PCA9629A

Fm+ I²C-bus advanced stepper motor controller

Rev. 2 — 21 March 2014 Product data sheet

1. General description

The PCA9629A is an I²C-bus controlled low-power CMOS device that provides all the logic and control required to drive a four phase stepper motor. PCA9629A is intended to be used with external high current drivers to drive the motor coils. The PCA9629A 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 PCA9629A. This is a useful feature in sensing home position of motor shaft or reference for step pulses. Upon interrupt, the PCA9629A can be programmed to automatically stop the motor, re-start motor, enable extra steps 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 PCA9629A 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 actions from 1 to 255 or continuous rotations and direction of rotation. Re-start motor for new speed and operation without waiting for motor stop. A ramp-up on start and/or ramp-down on stop is also provided with re-enable ramp-up or ramp-down to change the ramp rate curve on the fly.

The PCA9629A 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 unipolar stepper motor coils
  - Up to 1000 pF loads with 100 ns rise and fall times
- Built-in 1 MHz 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: 333.3 kpps to 0.3 pps with ±3 % accuracy
- Programmable ramp-up on start and ramp-down to stop
- Programmable re-enable ramp-up or ramp-down to change ramp rate curve on the fly
- Programmable re-start motor with new speed and operation while motor is still running
- Programmable motor action either multiple times (1 to 255) or continuously
- Programmable loop delay timer for motor reversal mode

- Programmable steps with clockwise and/or counter-clockwise control
- Direction control of motor shaft
- Selectable active hold (last state), power on, power off or released states for motor shaft
- 32-bit step counter to count output steps
- Interrupt features
  - Active LOW open-drain interrupt output
  - Programmable watchdog timer with option to generate interrupt, reset device or stop motor
  - Programmable motor stop interrupt
  - Sensor enabled drive control: linked to interrupt from GPIO pins
  - Programmable interrupt Mask Control for input sources
- Four stepper motor drive outputs: OUT0 to OUT3
  - Configured to drive stepper motor outputs and capable to read back the last output states when motor is stopped
  - Both output phase and state can be changed at any time
  - Programmable time-out timer to set all outputs to zeros when motor is stopped
  - Configured as general purpose outputs to drive (source/sink) loads up to 25 mA
- Four general purpose I/Os: P0 to P3
  - Configured to sense logic level outputs from optical interrupter photo transistor circuit
  - Programmable filter timer to suppress spike or noise for P0 and P1 inputs
  - Configured as outputs to drive (source/sink) LEDs or other loads up to 25 mA
- 4.5 V to 5.5 V operation
- 1 MHz Fast-mode Plus (Fm+) compatible I²C-bus serial interface with 30 mA high drive capability on SDA output for driving high capacitive buses
- Active LOW reset ($\overline{\text{RESET}}$) input pin resets device to power-up default state: can be used to recover from bus stuck condition
- 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
- Security and surveillance camera

- Variable-speed fans and pumps
- Vending machines

4. Ordering information

Table 1. Ordering information

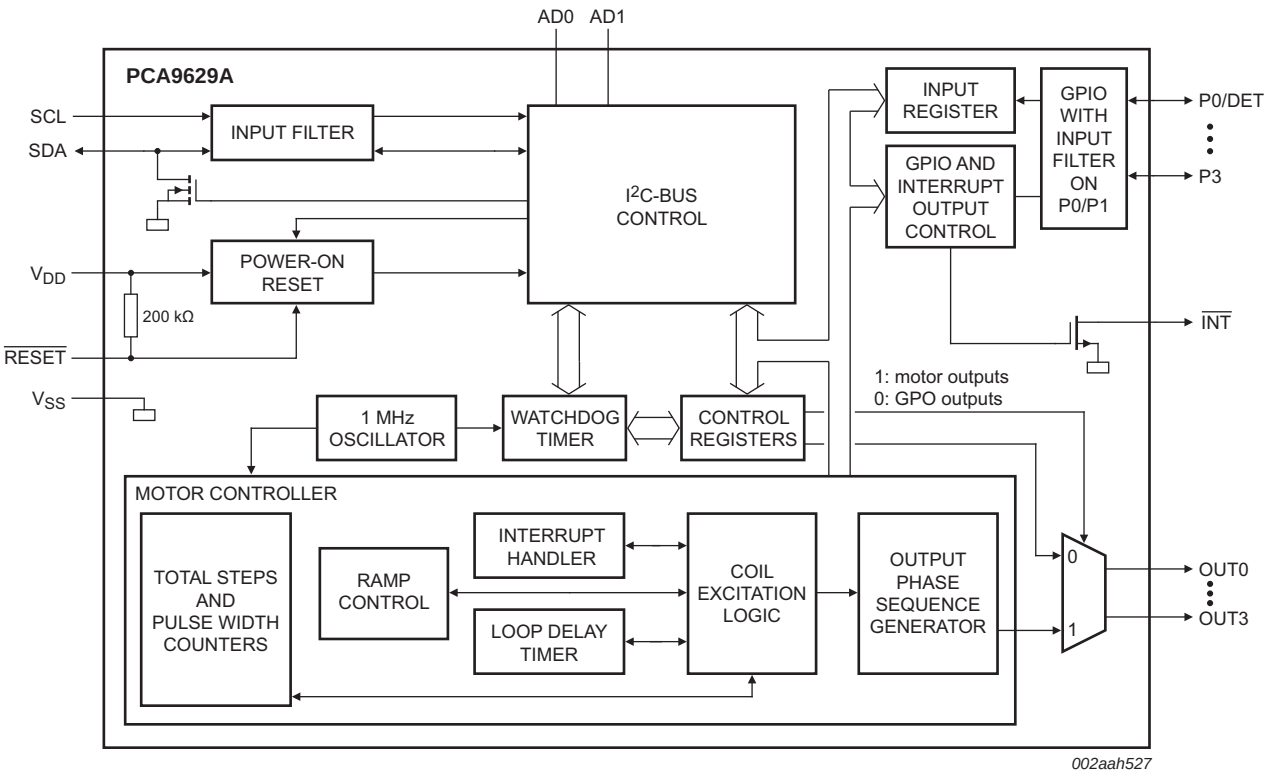
Type number	Topside marking	Package		
		Name	Description	Version
PCA9629APW	PA9629A	TSSOP16	plastic thin shrink small outline package; 16 leads; body width 4.4 mm	SOT403-1

4.1 Ordering options

Table 2. Ordering options

Type number	Orderable part number	Package	Packing method	Minimum order quantity	Temperature
PCA9629APW	PCA9629APWJ	TSSOP16	Reel 13” Q1/T1 *Standard mark SMD	2500	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. PCA9629A block diagram

6. Pinning information

6.1 Pinning

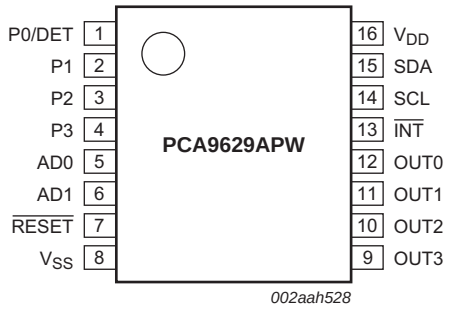


Fig 2. Pin configuration for TSSOP16

6.2 Pin description

Table 3. Pin description

Symbol	Pin	Type	Description
P0/DET	1	I/O	input/output 0 (output is 25 mA push-pull) or detection of motor position input (see details in Section 7.3.21 "MCNTL — Motor control register")
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 “PCA9629A 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 PCA9629A 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.

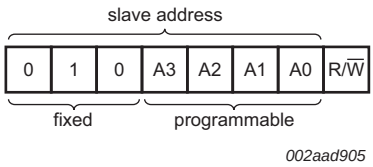


Fig 3. PCA9629A device address

The last bit of the first byte defines the reading from or writing to the PCA9629A. When set to logic 1 a read is selected, while logic 0 selects a write operation.

Table 4. PCA9629A 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 PCA9629A. 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 35 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 = 22h) 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	WDTOI	read/write	Watchdog time-out interval register
02h	0	0	0	0	1	0	WDCNTL	read/write	Watchdog control register
03h	0	0	0	0	1	1	IO_CFG	read/write	I/O Configuration register
04h	0	0	0	1	0	0	INTMODE	read/write	Interrupt mode register
05h	0	0	0	1	0	1	MSK	read/write	Mask interrupt register
06h	0	0	0	1	1	0	INTSTAT	read only	Interrupt status register
07h	0	0	0	1	1	1	IP	read only	Input port register
08h	0	0	1	0	0	0	INT_MTR_ACT	read/write	Interrupt motor action control register
09h	0	0	1	0	0	1	EXTRASTEPS0	read/write	Count value for extra steps for INTP0
0Ah	0	0	1	0	1	0	EXTRASTEPS1	read/write	Count value for extra steps for INTP1
0Bh	0	0	1	0	1	1	OP_CFG_PHS	read/write	Output port configuration and phase control register
0Ch	0	0	1	1	0	0	OP_STAT_TO	read/write	Output port state and time-out control register
0Dh	0	0	1	1	0	1	RUCNTL	read/write	Ramp up control register
0Eh	0	0	1	1	1	0	RDCTNL	read/write	Ramp down control register

Table 5. Register summary *...continued*

Register number	D5	D4	D3	D2	D1	D0	Name	Type	Function
0Fh	0	0	1	1	1	1	PMA	read/write	Perform multiple of actions control register
10h	0	1	0	0	0	0	LOOPDLY_CW	read/write	Loop delay time for reversing from CW to CCW register
11h	0	1	0	0	0	1	LOOPDLY_CCW	read/write	Loop delay time for reversing from CCW to CW register
12h	0	1	0	0	1	0	CWSCOUNTL	read/write	Number of steps CW low byte
13h	0	1	0	0	1	1	CWSCOUNTH	read/write	Number of steps CW high byte
14h	0	1	0	1	0	0	CCWSCOUNTL	read/write	Number of steps CCW low byte
15h	0	1	0	1	0	1	CCWSCOUNTH	read/write	Number of steps CCW 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	MCNTL	read/write	Motor start/stop and rotate direction control
1Bh	0	1	1	0	1	1	SUBADR1	read/write	I²C-bus subaddress 1
1Ch	0	1	1	1	0	0	SUBADR2	read/write	I²C-bus subaddress 2
1Dh	0	1	1	1	0	1	SUBADR3	read/write	I²C-bus subaddress 3
1Eh	0	1	1	1	1	0	ALLCALLADR	read/write	All Call I²C-bus address
1Fh	0	1	1	1	1	1	STEPCOUNT0	read only	Step counter byte 0
20h	1	0	0	0	0	0	STEPCOUNT1	read only	Step counter byte 1
21h	1	0	0	0	0	1	STEPCOUNT2	read only	Step counter byte 2
22h	1	0	0	0	1	0	STEPCOUNT3	read only	Step counter byte 3
23h to 3Fh	-	-	-	-	-	-	-	-	Reserved[1]

[1] These registers marked “Reserved” should not be written, and the master will not be acknowledged when accessed.

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	R/W	1	Low-power sleep mode. Oscillator off.
				0*	Normal mode.
		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 condition
		3	R/W	1	PCA9629A responds to I²C-bus subaddress 1
				0*	PCA9629A does not respond to I²C-bus subaddress 1
		2	R/W	1	PCA9629A responds to I²C-bus subaddress 2
				0*	PCA9629A does not respond to I²C-bus subaddress 2
		1	R/W	1	PCA9629A responds to I²C-bus subaddress 3
				0*	PCA9629A does not respond to I²C-bus subaddress 3
		0	R/W	1*	PCA9629A responds to All Call I²C-bus address
				0	PCA9629A does not respond to All Call I²C-bus address

7.3.1.1 Low-power sleep mode, oscillator off (bit 6)

This feature allows user to program the device in Low-power sleep mode (internal oscillator off) to save power when motor output pins are idle.

Writing ‘1’ to this bit will be ignored and no effect if the motor is running (MCNTL[7] = 1), otherwise the device will go into Low-power sleep mode and de-assert $\overline{\text{INT}}$ output pin if $\overline{\text{INT}}$ pin is asserted, but the interrupt status bits will not be cleared and keep the same value.

As soon as the device enters into the Low-power sleep mode, all motor output pins are forced driving LOW, all GPIO pins are set as input with 3-state (high-impedance) output, and interrupt status bits will not allow change (even GPIOs input signal edge is changed or reading/writing to access the INTSTAT register will not clear this register). User can still read or write to set motor parameter registers during Low-power sleep mode.

Writing ‘0’ to this bit will bring the device back to normal operation mode. It takes 300 μs maximum for the internal oscillator to be up and running back to normal operation mode. Timing on motor drive outputs and GPIOs control are not guaranteed if all registers are accessed within the 300 μs window. The $\overline{\text{INT}}$ output pin will be asserted again if any of the interrupt status bits are set before entering the Low-power sleep mode and interrupt output pin is enabled (bit 5 = 0).

7.3.1.2 Disable interrupt output pin (bit 5)

This feature is useful when the host/micro/master does not want the $\overline{INT}$ pin to toggle when interrupts occur. Within PCA9629A, 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 $\overline{INT}$ 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 $\overline{INT}$ pin.

The only exception to this rule is when the watchdog timer is enabled in the 'Interrupt and Reset' mode (see Section 7.3.2.2). 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.

7.3.1.3 Outputs change on STOP (bit 4)

This feature can be used to synchronize the starting of the motor across multiple PCA9629A devices on the bus at approximately the same time (within few microseconds of one another). The host controller can program all the PCA9629As on the bus and then issue the I²C-bus STOP condition. Upon receiving the STOP condition, all the PCA9629A 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 Watchdog timer

The purpose of the watchdog timer is to recover the PCA9629A 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 or stop motor, resets PCA9629A or motor is stopped if the user program fails to 'feed' the watchdog. To feed the watchdog, the user simply addresses the PCA9629A ([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 PCA9629A to POR state if it is not periodically addressed
- Can be programmed to stop the motor if it is not periodically addressed and motor is in start (running) state
- Can be enabled by software and select one of three watchdog interrupt modes
- Watchdog interrupt flag bit[5] in INTSTAT register
- Programmable 8-bit timer 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, interrupt and reset, or interrupt and stop motor) and enable the watchdog using the WDCNTL register
- Address the PCA9629A periodically before the watchdog timer underflows to prevent reset/interrupt/motor stop
- Read watchdog interrupt bit (WDINT) in INTSTAT register to check for watchdog event

7.3.2.1 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 7. WDTOI - Watchdog time-out interval register (address 01h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
01h	WDTOI	7:0	R/W	FFh*	Watchdog time-out interval

7.3.2.2 WDCNTL — WatchDog Control register

Table 8. WDCNTL - Watchdog control register (address 02h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
02h	WDCNTL	7:3	read only	00h*	Reserved.
		2:1	R/W	11 or 10	WDMOD: watchdog interrupt and stop motor mode.
				01	WDMOD: watchdog interrupt and reset mode.
				00*	WDMOD: watchdog interrupt only mode.
		0	R/W	1	WDEN: watchdog enabled.
				0*	WDEN: watchdog disabled.

This register controls the operation of the watchdog timer. Watchdog timer can be enabled by setting the WDEN bit of this register. Before enabling the watchdog timer, the watchdog interrupt flag WDINT **must** be cleared (if it is set). The interrupt flag, motor stop flag and the reset flag are shared same bit (WDINT) in INTSTAT register.

The WDMOD bit determines the mode of operation. There are three 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 bit) in INTSTAT register.
- Interrupt and reset mode: In this mode, when the watchdog timer times out, the reset flag is set (WDINT bit) in INTSTAT register and resets the chip to POR state. All registers are set to default value except the WDINT bit stay ‘1’ in INTSTAT register until read of or write to the INTSTAT register.
- Interrupt and stop motor mode: When the watchdog timer times out, the motor stop flag is set (WDINT bit) in INTSTAT register and motor is forced to stop (set OFF, logic 0 on OUT[3:0]) immediately regardless of the bit[3:0] setting in the OP_STAT_TO register. User can change the motor output state in OP_STAT_TO register after WDINT bit is cleared.

7.3.3 IO_CFG — 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 and the upper 4 bits of this register reflect the outgoing logic levels of these pins. At reset, the device's ports P0 to P3 are inputs.

Table 9. IO_CFG - I/O configuration register (address 03h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
03h	IO_CFG	7:4	R/W	0000*	Reflects output logic levels when P[3:0] are configured as general purpose outputs.
		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

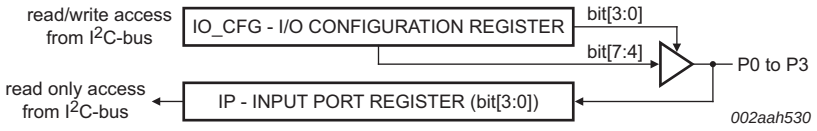


Fig 5. Simplified schematic for GPIO control

7.3.4 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.

Bits [6:4] are used to configure the pulse width of the input spike or noise to suppress for P0 to P1 as interrupt based inputs.

Table 10. INTMODE - Interrupt mode register (address 04h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
04h	INTMODE	7	-	0*	reserved
		6:4	R/W		pulse width of spike or noise that must be suppressed by the input filter for P0 and P1 inputs
				111 to 100	10 ms
				011	5 ms
				010	1 ms
				001*	500 μs
				000	0 s
		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

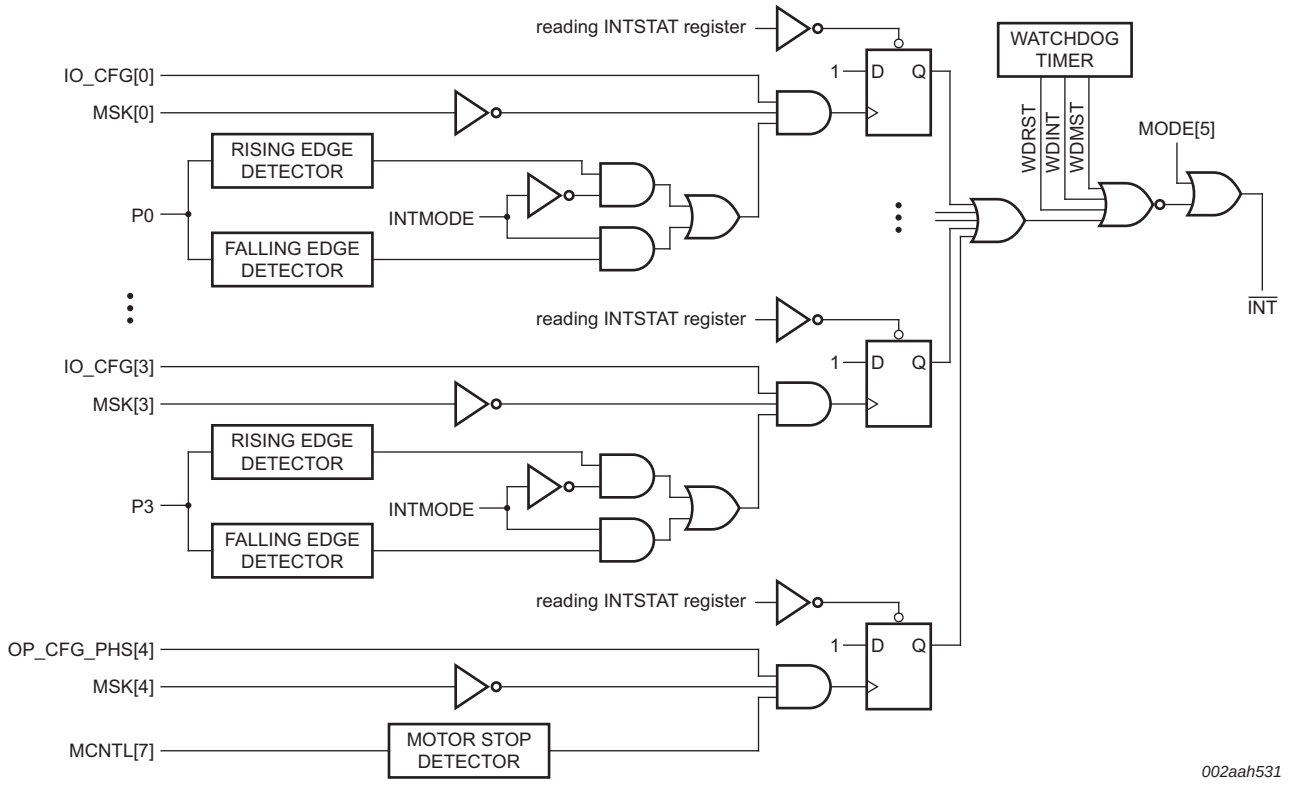


Fig 6. PCA9629A interrupt logic

7.3.5 MSK — Mask interrupt register

Upon power-up, all the internal interrupt latches are reset and interrupt flags cleared and interrupt mask bits [4:0] are set to logic 1, thus disabling interrupts from input ports P0 to P3 and motor stop caused by bit 7 in MCNTL register. Interrupts may be enabled by setting corresponding mask bits to logic 0.

Table 11. MSK - Interrupt mask register (address 05h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
05h	MSK	7:5	-	000*	reserved
		4	R/W	1*	Disable interrupt when motor is stopped
				0	Enable interrupt when motor is stopped
		3	R/W	1*	Disable interrupt for I/O P3
				0	Enable interrupt for I/O P3
		2	R/W	1*	Disable interrupt for I/O P2
				0	Enable interrupt for I/O P2
		1	R/W	1*	Disable interrupt for I/O P1
				0	Enable interrupt for I/O P1
		0	R/W	1*	Disable interrupt for I/O P0
				0	Enable interrupt for I/O P0

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

7.3.6 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, and motor stop interrupt caused by bit 7 in MCNTL register when this bit changes from 1 to 0.

Table 12. INTSTAT - Interrupt status register (address 06h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
06h	INTSTAT	7:6	-	00*	reserved
		5	read only	1	WDINT watchdog interrupt flag set[1][2]
				0*	WDINT watchdog interrupt flag clear
		4	read only	1	Motor stop interrupt flag set[1]
				0*	Motor stop interrupt flag clear
		3:0	read only	1	INTP3 flag set[1]
				0*	INTP3 flag clear
				1	INTP2 flag set[1]
				0*	INTP2 flag clear
				1	INTP1 flag set[1]
				0*	INTP1 flag clear
				1	INTP0 flag set[1]
				0*	INTP0 flag clear

[1] Reading or writing any value to this register will clear this bit.

[2] This bit will be cleared by any reset events except the watchdog reset event.

Upon power-up or any reset events, INTSTAT register bits [4:0] are cleared (= 0), thus clearing the interrupt flags. Change in logic level at GPIO pins P0 to P3 configured as inputs or motor stopped will cause generation of interrupt when not masked using MSK register. The corresponding flag bit in this register is set and latched until reading this register clears all bits.

7.3.7 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 13. 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

7.3.8 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
- Re-start the motor with new speed and ramp rate
- Move extra steps 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 register, INT_MTR_ACT, is 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_MTR_ACT register must be set to 1.

7.3.8.1 INT_MTR_ACT — Interrupt motor action control register

Table 14. INT_MTR_ACT - Interrupt motor action control register (address 08h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
08h	INT_MTR_ACT	7:5	R/W	111 or 110	re-start motor on INT caused by P0 or P1
				101	re-start motor on INT caused by P1
				100	re-start motor on INT caused by P0
				011	reverse motor on INT caused by P0 or P1
				010	stop motor on INT caused by P0 or P1
				001	stop motor on INT caused by P1
				000*	stop motor on INT caused by P0
		4:3	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
		2:1	R only	00*	reserved
		0	R/W	1	enable interrupt based control of motor
				0*	disable interrupt based control of motor

If the bit 0 interrupt based control of motor is disabled, then values programmed in the rest of bit [3:7] have no effect on the motor operation. The interrupt motor action control has no effect during ramp operation.

When an interrupt occurs, if the motor is programmed in bit [7:5] 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 EXTRASTEPS0/EXTRASTEPS1 register settings), then extra steps will occur.
2. If ramp down is enabled (see RDCNTL register setting), the motor starts ramping down.
3. Motor stops.

Remark: Setting ‘re-start motor’ will be ignored when ‘stop motor on INT’ is detected.

When an interrupt occurs, if the motor is programmed in bit [7:5] to reverse direction on that interrupt, the following sequence of events takes place:

1. If extra steps feature is enabled for that interrupt (see EXTRASTEPS0/EXTRASTEPS1 register setting), then extra steps occurs in the current direction of motion.
2. The motor stops for the amount of time specified in the LOOPDLY_CW or LOOPDLY_CCW timer register.
3. Motor reverses its direction of rotation.

When an interrupt occurs, if the motor is programmed in bit [7:5] to re-start motor on that interrupt, the following sequence of events takes place:

1. The current motor speed and operation is changing and re-start new motor speed and operation based on these new motor parameter registers setting.

The bit [4:3] setting provides a mechanism to clear the two interrupts (INTP0 and INTP1) automatically with the occurrence of one interrupt clears the other without the microcontroller. The auto clear feature is disabled by default.

This feature is only available for interrupts that directly affect the operation of the motor as defined by the bit [7:5]. 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 read the INTSTAT register). If both the interrupts are used to control the motor operation (bit [7:5] = 010, 011, or 110/111), then all options of the bit [4:3] setting are valid. Any interrupt that is not automatically cleared by its pair should be manually cleared through I²C-bus read the INTSTAT register.

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**.

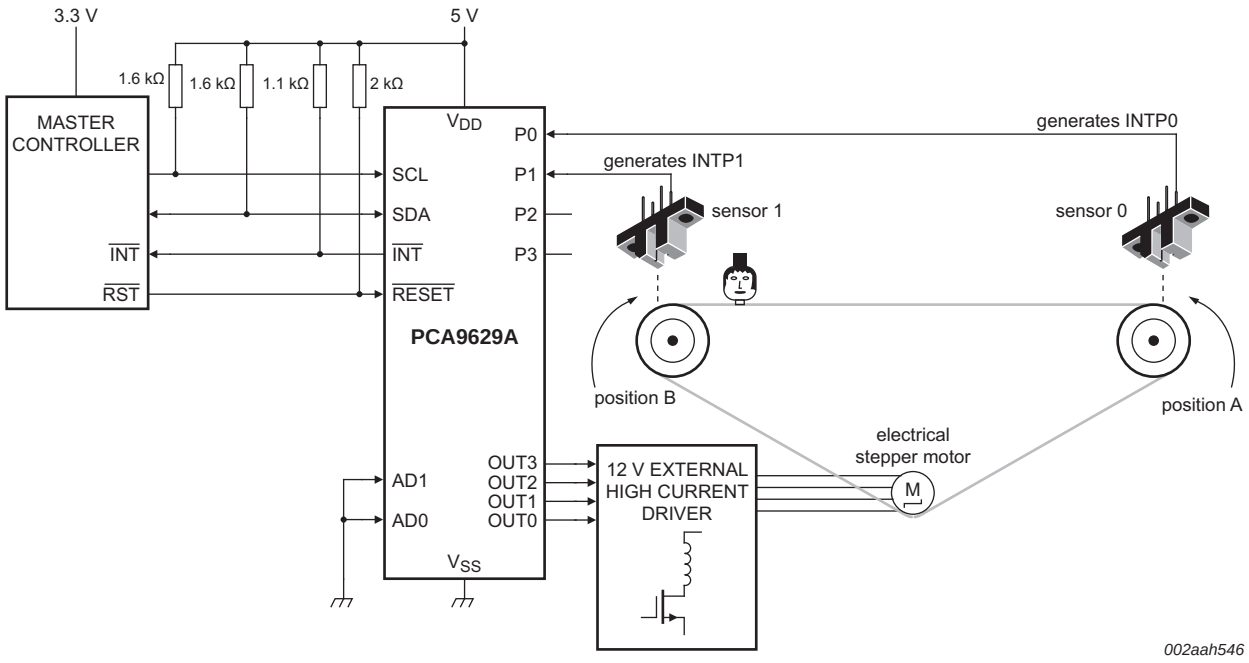


Fig 7. Example of controlling doll head movement between two home positions

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

Set IO_CFG[1:0] = 11, configure both P0 and P1 as inputs.

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

Set INT_MTR_ACT[7:5] = 011, reverse motor on interrupt caused by P0 or P1.

Set INT_MTR_ACT[4:3] = 01, INTP0 auto clears INTP1; INTP1 auto clears INTP0.

Set INTMODE[1:0] to select interrupt occurs on either falling or rising edge.

Set MSK[1:0] = 00, enable both interrupts for P0 and P1.

Start motor by writing bit 7 to 1 in 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_MTR_ACT[0] = 1 (enable interrupt based motor control) and INT_MTR_ACT[7:5] = 011 (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.

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

Table 15. EXTRASTEPS0, EXTRASTEPS1 - Extra steps count for INTP0, INTP1 register (address 09h, 0Ah) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
09h	EXTRASTEPS0	7:0	R/W	00h*	count value for EXTRASTEPS (steps) for INTP0
0Ah	EXTRASTEPS1	7:0	R/W	00h*	count value for EXTRASTEPS (steps) for INTP1

Extra steps feature is used to make the motor rotate a specified amount of steps from the point of an interrupt occurrence. The EXTRASTEPS0 register is used for P0 interrupt (INTP0) and the EXTRASTEPS1 register is used for P1 interrupt (INTP1).

This register has no effect if the interrupt based motor control is disabled (INT_MTR_ACT[0] = 0).

The 8-bit value in this register is used to determine the number of steps to be overdriven. Direction of rotation of motor is maintained. If the count value in this register = 0, the EXTRASTEPS feature is disabled.

7.3.10 OP_CFG_PHS — Output Port Configuration and Phase control register

This register is used to configure the OUT[3:0] pins output function and level. When bit 4 is set to 0 to select the OUT[3:0] as general purpose output pins and the lower 4 bits of this register reflect the outgoing logic levels of these pins. When bit 4 is set to 1 to select the OUT[3:0] as motor drive output pins and the lower 4 bits will reflect the output logic levels on OUT[3:0] pins when motor is stopped (read only).

The bits [7:6] are used to configure the phase of the output waveforms at the motor output pins 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 two bits:

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

Table 16. OP_CFG_PHS - Output Port Configuration and Phase control register (address 0Bh) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
0Bh	OP_CFG_PHS	7:6	R/W	11 or 10	half-step drive outputs
				01	two-phase drive outputs
				00*	one-phase drive outputs
		5	-	0*	reserved
		4	R/W	1*	OUT[3:0] will be configured as motor drive outputs
				0	OUT[3:0] will be configured as general purpose outputs (GPO)
		3:0	R/W	0000*	When bit 4 is set to 0, these 4 bits reflect output logic levels on OUT[3:0] pins for GPO function. When bit 4 is set to 1, these 4 bits reflect the last output logic levels on OUT[3:0] pins when motor is stopped (read only).

Remark: The phase of the output waveforms can be changed at any time by writing to bits [7:6] of this register.

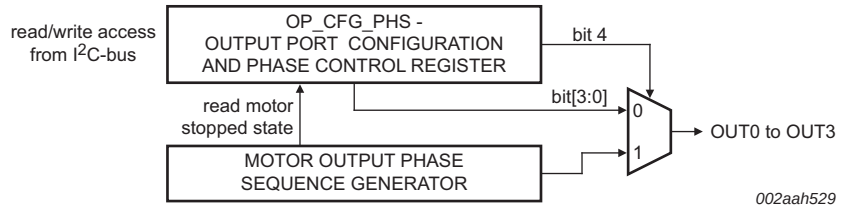


Fig 8. Simplified schematic for OUT[3:0] control

7.3.11 OP_STAT_TO

The bit [1:0] determines the condition of motor output pins when motor is stopped on clockwise and the bit [3:2] determines the condition of motor output pins when motor is stopped on counter-clockwise. One of logic 0, logic 1 or hold (last state) selects for motor output state.

The bit [7:5] is used to configure the motor stop time-out timer. The time-out timer is enabled and starts to count down when this value is non-zero and motor is stopped, the motor output pins will drive all logic 0s when counting down to zero. This time-out timer is counting down when motor is stopped and reload the value again when motor is started.

When motor is in stop condition (MCNTL[7] = 0), the OUT[3:0] pins are driving HIGH if output pins are set to logic 1 (ON) and the time-out timer is disabled in OP_STAT_TO register.

Table 17. OP_STAT_TO - Output state and time-out control register (address 0Ch) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
0Ch	OP_STAT_TO	7:5	R/W		Motor stop time-out timer. The output state will drive all logic 0 when motor is stopped and this timer counts down to zero.
				111	1020 ms
				110	508 ms
				101	252 ms
				100	124 ms
				011	60 ms
				010	28 ms
				001	12 ms
				000*	disable time-out timer
		4	R/W	0*	reserved
		3:2	R/W	11 or 10	output pins = logic 1 (ON) after CCW stop
				01	output pins = HOLD last state after CCW stop
				00*	output pins = logic 0 (OFF) after CCW stop[1]
		1:0	R/W	11 or 10	output pins = logic 1 (ON) after CW stop
				01	output pins = HOLD last state after CW stop
				00*	output pins = logic 0 (OFF) after CW stop[1]

[1] No need to enable the motor stop time-out timer.

Remark: Both output time-out control and output state can be changed at any time by writing to this register.

7.3.12 RUCNTL — Ramp-up control register

Table 18. RUCNTL - Ramp-up control register (address 0Dh) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
0Dh	RUCNTL	7:6	read only	00*	reserved
		5	R/W	1	enable ramp-up during start
				0*	disable ramp-up during start
		4	R/W	1	re-enable ramp-up to change ramp-up rate[1]
				0*	self clear after new rate of ramp-up start running
		3:0	R/W	0000*	ramp-up step multiplication factor

[1] No effect if bit 5 is set to 0, or when the motor is running on the final (top) speed.

The multiplication factor has a decimal range from 1 to 8192 as shown in Table 19.

Table 19. 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

The RUCNTL[5] enables/disables the speed ramp-up during starting of the motor.

The RUCNTL[4] re-enables the new speed ramp-up to replace the current ramp-up rate while the current ramp-up is running. User can write bit [5:4] = 11 with new ramp-up step multiplication factor in bit [3:0] to change new ramp-up rate and bit 4 is self-cleared to 0. Re-enabling the ramp-up will change the ramp-up rate immediately during the ramp-up operation only. User can set re-enable ramp-up without re-start motor operation to change the ramp-up rate curve on-the-fly. The prescaler range setting (eight ranges in CWPWH/CCWPWH registers) must be in the same range during re-enable ramp-up operation. See Figure 10.

The RUCNTL[3:0] defines the acceleration rate of the ramp-up control. If the value is small, the PWM width decrement (accelerating) is slower.

The pulse width decrement step is ‘smallest_pulse_step × RUCNTL[3:0] factor’. The smallest_pulse_step is defined by prescaler value of CWPWH [15:13] and CCWPWH[15:13]. Each prescaler setting’s smallest_pulse_step is given in Table 27 and Table 29 (the minimum value of the range).

The ramp up control will start in speed of maximum_pulse_step, which is the maximum value of the range given in Table 27 and Table 29.

The ramp-up is completed (final speed) 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 9

7.3.13 RDCNTL — Ramp-down control register

Table 20. RDCNTL - Ramp-down control register (address 0Eh) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
0Eh	RDCNTL	7:6	read only	00*	reserved
		5	R/W	1	enable ramp-down to stop
				0*	disable ramp-down to stop
		4	R/W	1	re-enable ramp-down to change ramp-down rate[1]
				0*	self clear after new rate of ramp-down start running
		3:0	R/W	0000*	ramp-down step multiplication factor

[1] No effect if bit 5 is set to 0, or when the motor is running on the final (top) speed.

The multiplication factor has a decimal range from 1 to 8192 as shown in Table 19.

The RDCNTL[5] enables/disables the speed ramp-down during stopping of the motor.

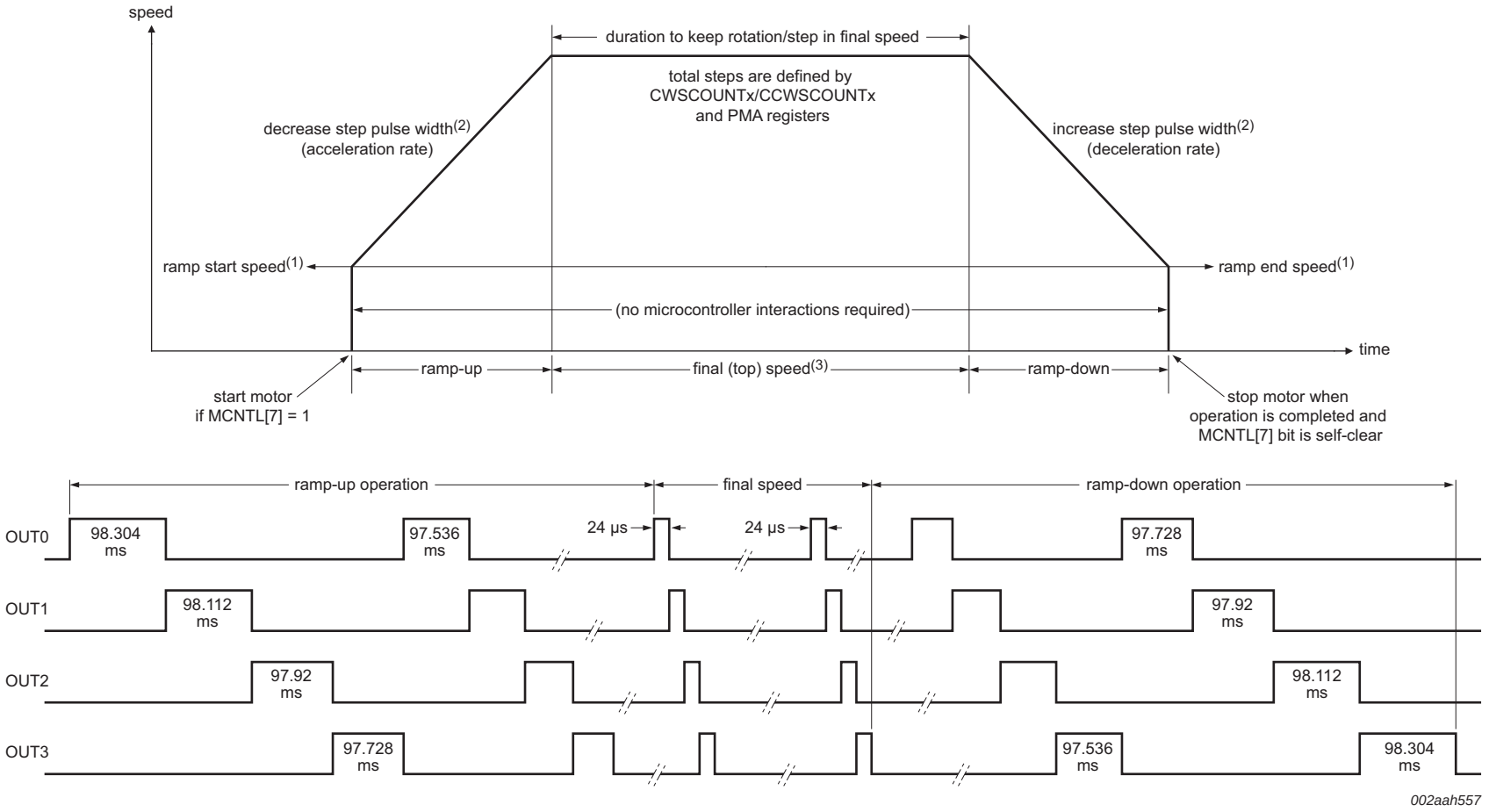
The RDCNTL[4] re-enables the new speed ramp-down to replace the current ramp-down rate while the current ramp down is running. User can write bit [5:4] = 11 with new ramp-down step multiplication factor in bit[3:0] to change new ramp-down rate and bit 4 is self-cleared to 0. Re-enabling the ramp-down will change the ramp-down rate immediately during the ramp down operation only. User can set re-enable ramp down without re-start motor operation to change the ramp-down rate curve on-the-fly. The prescaler range setting (eight ranges in CWPWH/CCWPWH registers) must be in the same range during re-enable ramp down operation. See Figure 10.

The RDCNTL[3:0] defines the decelerating rate of the ramp down control. If the value is small, the PWM width increment (decelerating) is slower.

The pulse width increment step is ‘smallest_pulse_step × RDCNTL[3:0] factor’. The smallest_pulse_step is defined by prescaler value of CWPWH[15:13] and CCWPWH[15:13]. Each prescaler setting’s smallest_pulse_step is given in Table 27 and Table 29 (the minimum value of the range).

The ramp-down control will end in speed of maximum_pulse_step, which is the maximum value of the range given in Table 27 and Table 29.

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 9.



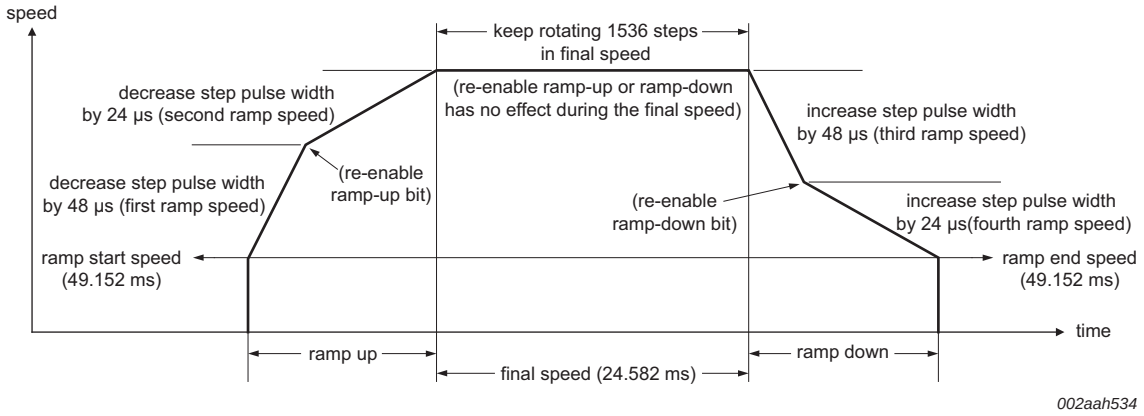
Example shown is one-phase drive for clockwise rotation.

(1) The ramp start or ramp end speed is defined as the maximum value of the range given in Table 27 and Table 29 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 27 and Table 29 based on prescaler bit s [15:13] in CWPWH/CCWPWH registers times the ramp step multiplication factor bits [3:0] in RUCNTL/RDCNTL registers. 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 RUCNTL/RDCNTL[3:0] = 0100 (multiplication factor 16).

(3) The ramp-up final speed is defined as the minimum value of the range given in Table 27 and Table 29 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 9. PCA9629A operation model for ramp-up (acceleration) and ramp-down (deceleration)



Example shown for how to program re-enable ramp-up and ramp-down during ramp operation.

(1) CWPWH = 30h, CWPWL = 00h (ramp start/stop speed is 49.152 ms and final speed is 24.582 ms.

(2) CWSCOUNTH = 06h, CWSCOUNTL = 00h (duration to keep rotation 1536 steps after final speed).

(3) RUCNTL = 23h (enable ramp-up and set acceleration rate 6 µs × 8 = 48 µs; this is the first ramp speed).

(4) RDCNTL = 23h (enable ramp-down and set deceleration rate 6 µs × 8 = 48 µs; this is the third ramp speed).

(5) PMA = 01h (perform motor action specified in bit[1:0] of MCNTL register once).

(6) MCNTL = 80h (start motor with rotate clockwise).

(7) RUCNTL = 32h (during ramp-up, re-enable ramp-up and change the multiplication factor value to the new ramp acceleration rate 6 µs × 4 = 24 µs; this is the second ramp speed).

(8) RDCNTL = 32h (during ramp-down, re-enable ramp-down and change the multiplication factor value to the new ramp deceleration rate 6 µs × 4 = 24 µs; this is the fourth ramp speed).

Fig 10. PCA9629A operation model for re-enable ramp-up and ramp-down

During ramp-up and ramp-down phase of operation, the interrupt based controls do not affect the motor run. A stop request from the microcontroller (writing MCNTL[7] = 0 or MCNTL[5] = 1) is the only event that affects the motor operation during ramp-up and ramp-down.

During ramp-up, the micro can issue a stop request either normal stop (MCNTL[7] = 0) or emergency stop (MCNTL[5] = 1). The following sequence of events takes place in the given order:

1. If emergency stop is enabled (MCNTL[5] = 1), the motor stops immediately (even if ramp-down is enabled) - Priority 1.
2. If normal stop is enabled (MCNTL[7] = 0) and ramp-down is enabled, then the motor starts to ramp down to stop - Priority 2.
3. If normal stop is enabled (MCNTL[7] = 0) and ramp-down is disabled, then motor stops immediately - Priority 3.

During ramp-down, the micro can issue a stop request either normal stop (MCNTL[7] = 0) or emergency stop (MCNTL[5] = 1). The following sequence of events takes place in the given order:

1. If emergency stop is enabled (MCNTL[5] = 1), the motor stops immediately (it does not finish ramping down) - Priority 1.
2. If normal stop is enabled (MCNTL[7] = 0) and ramp-down is enabled, then the motor continues to ramp down to a stop - Priority 2.

In the duration between end of ramp-up and beginning of ramp-down, the interrupt based controls (if enabled) can affect the operation of the motor. In this region, Section 7.3.8 gives the priority of events when both interrupt-based control and ramp control are enabled together.

7.3.14 PMA — Perform multiple of actions control register

Table 21. PMA - Perform multiple of actions control register (address 0Fh) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
0Fh	PMA	7:0	R/W	00h	perform motor action specified in bits [1:0] of MCNTL register continuously
				01h*	perform motor action specified in bits [1:0] of MCNTL register once
				02h to FFh	Perform motor action number of times from 2 (02h) to 255 (FFh) specified in bits [1:0] of MCNTL register

This register determines if the motor operation specified in bits [1:0] of MCNTL register is executed once, multiple times (2 ~ 255) or continuously. If continuous operation (PMA = 00h) is set, the motor can be stopped either by issuing a stop request or if an interrupt happens and the motor is programmed to stop on that interrupt. If multiple times operation is set from 1 to 255 in bit [7:0], the motor stops automatically after finishing the current multiple times operation. The number of action counter is always increased by 1 whenever the motor rotation direction is changed during the MCNTL[1:0] = 10 or 11.

7.3.15 LOOPDLY_CW — Loop delay timer for CW to CCW control register

This feature is used to make the motor wait for a certain amount of time before reversing its direction from clockwise to counter-clockwise rotation. There are two situations in which the motor must reverse its direction of rotation:

- The user requests both clockwise and counter-clockwise rotation (also known as auto reversal mode).
- On an interrupt (also known as interrupt reversal mode).

This register holds the wait time value in resolution of 4 ms. 00h = 0 second wait time, 01h = 4 ms wait time, and FFh = 1.02 seconds wait time.

Remark: LOOPDLY_CW timer has an accuracy of ±3 %.

Table 22. LOOPDLY_CW - Loop delay timer for CW to CCW control register (address 10h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
10h	LOOPDLY_CW	7:0	R/W	00h*	loop delay counter for reversing from clockwise to counter-clockwise rotation

7.3.16 LOOPDLY_CCW — Loop delay timer for CCW to CW control register

This feature is used to make the motor wait for a certain amount of time before reversing its direction from counter-clockwise to clockwise rotation. There are two situations in which the motor must reverse its direction of rotation:

- The user requests both clockwise and counter-clockwise rotation (also known as auto reversal mode).
- On an interrupt (also known as interrupt reversal mode).

This register holds the wait time value in resolution of 4 ms. 00h = 0 second wait time, 01h = 4 ms wait time, and FFh = 1.02 seconds wait time.

Remark: LOOPDLY_CCW timer has an accuracy of ±3 %.

Table 23. LOOPDLY_CCW - Loop delay timer for CCW to CW control register (address 11h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
11h	LOOPDLY_CCW	7:0	R/W	00h*	loop delay counter for reversing from counter-clockwise to clockwise rotation

7.3.17 CWSCOUNTL, CWSCOUNTH — Number of clockwise steps register

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

Table 24. CWSCOUNTL, CWSCOUNTH - Number of clockwise steps count register (address 12h, 13h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
12h	CWSCOUNTL	7:0	R/W	00h*	number of clockwise steps, low byte
13h	CWSCOUNTH	7:0	R/W	00h*	number of clockwise steps, high byte

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

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

Table 25. CCWSCOUNTL, CCWSCOUNTH - Number of counter-clockwise steps count register (address 14h, 15h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
14h	CCWSCOUNTL	7:0	R/W	00h*	number of counter-clockwise steps, low byte
15h	CCWSCOUNTH	7:0	R/W	00h*	number of counter-clockwise steps, high byte

7.3.19 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 26. 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 (±3 %).

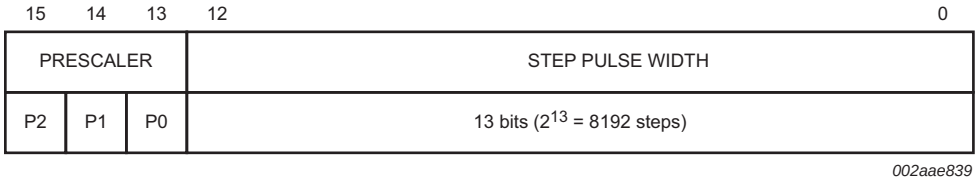


Fig 11. Step pulse width for clockwise rotation

The upper three bits of the register are the prescaler that determines the dynamic range for the step pulse width. Table 27 shows the range for each setting of the prescaler.

Table 27. 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 27 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).

7.3.20 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 28. 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 (±3 %).

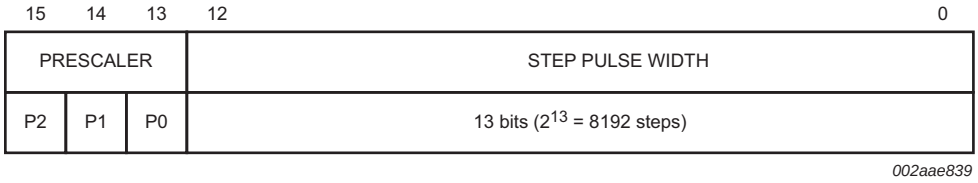


Fig 12. Step pulse width for counter-clockwise rotation

The upper three bits of the register are the prescaler that determines the dynamic range for the step pulse width. Table 29 shows the range for each setting of the prescaler.

Table 29. 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 29 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).

7.3.21 MCNTL — Motor control register

This register acts like the master control panel for driving the motor. It determines the type of motor operation and controls the starting/stopping or re-start new speed of the motor. The registers from address 08h (INT_MTR_ACT) to 19h (CCWPWH) are referred to as the motor parameter registers. The user must first program the motor parameter registers that are required for the current run of the motor. After that, this register should be programmed with the type of operation required in bits [1:0]. The motor starts when bit 7 of this register is set.

While the bit 7 of this register is still 1 (motor is running), the user also can re-program the motor parameter registers that are required to change the motor speed for the next run of the motor. The motor can re-start with new speed and operation without stopping motor when both bits [7:6] of this register are set to 1. The type of operation control in bit [1:0] is not allowed to change when re-starting motor.

Table 30. MCNTL - Motor control register (address 1Ah) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
1Ah	MCNTL	7	R/W	1	start motor
				0*	stop motor
		6	R/W	1	re-start motor for new speed and operation
				0*	self clear after new speed starts running
		5	R/W	1	emergency stop motor
				0*	self clear after motor stop and bit 7 also clears to 0
		4	W only	1	enable START (bit 7) ignore caused by P0 state
				0*	disable START (bit 7) ignore caused by P0 state
		3	W only		P0 polarity setting for START (bit 7) ignore
				1	set P0 input state is HIGH to ignore START bit 7
				0*	set P0 input state is LOW to ignore START bit 7
		2	R only	0*	reserved
		1:0	R/W	11	rotate counter-clockwise first, then clockwise
				10	rotate clockwise first, then counter-clockwise
				01	rotate counter-clockwise
				00*	rotate clockwise

7.3.21.1 MCNTL[7]: start/stop motor

This bit indicates the state of the motor. A ‘1’ indicates that the motor is running and ‘0’ indicates that the motor is in the stopped state.

To start the motor, write ‘1’ to this bit. Once the motor is started. Changing bits [1:0] of the MCNTL register do not affect motor operation. Only three bits that can affect motor operation in the MCNTL register while the motor is running are the start/stop bit, re-start bit, and emergency stop bit. Also, any start command (writing ‘1’ to this bit only when it is already set), before the completion of the current operation are ignored.

When the current operation is completed, the motor stops and this bit is cleared. The completion of motor operation can be checked by reading this bit or detecting motor stop interrupt if motor stop interrupt is enabled in MSK register (bit 4 = 0). After the motor has stopped, the motor parameter registers can be updated and the motor can be started again.

The microcontroller can stop the motor at any time by writing '0' to this bit (this is referred to as a normal stop request). Once the motor stops, this bit is cleared and it will generate interrupt if motor stop interrupt is enabled. Stop request issued when the motor is already in the stopped state is ignored.

7.3.21.2 MCNTL[6]: re-start motor

The 're-start motor' feature is designed to change the current motor speed and operation without stopping motor. So the re-start motor is changing current motor run based on the new motor parameter registers setting. User can re-program new speed and operation while the current motor operation is still running (bit 7 = 1), then write '1' to both bits [7:6] to re-start new speed/operation and this bit is self cleared after new speed and operation start running. The new speed of step pulse width will start on OUT(x + 1) output pin right after the current phase of output waveform finished on OUTx output pin. Once the motor is re-started by writing both bits [7:6] to '1', the motor parameter registers can be re-program again for another new speed and operation.

Re-start motor operation is allowed only if the total number of steps is not completed (in CWSCOUNT/CCWSCOUNT, PMA and EXTRASTEPS0/EXTRASTEPS1 registers), so re-start motor does not change the step counter (CWSCOUNT/CCWSCOUNT) value and will continuously count the remaining steps until all steps are completed or motor is stopped. The following registers can be re-programmed for re-start motor operation:

- INT_MTR_ACT
- LOOPDLY_CW, LOOPDLY_CCW
- RUCNTL, RDCNTL
- CWPW[12:0], CCWPW[12:0]

The following rules should be observed while programming this bit to re-start motor run:

- During the interrupt motor re-start operation, setting this re-start bit has no effect.
- The prescaler range setting (eight ranges in CWPWH/CCWPWH registers) must be in the same range during re-start motor operation.
- Set re-start motor bit to '1' while the current motor is running (bit 7 = 1) in the final (top) or target speed (S_0), the re-start motor action will take place immediately based on the new motor parameter registers setting. If both ramp-up/ramp-down are disabled, it will change to new speed of S_1 immediately as shown in Figure 13. If both ramp-up/ramp-down are enabled, it will go to new speed of S_1 with either ramp up ($S_0 < S_1$) or ramp down ($S_0 > S_1$) as shown in Figure 14.
- Set re-start motor bit to '1' while the current motor is running (bit 7 = 1) in the ramp operation, the re-start motor action will wait for the ramp-up or ramp-down to complete, then re-start the new motor operation based on the new motor parameter registers setting as shown in Figure 15.
- User can set both re-start motor and re-enable ramp operations at the same time. Motor will change the ramp rate first if motor is running in the ramp operation.

- Re-start motor request issued when the motor is already in the stopped state is ignored.
- Re-start motor request when the current motor is stopped and waiting for the loop delay time in auto-reversal mode is ignored.

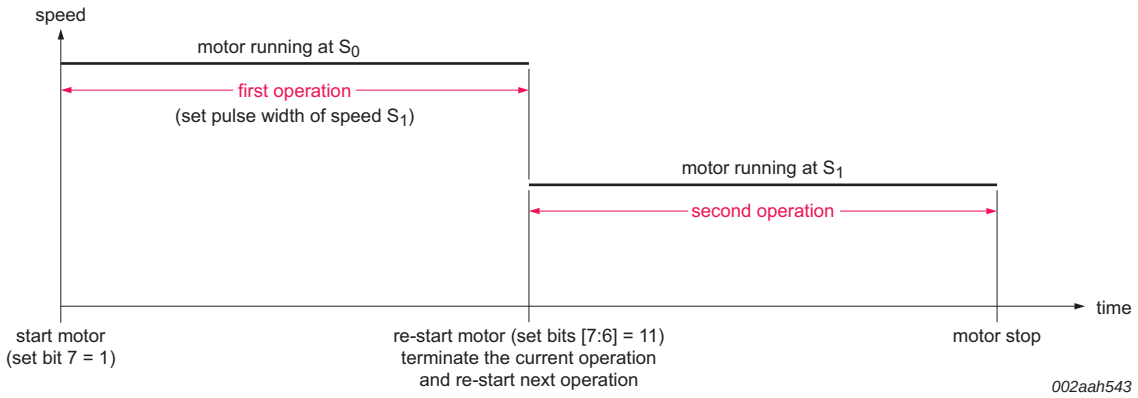


Fig 13. Re-start motor with ramp-up/ramp-down disable while current motor running in the final speed

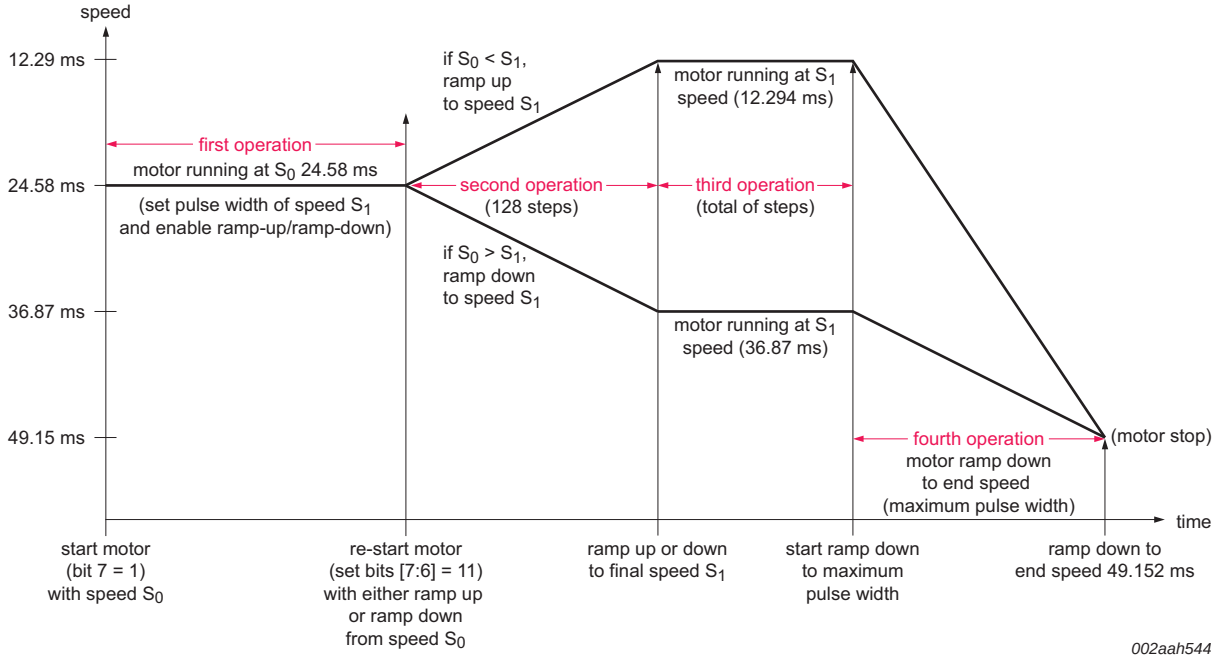


Fig 14. Re-start motor with ramp-up/ramp-down enable while current motor running in the final speed

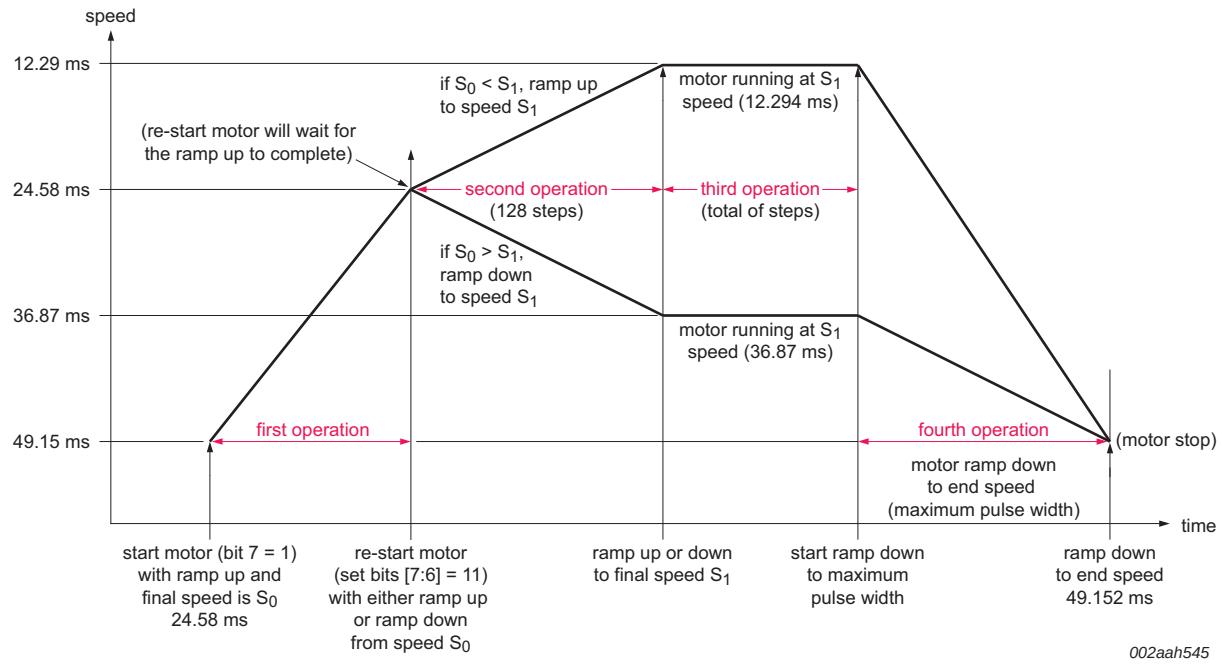


Fig 15. Re-start motor with ramp-up/ramp-down enable while current motor running in the ramp-up

7.3.21.3 MCNTL[5]: emergency stop

The ‘emergency stop’ feature is used to stop the motor immediately when this bit is set to 1. Emergency stop feature has a higher priority over ramp operation. So even if ramp operation is enabled, if the micro issues an emergency stop request, the motor stops immediately and does not ramp up or ramp down to stop. The micro should decide how the part should handle its stop request — either normal stop by setting bit 7 to 0, or emergency stop by setting this bit 5 to 1. This bit self clears after write 1 and bit 7 also clears to 0 after motor stop, it will generate an interrupt if motor stop interrupt is enabled. Emergency stop request issued when the motor is already in the stopped state is ignored.

7.3.21.4 MCNTL[4]: enable/disable START (bit 7) ignore caused by P0 state

The ‘START (bit 7) ignore caused by P0 state’ feature is designed to control and monitor motor home-position based on P0 input state. In normal operation, this bit is set to 0 to disable this function.

When the motor is in STOP condition (START bit 7 = 0), the microcontroller can set bit 7 to 1 and bit[4:3] = 10 to control motor operation in the following order:

1. If the P0 input state is LOW, then START bit 7 is ignored (motor is non-operational and in right position).
2. If P0 input state is HIGH, then motor is started until the P0 input state is detected as LOW (motor is back to right position).

When the motor is in STOP condition (START bit 7 = 0), the microcontroller can set bit 7 to 1 and bit[4:3] = 11 to control motor operation in the following order:

1. If the P0 input state is HIGH, then START bit 7 is ignored (motor is non-operational and in right position).

2. If P0 input state is LOW, then motor is started until the P0 input state is detected as HIGH (motor is back to right position).

User can periodically send this command to keep motor in home-position without polling the P0 input state.

7.3.21.5 MCNTL[3]: P0 polarity setting for START (bit 7) ignore

This bit is used to set P0 input state either LOW or HIGH for detection when bit 4 is set to 1 to enable the 'START (bit 7) ignore caused by P0 state' feature and P0 is configured as input in IO_CFG register.

The input filter for P0 input state change detection can be enabled to suppress a spike or noise in the range of 500 μs to 10 ms as shown in the bit[6:4] in INTMODE register. The input filter for P0 will be disabled if bit[6:4] is set to '00' in INTMODE register.

7.3.21.6 MCNTL[1:0]: clockwise (CW) / counter-clockwise (CCW)

These two bits are used to program the direction of the motor for current operation. The number of clockwise or counter-clockwise rotation steps are defined in CWSCOUNT or CCWSCOUNT registers and the number of repeat operations (from 1 to 255 or continuously) is defined in the PMA register. Number of steps should be non zero in the direction of operation (CW or CCW). In auto/interrupt based reversal modes (CW and CCW), the number of steps in both directions should be non zero. If this condition is not satisfied, the motor does not start. Options 10 and 11 are called auto reversal modes (to differentiate it from interrupt based reversal). In these modes, the motor starts rotating in one direction and after completing the required steps reverses the direction of rotation. If continuous mode of operation is programmed in PMA register (bits [7:0] = 00h) with auto reversal, then the motor keeps repeating the operation continuously until the micro issues a stop request or if an interrupt happens and the motor is programmed to stop on that interrupt.

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

Table 31. SUBADR1 to SUBADR3 - I²C-bus subaddress registers 1 to 3 (addresses 1Bh, 1Ch 1Dh) bit description

*Legend: * default value.*

Address	Register	Bit	Symbol	Access	Value	Description
1Bh	SUBADR1	7:1	A1[7:1]	R/W	1110 001*	I²C-bus subaddress 1
		0	A1[0]	R only	0*	reserved
1Ch	SUBADR2	7:1	A2[7:1]	R/W	1110 010*	I²C-bus subaddress 2
		0	A2[0]	R only	0*	reserved
1Dh	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.

7.3.23 ALLCALLADR — All Call I²C-bus address

Table 32. ALLCALLADR - All Call I²C-bus address register (address 1Eh) bit description

*Legend: * default value.*

Address	Register	Bit	Symbol	Access	Value	Description
1Eh	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 PCA9629As 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.

7.3.24 STEPCOUNT[0:3] — Step counter registers

Table 33. STEPCOUNT0, STEPCOUNT1, STEPCOUNT2, STEPCOUNT3 - Step counter registers (addresses 1Fh, 20h, 21h, 22h) bit description

*Legend: * default value.*

Address	Register	Bit	Access	Value	Description
1Fh	STEPCOUNT0	7:0	R only	00h*	step counter byte [7:0] (clear after read)
20h	STEPCOUNT1	7:0	R only	00h*	step counter byte [15:8] (clear after read)
21h	STEPCOUNT2	7:0	R only	00h*	step counter byte [23:16] (clear after read)
22h	STEPCOUNT3	7:0	R only	00h*	step counter byte [31:24] (clear after read)

This 32-bit counter is designed to continuously count total number of step pulses that drive the motor coils from output ports OUT0 to OUT3. This 32-bit step counter will be cleared after they are read, overflow, power-on reset, or hardware/software reset.

7.4 Motor coil excitation

Initially, after a power-up of the device, when the motor is started for the first time, the first coil that is energized is OUT0 (if the motor is turning in the clockwise direction), or OUT3 (if the motor is turning in the counter-clockwise direction). This very first step (after a power-up) is not counted towards the number steps the motor is required to move (it is the reference step). All subsequent steps are all counted. This applies only for the very first time the motor is started after the device is powered up.

For all subsequent starting of the motor, the first coil that is energized is the same coil where it had stopped. For example, consider the motor running in clockwise direction in the one-phase drive mode. If the last coil that was energized before the motor stopped was OUT2, then when the motor is started again OUT2 is energized first and after the pulse width time elapses the next coil in sequence, that is, OUT3 is energized.

7.5 Power-on reset

When power is applied to V_{DD}, an internal Power-On Reset (POR) holds the PCA9629A in a reset condition until V_{DD} has reached V_{POR}. At that point, the reset condition is released and the PCA9629A registers and state machine initialize to their default states. The power-on reset typically completes the reset and enables the part by the time the power supply is above V_{POR}. However, when it is required to reset the part by lowering the power supply, it is necessary to lower it below 2 V typical.

Remark: The system level reset pulse should be > 4 μs for the chip to guarantee reset condition.

7.6 $\overline{\text{RESET}}$ input

A reset can be accomplished by holding the $\overline{\text{RESET}}$ pin LOW for a minimum of $t_{w(rst)}$. The PCA9629A registers and I²C-bus state machine are held in their default state until the $\overline{\text{RESET}}$ input is once again HIGH. The $\overline{\text{RESET}}$ input has a 200 kΩ internal pull-up to V_{DD} pin.

The maximum wait time after RESET pin is released is 1 ms (typical).

7.7 Software reset

The Software Reset Call allows all the devices in the I²C-bus to be reset to the power-up state value through a specific formatted I²C-bus command. To be performed correctly, it implies that the I²C-bus is functional and that there is no device hanging the bus.

The maximum wait time after software reset is 1 ms (typical).

The SWRST Call function is defined as the following:

1. A START command is sent by the I²C-bus master.
2. The reserved General Call I²C-bus address '0000 000' with the R/$\overline{W}$ bit set to '0' (write) is sent by the I²C-bus master.
3. The PCA9629A device(s) acknowledge(s) after seeing the General Call address '0000 0000' (00h) only. If the R/$\overline{W}$ bit is set to '1' (read), no acknowledge is returned to the I²C-bus master.
4. Once the General Call address has been sent and acknowledged, the master sends one byte. The value of the byte must be equal to 06h. The PCA9629A acknowledges this value only. If the byte is not equal to 06h, the PCA9629A does not acknowledge it. If more than one byte of data is sent, the PCA9629A does not acknowledge anymore.
5. Once the right byte has been sent and correctly acknowledged, the master sends a STOP condition to end the software reset sequence: the PCA9629A then resets to the default value (power-up value) and is ready to be addressed again within the specified bus free time. If the master sends a Repeated START instead, no reset is performed. The I²C-bus master must interpret a non-acknowledge from the PCA9629A (at any time) as a 'Software Reset Abort'. The PCA9629A does not initiate a software reset.

7.8 Interrupt output

The open-drain active LOW interrupt is activated by the following three mechanisms:

- **Watchdog timer:** If the watchdog timer is enabled and the timer times out, then an interrupt is generated and the watchdog interrupt flag bit [5] is set in the interrupt status register (INTSTAT).
- **Motor stop:** If the motor stop interrupt is enabled in the mask interrupt register (MSK) and bit [7] in MCNTL register changes state from 1 to 0, then an interrupt is generated and interrupt flag bit [4] is set in the interrupt status register (INTSTAT).
- **GPIOs:** One or more of pins P0 to P3 can generate an interrupt if the following conditions are met:
  - The pin is configured as an input in the I/O configuration register (IO_CFG).
  - The interrupt from that pin is enabled in the mask interrupt register (MSK).
  - The pin's state change (rising edge or falling edge) is programmed to generate an interrupt in the interrupt mode register (INTMODE).

The interrupt $\overline{INT}$ pin output can be enabled or disabled using MODE register bit [5] (0 = enable; 1 = disable). The interrupt flag bit is set in the INTSTAT register when one of the interrupts is generated from P0 to P3, motor stops or watchdog timer time-out.

Remark: If the state of the pin does not match the contents of the Input port register, changing an I/O from an output to an input may cause a false interrupt to occur.

7.9 Phase sequence generator

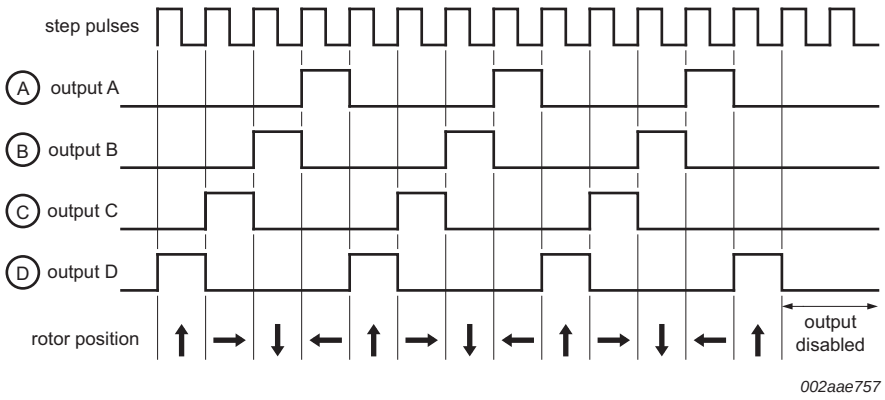
The PCA9629A phase sequence generator uses the on-chip oscillator and control logic to generate logic waveforms needed to support the following three types of stepper motor drive formats:

- One-phase drive, also called ‘wave drive’
- Two-phase drive
- Half-step drive

These logic level outputs are used to drive high current power driver stages to provide required drive current to the stepper motor coils.

7.9.1 One-phase drive (wave drive)

In one-phase drive method, only one winding is energized at any given time. The advantage of wave drive mode is its simplicity. The disadvantage of wave drive mode is that in the unipolar wound motor only 25 %, and in the bipolar motor only 50 % of the total motor winding are used at any given time. This means that maximum torque output from the motor is not made available. Since only one winding is energized, holding torque and working torque are reduced by 30 %. This can, within limits, be compensated by increasing supply voltage. The advantage of this form of drive is higher efficiency, but at the cost of reduced step accuracy.



Number of steps shown = 4 for simplicity.

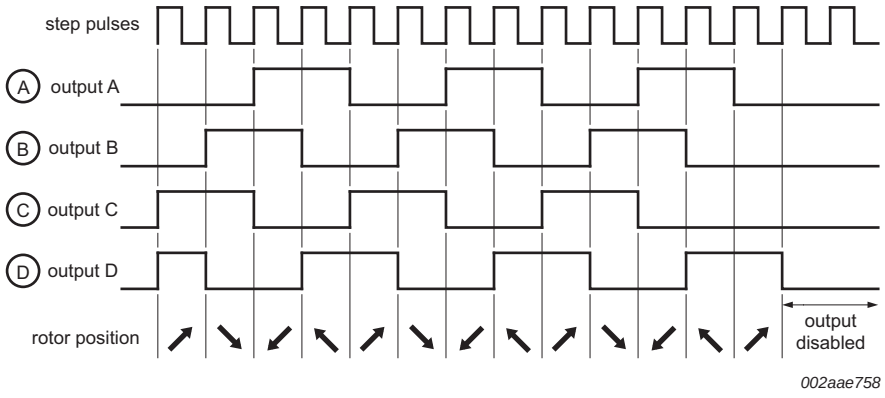
Fig 16. Wave drive step sequence waveforms

Table 34. Logic output sequence for wave drive

Winding	Step							
	1	2	3	4	5	6	7	8
Winding D	1	0	0	0	1	0	0	0
Winding C	0	1	0	0	0	1	0	0
Winding B	0	0	1	0	0	0	1	0
Winding A	0	0	0	1	0	0	0	1

7.9.2 Two-phase drive

In two-phase drive method, two windings are energized at any given time. In case of two-phase drive, the torque output of the unipolar wound motor is lower than the bipolar motor (for motors with the same winding parameters) since the unipolar motor uses only 50 % of the available winding, while the bipolar motor uses the entire winding.



Number of steps shown = 4 for simplicity.

Fig 17. Two-phase drive step sequence waveforms

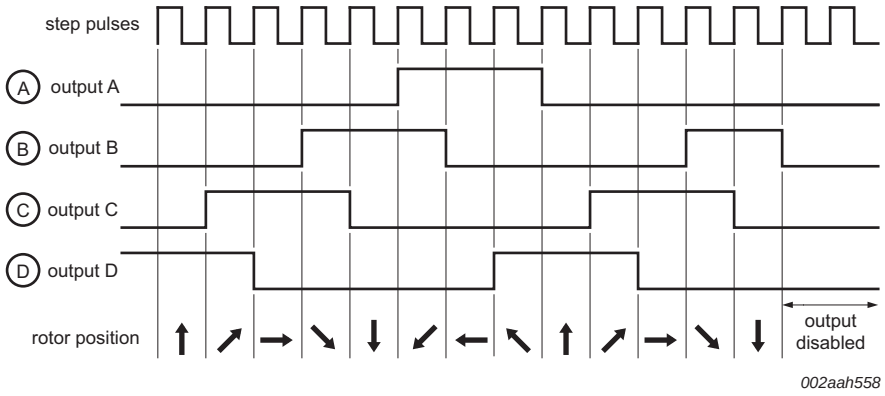
Table 35. Logic output sequence for two-phase drive

Winding	Step							
	1	2	3	4	5	6	7	8
Winding D	1	0	0	1	1	0	0	1
Winding C	1	1	0	0	1	1	0	0
Winding B	0	1	1	0	0	1	1	0
Winding A	0	0	1	1	0	0	1	1

7.9.3 Half-step drive (one-phase and two-phase on)

'Half-step drive' combines both wave and two-phase (one-phase and two-phase on) drive modes. This results in angular movements that are half of those in 1- or 2-phases-on drive modes. Half-stepping can reduce a phenomenon referred to as resonance, which can be experienced in 1- or 2-phases-on drive modes.

As the name implies, in this mode it is possible to step a motor in a half-step sequence, thus producing half steps, for example 3.75° steps from a 7.5° motor. A possible drawback for some applications is that the holding torque is alternately strong and weak on successive motor steps. This is because on full steps only one phase winding is energized, while on the half-steps two stator windings are energized. Also, because current and flux paths differ on alternate steps, accuracy is worse than when full stepping.



Four step stepper motor run with half-step waveforms increases the number of steps to eight.

Fig 18. Half-step drive sequence waveforms

Table 36. Logic output sequence for half-step drive

Winding	Step							
	1	2	3	4	5	6	7	8
Winding D	1	1	0	0	0	0	0	1
Winding C	0	1	1	1	0	0	0	0
Winding B	0	0	0	1	1	1	0	0
Winding A	0	0	0	0	0	1	1	1

8. Characteristics of the I²C-bus

The I²C-bus is for two-way, two-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

8.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals (see Figure 19).

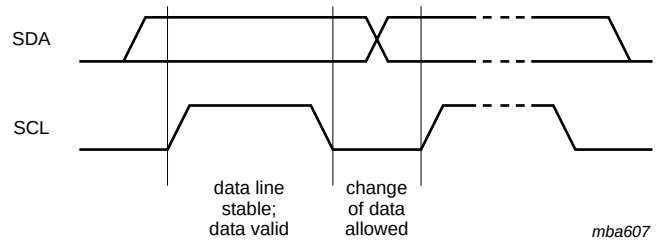


Fig 19. Bit transfer

8.1.1 START and STOP conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line while the clock is HIGH is defined as the START condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the STOP condition (P) (see Figure 20).

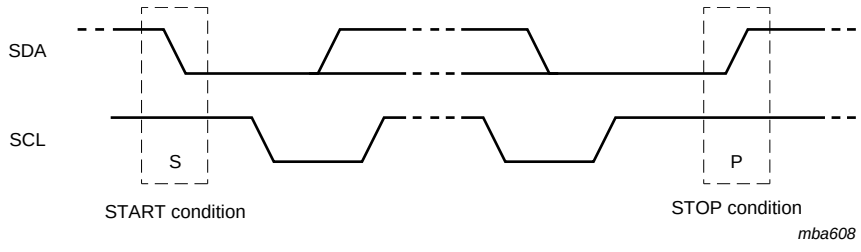


Fig 20. Definition of START and STOP conditions

8.2 System configuration

A device generating a message is a ‘transmitter’; a device receiving is the ‘receiver’. The device that controls the message is the ‘master’ and the devices which are controlled by the master are the ‘slaves’ (see Figure 21).

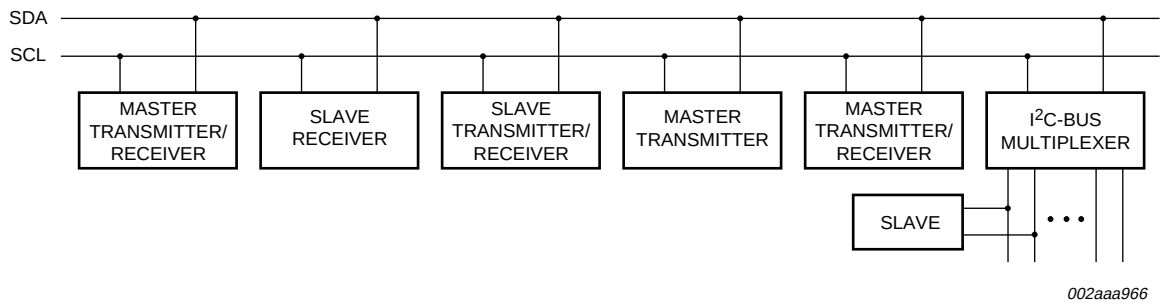


Fig 21. System configuration

8.3 Acknowledge

The number of data bytes transferred between the START and the STOP conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter, whereas the master generates an extra acknowledge related clock pulse.

A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse; set-up time and hold time must be taken into account.

A master receiver must signal an end of data to the transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this event, the transmitter must leave the data line HIGH to enable the master to generate a STOP condition.

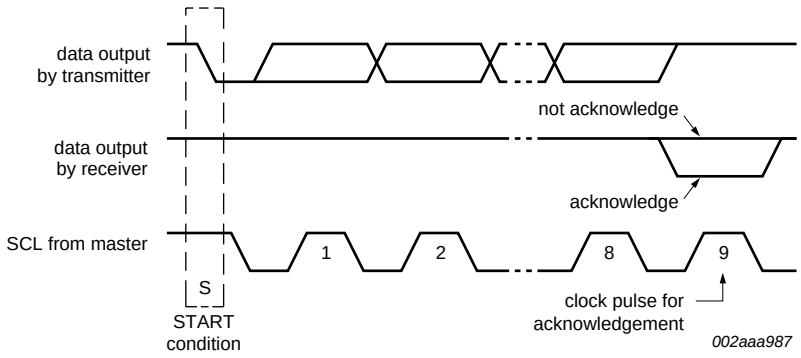


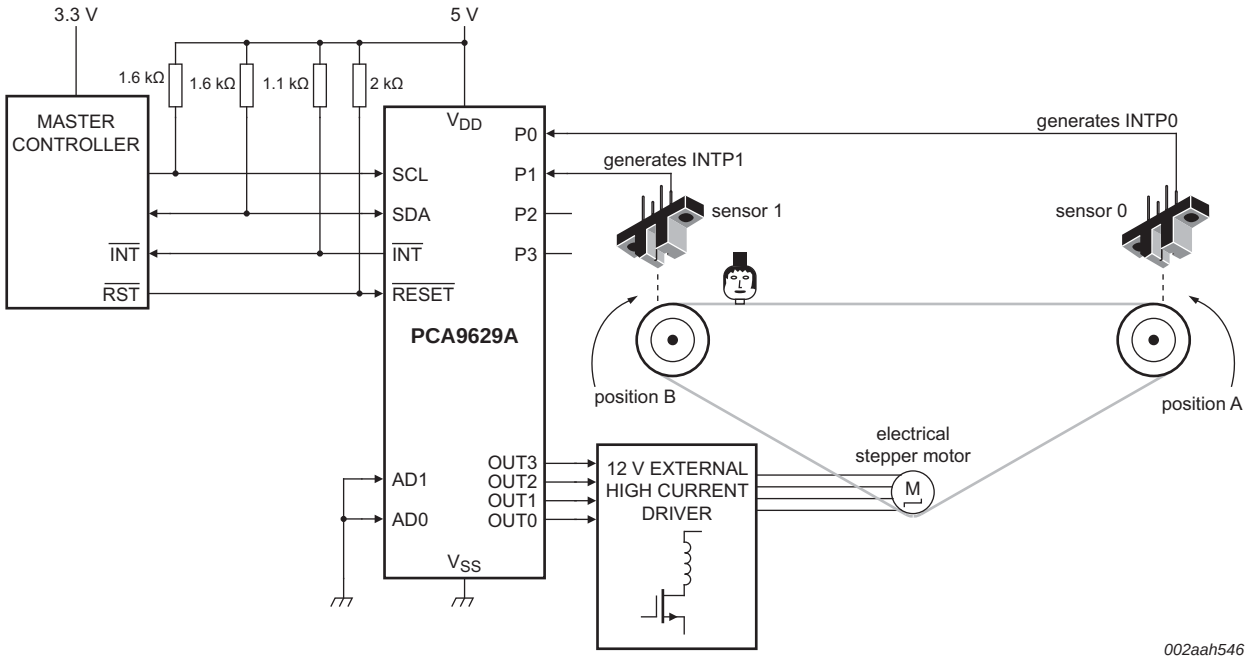
Fig 22. Acknowledgement on the I²C-bus

9. Bus transactions

Data is transmitted to the PCA9629A registers using ‘Write Byte’ transfers.

Data is read from the PCA9629A registers using ‘Read Byte’ transfers.

10. Application design-in information



Device address configured as 0100 0000b for this example.

Fig 23. Typical application

10.1 Stepper motor coil driver considerations

When choosing a motor and coil driver circuit for an application, it is necessary to choose the coil driver such that the minimum expected drive strength of the coil driver over the anticipated operating conditions exceeds the minimum coil current in the application. For the NMOS FETs, the gate voltage affects the FET drive strength, so it is necessary to evaluate the FET with its gate at the minimum V_{DD} planned for the PCA9629A application, because the PCA9629A cannot drive the gate higher than the V_{DD}.

For example, in most applications a 5 V power supply would have a specification like 5 V ± 10 % or 5 V ± 20 %, so it would be necessary to verify that the ON-resistance or current sinking capability of the FET with a gate voltage of 4.75 V or 4.5 V, whichever applies, is capable of sinking all of the current that the motor might require. Since FETs present a capacitive load to the outputs of the PCA9629A, the output asymptotically approaches the V_{DD} of the part, so eventually the full V_{DD} appears at the output. However, for Darlington bipolar coil drivers the input current represents a static current load that reduces the V_{OH}. So depending upon the input current of the Darlington bipolar coil driver, the PCA9629A output voltage will always be less than V_{DD}. This in turn reduces the input current and also reduces the available drive current from the Darlington bipolar coil driver, so the lowest gain for the driver and the input current gain product must be considered in verifying that the maximum motor current can be sunk by the driver.

10.2 Considerations when using GPIO pins P0 to P3 as inputs

For proper operation of GPIO pins as inputs, the signals at the inputs must be free from any glitches or noise. The signals must be logic level inputs.

For example, outputs from sensors must provide logic level signals at the input pins of PCA9629A. This may require signal conditioning at the outputs of sensors. Another example is when using P0 to P3 for key switch sensing. The inputs of PCA9629A do not provide key de-bouncing on P2 and P3 inputs. This is external to PCA9629A and is user-defined and supplied.

10.3 Priority of ramp control, interrupt-based control, loop delay and emergency stop

During ramp-up and ramp-down phases of operation, the interrupt-based controls do not affect the motor run. Interrupts that occur during ramp-up or ramp-down are ignored. Once the ramp-up operation is finished (when the motor is running at the final speed), then the interrupts that occur are acted upon. A stop request from the microcontroller (writing 0 to MCNTL[7], 1 to MCNTL[5] or watchdog stop motor mode) is the only event that affects the motor operation during ramp-up and ramp-down.

During ramp-up, the microcontroller can issue a stop request. The following sequence of events takes place in the given order:

1. If emergency stop is enabled (MCNTL[5] = 1), the motor stops immediately (even if ramp-down is enabled) or if watchdog stop motor mode is the same as emergency stop except the output phases are all zeros; Priority 1.
2. If normal stop is enabled (MCNTL[7] = 0) and ramp-down is enabled, then the motor starts to ramp down to a stop; Priority 2.
3. If normal stop is enabled (MCNTL[7] = 0) and ramp-down is disabled, then motor stops immediately; Priority 3.

During ramp-down, the microcontroller can issue a stop request. The following sequence of events takes place in the given order:

1. If emergency stop is enabled (MCNTL[5] = 1), the motor stops immediately (it does not finish ramping down) or if watchdog stop motor mode is the same as emergency stop except the output phases are all zeros; Priority 1.
2. If normal stop is enabled (MCNTL[7] = 0) and ramp-down is enabled, then the motor continues to ramp down to a stop; Priority 2.

In the duration between end of ramp-up and beginning of ramp-down, the interrupt-based controls (if enabled) can affect the operation of the motor. In this region, Section 7.3.8.1 gives the priority of events when both interrupt-based control and ramp control are enabled together. Consider the following example (the motor is programmed to reverse rotation on an interrupt):

- Motor programmed for CW rotations; ramp-up and ramp-down enabled; reverse rotation on interrupt P0/P1.
- When motor is started, it starts ramping up and when ramp-up is completed it rotates at the final speed.

- If interrupt P0 happens, then it reverses rotation right away and start rotating in the CCW direction for the specified number of steps.
- Before the specified number of steps is completed in the CCW direction, if interrupt P1 happens, then it again reverses its rotation right away and start rotating in the CW direction for the specified number of steps.
- If no other interrupt happens, the motor finishes executing the specified number of steps in the CW direction and then starts to ramp down.

In the above example, if extra steps are enabled for interrupts P0 and P1, then when the interrupts happen the motor executes the extra steps in the current direction of rotation and then reverses its direction.

11. Limiting values

Table 37. Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD}	supply voltage		–0.5	+6.0	V
$V_{I/O}$	voltage on an input/output pin		V_{SS} – 0.5	5.5	V
$I_{I/O}$	input/output current	Pn, OUTn, $\overline{INT}$, SCL, SDA	-	±50	mA
I_I	input current		-	±20	mA
I_{SS}	ground supply current		-	210	mA
P_{tot}	total power dissipation		-	400	mW
T_{stg}	storage temperature		–65	+150	°C
T_{amb}	ambient temperature		–40	+85	°C

12. Static characteristics

Table 38. Static characteristics

V_{DD} = 4.5 V to 5.5 V; V_{SS} = 0 V; T_{amb} = –40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Supply						
V_{DD}	supply voltage		4.5	-	5.5	V
I_{DD}	supply current	operating mode; no load; f_{SCL} = 1 MHz; V_{DD} = 5.5 V	-	6	10	mA
I_{stb}	standby current	no load; f_{SCL} = 0 kHz; MODE[6] = 1; OSC off; $V_I = V_{DD}$ or V_{SS}; V_{DD} = 5.5 V	-	600	800	µA
V_{POR}	power-on reset voltage	no load; $V_I = V_{DD}$ or V_{SS}	-	2.3	-	V
V_{PDR}	power-down reset voltage	no load; $V_I = V_{DD}$ or V_{SS} [1]	-	2.0	-	V
Input SCL; input/output SDA						
V_{IL}	LOW-level input voltage		–0.5	-	$+0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	-	5.5	V
I_{OL}	LOW-level output current	V_{OL} = 0.4 V; V_{DD} = 5.0 V	30	40	-	mA
I_L	leakage current	$V_I = V_{DD}$ or V_{SS}	–1	-	+1	µA
C_i	input capacitance	$V_I = V_{SS}$	-	6	-	pF

Table 38. Static characteristics *...continued*

V_{DD} = 4.5 V to 5.5 V; V_{SS} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
OUT0 to OUT3 outputs						
I_{OL}	LOW-level output current	V_{OL} = 0.4 V; V_{DD} = 4.5 V [2]	20	22	-	mA
		V_{OL} = 0.5 V; V_{DD} = 4.5 V [2]	25	28	-	mA
$I_{OL(tot)}$	total LOW-level output current	V_{OL} = 0.5 V; V_{DD} = 4.5 V [2]	-	-	120	mA
V_{OH}	HIGH-level output voltage	I_{OH} = −10 mA; V_{DD} = 4.5 V [3]	4.0	-	-	V
P0 to P3 I/Os						
I_{OL}	LOW-level output current	V_{OL} = 0.5 V; V_{DD} = 4.5 V [2]	25	28	-	mA
$I_{OL(tot)}$	total LOW-level output current	V_{OL} = 0.5 V; V_{DD} = 4.5 V [2]	-	-	120	mA
V_{OH}	HIGH-level output voltage	I_{OH} = −10 mA; V_{DD} = 4.5 V [3]	4.0	-	-	V
I_{OZ}	OFF-state output current	3-state; V_{OH} = V_{DD} or V_{SS}	−10	-	+10	μA
C_{io}	input/output capacitance	3-state pins as inputs	-	5	-	pF
Address inputs						
V_{IL}	LOW-level input voltage		−0.5	-	+0.3V_{DD}	V
V_{IH}	HIGH-level input voltage		0.7V_{DD}	-	5.5	V
I_{LI}	input leakage current		−1	-	+1	μA
C_i	input capacitance		-	3	-	pF
RESET input						
V_{IL}	LOW-level input voltage		−0.5	-	+0.3V_{DD}	V
V_{IH}	HIGH-level input voltage		0.7V_{DD}	-	5.5	V
I_{LI}	input leakage current		−1	-	+1	μA
C_i	input capacitance		-	3	-	pF
I_{LIL}	LOW-level input leakage current	V_I = V_{SS}	−7	-	−45	μA
INT output						
I_{OL}	LOW-level output current	V_{OL} = 0.5 V; V_{DD} = 4.5 V	24	28	-	mA
I_{OH}	HIGH-level output current	open-drain; V_{OH} = V_{DD}	−10	-	+10	μA
C_o	output capacitance		-	7	-	pF

[1] In order to reset part, V_{DD} must be lowered to 1.4 V.

[2] Each bit must be limited to a maximum of 25 mA and the total package limited to 210 mA due to internal busing limits.

[3] For I_{OH} = −25 mA, the minimum V_{OH} = V_{DD} − 0.7 V with V_{DD} = 4.5 V to 5.5 V.

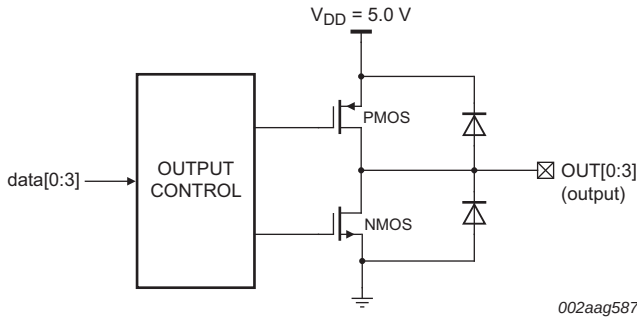


Fig 24. OUT[0:3] output equivalent circuit

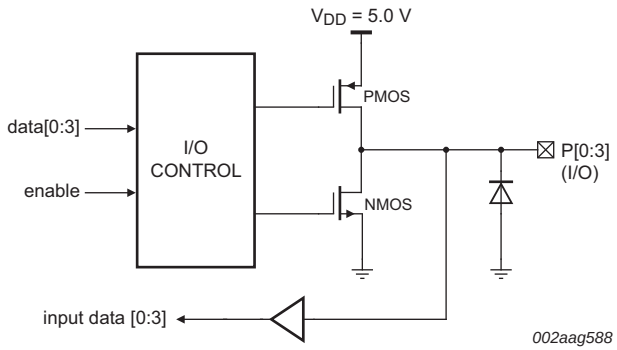


Fig 25. P[0:3] input/output equivalent circuit

13. Dynamic characteristics

Table 39. Dynamic characteristics

V_{DD} = 4.5 V to 5.5 V; V_{SS} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.
Oscillator frequency = 1 MHz ± 3 % at −40 °C to +85 °C (see Figure 28).

Symbol	Parameter	Conditions	Standard-mode I²C-bus		Fast-mode I²C-bus		Fast-mode Plus I²C-bus		Unit
			Min	Max	Min	Max	Min	Max	
f_{SCL}	SCL clock frequency	[1]	0	100	0	400	0	1000	kHz
t_{BUF}	bus free time between a STOP and START condition		4.7	-	1.3	-	0.5	-	µs
$t_{HD;STA}$	hold time (repeated) START condition		4.0	-	0.6	-	0.26	-	µs
$t_{SU;STA}$	set-up time for a repeated START condition		4.7	-	0.6	-	0.26	-	µs
$t_{SU;STO}$	set-up time for STOP condition		4.0	-	0.6	-	0.26	-	µs
$t_{HD;DAT}$	data hold time		0	-	0	-	0	-	ns
$t_{VD;ACK}$	data valid acknowledge time	[2]	0.3	3.45	0.1	0.9	0.05	0.45	µs
$t_{VD;DAT}$	data valid time	[3]	0.3	3.45	0.1	0.9	0.05	0.45	µs
$t_{SU;DAT}$	data set-up time		250	-	100	-	50	-	ns
t_{LOW}	LOW period of the SCL clock		4.7	-	1.3	-	0.5	-	µs
t_{HIGH}	HIGH period of the SCL clock		4.0	-	0.6	-	0.26	-	µs
t_f	fall time of both SDA and SCL signals	[4][5]	-	300	20 + 0.1C_b[6]	300	-	120	ns
t_r	rise time of both SDA and SCL signals		-	1000	20 + 0.1C_b[6]	300	-	120	ns
t_{SP}	pulse width of spikes that must be suppressed by the input filter	[7]	-	50	-	50	-	50	ns
$t_{d(o)}$	output delay time	interrupt based motor control latency [8]	5.7	7.4	5.7	7.4	5.7	7.4	µs
RESET									
$t_{w(rst)}$	reset pulse width		2.3	-	2.3	-	2.3	-	µs
$t_{rec(rst)}$	reset recovery time		1.2	-	1.2	-	1.2	-	ms

[1] Minimum SCL clock frequency is limited by the bus time-out feature, which resets the serial bus interface if either SDA or SCL is held LOW for a minimum of 25 ms. Disable bus time-out feature for DC operation.

[2] $t_{VD;ACK}$ = time for acknowledgement signal from SCL LOW to SDA (out) LOW.

[3] $t_{VD;DAT}$ = minimum time for SDA data out to be valid following SCL LOW.

[4] In order to bridge the undefined region of the SCL falling edge, a master device must internally provide a hold time of at least 300 ns for the SDA signal (refer to the V_{IL} of the SCL signal).

[5] The maximum t_f for the SDA and SCL bus lines is specified at 300 ns. The maximum fall time (t_f) for the SDA output stage is specified at 250 ns. This allows series protection resistors to be connected between the SDA and the SCL pins and the SDA/SCL bus lines without exceeding the maximum specified t_f.

[6] C_b = total capacitance of one bus line in pF.

[7] Input filters on the SDA and SCL inputs suppress noise spikes less than 50 ns.

[8] The time delay from one of the P[1:0] inputs edge changes to the motor control outputs OUT[3:0] change. Typical value = 6.5 µs.

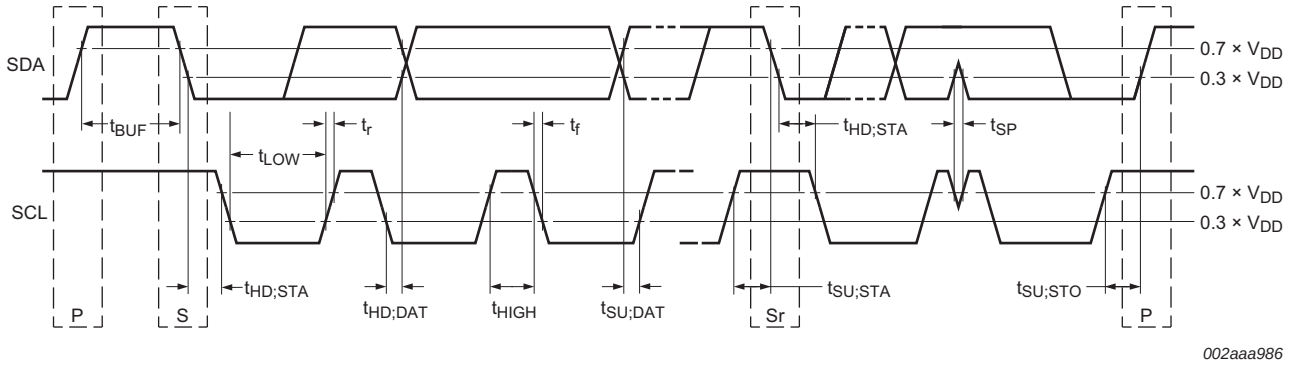


Fig 26. Definition of timing

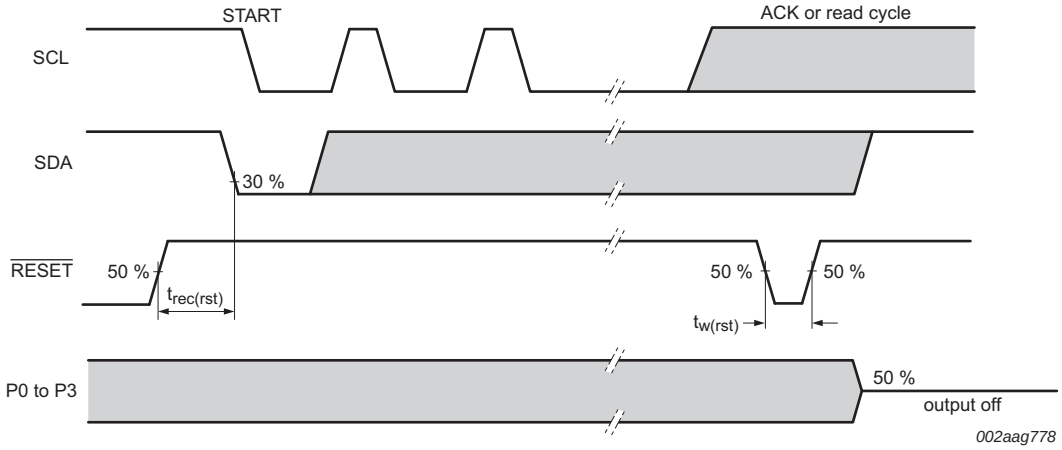
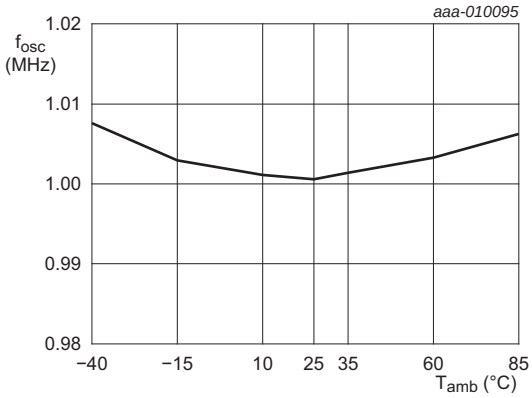


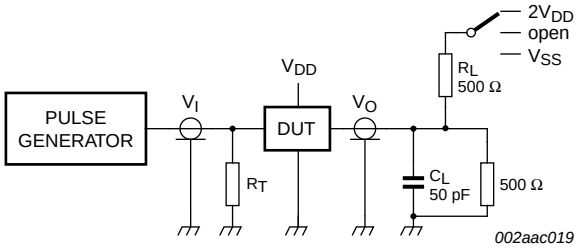
Fig 27. Reset timing



V_{DD} = 5 V ± 10 %

Fig 28. Typical oscillator frequency versus temperature

14. Test information

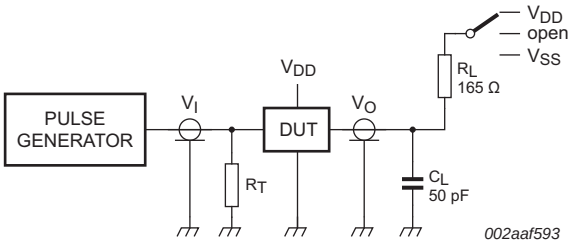


R_L = load resistance.

C_L = load capacitance includes jig and probe capacitance.

R_T = termination resistance should be equal to the output impedance Z_o of the pulse generators.

Fig 29. Test circuitry for switching times for GPIO pins P0 to P3



R_L for SDA and SCL = 165 Ω (30 mA or less current).

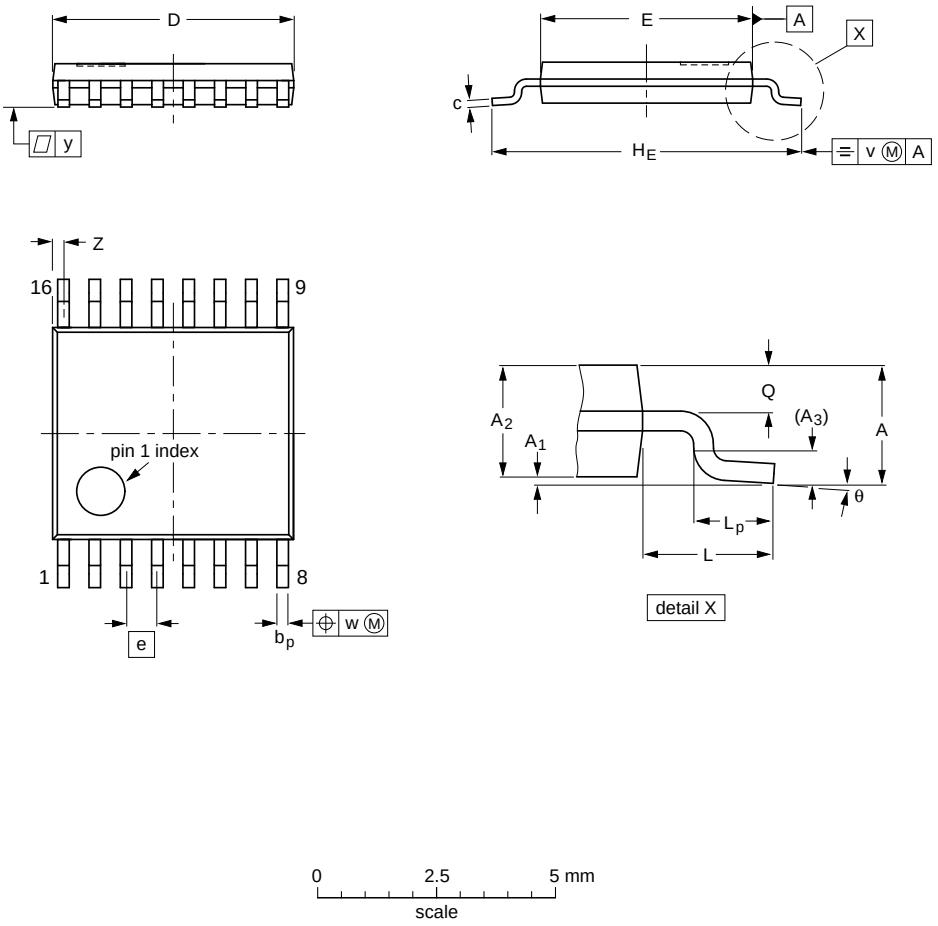
C_L = load capacitance includes jig and probe capacitance.

R_T = termination resistance should be equal to the output impedance Z_o of the pulse generators.

Fig 30. Test circuitry for switching times for SDA and SCL

15. Package outline

TSSOP16: plastic thin shrink small outline package; 16 leads; body width 4.4 mm **SOT403-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	D (1)	E (2)	e	H_E	L	L_p	Q	v	w	y	Z (1)	θ
mm	1.1	0.15 0.05	0.95 0.80	0.25	0.30 0.19	0.2 0.1	5.1 4.9	4.5 4.3	0.65	6.6 6.2	1	0.75 0.50	0.4 0.3	0.2	0.13	0.1	0.40 0.06	8^o 0^o

Notes

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.
2. Plastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT403-1		MO-153				~~99-12-27~~ 03-02-18

Fig 31. Package outline SOT403-1 (TSSOP16)

16. Handling information

All input and output pins are protected against ElectroStatic Discharge (ESD) under normal handling. When handling ensure that the appropriate precautions are taken as described in *JESD625-A* or equivalent standards.

17. Soldering of SMD packages

This text provides a very brief insight into a complex technology. A more in-depth account of soldering ICs can be found in Application Note *AN10365 "Surface mount reflow soldering description"*.

17.1 Introduction to soldering

Soldering is one of the most common methods through which packages are attached to Printed Circuit Boards (PCBs), to form electrical circuits. The soldered joint provides both the mechanical and the electrical connection. There is no single soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and Surface Mount Devices (SMDs) are mixed on one printed wiring board; however, it is not suitable for fine pitch SMDs. Reflow soldering is ideal for the small pitches and high densities that come with increased miniaturization.

17.2 Wave and reflow soldering

Wave soldering is a joining technology in which the joints are made by solder coming from a standing wave of liquid solder. The wave soldering process is suitable for the following:

- Through-hole components
- Leaded or leadless SMDs, which are glued to the surface of the printed circuit board

Not all SMDs can be wave soldered. Packages with solder balls, and some leadless packages which have solder lands underneath the body, cannot be wave soldered. Also, leaded SMDs with leads having a pitch smaller than ~0.6 mm cannot be wave soldered, due to an increased probability of bridging.

The reflow soldering process involves applying solder paste to a board, followed by component placement and exposure to a temperature profile. Leaded packages, packages with solder balls, and leadless packages are all reflow solderable.

Key characteristics in both wave and reflow soldering are:

- Board specifications, including the board finish, solder masks and vias
- Package footprints, including solder thieves and orientation
- The moisture sensitivity level of the packages
- Package placement
- Inspection and repair
- Lead-free soldering versus SnPb soldering

17.3 Wave soldering

Key characteristics in wave soldering are:

- Process issues, such as application of adhesive and flux, clinching of leads, board transport, the solder wave parameters, and the time during which components are exposed to the wave
- Solder bath specifications, including temperature and impurities

17.4 Reflow soldering

Key characteristics in reflow soldering are:

- Lead-free versus SnPb soldering; note that a lead-free reflow process usually leads to higher minimum peak temperatures (see Figure 32) than a SnPb process, thus reducing the process window
- Solder paste printing issues including smearing, release, and adjusting the process window for a mix of large and small components on one board
- Reflow temperature profile; this profile includes preheat, reflow (in which the board is heated to the peak temperature) and cooling down. It is imperative that the peak temperature is high enough for the solder to make reliable solder joints (a solder paste characteristic). In addition, the peak temperature must be low enough that the packages and/or boards are not damaged. The peak temperature of the package depends on package thickness and volume and is classified in accordance with Table 40 and 41

Table 40. SnPb eutectic process (from J-STD-020D)

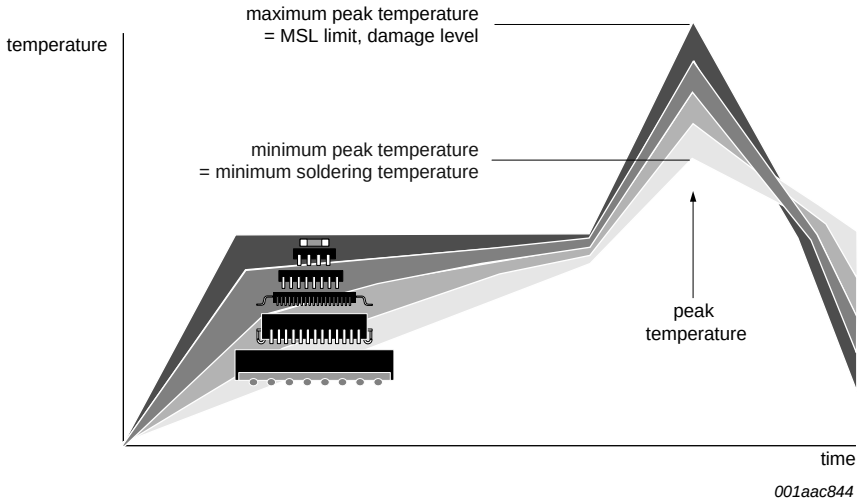
Package thickness (mm)	Package reflow temperature (°C)	
	Volume (mm³)	
	< 350	≥ 350
< 2.5	235	220
≥ 2.5	220	220

Table 41. Lead-free process (from J-STD-020D)

Package thickness (mm)	Package reflow temperature (°C)		
	Volume (mm³)		
	< 350	350 to 2000	> 2000
< 1.6	260	260	260
1.6 to 2.5	260	250	245
> 2.5	250	245	245

Moisture sensitivity precautions, as indicated on the packing, must be respected at all times.

Studies have shown that small packages reach higher temperatures during reflow soldering, see Figure 32.



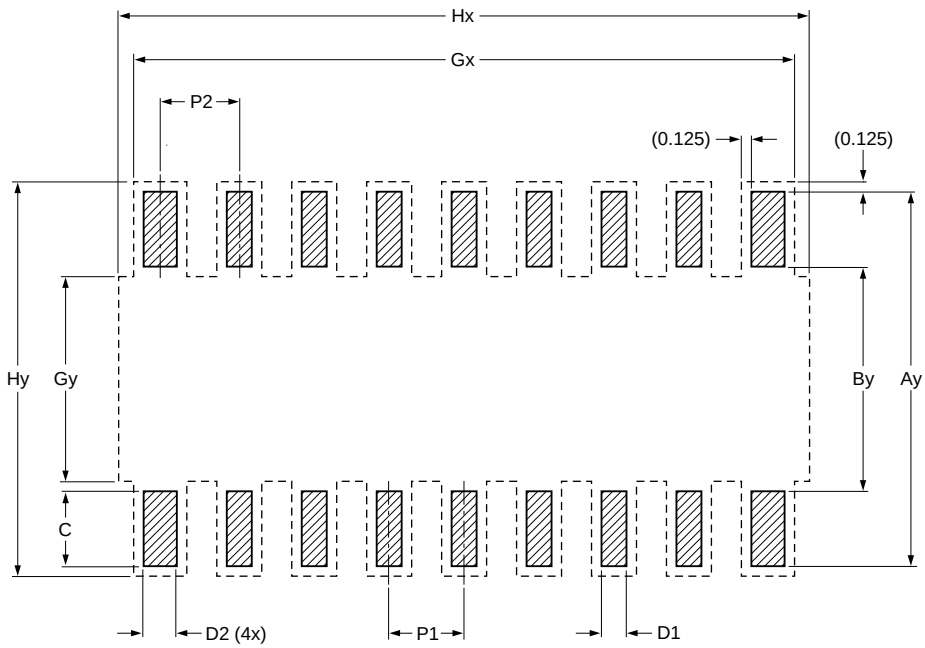
MSL: Moisture Sensitivity Level

Fig 32. Temperature profiles for large and small components

For further information on temperature profiles, refer to Application Note *AN10365 "Surface mount reflow soldering description"*.

18. Soldering: PCB footprints

Footprint information for reflow soldering of TSSOP16 package **SOT403-1**



Generic footprint pattern

Refer to the package outline drawing for actual layout

solder land

occupied area

DIMENSIONS in mm

P1	P2	Ay	By	C	D1	D2	Gx	Gy	Hx	Hy
0.650	0.750	7.200	4.500	1.350	0.400	0.600	5.600	5.300	5.800	7.450

sot403-1_fr

Fig 33. PCB footprint for SOT403-1 (TSSOP16)

19. Abbreviations

Table 42. Abbreviations

Acronym	Description
AI	Auto-Increment
CCW	Counter-ClockWise
CDM	Charged-Device Model
CMOS	Complementary Metal-Oxide Semiconductor
CPU	Central Processing Unit
CW	ClockWise
DMOS	double-Diffused Metal-Oxide Semiconductor
DUT	Device Under Test
ESD	ElectroStatic Discharge
FET	Field-Effect Transistor
Fm+	Fast-mode Plus
GPIO	General Purpose Input/Output
HBM	Human Body Model
HVAC	Heating, Venting and Air Conditioning
I/O	Input/Output
I²C-bus	Inter-Integrated Circuit bus
IC	Integrated Circuit
LED	Light Emitting Diode
LSB	Least Significant Bit
NMOS	Negative-channel Metal-Oxide Semiconductor
MSB	Most Significant Bit
PCB	Printed-Circuit Board
pps	pulses per second
PWM	Pulse Width Modulator
POR	Power-On Reset

20. Revision history

Table 43. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
PCA9629A v.2	20140321	Product data sheet	-	PCA9629A v.1
Modifications:	• Section 2 “Features and benefits”, sixth bullet item: changed from “344.8 kpps” to “333.3 kpps” (this is a correction to documentation only; no change to device)			
PCA9629A v.1	20140225	Product data sheet	-	-

21. Legal information

21.1 Data sheet status

Document status[1][2]	Product status[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term 'short data sheet' is explained in section "Definitions".

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL http://www.nxp.com.

21.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

Product specification — The information and data provided in a Product data sheet shall define the specification of the product as agreed between NXP Semiconductors and its customer, unless NXP Semiconductors and customer have explicitly agreed otherwise in writing. In no event however, shall an agreement be valid in which the NXP Semiconductors product is deemed to offer functions and qualities beyond those described in the Product data sheet.

21.3 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

21.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I²C-bus — logo is a trademark of NXP Semiconductors N.V.

22. Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

23. Contents

continued >>

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 21 March 2014
Document identifier: PCA9629A