Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

PCA9629

Fm+ I²C-bus stepper motor controller

Rev. 1 — 29 February 2012 Product data sheet

1. General description

The PCA9629 is an I²C-bus controlled low-power CMOS device that provides all the logic and control required to drive a four phase stepper motor. PCA9629 is intended to be used with external high current drivers to drive the motor coils. The PCA9629 supports three stepper motor drive formats: one-phase (wave drive), two-phase, and half-step. In addition, when used as inputs, four General Purpose Input/Outputs (GPIOs) allow sensing of logic level output from optical interrupter modules and generate active LOW interrupt signal on the $\overline{\text{INT}}$ pin of PCA9629. This is a useful feature in sensing home position of motor shaft or reference for step pulses. Upon interrupt, the PCA9629 can be programmed to automatically stop the motor or reverse the direction of rotation of motor.

Output wave train is programmable using control registers. The control registers are programmed via the I²C-bus. Features built into the PCA9629 provide highly flexible control of stepper motor, off-load bus master/micro and significantly reduce I²C-bus traffic. These include control of step size, number of steps per single command, number of full rotations and direction of rotation. A ramp-up on start and/or ramp-down on stop is also provided.

The PCA9629 is available in a 16-pin TSSOP package and is specified over the −40 °C to +85 °C industrial temperature range.

2. Features and benefits

- Generate motor coil drive phase sequence signals with four outputs for use with external high current drivers to off-load CPU
- Four balanced push-pull type outputs capable of sinking 25 mA or sourcing 25 mA for glueless connection to external high current drivers needed to drive motor coils
  - Up to 1000 pF loads with 100 ns rise and fall times
- Built-in oscillator requires no external components
- Stepper motor drive control logic
- One-phase (wave drive), two-phase, and half-step drive format logic level outputs
- Programmable step rate: 344.8 kpps to 0.3 pps with ±5 % accuracy
- Programmable ramp-up on start and ramp-down to stop
- Programmable steps and rotation control
- Sensor enabled drive control: linked to interrupt from I/O pins
- Direction control of motor shaft
- Selectable active hold, power off or released states for motor shaft

- Four general purpose I/Os:
  - Configured to sense logic level outputs from optical interrupter photo transistor circuit
  - Configured as outputs to drive (source/sink) LEDs or other loads up to 25 mA
  - Programmable interrupt Mask Control for input pins
- 4.5 V to 5.5 V operation
- 1 MHz Fast-mode Plus (Fm+) I²C-bus serial interface with 30 mA high drive capability on SDA output for driving high capacitive buses
- Compliant with I²C-bus Standard-mode (100 kHz) and Fast-mode (400 kHz) speeds
- Active LOW open-drain interrupt output
- Active LOW reset ($\overline{\text{RESET}}$) input pin resets device to power-up default state: can be used to recover from bus stuck condition
- Programmable watchdog timer
- All Call address allows programming of more than one device at the same time with the same parameters
- 16 programmable slave addresses using two address pins
- −40 °C to +85 °C operation
- ESD protection exceeds 2000 V HBM per JESD22-A114 and 1000 V CDM per JESD22C101
- Latch-up testing is done to JEDEC Standard JESD78 which exceeds 100 mA
- Package offered: TSSOP16

3. Applications

- Amusement machines
- Gaming and slot machines
- Consumer home appliances or toys
- Industrial automation
- HVAC and building climate control systems
- Robotics

4. Ordering information

Table 1. Ordering information

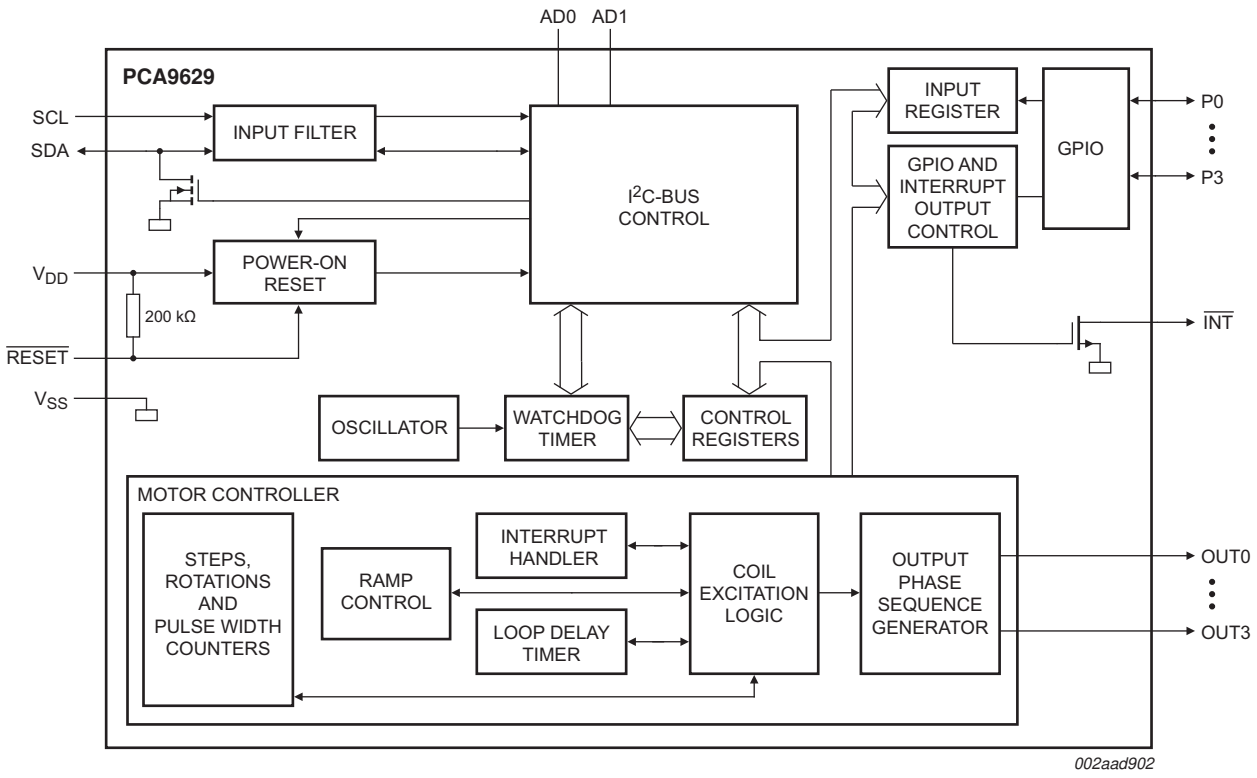
Type number	Package		
	Name	Description	Version
PCA9629PW	TSSOP16	plastic thin shrink small outline package; 16 leads; body width 4.4 mm	SOT403-1

4.1 Ordering options

Table 2. Ordering options

Type number	Topside mark	Temperature range
PCA9629PW	PCA9629	T_{amb} = −40 °C to +85 °C

5. Block diagram



Remark: All I/Os are set to inputs at power-up and reset.

Fig 1. PCA9629 block diagram

6. Pinning information

6.1 Pinning

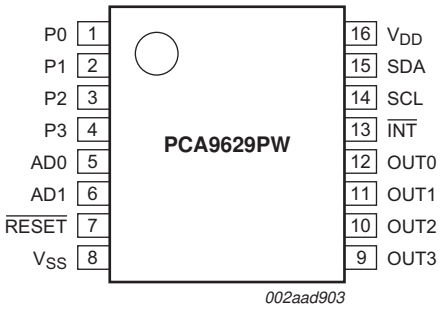


Fig 2. Pin configuration for TSSOP16

6.2 Pin description

Table 3. Pin description

Symbol	Pin	Type	Description
P0	1	I/O	input/output 0 (output is 25 mA push-pull)
P1	2	I/O	input/output 1 (output is 25 mA push-pull)
P2	3	I/O	input/output 2 (output is 25 mA push-pull)
P3	4	I/O	input/output 3 (output is 25 mA push-pull)
AD0	5	I	address input 0
AD1	6	I	address input 1
$\overline{\text{RESET}}$	7	I	active LOW reset input with 1 μs filter
V_{SS}	8	ground	supply ground
OUT3	9	O	control 25 mA push-pull output 3
OUT2	10	O	control 25 mA push-pull output 2
OUT1	11	O	control 25 mA push-pull output 1
OUT0	12	O	control 25 mA push-pull output 0
$\overline{\text{INT}}$	13	O	active LOW interrupt output; open-drain
SCL	14	I	serial clock line
SDA	15	I/O	serial data line; open-drain capable of sinking 30 mA
V_{DD}	16	power supply	supply voltage

7. Functional description

Refer to Figure 1 “PCA9629 block diagram”.

7.1 Device address

Following a START condition, the bus master must send the target slave address followed by a read or write operation. The slave address of the PCA9629 is shown in Figure 3. Slave address pins AD1 and AD0 choose one of 16 slave addresses. To conserve power, no internal pull-up resistors are incorporated on AD1 and AD0. Table 4 shows all 16 slave addresses by connecting the AD0 and AD1 to V_{DD}, V_{SS}, SCL or SDA.

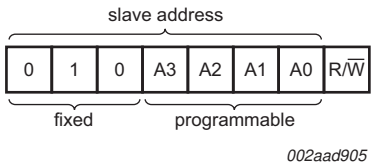


Fig 3. PCA9629 device address

The last bit of the first byte defines the reading from or writing to the PCA9629. When set to logic 1 a read is selected, while logic 0 selects a write operation.

Table 4. PCA9629 address map

AD1	AD0	Device family high-order address bits			Variable portion of address				Address
		A6	A5	A4	A3	A2	A1	A0	
V_{SS}	V_{SS}	0	1	0	0	0	0	0	40h
V_{SS}	V_{DD}	0	1	0	0	0	0	1	42h
V_{DD}	V_{SS}	0	1	0	0	0	1	0	44h
V_{DD}	V_{DD}	0	1	0	0	0	1	1	46h
V_{SS}	SCL	0	1	0	0	1	0	0	48h
V_{SS}	SDA	0	1	0	0	1	0	1	4Ah
V_{DD}	SCL	0	1	0	0	1	1	0	4Ch
V_{DD}	SDA	0	1	0	0	1	1	1	4Eh
SCL	V_{SS}	0	1	0	1	0	0	0	50h
SDA	V_{SS}	0	1	0	1	0	0	1	52h
SCL	V_{DD}	0	1	0	1	0	1	0	54h
SDA	V_{DD}	0	1	0	1	0	1	1	56h
SCL	SCL	0	1	0	1	1	0	0	58h
SCL	SDA	0	1	0	1	1	0	1	5Ah
SDA	SCL	0	1	0	1	1	1	0	5Ch
SDA	SDA	0	1	0	1	1	1	1	5Eh

7.2 Command register

Following the successful acknowledgement of the slave address and a write bit, the bus master sends a byte to the PCA9629. This byte is stored in the Command register.

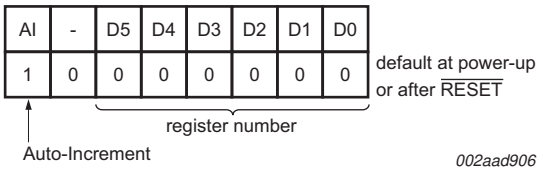


Fig 4. Command register

At power-up, the Command register defaults to 80h, with the AI bit set to '1' and the lowest seven bits set to '0'. The lowest six bits are used as a pointer to determine which register will be accessed. Only a command register code with the six least significant bits equal to the 39 allowable values as defined in Table 5 "Register summary" are acknowledged. Reserved or undefined command codes are not acknowledged.

The most significant bit of the Command register is for Auto-Increment. If the Auto-Increment flag is set, the six low-order bits of the Control register are automatically incremented after a read or write. This allows the user to program the registers sequentially. The contents of these bits will roll over to '00 0000' after the last register (address = 26h) is accessed. Only the six least significant bits are affected by the AI flag. Unused bits must be programmed with zeroes.

7.3 Register definitions

Table 5. Register summary

Register number	D5	D4	D3	D2	D1	D0	Name	Type	Function
00h	0	0	0	0	0	0	MODE	read/write	Mode register
01h	0	0	0	0	0	1	SUBADR1	read/write	I²C-bus subaddress 1
02h	0	0	0	0	1	0	SUBADR2	read/write	I²C-bus subaddress 2
03h	0	0	0	0	1	1	SUBADR3	read/write	I²C-bus subaddress 3
04h	0	0	0	1	0	0	ALLCALLADR	read/write	All Call I²C-bus address
05h	0	0	0	1	0	1	WDTOI	read/write	Watchdog time-out interval register
06h	0	0	0	1	1	0	WDCNTL	read/write	Watchdog control register
07h	0	0	0	1	1	1	IP	read only	Input Port register
08h	0	0	1	0	0	0	INTSTAT	read only	Interrupt status register
09h	0	0	1	0	0	1	OP	read/write	Output Port register
0Ah	0	0	1	0	1	0	IOC	read/write	I/O Configuration register
0Bh	0	0	1	0	1	1	MSK	read/write	Mask interrupt register
0Ch	0	0	1	1	0	0	CLRINT	write only	Clear interrupts
0Dh	0	0	1	1	0	1	INTMODE	read/write	Interrupt mode register
0Eh	0	0	1	1	1	0	INT_ACT_SETUP	read/write	Interrupt action setup control register
0Fh	0	0	1	1	1	1	INT_MTR_SETUP	read/write	Interrupt motor setup control register

Table 5. Register summary *…continued*

Register number	D5	D4	D3	D2	D1	D0	Name	Type	Function
10h	0	1	0	0	0	0	INT_ES_SETUP	read/write	Interrupt extra steps setup control register
11h	0	1	0	0	0	1	INT_AUTO_CLR	read/write	Interrupt auto clear control register
12h	0	1	0	0	1	0	SETMODE	read/write	Output state on STOP
13h	0	1	0	0	1	1	PHCNTL	read/write	Phase control register
14h	0	1	0	1	0	0	SROTNL	read/write	Steps per rotation low byte
15h	0	1	0	1	0	1	SROTNH	read/write	Steps per rotation high byte
16h	0	1	0	1	1	0	CWPWL	read/write	Step pulse width for CW rotation low byte
17h	0	1	0	1	1	1	CWPWH	read/write	Step pulse width for CW rotation high byte
18h	0	1	1	0	0	0	CCWPWL	read/write	Step pulse width for CCW rotation low byte
19h	0	1	1	0	0	1	CCWPWH	read/write	Step pulse width for CCW rotation high byte
1Ah	0	1	1	0	1	0	CWSCOUNTL	read/write	Number of steps CW low byte
1Bh	0	1	1	0	1	1	CWSCOUNTH	read/write	Number of steps CW high byte
1Ch	0	1	1	1	0	0	CCWSCOUNTL	read/write	Number of steps CCW low byte
1Dh	0	1	1	1	0	1	CCWSCOUNTH	read/write	Number of steps CCW high byte
1Eh	0	1	1	1	1	0	CWRCOUNTL	read/write	Number of rotations CW low byte
1Fh	0	1	1	1	1	1	CWRCOUNTH	read/write	Number of rotations CW high byte
20h	1	0	0	0	0	0	CCWRCOUNTL	read/write	Number of rotations CCW low byte
21h	1	0	0	0	0	1	CCWRCOUNTH	read/write	Number of rotations CCW high byte
22h	1	0	0	0	1	0	EXTRASTEPS0	read/write	Count value for extra steps or rotations for INTP0
23h	1	0	0	0	1	1	EXTRASTEPS1	read/write	Count value for extra steps or rotations for INTP1
24h	1	0	0	1	0	0	RMPCNTL	read/write	Ramp control register
25h	1	0	0	1	0	1	LOOPDLY	read/write	Loop delay time register
26h	1	0	0	1	1	0	MCNTL	read/write	Control start/stop motor
27h to FFh	-	-	-	-	-	-	-	-	Reserved

7.3.1 MODE — Mode register

Table 6. MODE - Mode register (address 00h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
00h	MODE	7	-	0*	not used
		6	-	0*	not used
		5	R/W	1	Disable $\overline{\text{INT}}$ output pin
				0*	Enable $\overline{\text{INT}}$ output pin
		4	R/W	1	outputs change on I²C-bus ACK
				0*	outputs change on I²C-bus STOP command
		3	R/W	1	PCA9629 responds to I²C-bus subaddress 1
				0*	PCA9629 does not respond to I²C-bus subaddress 1
		2	R/W	1	PCA9629 responds to I²C-bus subaddress 2
				0*	PCA9629 does not respond to I²C-bus subaddress 2
		1	R/W	1	PCA9629 responds to I²C-bus subaddress 3
				0*	PCA9629 does not respond to I²C-bus subaddress 3
		0	R/W	1*	PCA9629 responds to All Call I²C-bus address
				0	PCA9629 does not respond to All Call I²C-bus address

7.3.1.1 Disable interrupt output pin (bit 5)

This feature is useful when the host/micro/master does not want the $\overline{\text{INT}}$ pin to toggle when interrupts occur. Within PCA9629, when interrupts are enabled and interrupt event occurs, the actions related to the interrupt event are still carried out. However, if bit 5 = 1, the $\overline{\text{INT}}$ pin does not show the activation of interrupt because the pin is disabled. If bit 5 = 0, the micro sees the actual status of the $\overline{\text{INT}}$ pin.

The only exception to this rule is when the watchdog timer is enabled in the ‘Interrupt and Reset’ mode (see Section 7.3.4.2). In this case, the interrupt line toggles when the watchdog timer times out (even though bit 5 of this register is a ‘1’). This is because in the ‘Interrupt and Reset mode’ the part gets reset (and hence bit 5 is cleared) when the timer times out.

7.3.1.2 Outputs change on STOP (bit 4)

This feature can be used to synchronize the starting of the motor across multiple PCA9629 devices on the bus at approximately the same time (within few microseconds of one another). The host controller can program all the PCA9629s on the bus and then issue the I²C-bus STOP command. Upon receiving the STOP command, all the PCA9629 devices on the bus start generating pulse sequences required to turn the motor. This feature is applicable only to the motor coil outputs of the device namely, OUT0 to OUT3. It is **not** applicable to the general purpose I/Os (P0 to P3).

7.3.2 SUBADR1 to SUBADR3 — I²C-bus subaddress 1 to 3

Table 7. SUBADR1 to SUBADR3 - I²C-bus subaddress registers 1 to 3 (addresses 01h, 02h 03h) bit description

*Legend: * default value.*

Address	Register	Bit	Symbol	Access	Value	Description
01h	SUBADR1	7:1	A1[7:1]	R/W	1110 001*	I²C-bus subaddress 1
		0	A1[0]	R only	0*	reserved
02h	SUBADR2	7:1	A2[7:1]	R/W	1110 010*	I²C-bus subaddress 2
		0	A2[0]	R only	0*	reserved
03h	SUBADR3	7:1	A3[7:1]	R/W	1110 100*	I²C-bus subaddress 3
		0	A3[0]	R only	0*	reserved

Subaddresses are programmable through the I²C-bus. Default power-up values are E2h, E4h, E8h, and the device(s) will not acknowledge these addresses right after power-up (the corresponding bits [3:1] in MODE register is equal to 0).

Once subaddresses have been programmed to their right values, bits [3:1] (MODE register) must be set to logic 1 in order to have the device acknowledging these addresses. Only the seven MSBs representing the I²C-bus subaddress are valid. The LSB in SUBADRx register is a read-only bit (0). When subaddress control bits [3:1] in MODE register is set to logic 1, the corresponding I²C-bus subaddress can be used during either an I²C-bus read or write sequence.

7.3.3 ALLCALLADR — All Call I²C-bus address

Table 8. ALLCALLADR - All Call I²C-bus address register (address 04h) bit description

*Legend: * default value.*

Address	Register	Bit	Symbol	Access	Value	Description
04h	ALLCALLADR	7:1	AC[7:1]	R/W	1110 000*	ALLCALL I²C-bus address register
		0	AC[0]	R only	0*	reserved

The All Call I²C-bus address allows all the PCA9629s on the bus to be programmed at the same time (bit 0 in register MODE must be equal to 1 (power-up default state)). This address is programmable through the I²C-bus and can be used during either an I²C-bus read or write sequence. Only the seven MSBs representing the All Call I²C-bus address are valid. The LSB in ALLCALLADR register is a read-only bit (0). If bit 0 in MODE register = 0, the device does not acknowledge the address programmed in register ALLCALLADR.

7.3.4 Watchdog timer

The purpose of the watchdog timer is to recover the PCA9629 if the system it is used in enters an erroneous state. When the timer times out, the watchdog generates an interrupt to the host controller and, if programmed for reset, resets PCA9629 if the user program fails to 'feed' the watchdog. To feed the watchdog, the user simply addresses the PCA9629 ([START + slave address + START] or [START + slave address + STOP]) within the watchdog time-out interval. Only this sequence resets the watchdog.

Watchdog timer features:

- Can be programmed to reset the PCA9629 to POR state if it is not periodically addressed
- Enabled by software, but requires a hardware reset or a watchdog reset to be disabled
- Flag to indicate watchdog reset
- Programmable 8-bit timer with internal prescaler
- Selectable time period from one second to 255 seconds

The watchdog timer should be used in the following manner:

- Set the time-out interval value in WDTOI register
- Set the mode of operation (interrupt only or interrupt and reset) and enable the watchdog using the WDCNTL register
- Watchdog should be fed by periodically addressing PCA9629 before the watchdog timer underflows to prevent reset/interrupt
- Watchdog control register, WDCNTL, can be read at any time to determine the status of the watchdog operation

7.3.4.1 WDTOI — WatchDog Time-Out Interval register

The watchdog time-out interval should be programmed in this register. The default value is FFh, which indicates a 255 second time-out interval. The smallest value for the time-out interval is 01h, which indicates a one-second time-out interval. Watchdog operation cannot be enabled with a zero second time-out interval. If user writes a zero value to this register, the timer does not start.

Table 9. WDTOI - Watchdog time-out interval register (address 05h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
05h	WDTOI	7:0	R/W	FFh*	Watchdog time-out interval

7.3.4.2 WDCNTL — WatchDog Control register

Table 10. WDMOD - Watchdog control register (address 06h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
06h	WDCNTL	7:5	read only	000*	Reserved.
		4	write only	1	Clear WDINT flag.
				0*	Read value.
		3	read only	1	WDINT: watchdog interrupt flag set.[1]
				0*	WDINT: watchdog interrupt flag not set.
		2	read only	1	WDRST: watchdog reset flag.[2]
				0*	WDRST: watchdog reset flag not set.
		1	R/W	1	WDMOD: watchdog interrupt and reset mode (set only).
				0*	WDMOD: watchdog interrupt only mode.
		0	R/W	1	WDEN: watchdog enabled (set only).
				0*	WDEN: watchdog disabled.

[1] Use bit 4 to clear this bit.

[2] Reading WDCNTL register clears this bit.

This register controls the operation of the watchdog timer. Watchdog timer can be enabled by setting the WDEN bit of this register. WDEN is a set-only bit. Once set (enabled), this bit cannot be cleared by software. It can be cleared only with a hardware reset or watchdog reset.

The WDMOD bit determines the mode of operation. This bit is a set-only bit. There are two modes of operation:

- Interrupt only mode: This is the default mode of operation. In this mode, when the watchdog timer times out, the interrupt flag is set (WDINT) and an interrupt is generated to the host controller.
- Interrupt and reset mode: In this mode, when the watchdog timer times out, the reset flag is set (WDRST) and an interrupt is generated to host controller and resets the chip to POR state.

WDINT flag: This flag can be cleared by writing a ‘1’ to bit 4 of this register.

WDRST flag: This flag indicates that a watchdog reset has occurred. This flag does not get cleared by the watchdog reset. After a watchdog reset event, the host controller can read this bit to determine if a reset had occurred. The WDRST flag gets cleared after it is read or after an external reset is applied.

Before enabling the watchdog timer, the watchdog flags (interrupt flag and reset flag) **must** be cleared (if they are set). The interrupt flag is cleared by using bit 4 of the WDCNTL register and the reset flag is cleared just by reading the WCNTL register.

7.3.5 GPIOs and interrupts

7.3.5.1 IP — Input Port register

This register is read-only. They reflect the incoming logic levels of the port pins P0 to P3, regardless of whether the pin is defined as an input or an output by the I/O configuration register. Writes to this register have no effect.

Table 11. IP - Input Port register (address 07h) bit description

*Legend: * default value 'X' is determined by the externally applied logic level.*

Address	Register	Bit	Access	Value	Description
07h	IP	7:4	read only	0h*	reserved
		3:0	read only	Xh*	reflects incoming logic levels of I/O P0 to P3

7.3.5.2 INTSTAT — Interrupt Status register

This register reflects the status of an interrupt. INTSTAT is a read-only register.

INTP0 to INTP3 interrupt caused by input port pins P0 to P3, respectively.

Table 12. INTSTAT - Interrupt status register (address 08h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
08h	INTSTAT	7:4	-	0*	reserved
		3:0	read only	1	INTP3 flag set
				0*	INTP3 flag clear
				1	INTP2 flag set
				0*	INTP2 flag clear
				1	INTP1 flag set
				0*	INTP1 flag clear
				1	INTP0 flag set
				0*	INTP0 flag clear

Upon power-up or activation of hardware reset by $\overline{\text{RESET}}$ pin, INTSTAT register bits [3:0] are cleared (= 0), thus clearing the interrupt flags. Change in logic level at GPIO pins P0 to P3 configured as inputs will cause generation of interrupt when not masked using MSK register. The corresponding flag bit in this register is set and latched until cleared.

7.3.5.3 OP — Output Port register

This register is an output-only port. It reflects the outgoing logic levels of the pins defined as outputs by IOC register. Bit values in this register have no effect on pins defined as inputs. In turn, reads from this register reflect the value that is in the flip-flop controlling the output selection, not the actual pin value. Only the lower four bits are used and P0 to P3 are affected by this register.

Table 13. OP - Output Port register (address 09h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
09h	OP	7:4	-	0000*	reserved
		3:0	R/W	0000*	reflects outgoing logic levels of I/O P0 to P3

7.3.5.4 IOC — I/O Configuration register

The lower four bits of this register configures the direction of the I/O pins P0 to P3. If a bit in [3:0] is set (written with logic 1), the corresponding port pin is enabled as an input with high-impedance output driver. If the bit is cleared (written with logic 0), the corresponding port pin is enabled as an output. At reset, the device's ports P0 to P3 are inputs.

Table 14. IOC - I/O configuration register (address 0Ah) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
0Ah	IOC	7:4	-	0*	reserved
		3	R/W	1*	P3 will be configured as input
				0	P3 will be configured as output
		2	R/W	1*	P2 will be configured as input
				0	P2 will be configured as output
		1	R/W	1*	P1 will be configured as input
				0	P1 will be configured as output
		0	R/W	1*	P0 will be configured as input
				0	P0 will be configured as output

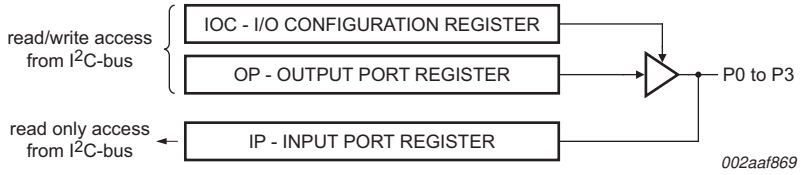


Fig 5. Simplified schematic for GPIO control

7.3.5.5 MSK — Mask interrupt register

Upon power-up, all the internal interrupt latches are reset and interrupt flags cleared and interrupt mask bits [3:0] are set to logic 1, thus disabling interrupts from input ports P0 to P3. Interrupts may be enabled by setting corresponding mask bits to logic 0.

Table 15. MSK - Interrupt mask register (address 0Bh) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
0Bh	MSK	7:4	-	0*	reserved
		3	R/W	1*	disables interrupt for I/O P3
				0	enables interrupt for I/O P3
		2	R/W	1*	disables interrupt for I/O P2
				0	enables interrupt for I/O P2
		1	R/W	1*	disables interrupt for I/O P1
				0	enables interrupt for I/O P1
		0	R/W	1*	disables interrupt for I/O P0
				0	enables interrupt for I/O P0

An additional control to enable or disable the $\overline{INT}$ pin is provided by MODE control register bit 5 (MODE[5]). Refer to Table 6.

7.3.5.6 CLRINT — Clear Interrupts register

Interrupt flags can be cleared by bits [3:0] when set to logic 1.

Table 16. CLRINT - Clear interrupts register (address 0Ch) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
0Ch	CLRINT	7:4	-	0*	reserved
		3	write only	1	clear INTP3 flag
				0*	read value
		2	write only	1	clear INTP2 flag
				0*	read value
		1	write only	1	clear INTP1 flag
				0*	read value
		0	write only	1	clear INTP0 flag
				0*	read value

7.3.5.7 INTMODE — Interrupt Mode register

When interrupt(s) are enabled, bits [3:0] determine whether rising edge or falling edge of signal at P0 to P3 causes the interrupt to be generated. Interrupts are latched and flag(s) are set in the corresponding bits of INTSTAT register. When interrupts are masked using MSK register, these bits have no effect.

Table 17. INTMODE - Interrupt mode register (address 0Dh) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
0Dh	INTMODE	7:4	-	0*	reserved
		3	R/W	1	interrupt occurs on falling edge for P3
				0*	interrupt occurs on rising edge for P3
		2	R/W	1	interrupt occurs on falling edge for P2
				0*	interrupt occurs on rising edge for P2
		1	R/W	1	interrupt occurs on falling edge for P1
				0*	interrupt occurs on rising edge for P1
		0	R/W	1	interrupt occurs on falling edge for P0
				0*	interrupt occurs on rising edge for P0

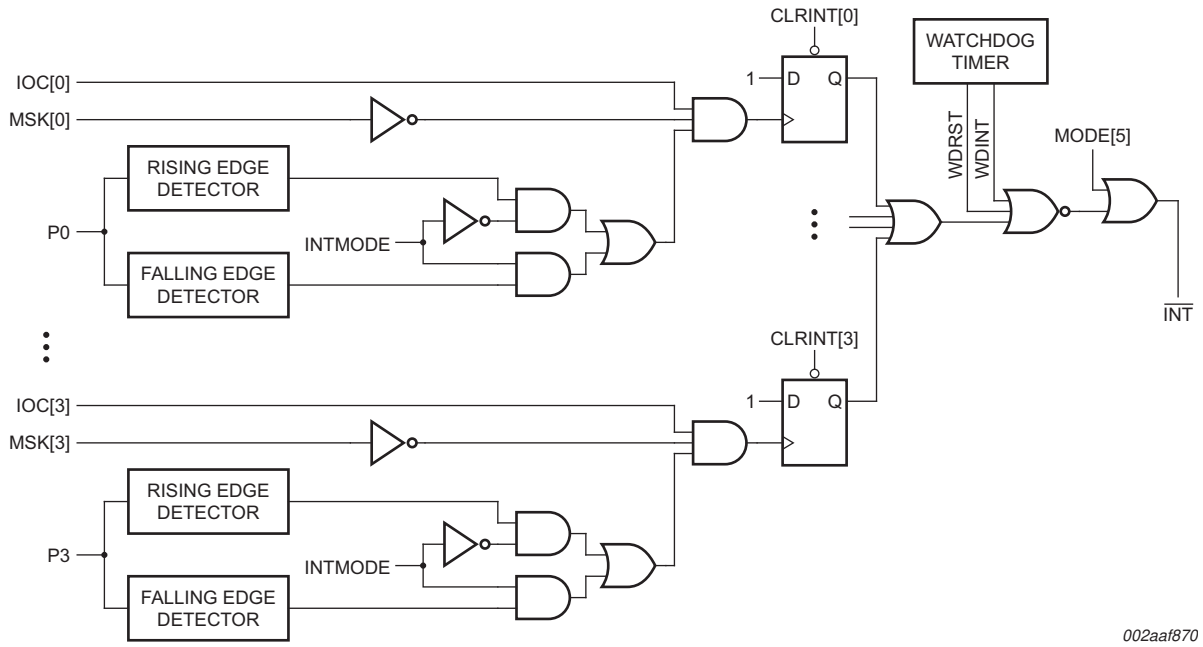


Fig 6. PCA9629 interrupt logic

7.3.6 Interrupt based motor control

Interrupt mechanisms from GPIOs 0 and 1 (INTP0 and INTP1) can be used to control the motor operation. Interrupts from GPIOs 2 and 3 are not used for motor control. They behave as normal GPIO interrupts. In the following sections, the word interrupt refers only to INTP0 and INTP1. The following actions can be performed upon the occurrence of an interrupt:

- Stop the motor
- Reverse the direction of motion
- Move extra steps/rotations and then, stop the motor or reverse its direction.

Only interrupts that occurred after the motor was started are acted upon. When an interrupt occurs, it is latched and the programmed action is performed. The microcontroller has to clear the interrupt before another occurrence of the same interrupt otherwise the second occurrence will not be acted upon. The following four registers, INT_ACT_SETUP, INT_MTR_SETUP, INT_ES_SETUP and INT_AUTO_CLR are used to program the various interrupt based control features of the motor. To enable the interrupt based control of the motor, bit 0 of the INT_ACT_SETUP register must be set.

7.3.6.1 INT_ACT_SETUP — Interrupt Action Setup control register

Table 18. INT_ACT_SETUP - Interrupt action setup control register (address 0Eh) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
0Eh	INT_ACT_SETUP	7:5	-	-	not used
		4	R/W	1	unit for EXTRASTEPS for both P0 and P1 counter is number of full rotations
				0*	unit for EXTRASTEPS for both P0 and P1 counter is number of steps
		3:1	-	-	not used
		0	R/W	1	enable interrupt based control of motor
				0*	disable interrupt based control of motor

If the interrupt based control is disabled, then values programmed in the following three registers (INT_MTR_SETUP, INT_ES_SETUP and INT_AUTO_CLR) have no effect on the motor operation.

Bit 4 of this register determines whether the values programmed in EXTRASTEPS0 and EXTRASTEPS1 registers represent the number of steps or number of rotations (see Section 7.3.16).

7.3.6.2 INT_MTR_SETUP — Interrupt Motor Setup control register

Table 19. INT_MTR_SETUP - Interrupt motor setup control register (address 0Fh) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
0Fh	INT_MTR_SETUP	7:2	R	-	reserved
		1:0	R/W	11	Reverse motor on INT caused by P0 or P1
				10	Stop motor on INT caused by P0 or P1
				01	Stop motor on INT caused by P1
				00*	Stop motor on INT caused by P0

When an interrupt occurs, if the motor is programmed to stop on that interrupt, the following sequence of events takes place in the given order:

1. If extra steps feature is enabled for that interrupt (see INT_ES_SETUP, Section 7.3.6.3) then extra steps (/rotations) will occur.
2. If ramp down is enabled (see RMPCNTL, Section 7.3.17), the motor starts ramping down.
3. Motor stops.

When an interrupt occurs, if the motor is programmed to reverse direction on that interrupt, the following sequence of events takes place:

1. If extra steps feature is enabled for that interrupt (see INT_ES_SETUP, Section 7.3.6.3) then extra steps (/rotations) occurs in the current direction of motion.
2. The motor stops for the amount of time specified in the LOOPDLY timer register.
3. Motor reverses its direction of rotation.

7.3.6.3 INT_ES_SETUP — Interrupt Extra Steps Setup control register

Table 20. INT_ES_SETUP - Interrupt extra steps setup control register (address 10h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
10h	INT_ES_SETUP	7:2	R	0000 00	reserved
		1:0	R/W	11	Enable EXTRASTEPS on both INTP0 and INTP1
				10	Enable EXTRASTEPS only on INTP1
				01	Enable EXTRASTEPS only on INTP0
				00*	Disable EXTRASTEPS for both INTP0 and INTP1

This register can be used to enable / disable the extra steps feature for each interrupt. Extra steps feature is used to make the motor rotate a specified amount of steps/rotations from the point of an interrupt occurrence.

7.3.6.4 INT_AUTO_CLR — Interrupt Auto Clear register

This register provides a mechanism to clear the two interrupts (INTP0 and INTP1) automatically without the occurrence of one interrupt clears the other without the microcontroller. The auto clear feature is disabled by default.

Table 21. INT_AUTO_CLR - Interrupt auto clear register (address 11h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
11h	INT_AUTO_CLR	7:2	-	0*	reserved
		1:0	R/W	11	INTP0 auto clears INTP1
				10	INTP1 auto clears INTP0
				01	INTP0 auto clears INTP1; INTP1 auto clears INTP0
				00*	INT auto clear for INTP0, INTP1 disabled

This feature is only available for interrupts that directly affect the operation of the motor as defined by the INT_MTR_SETUP register (see Section 7.3.6.2). For example, if INTP0 is used to stop the motor then it can be automatically cleared by its pair INTP1. However INTP1 should be manually cleared (through I²C-bus write to the CLRINT register). If both the interrupts are used to control the motor operation (INT_MTR_SETUP = 10 or 11), then all options of this register are valid. Any interrupt that is not automatically cleared by its pair should be manually cleared through I²C-bus write.

The auto clear mechanism can be used to create various motor movement patterns without being supervised by the microcontroller. For example, consider an application where the direction of motor rotation must be automatically reversed based on signals from two sensors placed apart from each other (sometimes referred to as ‘HOME’ positions) in a continuous manner without involving the microcontroller. The following example shows how to program the device for such an operation.

Example: This example assumes that two position sensors are located spaced apart and a drive mechanism is needed to move an object back and forth between these two sensors. Figure 7 shows this application use case. Driving the stepper motor causes movement of the object toward one of the sensors. Logic level output of one sensor is connected to input pin P0 and the other to P1. P0 and P1 are configured as **inputs**.

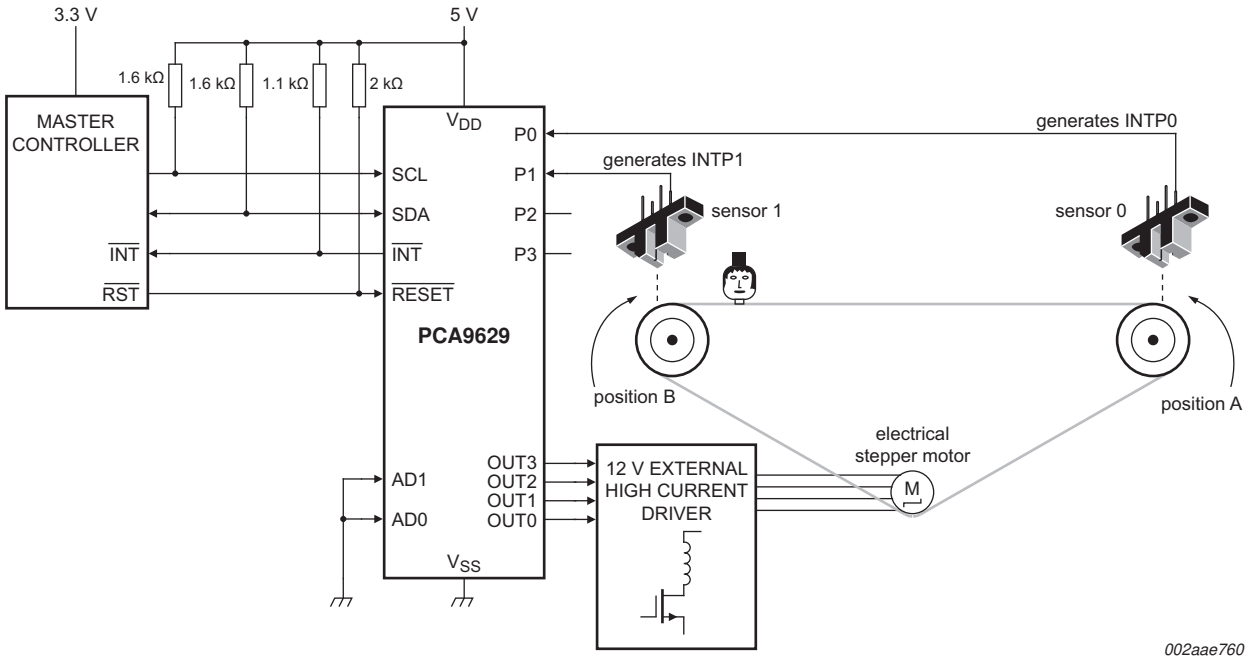


Fig 7. Example of controlling doll head movement between two home positions

At power-up, INTP0 to INTP3 flags INTSTAT[3:0] are clear (= 0).

Set INT_ACT_SETUP[0] = 1, enable interrupt based motor control.

Set INT_MTR_SETUP[1:0] = 11, Reverse motor on interrupt caused by P0 or P1.

Set INT_AUTO_CLR[1:0] = 01, INTP0 clears INTP1; INTP1 clears INTP0.

Start motor by writing MCNTL register and after some time, position sensor causes input logic at P0 to toggle.

When the input logic level at P0 changes, the interrupt caused by P0 is latched; INTP0 flag in INTSTAT is set (= 1).

Since INT_ACT_SETUP[0] = 1 and INT_MTR_SETUP[1:0] = 11 (reverse motor on interrupt caused by P0 or P1), the motor direction is reversed and the INTP1 flag is cleared (since INTP0 clears INTP1). This allows interrupt generation at the end of reverse movement by sensor at P1.

7.3.7 SETMODE — output state on STOP control register

This register determines the condition of motor output pins when STOPPED, one of logic 0 or Hold (last state).

Table 22. SETMODE - Output state on STOP control register (address 12h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
12h	SETMODE	7:2	R/W	-	reserved
		1	R/W	1	outputs = HOLD after CCW STOP
				0*	outputs = logic 0 after CCW STOP
		0	R/W	1	outputs = HOLD after CW STOP
				0*	outputs = logic 0 after CW STOP

7.3.8 PHCNTL — Phase Control register

This register is used to configure the phase of the output waveforms at the output ports OUT0 to OUT3 to drive the motor coils (with external high current drivers). One of the following three modes of drive method can be selected using these bits:

- One-phase drive (wave drive)
- Two-phase drive
- Half-step drive

Table 23. PHCNTL - Phase control register (address 13h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
13h	PHCNTL	7:2	-	0*	reserved
		1:0	R/W	11 or 10	half-step drive outputs
				01	two-phase drive outputs
				00*	one-phase drive outputs

The phase drive can be changed at any time by writing to PHCNTL[1:0] bits.

7.3.9 SROTNL, SROTNH — Steps per rotation registers

This register determines how many steps are needed to execute one full turn of motor shaft (360°). This register should have a non-zero value if the requested operation is rotations (see Section 7.3.19).

Remark: If the motor has built-in gear, the number of steps needed to complete one full turn at the output shaft depends on the gear ratio used.

Table 24. SROTNL, SROTNH - Steps per rotation control registers (address 14h, 15h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
14h	SROTNL	7:0	R/W	00h*	number of steps per one rotation, low byte
15h	SROTNH	7:0	R/W	00h*	number of steps per one rotation, high byte

7.3.10 CWPWL, CWPWH — Clockwise step pulse width register

This register determines the step pulse width used for the phase sequence output waveforms during ClockWise (CW) rotation.

Table 25. CWPWL, CWPWH - Clockwise step pulse width control register (address 16h, 17h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
16h	CWPWL	7:0	R/W	00h*	step pulse width, low byte
17h	CWPWH	7:0	R/W	00h*	step pulse width, high byte

This register sets the pulse width value between 3 μs and 3145 ms (±5 %).

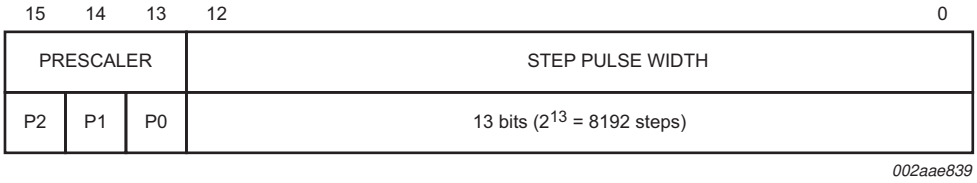


Fig 8. Step pulse width

The upper three bits of the register are the prescaler that determines the dynamic range for the step pulse width. Table 26 shows the range for each setting of the prescaler.

Table 26. Prescaler range settings

Prescaler [P2:P0]	Decimal value (D)	2^D	Range
000	0	1	3 μs to 24.576 ms
001	1	2	6 μs to 49.152 ms
010	2	4	12 μs to 98.304 ms
011	3	8	24 μs to 196.608 ms
100	4	16	48 μs to 393.216 ms
101	5	32	96 μs to 786.432 ms
110	6	64	192 μs to 1572.864 ms
111	7	128	384 μs to 3145.728 ms

Remark: The values given in Table 26 are based on nominal 1 MHz internal clock.

This method gives the user access to the entire range with the smallest pulse width (fastest speed) of 3 μs at the lower end, and the largest pulse width (slowest speed) of 3145 ms at the higher end.

The prescaler value defines the range of the ramp control. The ramp-up starts from its maximum pulse width and ramp-down ends at same maximum pulse width. The top speed of the ramp control is defined by both PRESCALER and STEP_PULSE_WIDTH values.

Final (top) speed = (minimum pulse width in the range defined by PRESCALER[15:13]) × (STEP_PULSE_WIDTH[12:0] + 1).

7.3.11 CCWPWL, CCWPWH — Counter-clockwise step pulse width register

This register determines the step pulse width used for the phase sequence output waveforms during Counter-ClockWise (CCW) rotation.

Table 27. CCWPWL, CCWPWH - Counter-clockwise step pulse width control register (address 18h, 19h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
18h	CCWPWL	7:0	R/W	00h*	step pulse width, low byte
19h	CCWPWH	7:0	R/W	00h*	step pulse width, high byte

The 16-bit value sets the pulse width between 3 μs and 3145 ms (±5 %).

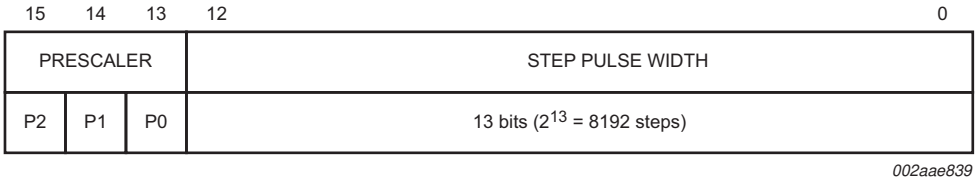


Fig 9. Step pulse width

The upper three bits of the register are the prescaler that determines the dynamic range for the step pulse width. Table 28 shows the range for each setting of the prescaler.

Table 28. Prescaler range settings

Prescaler [P2:P0]	Decimal value (D)	2^D	Range
000	0	1	3 μs to 24.576 ms
001	1	2	6 μs to 49.152 ms
010	2	4	12 μs to 98.304 ms
011	3	8	24 μs to 196.608 ms
100	4	16	48 μs to 393.216 ms
101	5	32	96 μs to 786.432 ms
110	6	64	192 μs to 1572.864 ms
111	7	128	384 μs to 3145.728 ms

Remark: The values given in Table 28 are based on nominal 1 MHz internal clock.

This method gives the user access to the entire range with the smallest pulse width (fastest speed) of 3 μs at the lower end, and the largest pulse width (slowest speed) of 3145 ms at the higher end.

The prescaler value defines the range of the ramp control. The ramp-up is started from its maximum pulse width and ramp-down ends at same maximum pulse width. The top speed of the ramp control is defined by both PRESCALER and STEP_PULSE_WIDTH values.

Final (top) speed = (minimum pulse width in the range defined by PRESCALER[15:13]) × (STEP_PULSE_WIDTH[12:0] + 1).

7.3.12 CWSCOUNTL, CWSCOUNTH — Number of clockwise steps register

This register determines the number of steps the motor should turn in clockwise direction.

Table 29. CWSCOUNTL, CWSCOUNTH - Number of clockwise steps count register (address 1Ah, 1Bh) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
1Ah	CWSCOUNTL	7:0	R/W	00h*	number of clockwise steps, low byte
1Bh	CWSCOUNTH	7:0	R/W	00h*	number of clockwise steps, high byte

7.3.13 CCWSCOUNTL, CCWSCOUNTH — Number of counter-clockwise steps register

This register determines the number of steps the motor should turn in counter-clockwise direction.

Table 30. CCWSCOUNTL, CCWSCOUNTH - Number of counter-clockwise steps count register (address 1Ch, 1Dh) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
1Ch	CCWSCOUNTL	7:0	R/W	00h*	number of counter-clockwise steps, low byte
1Dh	CCWSCOUNTH	7:0	R/W	00h*	number of counter-clockwise steps, high byte

7.3.14 CWRCOUNTL, CWRCOUNTH — Number of clockwise rotations register

This register determines the number of full rotations the motor should turn in clockwise direction.

Table 31. CWRCOUNTL, CWRCOUNTH - Number of clockwise rotations count register (address 1Eh, 1Fh) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
1Eh	CWRCOUNTL	7:0	R/W	00h*	number of clockwise rotations, low byte
1Fh	CWRCOUNTH	7:0	R/W	00h*	number of clockwise rotations, high byte

7.3.15 CCWRCOUNTL, CCWRCOUNTH — Number of counter-clockwise rotations register

This register determines the number of full rotations the motor should turn in counter-clockwise direction.

Table 32. CCWRCOUNTL, CCWRCOUNTH - Number of counter-clockwise rotations count register (address 20h, 21h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
20h	CCWRCOUNTL	7:0	R/W	00h*	number of counter-clockwise rotations, low byte
21h	CCWRCOUNTH	7:0	R/W	00h*	number of counter-clockwise rotations, high byte

7.3.16 EXTRASTEPS0, EXTRASTEPS1 — Extra steps count for INTP0, INTP1 control register

Table 33. EXTRASTEPS0, EXTRASTEPS1 - Extra steps count for INTP0, INTP1 register (address 22h, 23h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
22h	EXTRASTEPS0	7:0	R/W	00h*	count value for EXTRASTEPS (steps or rotations) for INTP0
23h	EXTRASTEPS1	7:0	R/W	00h*	count value for EXTRASTEPS (steps or rotations) for INTP1

This register has no effect if the interrupt based motor control is disabled or if the EXTRASTEPS feature for that interrupt is disabled.

When EXTRASTEPS feature is selected using INT_ES_SETUP register bits [1:0], the 8-bit value in this register is used to determine the number of steps or rotations to be overdriven. Direction of rotation of motor is maintained. If the count value in this register = 0, no EXTRASTEPS occurs. Whether the count indicates the number of extra steps or number of full rotations depends on the value of INT_ACT_SETUP control register bit 4.

If INT_ACT_SETUP[4] = 0 (default value), then EXTRASTEPSn value indicates number of extra steps that will occur after the corresponding interrupt.

If INT_ACT_SETUP[4] = 1, then EXTRASTEPSn value indicates number of full rotations that will occur after the corresponding interrupt.

7.3.17 RMPCNTL — Ramp control register

Table 34. RMPCNTL - Ramp control register (address 24h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
24h	RMPCNTL	7:6	R only	00*	reserved
		5	R/W	1	enable ramp-up during start
				0*	disable ramp-up during start
		4	R/W	1	enable ramp-down to stop
				0*	disable ramp-down to stop
		3:0	R/W	0000*	ramp step multiplication factor

The multiplication factor has a decimal range from 1 to 8192 as shown in Table 35.

Table 35. Multiplication factor value for ramp-up, ramp-down control

Register value [3:0]	Decimal value (D)	Ramp step multiplication factor (2^D)
0000	0	1
0001	1	2
0010	2	4
0011	3	8
0100	4	16
0101	5	32
0110	6	64
0111	7	128
1000	8	256
1001	9	512
1010	10	1024
1011	11	2048
1100	12	4096
1101	13	8192
1110, 1111	14, 15	reserved and do not use

RMPCNTL[5:4] enables/disables the speed ramp-up during starting of the motor and speed ramp-down during stopping of the motor.

The RMPCNTL[3:0] defines the acceleration/decelerating rate of the ramp control. If the value is small, the PWM width decrement (accelerating)/increment (decelerating) is slower.

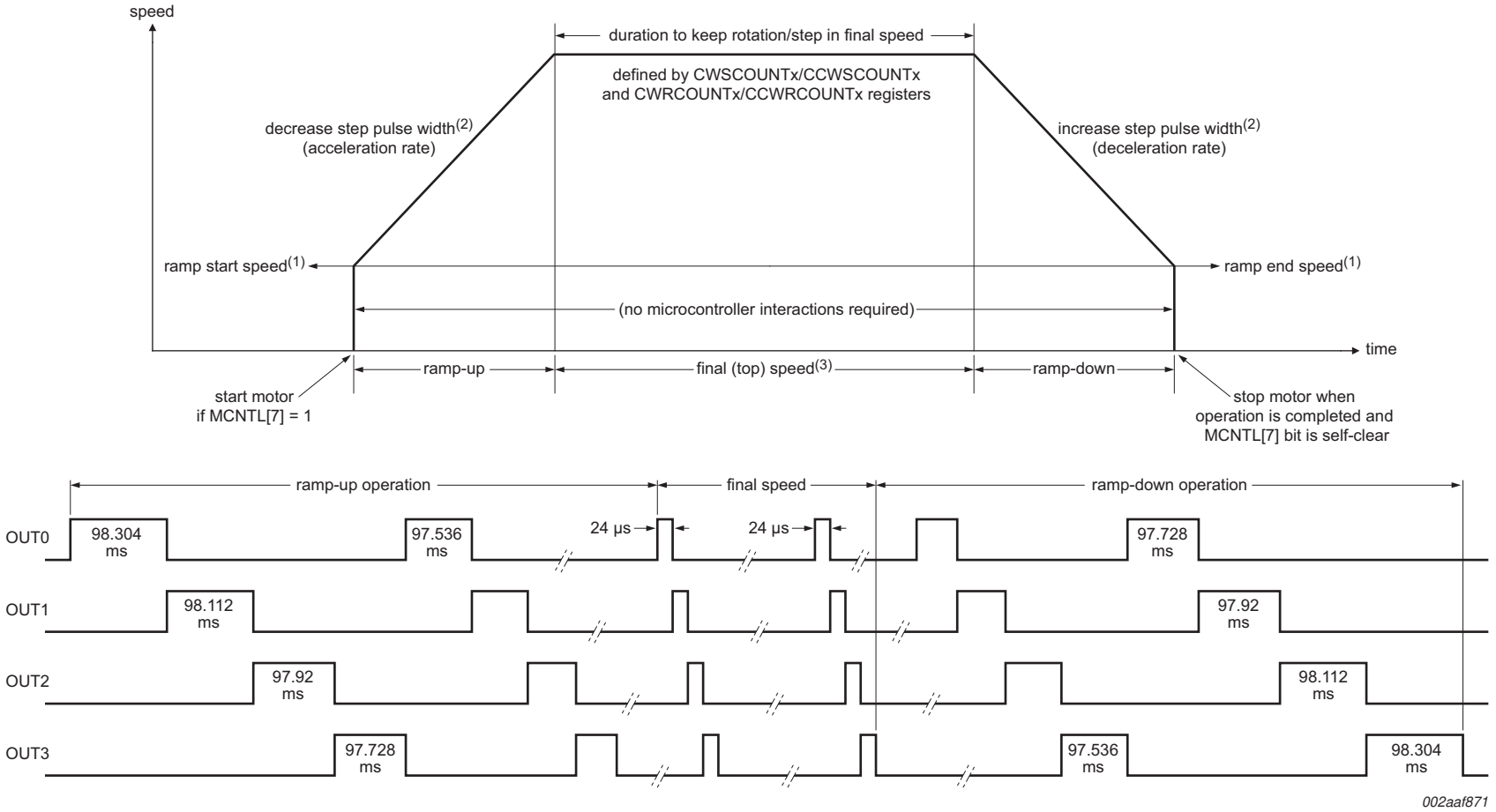
The pulse width decrement and increment step is 'smallest_pulse_step × RMPCNTL[3:0]'. The smallest_pulse_step is defined by prescaler value of CWPWH and CCWPWH. Each prescaler setting's smallest_pulse_step is given in Table 26 and Table 28 (the minimum value of the range).

The ramp control will start and end in speed of maximum_pulse_step, which is the maximum value of the range given in Table 26 and Table 28.

The ramp-up is completed when the pulse width gets the width that is set by CWPWL/CWPWH or CCWPWL/CCWPWH registers.

During ramp-up, the step pulse width is automatically decremented (from the maximum value for step pulse width in the chosen range) until the value in CWPW or the CCWPW register is reached, depending on the direction of rotation. See Figure 10.

During ramp-down, the step pulse width is automatically incremented from the current value in CWPW or the CCWPW, depending on the direction of rotation, until it reaches the maximum value for step pulse width in the chosen range. See Figure 10.



Example shown is one-phase drive for clockwise rotation.

(1) The ramp start or ramp end speed is defined as the maximum value of the range given in Table 26 and Table 28 based on prescaler bits [15:13] in CWPWH/CCWPWH registers. For example, the ramp start or ramp end speed is 98.304 ms if the CWPWH/CCWPWH[15:13] = 010.

(2) The decrease/increase step pulse width is defined as the minimum value of the range given in Table 26 and Table 28 based on prescaler bit s [15:13] in CWPWH/CCWPWH registers times the ramp step multiplication factor bits [3:0] in RMPCNTL register. For example, the decrease/increase step pulse width is 192 µs (12 µs × 16) if the CWPWH/CCWPWH[15:13] = 010 (minimum value 12 µs) and RMPCNTL[3:0] = 0100 (multiplication factor 16).

(3) The ramp-up final speed is defined as the minimum value of the range given in Table 26 and Table 28 based on prescaler bit s [15:13] times the step pulse width value bits [12:0] plus 1 in CWPWH/L and CCWPWH/L registers. For example, the ramp-up final speed is 24 µs (12 µs × 2) if the CWPWH/CCWPWH[15:13] = 010 (minimum value 12 µs) and the CWPWH/L or CCWPWH/L = 0x0001 (1 + 1).

Fig 10. PCA9629 operation model for ramp-up (acceleration) and ramp-down (deceleration)