APPLICATION NOTE

ABSTRACT

One of the more common functions of the I²C bus is to provide additional I/O port capability for a controller without increasing the number of pins on the controller package. Philips Semiconductors offers 9 different flavors of I²C I/O port expanders. The following is a selection guide that provides the designer with an overview of the similarities and important differences of selecting the best port expander for the job.

AN469
I²C I/O ports

Author: Bob Marshall

2001 Aug 15

Philips Semiconductors

PHILIPS

I²C I/O ports AN469

Author: Bob Marshall

I²C PORTS

One of the more common functions of the I²C bus is to provide additional I/O port capability for a controller without increasing the number of pins on the controller package. Philips Semiconductors offers 9 different flavors of I²C I/O port expanders. The following is a selection guide that provides the designer with an overview of the similarities and important differences of selecting the best port expander for the job. This is only a general overview and the reader is encouraged to thoroughly review the data sheets for specific information.

16 BIT I/O PORTS

One of the first decisions for a designer is to determine how many I/O ports are needed or how big an expander is required. There are 2 different devices with 16 bits of I/O expansion, the PCF8575 and the PCA9555. These devices are capable of supporting both standard-mode (100 KHz) and fast-mode (400 KHz) bus speeds. They also have the same I²C address range with capability to support 8 different addresses so you can control up to 128 I/O ports on a single I²C bus. (You can easily control even more if you use an I²C multiplexer or I²C hub device.)

The V_{DD} ranges of the parts are similar. The PCF8575 operates from 2.5 to 5.5 volts and the PCA9555 operates from 2.3 to 5.5 volts. The PCA9555, being a newer device, has the advantage when working in mixed V_{DD} systems due to its 5 volt tolerant I/Os. The maximum input voltage on the PCA9555 ports is 5.5 V over the entire voltage range. Both devices can tolerate 5 volts on the SDA and SDL lines. The PCA9555 has a slightly higher sink current of 25 mA on the I/O pins and the PCF8575 has a sink current of 20 mA. The PCF8575 requires the user to limit the total available current sink from the I/O ports to 100mA while the PCA9555 limits the user to sink a maximum of 200 mA without damage, however, the PCF8575 maintains an advantage of a lower supply current. Because of the low leakage currents on the SCL and SDA lines, the PCA9555 can meet the SMBus electrical requirements.

Another difference between the two is the I/O port structure. The PCF8575 uses a quasi-bi-directional I/O port. When the device initializes, all I/O pins default high, effectively enabling all ports to be used as inputs. An I/O pin can only be used as an input to a port that in the write mode would be active high. To use the I/O port as an input, the system should write an output high to the port. Applying a high signal to a pin programmed as an output low will cause a large IOL to flow to V_{SS} and could damage the device. When a port that is being used as an output port, it can be switched either high or low. When switched from a low to a high, there is an additional strong pull up circuit turned on during switching to allow a fast rising edge into heavily loaded outputs. This strong driver is switched off on the falling edge of SCL.

For the PCA9555, a configuration register controls the I/O structure. This register is used to program an I/O pin as either an input or output. As an output, the device can sink up to 24 mA or source up to 10 mA. The PCF8575 can only source 100 μA. Because the PCA9555 can sink and source current, applying a high signal to a pin programmed as an output low or a low signal applied to a pin programmed as an output high will cause a large I/O currents to flow. The PCA9555 also provides a polarity inversion register. The value read back from the Input Port Register can be programmed to be a true or complement of the input signal. Because of the configuration and polarity registers, there can be a slightly higher software overhead when writing to or reading from the PCA9555. The PCA9555 does allow the user to read or write one port at a time, which can actually reduce the software overhead in some applications. The PCF8575 always requires the user to read or write both the upper and lower port. (See appendix A for a more complete software discussion). Another consideration when selecting the best part to use is the package size. The PCA9555 is available in a very small TSSOP package. There are other minor differences and the designer should consult both data sheets for details of how the devices operate.

8 BIT I/O PORTS

There are two reasons that a designer may select an 8-bit I/O port over the 16-bit device. The most obvious is that the design only needs an 8-bit port. Another reason may be that an additional feature is currently only available on the 8-bit device such as a hardware reset or an on-board EEPROM. The PCF8574 is the oldest I²C port expander device. As was the case with the 16 I/O ports, there exist many similarities between the PFC8574 and the PCA 9554. Both have A-version devices, which gives the user a choice of two different base addresses. On both there are three programmable address pins so either device can support up to 16 separate 8 bit I/O ports on a single I²C system or up to 64 I/O ports without the use of a multiplexer or hub device. The PCF8574 is only designed to support the standard mode (100 KHz) while the PCA9554 can support both Standard and Fast-mode (400 kHz) operation.

The V_{DD} ranges of the parts are similar. The PCF8574 operates from 2.5 to 5.5 volts and the PCA9555 operates from 2.3 to 5.5 volts. In low power applications, the standby current for the PCA9554 is 1 μA while the PCF8574 powers down to 10 μA. The PCA9554, being a newer device, has the advantage when working in mixed V_{DD} systems. The I/O ports are 5 volt tolerant regardless of the device's operating voltage. Both devices can tolerate 5 volts on the SDA and SDL lines. The PCA9554 offers the same I/O structure as the PCA9555 with the increased flexibility and drive of a truly programmable Input/output port structure but with slightly higher software overhead in some cases. (See appendix B for a more complete software discussion). In situations where board space is at a premium, the PCA9554 offers the smallest package option being available in a TSSOP16 package.

In the cases where board space is at a minimum and the designer has an additional requirement of a serial EEPROM and an 8-bit I/O port, they can use the PTN3500. The PTN3500 has the same I²C address as the PCF8574 or PCA9554 but the chip includes a 2K byte I²C EEPROM on chip. It has 5 volt tolerant I/O but the maximum supply voltage is limited to 3.3 volts.

The PCF8574 and PTN3500 have the same quasi-bi-directional I/O port structure as the PCF8575 described earlier. The PTN3500 offers pin compatibility with the PCF8574 except for the interrupt pin. On the PTN3500, the interrupt pin is sacrificed to provide the read/write control pin for an on-board EEPROM. The PTN3501 is identical to the PTN3500 except it is available in a 20-pin TSSOP package. This allows for both the EEPROM read/write command and interrupts function. With the additional pins available, the PTN3501has 6 address and can be programmed to any one of 64 addresses.

All of the devices so far have a feature that they all have a power on reset function, which places the internal I/O ports to a default condition on power up. The PCA9556, like the PTN3500, does not have an interrupt function. The PTN3500 used the interrupt pin as a read/write control. The PCA9556 uses that pin for an active Low hardware reset function. Typically, the reset pin is pulled to V_{DD} with an external resistor so the device reset is held high in a default condition. This is especially important if the hardware reset will not automatically clear the reset signal. The I/O structure is similar to the PCA9554 except that I/O 0 is an open drain output. This allows the I/O 0 port to function as an interrupt generation port or could be used if the system wanted to use an I²C signal as a hardware reset by connecting the I/O 0 port to the reset pin. The PCA9556 is limited to 100 KHz operation however the PCA9557 is the same basic part but supports the Fast Mode I²C. Both devices point to input register on power up or external reset. If the user performs a ‘read’ operation directly after a reset, the parts will read out the value on the input register. There are some minor differences in the way these two devices acknowledge interrupts and should be reviewed before a final decision is made.

All of the devices thus far described are designed to operate over the industrial (–40 to +85 °C) temperature range. There is one more I/O port device specifically designed for use on PC motherboards and works only over the commercial (0-70 °C) temperate range. The PCA9558 contains an 8-bit I/O port, a Multiplexed/latched EEPROM register and a 2 Kbyte EEPROM. The I/O port portion of the device has limited drive capability but otherwise functions the same as the PCA9554.

Table 1 summarizes the features of the different I/O port devices. For additional information on the I²C I/O port devices described, and to get the most current data sheets, you can go to the Philips Logic web site at www.philipslogic.com. While this paper has summarized some of the more significant differences between various devices, there are other minor differences, which should be considered before final part selection. Philips applications engineers are available to answer any questions you may have. You can also email us at logicpg@philips.com

For additional information about the I²C bus including the most recent copy of the complete standard, you can go to the Philips Semiconductor web site at www.semiconductors.philips.com/i2c/

Table 1. Quick Features Guide

Part	I/O	Number of Addresses	Int	H/W Reset	V_{DD} Range	I/O Sink mA		Pin Count	Package	Max Bus Speed	I/O Type
						Bit	Total				
PCA9554/54A	8	8 or 16	Yes	No	2.3–5.5	10–24	100	16	SO, SSOP, TSSOP	400 KHz	True
PCA9555	16	8	Yes	No	2.3–5.5	10–24	200	24	SO, SSOP, TSSOP	400 KHz	True
PCA9556	8	8	No	Yes	3.0–5.5	10	80	16	TSSOP	100 KHz	True
PCA9557	8	8	No	Yes	2.3–5.5	10–24	100	16	SO, TSSOP	400 KHz	True
PCA9558	8	2	No	No	3.0–3.6	4	32	28	TSSOP	400 KHz	True
PCF8574/74A	8	8	Yes	No	2.5–6.0	20	100	16, 20	DIP, SO, SSOP	100 KHz	Quasi
PCF8575	16	8	Yes	No	2.5–5.5	20	100	24	SSOP	400 KHz	Quasi
PTN3500	8 + EEPROM	8	No	No	2.5–3.6	25	100	16	SO, TSSOP	400 KHz	Quasi
PTN3501	8 + EEPROM	64	Yes	No	2.5–3.6	25	100	20	TSSOP	400 KHz	Quasi

APPENDIX A — SOFTWARE CONSIDERATIONS IN USING A PCA9555 IN PLACE OF A PCF8575

Because the PCA9555 and the PCF9575 are pin compatible and have the same I²C address, they can often be used in the same board design, however the software to use these devices differ a bit. At power up, both devices look very similar to the target application. The PCF8575 powers up with all I/O active high and so can immediately be used as inputs. The PCA9555 also powers up with all of the I/O pins configured as inputs. How you read or write to the individual device is slightly different as shown in the following table:

	PCF8575	PCA9555
Write to the device	ST 40 00 00 SP (3 bytes)	ST 40 02 00 00 SP (4 bytes)
Read from device	ST 41 xx xx SP (3 bytes)	ST 40 00 ST 41xx xx SP (5 bytes)

Where ST = start, SP = Stop, and xx is data from device. In the write example, all outputs are programmed low. (0 to all ports in this example)

The PCF8575 requires that you send a total of 3 bytes to either read or write to the device. When writing to the PCF8575 the master sends a start bit, followed by the device address with the least significant bit set to 0. The following two bytes will set the condition of the output ports and then followed by a stop bit. A total of 3 bytes are required.

To read from the PCF8575, all of the ports must be set to logic 1. The master then sets the last bit of the byte containing the address to logic 1. The next two bytes will contain the status of the input ports. A total of 3 bytes are used.

In the PCA9555, data is also transmitted by sending the device address and setting the least significant bit to logic 0. But the next byte after the address will be a command byte. Internal to PCA9555 are eight octal registers configured to operate in 4 registered pairs. The four pairs are Input ports (commands 0 and 1), output ports (commands 2 and 3), polarity inversion ports (command 4 and 5) and configuration ports commands 6 and 7). After sending data to one register, the next byte will automatically be sent to the other register pair. There is no limitation to the number of data bytes that can be sent in one write transmission

The table above then shows a write sequence for the PCA9555. Note that the master generates a start bit followed by the address with the least significant bit set to 0. The next byte is the command to write to the output register (command 2). The following two bytes then write the output (in this case, the data is 00). A total of 4 bytes are used.

To be able to use the current source feature of the PCA9555, the master must first configure the port or ports as an output because the power up default condition for the PCA9555 is that the ports are all set as inputs. For example, to program all of the ports as outputs, the master first generates a start bit followed by the device address with the least significant bit set to 0. The following byte addresses the configuration port (0x06) and the next two bytes set all of the ports to outputs (00 00). The command sequence would be ST 40 06 00 00 SP. The PCA9555 is now set as a 16-bit output port and can sink up to 24 mA active low or source up to 10 mA active high. Caution must be used if the user plans on using the port pin as both an input and output. If the port pin is programmed to an active high and the pin is driven by an external signal low, excessive current could flow because of the conflict.

To read from the PCA9555, requires that you set the command register to 0x00 (input register) before you actually read the data. First the address is sent with the least significant byte set to 0 followed by the command byte 0x00. A restart condition is then sent by the master followed by the read address with the least significant bit set to 1. The following two bytes is the port data. A total of 5 bytes are used.

It should be noted that the next time input port data is needed, the input port will still be accessed until a new command byte has been transmitted., so only two bytes will be needed if no change to the command register is made. Although the PCA9555 appears to require a little more overhead to complete a transaction, it does have the advantage of being able to access either the true or the compliment of each of its data registers independently. Also, since there is no limit to the number of reads in a single transmission, the master can continuously monitor alternate ports until a stop bit is sent.

APPENDIX B — SOFTWARE CONSIDERATIONS IN USING A PCA9554 IN PLACE OF A PCF8574

Because the PCA9554 and the PCF9574 are pin compatible and have the same I²C address, they can often be used in the same board design, however the software to use these devices differ a bit. At power up, both devices look very similar to the target application. The PCF8574 powers up with all I/O active high and so can immediately be used as inputs. The PCA9554 also powers up with all of the I/O pins configured as inputs. How you read or write to the individual device is slightly different as shown in the following table:

	PCF8574	PCA9555
Write to the device	ST 40 00 SP (2 bytes)	ST 40 01 00 SP (3 bytes)
Read from device	ST 41 xx SP (2 bytes)	ST 40 00 ST 41xx SP (4 bytes)

Where ST = start, SP = Stop, and xx is data from device. In the write example, all outputs are programmed low. (0 to all ports in this example)

The PCF8574 requires that you send a total of 2 bytes to either read or write to the device. When writing to the PCF8574 the master sends a start bit, followed by the device address with the least significant bit set to 0. The following byte will set the condition of the output port and then followed by a stop bit. A total of 2 bytes are required.

To read from the PCF8574, all of the ports must be set to logic 1. The master then sets the last bit of the byte containing the address to logic 1. The next byte will contain the status of the input port. A total of 2 bytes are used.

In the PCA9554, data is also transmitted by sending the device address and setting the least significant bit to logic 0. But the next byte after the address will be a command byte. Internal to PCA9554 are 4 octal registers. The four registers are Input (commands 0), output (commands 1), polarity inversion (command 2) and configuration 3).

The table above then shows a write sequence for the PCA9554. Note that the master generates a start bit followed by the address with the least significant bit set to 0. The next byte (0x01) then tells the device that it should write the following byte to the output port and the last byte is the data to be written to the port.

To be able to use the current source feature of the PCA9554, the master must first configure the port as an output because the power up default condition for the PCA9554 is that the ports are all set as inputs. For example, to program all of the ports as outputs, the master first generates a start bit followed by the device address with the least significant bit set to 0. The following byte addresses the configuration port (0x04) and the next byte set all of the ports to outputs (00). The command sequence would be ST 40 06 00 SP. The PCA955 is now set as an 8-bit output port and can sink up to 24 mA active low or source up to 10 mA active high. Caution must be used if the user plans on using the port pin as both an input and output. If the port pin is programmed to an active high and the pin is driven by an external signal low, excessive current could flow because of the conflict.

To read from the PCA9554, requires that you set the command register to 0x00 (input register) before you actually read the data. First the address is sent with the least significant byte set to 0 followed by the command byte 0x00. A restart condition is then sent by the master followed by the read address with the least significant bit set to 1. The following byte is the port data. A total of 4 bytes are used.

It shout be noted that the next time input port data is needed, the input port will still be accessed until a new command byte has been transmitted., so only two bytes will be needed if no change to the command register is made. By using the command register to address the polarity inversion register (command 3), the PCA9554 can be programmed to either read or write the true or compliment of the actual value at the port pin. The command port structure for the PCA9554 is also found in the PCA9556 and PCA9557.

I²C I/O ports

AN469

Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specifications defined by Philips. This specification can be ordered using the code 9398 393 40011.

Philips Semiconductors
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088–3409
Telephone 800-234-7381

Date of release: 08-01

Document order number:

Let's make things better.

Philips Semiconductors

PHILIPS