Version 1.0

Configurable Port Expander (CoPoX)

August 2013

Tom Crofton
8/26/2013

Contents

Introduction

The configurable port expander provides easy access to up to 32 I/O pins over the I^2C protocol. This documentation describes the Configurable Port Expander (CoPoX) standard and the I^2C commands used to read from, write to, and configure a device. A complete discussion of the I^2C bus is outside the scope of this document. The goal is to provide an easy-to-use standard that provides maximum flexibility and allows for rapid prototyping.

Features

The capabilities will vary based on the physical device being used, but the following are possible:

- Digital Input (with or without use of pull ups/downs)
- Digital Output (with or without soft start/stop)
- Analog Input
- Analog Output
- PWM (high and low speed)
- Quantity Pulse Mode (QPM)
- Pulse Count

Quantity Pulse Mode (QPM) is used for sending specific quantities of pulses to a pin instead of sending many commands to turn the pin on and then off. In a practical application this could be used tell a stepper motor driver to step the motor a specific number of times.

Organization

Each device will have 256 registers that each contains an 8 bit byte or 16 bit word. There are 5 groups of registers described below. Not all registers may be implemented, but all will return a value.

Register Organization

Data I/O		0x00 0x1F
Pin Configuration		0x20 0x3F
Pin Capabilities	Digital Capabilities	0x40 0x5F
	Analog Capabilities	0x60 0x7F
Device Configuration		0x80 0x9F
Device Information		0xA0 0xBF

Data I/O are read/write registers used to read or change the setting for a pin on the device.

Pin Configuration are read/write registers used to set the individual pins to inputs or outputs dependent upon what is available as seen in the Pin Capability registers. Attempting to configure a pin to a mode it does not support will result in no action being taken.

The Pin Capability registers are read only and correspond to individual pins on the device. They can be used to determine what features are available for each pin.

The Device Configuration registers allow for general configuration of the device. This includes setting of the I^2C address and timing adjustments such as the frequency of the high speed PWM output.

The Device Information registers are used to obtain information that applies at the device level. This includes information such as the number of pins that can be configured and which version of this specification is implemented. These registers are read only.

The following table lists all available registers and their function names.

List of Registers

Addr	Name	Addr	Name	Addr	Name	Addr	Name	Addr	Name	Addr	Name
0x00	DATA0	0x20	PCON0	0x40	DCAP0	0x60	ACAP0	0x80	ADDR	0xA0	SPECV
0x01	DATA1	0x21	PCON1	0x41	DCAP1	0x61	ACAP1	0x81	PTWEAK	0xA1	SOFTV
0x02	DATA2	0x22	PCON2	0x42	DCAP2	0x62	ACAP2	0x82	QPMPW	0xA2	PINS
0x03	DATA3	0x23	PCON3	0x43	DCAP3	0x63	ACAP3	0x83	PCONF	0xA3	PTMIN
0x04	DATA4	0x24	PCON4	0x44	DCAP4	0x64	ACAP4	0x84	PWMDIV	0xA4	PTMAX
0x05	DATA5	0x25	PCON5	0x45	DCAP5	0x65	ACAP5	0x85	PWMPER	0xA5	PWMLX
0x06	DATA6	0x26	PCON6	0x46	DCAP6	0x66	ACAP6	0x86		0xA6	QPWMIN
0x07	DATA7	0x27	PCON7	0x47	DCAP7	0x67	ACAP7	0x87		0xA7	QPWMLX
0x08	DATA8	0x28	PCON8	0x48	DCAP8	0x68	ACAP8	0x88		0xA8	
0x09	DATA9	0x29	PCON9	0x49	DCAP9	0x69	ACAP9	0x89		0xA9	
0x0A	DATA10	0x2A	PCON10	0x4A	DCAP10	0x6A	ACAP10	0x8A		0xAA	
0x0B	DATA11	0x2B	PCON11	0x4B	DCAP11	0x6B	ACAP11	0x8B		0xAB	
0x0C	DATA12	0x2C	PCON12	0x4C	DCAP12	0x6C	ACAP12	0x8C		0xAC	
0x0D	DATA13	0x2D	PCON13	0x4D	DCAP13	0x6D	ACAP13	0x8D		0xAD	
0x0E	DATA14	0x2E	PCON14	0x4E	DCAP14	0x6E	ACAP14	0x8E		0xAE	
0x0F	DATA15	0x2F	PCON15	0x4F	DCAP15	0x6F	ACAP15	0x8F		0xAF	
0x10	DATA16	0x30	PCON16	0x50	DCAP16	0x70	ACAP16	0x90		0xB0	
0x11	DATA17	0x31	PCON17	0x51	DCAP17	0x71	ACAP17	0x91		0xB1	
0x12	DATA18	0x32	PCON18	0x52	DCAP18	0x72	ACAP18	0x92		0xB2	
0x13	DATA19	0x33	PCON19	0x53	DCAP19	0x73	ACAP19	0x93		0xB3	
0x14	DATA20	0x34	PCON20	0x54	DCAP20	0x74	ACAP20	0x94		0xB4	
0x15	DATA21	0x35	PCON21	0x55	DCAP21	0x75	ACAP21	0x95		0xB5	
0x16	DATA22	0x36	PCON22	0x56	DCAP22	0x76	ACAP22	0x96		0xB6	
0x17	DATA23	0x37	PCON23	0x57	DCAP23	0x77	ACAP23	0x97		0xB7	
0x18	DATA24	0x38	PCON24	0x58	DCAP24	0x78	ACAP24	0x98		0xB8	
0x19	DATA25	0x39	PCON25	0x59	DCAP25	0x79	ACAP25	0x99		0xB9	
0x1A	DATA26	0x3A	PCON26	0x5A	DCAP26	0x7A	ACAP26	0x9A		0xBA	
0x1B	DATA27	0x3B	PCON27	0x5B	DCAP27	0x7B	ACAP27	0x9B		0xBB	
0x1C	DATA28	0x3C	PCON28	0x5C	DCAP28	0x7C	ACAP28	0x9C		0xBC	
0x1D	DATA29	0x3D	PCON29	0x5D	DCAP29	0x7D	ACAP29	0x9D		0xBD	
0x1E	DATA30	0x3E	PCON30	0x5E	DCAP30	0x7E	ACAP30	0x9E		0xBE	
0x1F	DATA31	0x3F	PCON31	0x5F	DCAP31	0x7F	ACAP31	0x9F		0xBF	

Data I/O

Each data I/O register represents a single pin on the device, but the physical pin number may not be the same as the logical pin number.

Reading Data

Reading any register that is not assigned to a physical pin will return 0xFF. For example, register 0x1A on a device that has only 18 pins will read 0xFF. A register will hold either one or two bytes based on the pin's configuration. Reading a single byte will always return the low byte value. If the register is holding one byte and two are read, the high byte will always be zero.

Writing Data

Writing to any register that is not assigned to a physical pin will result in no action being taken. Writing an invalid setting *may* result in *either* no action being taken, or the valid portion of the value may be used. For example, writing 0x83CA to a register that is expecting a single byte may result in 0xCA being used and the high byte discarded. Care should be taken to validate the value before sending as invalid data may cause unexpected results.

Pin Configuration

Each configuration register represents a single pin on the device. Reading a register that is not assigned to a pin will result in a zero code (not connected). Attempts to select an unsupported mode will result in no action being taken.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCONx	–	–	–	–	MODE3	MODE2	MODE1	MODE0

Modes

Mode	Description	MODE3	MODE2	MODE1	MODE0
0	Ignore / not connected	0	0	0	0
1	Digital Input	0	0	0	1
2	Digital Input enable internal pull up	0	0	1	0
3	Digital Input enable internal pull down	0	0	1	1
4	Digital Output	0	1	0	0
5	Digital Output enable soft start/stop	0	1	0	1
6	Quantity Pulse Mode (QPM)	0	1	1	0
7	PWM low speed	0	1	1	1
8	PWM high speed	1	0	0	0
9	Analog Input	1	0	0	1
10	Analog Output	1	0	1	0
11	Pulse Count	1	0	1	1

Mode 0: Ignore / not connected
Setting a pin to this mode will leave the pin in its current state (except for low speed PWM), but no further action will be taken on that pin. Any further write operation to the pin's Data I/O will be ignored. For example, if the pin is first configured to digital output and set to high, changing the configuration to ignore will leave the pin set high, but an attempt to set the pin low will be ignored. If the pin is in low speed PWM mode when configured to ignore, PWM will stop.

Mode 1: Digital Input
Writing to the Data I/O register will have no effect. Reading the register will return 0 if the pin is set low, or 1 if it is high.

Mode 2: Digital Input enable internal pull up
Same as above, only the device's internal pull up resistor is used.

Mode 3: Digital Input enable internal pull down
Same as above, only the device's internal pull down resistor is used.

Mode 4: Digital Output

Writing to the Data I/O register will set the pin high if the value written is > 0 or low if it is 0. Reading the register will return 1 if the pin is set high, 0 if it is set low.

Mode 5: Digital Output enable soft start/stop

The soft start sequence begins with the pin output at logical 0 and pulses the output with increasing duty cycle until it reaches 100% at which point the pin is left at logical high until another command is received. The soft stop sequence is the same process in reverse. Writing any value other than 0x00 or 0xFF to the Data I/O register will cause the sequence to begin at the given duty cycle and continue towards full on, or full off depending on the current setting. For example, if the pin is currently off, setting the pin to 0XFF will start the duty cycle at 1 and continue to full on. However, if the pin is currently off, setting the pin to 0X80 will start the duty cycle at 50% and increase until full on. Reading the Data I/O register will return a single byte indicating the current duty cycle of the pin.

Mode 6: Quantity Pulse Mode (QPM)

In Quantity Pulse Mode (QPM) writing to the Data I/O register will set the amount of pulses to send. Reading the Data I/O register will give how many pulses are remaining to be sent. QPM uses the same period values as low speed PWM, including PTWEAK. The pulse width is set using the QPMPW register.

Mode 7: PWM low speed

In low speed PWM mode, the period is fixed at 50Hz, but the device configuration register PTWEAK can be used to increase or decrease this value. The duty cycle is set by writing one or two bytes to the Data I/O register depending upon how many bits of resolution are available. If only one byte is sent, the byte will be assigned to the low byte of the duty cycle. Any bits beyond the capability will be ignored. For example, a pin capable of 12 bit resolution will apply a mask of 0x0FFF to the number received before setting the duty cycle. Reading from the register while in PWM mode will return the current duty cycle setting.

Mode 8: PWM high speed

Same as above, only PWM operates using a shorter period (more pulses per second). Two registers are available for changing the hardware configuration. They are PWMPER and PWMDIV. See the section on device configuration for more on these registers.

Mode 9: Analog Input

In analog input mode, reading the Data I/O register will return a 16 bit value to the level of precision as given by the pins capability register. Input values must be between ground and Vcc.

Mode 10: Analog Output

In analog output mode the value is set by writing one or two bytes to the register. If only one byte is sent, the byte will be assigned to the low byte of the output level. Any

bits beyond the capability will be ignored. For example a pin capable of 12 bit resolution will apply a mask of 0x0FFF to the number received before setting the value. Output values will be between ground an Vcc.

Mode 11: Pulse Count

Writing to the Data I/O register sets the initial value. Reading the register returns a 16 bit value of the number of pulses counted. The PCONF register is used to determine if the count is increased on the rising or falling of the input level. Pulse Count can use the devices internal pull up/down by first configuring the pin to digital input with the pull up or down enabled, then switching to Pulse Count mode. If a pin is not first configured for any type of digital input, it will be set to digital input (Mode 1) before being set to Pulse Count.

Pin Capabilities

Because of the amount of data, pin capabilities are divided into analog and digital sections.

Digital Capabilities

Each capability register corresponds to a single I/O pin on the device. Reading the register will give a two byte word indicating the capabilities of the pin. Writing to the register will result in no action being taken.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DCAPL	–	–	–	QPM	DSOFT	DOUT	DIN1	DIN0
DCAPH	PWMH3	PWMH2	PWMH1	PWMH0	PWML3	PWML2	PWML1	PWML0

Digital Input

DIN1	DIN0	
0	0	No digital input
0	1	Digital input without pull up/down
1	0	Digital input with pull up
1	1	Digital input with pull down

Digital Output

DOUT	
0	No digital input
1	Digital output

DSOFT	
0	Does not support soft start/stop
1	Soft start/stop capable

QPM	
0	No QPM support
1	QPM capable

If either DSOFT or QPM are available (bit is one), then DOUT must also be available. If DOUT is zero (not available) then DSOFT and DOUT must also be zero

PWML Low Speed PWM Output

The output period of low speed PWM is fixed at 50 Hz, but the PTWEAK register can be used to increase or decrease this value. If the number represented the PWML bits is

zero, the pin has no PWM capability. If the number is non-zero, it indicates the number of bits of resolution minus one.

PWML3	PWML2	PWML1	PWML0	Resolution
0	0	0	0	No PWM Output
0	0	0	1	2 Bits
0	0	1	0	3 Bits
0	0	1	1	4 Bits
0	1	0	0	5 Bits
0	1	0	1	6 Bits
0	1	1	0	7 Bits
0	1	1	1	8 Bits
1	0	0	0	9 Bits
1	0	0	1	10 Bits
1	0	1	0	11 Bits
1	0	1	1	12 Bits
1	1	0	0	13 Bits
1	1	0	1	14 Bits
1	1	1	0	15 Bits
1	1	1	1	16 Bits

PWMH High Speed PWM Output

The output of PWMH is variable and is determined by PWMPER and PWMDIV. If the number represented the PWMH bits is zero, the pin has no PWM capability. If the number is non-zero, it indicates the number of bits of resolution minus one.

PWMH3	PWMH2	PWMH1	PWMH0	Resolution
0	0	0	0	No PWM Output
0	0	0	1	2 Bits
0	0	1	0	3 Bits
0	0	1	1	4 Bits
0	1	0	0	5 Bits
0	1	0	1	6 Bits
0	1	1	0	7 Bits
0	1	1	1	8 Bits
1	0	0	0	9 Bits
1	0	0	1	10 Bits
1	0	1	0	11 Bits
1	0	1	1	12 Bits
1	1	0	0	13 Bits
1	1	0	1	14 Bits
1	1	1	0	15 Bits
1	1	1	1	16 Bits

Analog Capabilities

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ACAPx	ANOUT3	ANOUT2	ANOUT1	ANOUT0	ANIN3	ANIN2	ANIN1	ANIN0

Analog Input

If the number represented the ANIN bits is zero, the pin has no analog input capability. If the number is non-zero, it indicates the number of bits of resolution minus one.

ANIN3	ANIN2	ANIN1	ANIN0	Input Resolution
0	0	0	0	No Analog Input
0	0	0	1	2 Bits
0	0	1	0	3 Bits
0	0	1	1	4 Bits
0	1	0	0	5 Bits
0	1	0	1	6 Bits
0	1	1	0	7 Bits
0	1	1	1	8 Bits
1	0	0	0	9 Bits
1	0	0	1	10 Bits
1	0	1	0	11 Bits
1	0	1	1	12 Bits
1	1	0	0	13 Bits
1	1	0	1	14 Bits
1	1	1	0	15 Bits
1	1	1	1	16 Bits

Analog Output

If the number represented the ANOUT bits is zero, the pin has no analog output capability. If the number is non-zero, it indicates the number of bits of resolution minus one.

ANOUT3	ANOUT2	ANOUT1	ANOUT0	Resolution
0	0	0	0	No Analog Output
0	0	0	1	2 Bits
0	0	1	0	3 Bits
0	0	1	1	4 Bits
0	1	0	0	5 Bits
0	1	0	1	6 Bits
0	1	1	0	7 Bits

0	1	1	1	8 Bits
1	0	0	0	9 Bits
1	0	0	1	10 Bits
1	0	1	0	11 Bits
1	0	1	1	12 Bits
1	1	0	0	13 Bits
1	1	0	1	14 Bits
1	1	1	0	15 Bits
1	1	1	1	16 Bits

Device Configuration

Address	Name	Description
0x80	ADDR	I^2C address to use
0x81	PTWEAK	Slow PWM Tweak value
0x82	QPMPW	QPM Pulse Width
0x83	PCONF	Pulse Count Configuration
0x84	PWMDIV	PWM Divider
0x85	PWMPER	PWM Period

ADDR

The ADDR register is used to set the I^2C address of the chip. Only 7 bit addressing is supported. Changing the address will cause the device to immediately stop responding on the current address and switch to the new address. If the device has persistent memory, the address will be stored and used the next time the device starts. The I^2C address must be greater than 0x07 and less than 0x78, with the default being 0x18.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADDR	–	I^2C Address						

PTWEAK

The PTWEAK is a single byte register is used to adjust the low speed PWM period. If the device has persistent memory, the value will be stored and used the next time the device starts. The maximum and minimum values of PTWEAK are stored in the PTMIN and PTMAX registers. Increasing the value of PTWEAK will shorten the period, and thus increase the frequency.

QPMPW

The QPMPW register is used to set the QPM duty cycle length. The minimum and maximum allowable values are located in the QPMMIN and QPMMAX registers. The QPM period is the same as slow PWM period. If the default PTWEAK setting is used, the total pulse period time will be about 20ms. Setting QPMPW to 64 (25%) would mean each pulse would consist of approximately 5ms on time, followed by 15ms of off time, and repeat for each pulse. If the device has persistent memory, the value will be stored and used the next time the device starts.

PCONF

The PCOUNT register is used to control the behavior of pulse counting. If the PCHL bit is set, pulse counts will occur on high to low transition. If the PCHL is clear (default),

counts occur on low to high transitions. If the device has persistent memory, the value will be stored and used the next time the device starts.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCONF	–	–	–	–	–	–	–	PCHL

PWMDIV

This register is used to set the divider for Mode 8 (PWM High Speed). If the device has persistent memory, the value will be stored and used the next time the device starts. See the individual device documentation for more on its implementation.

PWMPER

This register is used to set the period for Mode 8 (PWM High Speed). If the device has persistent memory, the value will be stored and used the next time the device starts. See the individual device documentation for more on its implementation.

Device Information

Address	Name	Description
0xA0	SPECV	Specification version implemented
0xA1	SOFTV	Software Release Number
0xA2	PINS	Number of pins available
0xA3	PTMIN	Slow PWM Tweak Minimum value
0xA4	PTMAX	Slow PWM Tweak Maximum value
0xA5	PWMLX	Maximum number of pins assigned to slow PWM
0xA6	QPWMIN	QPM Minimum Pulse Width
0xA7	QPWMAX	QPM Maximum Pulse Width

Appendix A: Typical Examples

Below are typical examples run on a Raspberry Pi using the I^2C utilities. This location has an article about enabling I^2C on the Pi. http://www.instructables.com/id/Raspberry-Pi-I2C-Python/step2/Enable-I2C/ Please refer to the man pages complete information on the I^2C commands and their usage.

Example One:
In this example we will query the I^2C registers to find out if a pin supports simple digital output, then use it to turn the pin on and off. If you connect a resistor and an LED to the pin you will see it turn on with the 3rd command and off with the next. This example assumes the device is at the default address of 0x20.

```
//get the capabilities of pin 1
i2cget –y 1 0x20 0x41
// returns 0x1D – logical and with 0x04 shows the DOUT bit is 1

//set pin1 to digital output
i2cset –y 1 0x20 0x21 0x04

i2cset –y 1 0x20 0x01 0x01
//the pin is ON
i2cset –y 1 0x20 0x01 0x00
//the pin is OFF
```

Example Two:
Using the same setup, use slow PWM to make the LED glow at different intensities.

```
//set pin1 to slow PWM output (50 Hz)
i2cset –y 1 0x20 0x21 0x07

i2cset –y 1 0x20 0x01 0x30
//the pin is ON dim

i2cset –y 1 0x20 0x01 0xCC
//the pin is ON brighter

i2cset –y 1 0x20 0x01 0x00
//the pin is OFF
```

Example Three:
Use hardware high speed PWM.

```
i2cget –y 1 0x20 0x41 w
```

//returns 0x971D – fir 4 bits shows 10 bit PWM

//set pin1 to hardware pwm
i2cset –y 1 0x20 0x21 0x08

//set the duty cycle light
i2cset –y 1 0x20 0x01 0x0002 w
//set the duty cycle to more
i2cset –y 1 0x20 0x01 0x030F w

Example Four:
Use QPM to send a number of pulses and use pulse count on another pin to check the accuracy
//pulse count test
//set pin 1 to qpm
i2cset –y 1 0x20 0x21 0x06

//set pin 0 to input no pull up/down
i2cset –y 1 0x20 0x20 0x01
//set pin 0 to count mode
i2cset –y 1 0x20 0x20 0x0B

//pulse pin 1 10 times
i2cset –y 1 0x20 0x01 0x0A

//read the count
i2cget –y 1 0x20 0x00

Example Five:
//Testing Soft Start Stop
i2cset –y 1 0x20 0x21 0x05
//in soft start/stop mode
i2cset –y 1 0x20 0x01 0xFF
//fades to full on
i2cset –y 1 0x20 0x01 0x00
//fades to full off

I2cset –y 1 0x20 0x20 0x09
//set to analog input
I2cget –y 1 0x20 0x20 w
0x02ba
Meter reads 2.25 of 3.28 volts
3.28*(698/1023)=2.24 (1/100 volt difference is less than 0.5%)

```
//analog out test (running at 3.3v)
i2cset –y 1 0x20 0x20 0x0A
//set to analog input
i2cset –y 1 0x20 0x00 0x09
// 1 volt is output
i2cset –y 1 0x20 0x00 0x0e
// 1.49 volt is output
```

Appendix B: Implementation on PIC16F1827

Pin Out of implementation on a pic16f1827

Logical Pin	Physical Pin	Name	Features
0	1	RA2	Analog Out, Analog In
1	2	RA3	PWM, Analog In
2	3	RA4	PWM, Analog In
3	4	RA5	Digital Input Only
4	6	RB0	PWM, PullUp
5	8	RB2	PullUp, Analog In
6	9	RB3	PullUp, Analog In
7	11	RB5	PullUp, Analog In
8	12	RB6	PullUp, Analog In
9	13	RB7	PullUp, Analog In
10	15	RA6	
11	16	RA7	PWM
12	17	RA0	Analog In
13	18	RA1	Analog In