time (tSCLhigho) (Note 8)	After SCL RE	16			mclk
SDA Hold Time (tSDAho) (Note 8)	After SCL FE	7			mclk
SDA Valid Time (tSDAso) (Note 8)	Before SCL RE	7			mclk

Note 8: Guaranteed by design, not tested.

Note 9: The ACCESS.bus interface implements and meets the timing necessary for interface to the I²C and SMBus protocol at logic levels. The bus drivers are designed with open-drain output for bidirectional operation. The will not meet the AC timing and current/voltage requirements of the full bus specification.

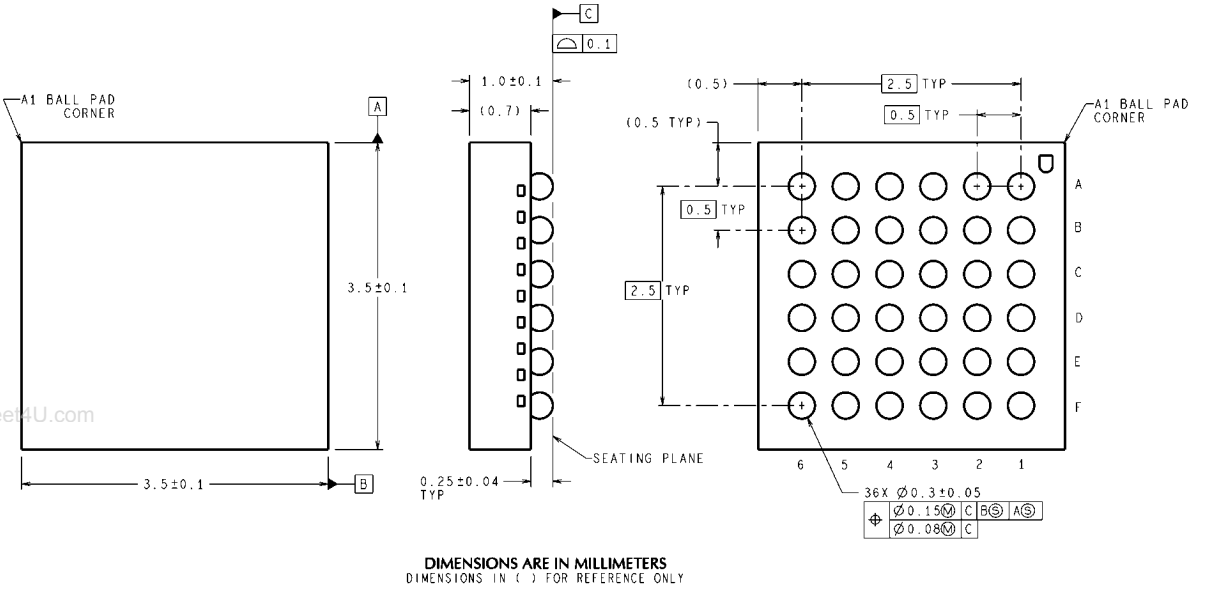


Note: In the timing tables the parameter name is added with an "o" for output signal timing and "i" for input signal timing.

30013619

FIGURE 19. ACB Start and Stop Condition Timing

21.0 Physical Dimensions inches (millimeters) unless otherwise noted



DIMENSIONS ARE IN MILLIMETERS
DIMENSIONS IN ( ) FOR REFERENCE ONLY

GRA36A (Rev A)

Micro Array Package
Order Number LM8322GGR8
NS Package Number GRA36A

Notes

National Semiconductor Americas Customer Support Center
Email: new.feedback@nsc.com
Tel: 1-800-272-9959

National Semiconductor Europe Customer Support Center
Fax: +49 (0) 180-530-85-86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +49 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Customer Support Center
Email: ap.support@nsc.com

National Semiconductor Japan Customer Support Center
Fax: 81-3-5639-7507
Email: jpn.feedback@nsc.com
Tel: 81-3-5639-7560