AN214818

Keyscan Matrix

Associated Part Family: CYW20730/CYW20733

This application note describes the keyscan matrix implemented in the Cypress CYW20730/CYW20733 Bluetooth transceiver chips. It is intended for software engineers who are writing code for devices that include the CYW20730/CYW20733, and who have experience working with keyboard scanners and with the CYW20730/CYW20733 chips.

Contents

1 Overview

The keyboard scanner is designed to autonomously sample keys and store them in buffer registers without the need for intervention by the host microcontroller

1.1 Cypress Part Numbering Scheme

Cypress is converting the acquired IoT part numbers from Broadcom to the Cypress part numbering scheme. Due to this conversion, there is no change in form, fit, or function as a result of offering the device with Cypress part number marking. The table provides Cypress ordering part number that matches an existing IoT part number.

Table 1. Mapping Table for Part Number between Broadcom and Cypress

Broadcom Part Number	Cypress Part Number
BCM20730	CYW20730
BCM20733	CYW20733

1.2 Features

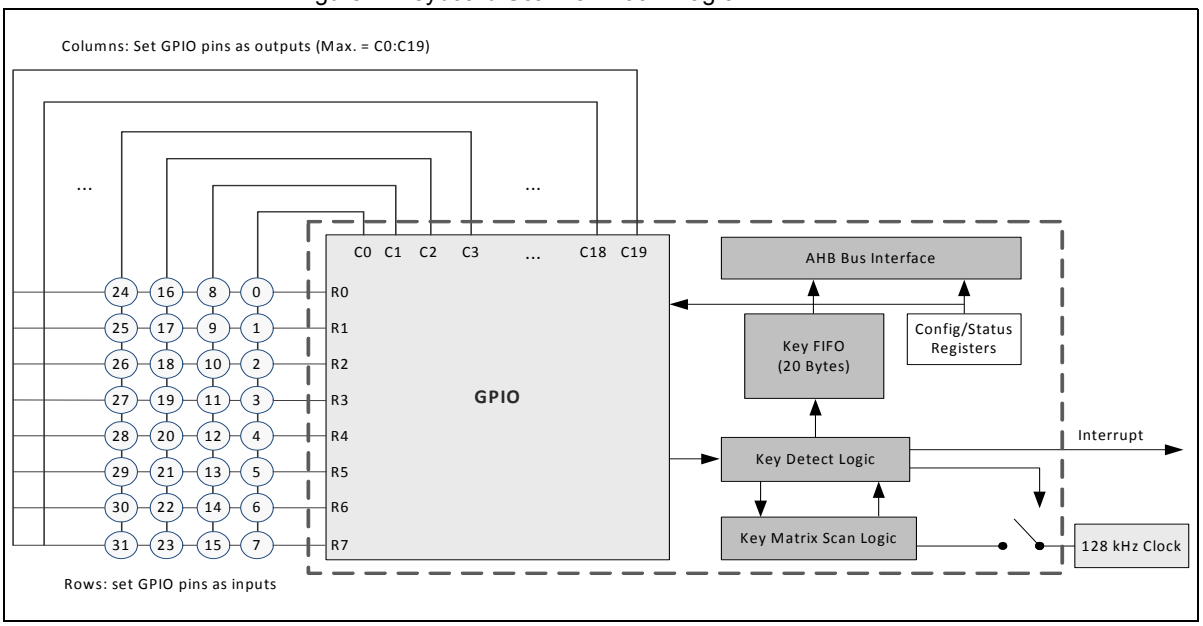
Keyboard scanner features are listed below:

- Ability to turn off its clock if no keys are pressed (see the kysclk_stayon bit in Table 4).
- Sequential scanning of up to 160 keys in an 8 × 20 matrix.
- Programmable number of columns (from 1 to 20).
- Programmable number of rows (from 1 to 8).
- 20-byte keycode buffer.
- A 128 kHz clock allows scanning of a full 160 key matrix in ~1.28 ms.
- Key actions are buffered until the host microcontroller has a chance to read them, or until an overflow occurs.
- Hardware debouncing and noise/glitch filtering.

1.3 Block Diagram

A block diagram of the keyboard scanner is shown in Figure 1.

Figure 1. Keyboard Scanner Block Diagram



The Scan matrix support up to 8 × 20 matrix, or a maximum of 160 keys. Any key press will be translated into an index corresponding to the appropriate column and row.

Key Matrix Scan logic is disabled prior to any key being pressed. Once a key press is detected by the Key Detection Logic, it will enable the gate for the clock to drive the Key Matrix Scan logic for GPIO scanning. GPIO scanning is done one column at a time by driving each column “low” and reading from each row of GPIO pins to find out which input is low.

After the Key Scan logic had scanned through the matrix for a specific number of debounce times (configured by firmware through a configuration register), the keycode representing the pressed key is pushed into the key FIFO for firmware to read, and an interrupt will be generated to notify the CPU.

Two types of debounce mechanism are built into the scan matrix block. Microdebounce logic provides a small debounce period to debounce the noise generated by the key-break action. The macrodebounce logic scans through the key matrix multiple times to determine whether a key has been pressed.

The number of rows and columns in the Scan Matrix is software configurable. Rows must start from port 0, pin 1, to max port 0, pin 7. Columns must start from port 0, pin 8, to max port 1, pin 11.

False detection of key-up or key-down event may occur due to the way key scan matrix detection is implemented. For example, if three keys are pressed in such a way that they would form a rectangle with a fourth, unpressed key, the fourth key may be falsely detected as having been pressed. This kind of false detection cannot be resolved. The Key Scan logic will notify the firmware of the false detection event through an interrupt with the ghost_detected status bit set to 1, and will put a special keycode (0xF5) into the keycode FIFO. The ghost detection feature can be turned off through the ghost_enable bit in the control register.

After the firmware has received the key detection interrupt, it must set the freeze_MIA bit to freeze the clock on the scan matrix, which prevents the buffer from being updated. The firmware must check whether the keyscan module freeze is in effect by polling the MIA_clock_freezed_indicator bit in the MIA_status register.

Once the MIA_clock_freezed_indicator bit has been set, firmware can start reading the keycode entries from the key FIFO register. The keyfifo_cnt register (see Table 6 indicates how may keys are stored in the key FIFO. The freeze_MIA bit and reported_clear_kys bits must be cleared when all keycode FIFO entries are read. The reported_clear_kys bits are used to clear the mia_status register (see Table 9).

2 IoT Resources

Cypress provides a wealth of data at http://www.cypress.com/internet-things-iot to help you to select the right IoT device for your design, and quickly and effectively integrate the device into your design. Cypress provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates. Customers can acquire technical documentation and software from the Cypress Support Community website (http://community.cypress.com/).

3 Micro Debounce and Extend Keyscan Cycle Control

3.1 Key-Down

Microdebounce is a preliminary filter for a key-down event. It is used to filter out short-duration events such as an Electrical Fast Transient (EFT) event. Once a key is detected as having been pressed down, the key is tested again in the next few cycles based on the value defined in the u_debonce[1:0] register (refer to the debounce_adr[9:8] bits).

The u_debounce register can be set as shown below:

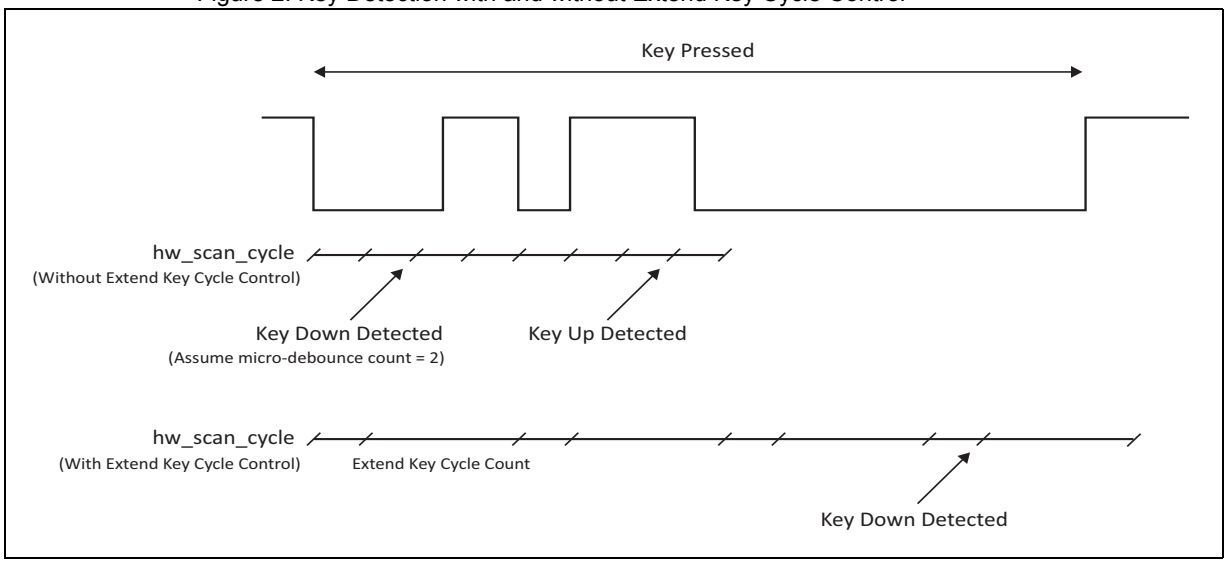
- u_debounce[1:0] = 0: Checks for key-down one time.
- u_debounce[1:0] = 1: Checks for key-down two times.
- u_debounce[1:0] = 2: Checks for key-down three times.
- u_debounce[1:0] = 3: Checks for key-down four times (default).

A key event that satisfies the microdebounce down criteria is stored in the debounce buffer for verification by the macrodebounce. Each entry in the debounce buffer consists of an 8-bit key index and a bit to indicate whether key is up or down.

In some cases, depending on the debounce characteristics of the key switch, a single key-down event can be mistaken as multiple key press events. The Extend Key Cycle Control is used to space the hardware scan activity so that this false scenario is less likely to occur.

Figure 2 shows a key detection scenario with and without the Extend Key Cycle Control.

Figure 2. Key Detection with and without Extend Key Cycle Control



3.2 Key-Up

Once a key is detected as being pressed down, one bit of the debounce buffer is set to indicate that the key in the buffer is ready for the key-up test. This bit switches the debounce counter to a down-counting mode. The u_debounce value is also used for the detection of a key-up event. The u_debounce counter counts down once, whenever a key is up for [extend key cycle] cycles. For example, if u_debounce[1:0]= 2, the countdown for the u_debounce counter will last for [Extend Key Cycle × 2]. The u_debounce countdown will decrease to [extend key cycle × 1] when key-up is detected for consecutive [Extend Key Cycle] cycles. When the debounce counter reaches 0, the key index will be transferred to the key buffer to be processed by the macro debounce.

4 Macro Debounce and Key Index

4.1 Key-down

After being qualified through the microdebounce, a given key is tested for the macrodebounce criteria defined by the key-down macrodebounce register, md_debounce[3:0]. If a key is detected for the number of consecutive hardware scan cycle as defined in the md_debounce counter, it is moved to the key event buffer to give firmware access to it.

When a key does not pass the macrodebounce criteria in a side-by-side hardware scan cycle, the macrodebounce counter is reset to zero. For example, if the md_debounce[3:0] = 3 and a key is detected as down/down/up/down/down/down, then the md_debounce counter contains the following values: 1, 2, 0, 1, 2, 3. At count 3, the key is copied to the key event buffer. Each entry of the key event buffer is defined in Table 2.

Table 2. Key Event Buffer

Bit	Description
31:31	Key-up/down indicator: 0 = Key-down 1 = Key-up
30:30	Scan cycle toggle bit
7:0	Key index

4.2 Key-Up

The key has to be up for the mu_debounce consecutive hardware scan cycle to be considered released. The released key is copied into the key event buffer with bit 31 set to one to indicate a key-up event.

4.3 Key Index

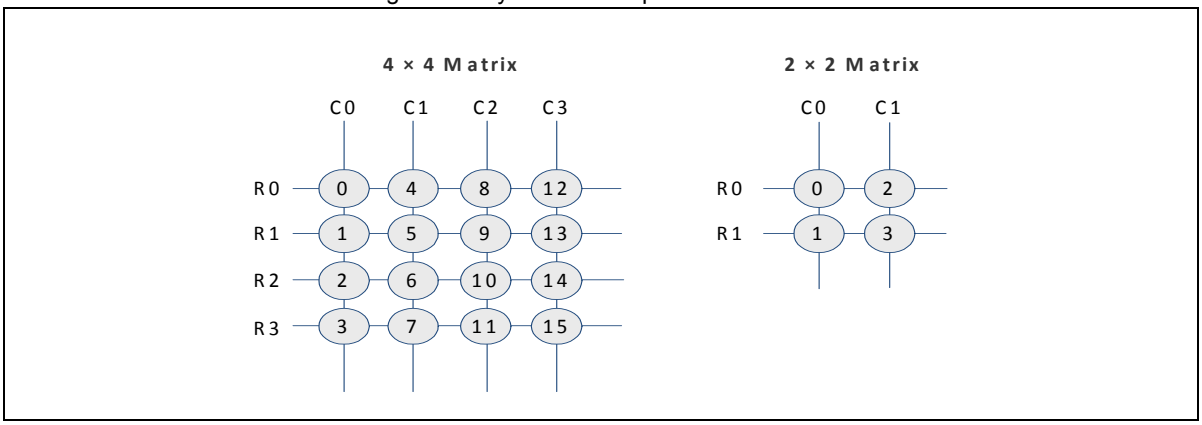
The key code (32-bit DWORD) is defined in Figure 3.

Table 3. Key Index

Bit	Description
31:31	Up or down indication for each key entry: 1 = Key-up event 0 = Key-down event
30:30	A 1 or 0 indicates the hardware scan cycle. Any key detected in the same hardware scan cycle will have same Scan Cycle value. This bit toggles between 0 and 1 whenever a key in a scan cycle is detected; in other words, this value does not change when no key-up or key-down event has been detected in the current scan cycle.
7:0	Indicates which key has been pressed or released, depending on the position of the column and row. The Key Index starts from 0 (column 0, row 0) and counts down the row (see Figure 3).

Key index examples are shown in Figure 3.

Figure 3. Key Index Examples



5 Keyscan Registers

5.1 keyscan_ctl_adr

Reset Value: 0x0007_9FCD

Firmware Access: R/W

Hardware Access: R

Table 4. keyscan_ctl_adr Register — 0x390000

Bit(s)	Name	Description
0:0	ks_en	Enables the key scan module for keyboard function (specifically, the 128 kHz clock to the keyscan module is enabled). In addition, if ks_en = 1, the 128 kHz LPO can be powered down after 8 ms of idling if the lpo_pd_en bit and the hid_off bit are both set to 1.
1:1	Reserved	–
2:2	ghost_en	Enables ghost key event detection.
3:3	ks_int_en	Enables the keyscan block to wake the baseband PMU module if a key-up or key-down event is detected. Waking the baseband PMU is further controlled by bit 9 of the cr_wake_int_en_2_adr register. For an HID-related wake-up, the source mapping for the cr_wake_int_en_2_adr register is: cr_wake_int_en_2_adr[10:8] ≥ {mif_int_raw, kys_int_raw, lhl_gpio_asynch_wakeup} The above keyscan raw interrupt signal is resynchronized to a free-running 24 MHz clock and is then fed to the CM3 interrupt. The HID-related interrupt sources and their line numbers for CM3 are: {IR_tx_FIFO_int, mif_int, kys_int, LHL_gpio_int} == {CM3_int #34, CM3_int #33, CM3_int #32, CM3_int #23}.
4:4	kys_rst_en	Enables the reset of the debounce counter with the kys_reported_clr bit. The reset clears the following registers: ■ Read/write counters for the event FIFO ■ Debounce buffers • Debounce counters • Debounce up/down registers ■ Cycle tracking registers ■ Keyscan read counter For reset to take effect, the following two bits have to be set during a clock unfreeze operation: ■ kys_rst_en ■ reported_clear_kys **Note:** ks_en can be in either state.
5:5	reserved	–
7:6	rc_ext[1:0]	Programmable idle duration between column scans. For example, rc_ext[1:0] = 1 provides 1 clock cycle of idle time in which no columns will be scanned. This is to alleviate the problem of a long RC delay on some keyboard designs. Valid values for rc_ext are 1, 2, and 3 (default).
10:8	rctc: row	Sets the number of rows for the key matrix. Should be programmed to a value of (the number of rows in the keyboard) – 1.
15:11	rctc: column	Sets the number of columns for the key matrix. Should be programmed to a value of (the number of column in the keyboard) – 1.
16:16	pull_high	Used to pull the columns high after each column scan to alleviate slow rise times due to a large amount of capacitance in the key matrix. This bit is turned on by default.
17:17	ksi_drv_high	For keyboard applications, this bit enables the key scan interface rows to be outputs. The rows are driven high to accelerate the pull-up resistor effect for avoiding false key detection due to otherwise slow-rising nodes.
18:18	kysclk_stayon	The keyscan clock will stay on when set; otherwise, the clock will be gated off by multiple interface agent (MIA) when no activity is detected.
23:18	Reserved	–

5.2 debounce_adr

Reset Value: 0x0000_0333

Firmware Access: R/W

Hardware Access: R

Table 5. debounce_adr Register — 0x390004

Bit(s)	Name	Description
3:0	md_debounce	Macro down debounce count
7:4	mu_debounce	Macro up debounce count
9:8	u_debounce	Sets the microdebounce count
16:10	ksc_extend_cnt	These bits are used to place an idle time between two consecutive hardware scan cycles in units of 4 multiple interface agent (MIA) clocks. Therefore, the time gap is from 4 (ksc_extend = 1) to 512 (ksc_extend = 127) MIA clocks.
17:17	ksc_extend_en	Enables Key Scan Cycle Extend, which puts idle time between successive hardware key scans. This signal is resynchronized to the 128 kHz MIA clock, so it can be changed by firmware at any time.

5.3 keyfifo_cnt_adr

Reset Value: 0x0000_0000

Firmware Access: R

Hardware Access: R

Table 6. keyfifo_cnt_adr Register — 0x390008

Bit(s)	Name	Description
5:0	keyfifo_cnt	This register indicates the number of events that are ready for firmware access in the keycode event FIFO. Firmware reads this register before accessing the keycode event FIFO register. For example, a 1 in this register indicates that there is one event in the keycode event FIFO, and firmware can issue a read to the keycode event FIFO once. The accumulated number of key events is latched in a buffer register at the end of each hardware scan frame. This number is then transferred to the keyfifo_cnt_adr register when the freeze_MIA bit is set to 1. The value in this register is cleared when the reported_clear_kys bit is set in the mia_ctl_adr register (see Table 7 on page 17). **Note:** An overflow condition (more than 20 keys detected in a given scan cycle) may occur before the entire frame is scanned. If the overflow bit is set and the keyfifo_cnt_adr register contains a value that is less than 20, firmware must read this register a second time. The remaining balance of the event count will be loaded into this register automatically when the firmware sets the reported_clear_kys and unfreeze bits if the hardware scan frame where overflow occurred has completed. In some cases a reset (kys_rst_en; see Table 3 on page 13) may be required to clear the overflow condition.

5.4 keyfifo_adr

Reset Value: 0x0000_0000

Firmware Access: R

Hardware Access: R

Table 7. keyfifo_adr Register — 0x39000C

Bit(s)	Name	Description
7:0	keyfifo	The detected key index. The event FIFO is 20 bytes deep. After power-up reset or a soft reset (as defined by the kys_rst_en bit), the event FIFO will contain 0xFF values.
29:8	Reserved	–
30:30	track_scan_cycle	The track_scan_cycle bit toggles between 1 and 0. For example, if the current cycle is set to 1, all scanned keys in the key FIFO will have this bit set to 1. In the next scan cycle, all scanned keys in the key FIFO will be set to 0, and so on. All keys detected in the same hardware scan cycle will have the same track_cycle bit value. The value of this bit does not change when no keys are detected in the current scan cycle.
31:31	key_up_down	Key-down (0) or key-up (1) for each key entry in the FIFO. This bit is associated with the keyfifo bits (bit[7:0] above).

5.5 mia_ctl_adr

Reset Value: 0x0000_0000

Firmware Access: R

Hardware Access: R

Table 8. mia_ctl_adr Register — 0x390014

Bit(s)	Name	Description
0:0	freeze_MIA	When set to 1, this bit will latch the accumulated key event count for firmware access through the keyfifo_cnt_adr register. After setting the freeze_MIA bit to 1, firmware must poll the clkrc_freezed bit (synchronized to the 24 MHz clock in hardware) in the mia_status_adr register (see Table 9). When the clkrc_freezed bit goes high, read access to the keyfifo_cnt_adr and keyfifo_adr registers is granted. After reading these registers, firmware must set the freeze_MIA bit low. The typical latency for the freeze state is 2.5 cycles of the 128 kHz clock (~ 19.5 µs). The clock in the keyscan module is not halted when the freeze bit is set.
1:1	reported_clear_kys	After reading the multiple interface agent (MIA) registers, firmware sets this bit to instruct the MIA keyscan module to clear the keycode status bits, ghost status bits, and other internal bits. It is important that firmware clears and sets the bit properly (readable/writable by the CPU). **Note:** This bit takes effect only when the unfreeze occurs. A logical AND between this bit and the freeze_MIA bit clears the registers. Cypress engineers recommend that the reported_clear_kys bit be cleared before the firmware enters into any freeze operation (readable/writable by uP, read-only by MIA).
2:2	report_clear_mif	After reading the MIA registers, firmware sets this bit to 1 to instruct the MIA mouse module to clear the quadrature count registers. It is important that firmware clears and sets the bit properly (readable/writable by uP). **Note:** This bit takes effect only when the unfreeze occurs. A logical AND between this bit and the freeze_MIA bit clears the registers. Cypress engineers recommend that the reported_clear_mif bit be cleared before the firmware enters into any freeze operation (readable/writable by the CPU, read-only by the MIA).

5.6 mia_status_adr

Reset Value: 0x0000_0000

Firmware Access: R

Hardware Access: W

Table 9. mia_status_adr Register — 0x390018

Bit(s)	Name	Description
0:0	MIA_clock_freezed_indicator	Firmware must poll the MIA_clock_freezed_indicator bit when trying to access the multiple interface agent (MIA) registers. When this bit is high, the freeze is successful (see bit 0 in Table 8), and read access will be granted; otherwise, firmware must wait and poll this bit until it is high. This bit is writable by the MIA and readable by the CPU.
7:1	keyscan_status	{ir_int_sync, kys_int_sync, mif_int_sync, mif_event_set, ghost, key FIFO overflow, keycode_set} **keycode_set**: Indicates that there is an event in the key event FIFO. **key FIFO overflow**: Indicates an overflow condition in the 20-byte key event FIFO (more than 16 keys were pressed during current scan interval). **Ghost**: Indicates that ghost key events have been detected. 0xF5 is inserted into the event FIFO (see Table 4 for details on the ghost enable bit). **mif_int_sync**: Status indicating that the quadrature interrupt is set. Refer to the qctl_adr register for the qd_int_en interrupt enable bit. **kys_int_sync**: Status indicating that the keyscan interrupt is set. Refer to Table 4 for details on the keyscan interrupt enable bit (ks_int_en). **ir_int_sync**: Status indicating that the predefined watermark for the IR transmit FIFO has been reached. **Note:** The keyscan_status bits can be read at any time. It is not necessary to freeze the MIA clock first.

5.7 lhl_ctl_adr

Reset Value: 0xE000_0000

Firmware Access: R/W

Hardware Access: R

Table 10. lhl_ctl_adr Register — 0x390130

Bit(s)	Name	Description
	HIDOFF	Human Interface Device Off. When this bit is set to 1, the main LDO will be powered down.
	ks_en	Keyscan enable for the LHL domain. Since the Multiple Interface Agent (MIA) Keyscan module is located in the baseband core, this bit must be set to enable waking up from the Human Interface device Off state. **Note:** This definition is the same as that of the ks_en register defined in MIA keyscan_ctl_adr[0] in order to support Human Interface device Off (HIDOFF) mode.
	rtl_cnt_en	Real-time clock counter enable and 32 kHz crystal oscillator power-down. 1: Enable the real-time clock counter and power up the 32 kHz crystal oscillator. 0: The real-time clock counter is disabled and the 32 kHz crystal oscillator is powered down (default).
	Reserved	–
	qd_z_en	Quadrature enable signal for the LHL domain. Since the MIA quadrature module is located in the baseband core, this bit must be set to enable waking from the Human Interface device Off state. **Note:** This definition is the same as that of the qd_en register defined in MIA qctl_adr[0] in order to support HIDOFF mode.
	qd_y_en	Quadrature enable signal for the LHL domain. Since the MIA quadrature module is located in the baseband core, this bit must be set to enable waking from the Human Interface device Off state. **Note:** This definition is the same as that of the qd_en register defined in MIA qctl_adr[0] in order to support HIDOFF mode.

Bit(s)	Name	Description
	qd_x_en	Quadrature enable signal for the LHL domain. Since the MIA quadrature module is located in the baseband core, this bit must be set to enable waking from the Human Interface device Off state. **Note:** This definition is the same as that of the qd_en register defined in MIA qctl_adr[0] in order to support HIDOFF mode.
	Reserved	–
	ksi_used	Defines the number of rows used for the keyscan. **Note:** This definition is the same as that of the rctc register defined in MIA keyscan_ctl_adr[10:8] in order to support HIDOFF mode.

6 Additional Information

6.1 Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use. For a comprehensive list of acronyms and other terms used in Cypress documents, go to: http://www.cypress.com/glossary.

6.2 References

The references in this section may be used in conjunction with this document.

Note: Cypress provides customer access to technical documentation and software through its Cypress Developer Community and Downloads & Support site (see IoT Resources).

Document (or Item) Name	Broadcom Document Number	Cypress Document Number	Source
Single-Chip Bluetooth Transceiver for Wireless Input Devices	20730-DS1xx-R	002-14824, 002-15291	Cypress Developer Community
Single-Chip Bluetooth Transceiver for Wireless Input Devices	20733-DSxx-R	002-14859, 002-14860	Cypress Developer Community

Document History Page

Document Title: AN214818 - Keyscan Matrix Document Number: 002-14818				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
**	–	–	02/29/2012	20730_20733-AN100-R Initial release
*A	5471476	UTSV	10/13/2016	Updated to Cypress template. Added Cypress Part Numbering Scheme.
*B	5881510	AESATMP8	09/13/2017	Updated logo and Copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Video | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709