PmodACL Programmer's Reference Manual

Revision: December 16, 2011

www.digilentinc.com

1300 NE Henley Court, Suite 3
Pullman, WA 99163
(509) 334 6306 Voice | (509) 334 6300 Fax

Introduction

This document describes the programming interface to the PmodACL library that is included as part of the PmodLib library. It describes the capabilities of the PmodACL and all the API functions used to access its features.

The purpose this library is to offer access to the functionality of the PmodACL for the Cerebot 32MX4 or 32MX7 microcontrollers. For additional information on operation and signaling see the Digilent PmodACL Reference Manual and the Analog Devices ADXL345 Reference Manual.

PmodACL Basic API Functions

void PmodACLInitSpi(SpiChannel chn, uint32_t pbClock, uint32_t bitRate)

Parameters

SpiChannel chn - Spi Channel
uint32_t pbClock - peripheral bus clock frequency in Hz
uint32_t bitRate - bit rate desired in Hz

Return Value

none

Initializes the PmodACL for SPI 4 wire in Mode 3 as follows: master, enable slave select, 8 bit mode, CKP high (SPI Mode 3 -> CKP-1 CKE-0)

NOTE: See SERIAL COMMUNICATIONS->SPI in the ADXL345 reference manual for additional information.

void PmodACLGetAxisData(SpiChannel chn, PMODACL_AXIS *pmodACLAxis)

Parameters

SpiChannel chn - Spi Channel
PMODACL_AXIS *pmodACLAxis - pointer to data structure used to store axis data

Return Value

none

Reads the values from the axis values and writes to a PMODACL_AXIS struct

A multi-byte read of the axis registers is performed, low and high bytes are shifted and combined to create 16 bit signed values then stored in the corresponding axis field in pmodACLAxis.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x32 to Register 0x37 DATAX0, DATAX1, DATAY0, DATAY1, DATAZ0, DATAZ1 in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

uint8_t PmodACLReadReg(SpiChannel chn, uint8_t address)

Parameters

SpiChannel chn - Spi Channel
uint8_t address - address of register to read

Return Value

uint8_t - value stored in register

Reads and returns a value from the register specified in the address parameter, valid register address are located in Table 1: PmodACL Register/Configuration Definitions.

void PmodACLReadRegMultiByte(SpiChannel chn, uint8_t startAddress, uint8_t *data, uint8_t numBytes)

Parameters

SpiChannel chn - Spi Channel
uint8_t startAddress - register start address
uint8_t *data - pointer to array of uint8_t, register values are stored here
uint8_t numBytes - number of bytes to read

Return Value

none

Reads a series of bytes starting at a base register address and stores them in a array.

Example:

```
uint8_t data[PMODACL_NUM_OFSXYZ_OFFSET_BYTES];

PmodACLReadRegMultiByte(SPI2,PMODACL_REG_OFSX,data,
                        PMODACL_NUM_OFFSET_BYTES);
```

This preceding will start at register PMODACL_REG_OFSX and read PMODACL_NUM_OFFSET_BYTES into corresponding indices in data, incrementing register address by 1 each time. This function is best used when multiple contiguous values must be read in an atomically before register values change. See Table 1: PmodACL Register/Configuration Definitions.

int32_t PmodACLCalibrate(SpiChannel chn,uint8_t numSamples,
uint8_t oneGaxisOrienatation)

Parameters

SpiChannel chn - Spi Channel
uint8_t numSamples -number of samples to take during calibration
uint8_t oneGaxisOrienatation - 1G axis orientation during calibration Acceptable values are prefixed with PMODACL_CALIBRATE in Table 1: PmodACL Register/Configuration Definitions.

Return Value

int32_t - signed integer representing calculated offset values <AXIS:byte>: X:2, Y:1, Z:0

Performs a calibration of the PmodACL axis utilizing the OFFSET register

Axis value sample are taken and averaged to achieve a baseline reprentation of all axes, the OFFSET register is then set to automatically adjust axis readings. PmodACL should be placed such that one axis is positioned to read 1g and the others 0g, typically this is the Z axis. The 1G orientation axis is specified using values defined in Table 1: PmodACL Register/Configuration Definitions prefixed with PMODACL_CALIBRATE.

NOTE: For a full description of the calibration procedure see "Offset Calibration" in the ADXL345 reference manual.

void PmodACLWriteRegMultiByte(SpiChannel chn,uint8_t startAddress,uint8_t *data,
uint8_t numBytes)

Parameters

SpiChannel chn - Spi Channel
uint8_t startAddress - register start address
uint8_t *data - pointer to array of uint8_t, new register values
uint8_t numBytes - number of bytes to write

Return Value

none

Writes a series of bytes starting at a base register address and from an array.

Example:

```
uint8_t data[PMODACL_NUM_OFSXYZ_OFFSET_BYTES] = {1,2,3}
PmodACLWriteRegMultiByte(SPI2,PMODACL_REG_OFSX,data,
                                    PMODACL_NUM_OFFSET_BYTES)
```

This preceding will start at register PMODACL_REG_OFSX and write PMODACL_NUM_OFFSET_BYTES from corresponding indices in data to PMODACL_NUM_OFFSET_BYTES registers, the address is increased by 1 each time a byte is written. This function is best used when multiple contiguous values must be written atomically for a specific command. See Table 1: PmodACL Register/Configuration Definitions.

void PmodACLWriteReg(SpiChannel chn,uint8_t address,uint8_t dataBits)

Parameters

SpiChannel chn - Spi Channel
uint8_t address - address of register to write

Return Value

none

Writes a value to the register specified in the address parameter, valid register address are located in Table 1: PmodACL Register/Configuration Definitions and are prefixed with PMODACL_REG

PmodACL Macros

PmodACLSetDataFormat(CHN,DATA_FORMAT)

Parameters

SpiChannel CHN - Spi channel
uint8_t DATA_FORMAT - Combination of ORed values prefixed with PMODACL_BIT_DATA_FORMAT defined in Table 1: PmodACL Register/Configuration Definitions

Return Value

none

Sets the representation of data to the registers defined in PMODACL_REG_DATA<AXIS><BIT>, sets SPI 3 or 4 wire mode, interrupt active high or low, full resolution mode, bit justification and range bits

DATA_FORMAT Register *

D7	D6	D5	D4	D3	D2	D1	D0
SELF_TEST	SPI	INT_INVERT	0	FULL_RES	JUSTIFY	RANGE BITS	

*Values prefixed with PMODACL_BIT_DATA_FORMAT in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x31—DATA_FORMAT (Read/Write) in the ADXL345 reference manual for additional information. Only 4 Wire SPI is supported on the PmodACL. See Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetDataFormat(CHN)

Parameters

uint8_t - PMODACL_REG_DATA_FORMAT register contents

Return Value

uint8_t - PMODACL_REG_DATA_FORMAT register contents

Returns the DATA_FORMAT register (defined in PMODACL_REG_DATA_FORMAT)

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x31—DATA_FORMAT (Read/Write) in the ADXL345 reference manual for additional information. See Table 1: PmodACL Register/Configuration Definitions prefixed with PMODACL_REG.

PmodACLSetPowerCtl(CHN,POWER_CTL)

Parameters

uint8_t POWER_CTL - Combination of ORed values prefixed with PMODACL_BIT_POWER_CTL in see Table 1: PmodACL Register/Configuration Definitions

Return Value

none

Sets the PMODACL_REG_POWER_CTL register

POWER_CTL Register *

D7	D6	D5	D4	D3	D2	D1	D0
0	0	LINK	AUTO_SLEEP	MEASURE	SLEEP	WAKEUP	

*Values prefixed with PMODACL_BIT_POWER_CTL in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x2D—POWER_CTL (Read/Write) in the ADXL345 reference manual for additional information. See Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetPowerCtl(CHN)

Parameters

SpiChannel CHN - spiChannel

Return Value

uint8_t - PMODACL_REG_POWER_CTL register contents see Table 1: PmodACL Register/Configuration Definitions, prefixed with PMODACL_BIT_POWER_CTL

Returns the POWER_CTL register (defined in PMODACL_REG_POWER_CTL)

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x2D—POWER_CTL (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetDeviceID(CHN)

Parameters

SpiChannel CHN - spiChannel

Return Value

uint8_t - PMODACL_DEVICE_ID (value)

Returns the contents of DEVID register, which is a fixed value defined in PMODACL_DEVICE_ID listed in Table 1: PmodACL Register/Configuration Definitions.

PmodACLSetFIFOCtl(CHN,FIFO_CTL)

Parameters

SpiChannel CHN - Spi channel

uint8_t FIFO_CTL - Combination of ORed values prefixed with PMODACL_BIT_FIFO_CTL defined in "Local Type Definitions"

Return Value

none

Sets the PMODACL_REG_FIFO_CTL register

FIFO_CTL Register *

D7	D6	D5	D4	D3	D2	D1	D0
FIFO_MODE		TRIGGER	Samples				

*Values prefixed with PMODACL_BIT_FIFO_CTL in Table 1: PmodACL Register/Configuration Definitions.

FIFO Modes*

D7	D6	Mode
PMODACL_BIT_FIFO_CTL_BYPASS		Bypass
PMODACL_BIT_FIFO_CTL_FIFO		FIFO
PMODACL_BIT_FIFO_CTL_STREAM		Stream
PMODACL_BIT_FIFO_CTL_TRIGGER		Trigger

*Values prefixed with PMODACL_BIT_FIFO_CTL in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x38—FIFO_CTL (Read/Write) in the ADXL345 reference manual for additional information.

PmodACLGetFIFOCtl(CHN)

Parameters

SpiChannel CHN - Spi channel

Return Value

uint8_t - Combination of ORed values prefixed with PMODACL_BIT_FIFO_CTL defined in Table 1: PmodACL Register/Configuration Definitions.

Gets the contents of the PMODACL_REG_FIFO_CTL register

FIFO_CTL Register*

D7	D6	D5	D4	D3	D2	D1	D0
FIFO_MODE		TRIGGER	Samples				

*Values prefixed with PMODACL_BIT_FIFO_CTL in Table 1: PmodACL Register/Configuration Definitions.

FIFO Modes*

D7	D6	Mode
PMODACL_BIT_FIFO_CTL_BYPASS		Bypass
PMODACL_BIT_FIFO_CTL_FIFO		FIFO
PMODACL_BIT_FIFO_CTL_STREAM		Stream
PMODACL_BIT_FIFO_CTL_TRIGGER		Trigger

*Values prefixed with PMODACL_BIT_FIFO_CTL in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x38—FIFO_CTL (Read/Write) in the ADXL345 reference manual for additional information.

PmodACLSetOffset(CHN,OFFSET_BYTES)

Parameters

SpiChannel CHN - Spi Channel
int8_t OFFSET_BYTES[PMODACL_NUM_OFFSET_BYTES] - 3 byte array containing offset values

Return Value

none

Set the contents of the PMODACL_REG_OFSX,PMODACL_REG_OFSY, and PMODACL_REG_OFSZ registers

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x1E, Register 0x1F, Register 0x20—OFSX, OFSY, OFSZ (Read/Write) in the ADXL345 reference manual for additional information.

Function PmodACLCalibrate supplied in this library will perform the calibration calculations and set the OFFSET registers. Array passed in should be of size PMODACL_NUM_OFFSET_BYTES listed in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetOffset(CHN,OFFSET_BYTES)

Parameters

SpiChannel CHN - Spi Channel
int8_t OFFSET_BYTES[PMODACL_NUM_OFFSET_BYTES] - 3 byte array to fill with offset bytes

Return Value

none

Fills a 3 byte array with the contents of the PMODACL_REG_OFSX,PMODACL_REG_OFSY, and PMODACL_REG_OFSZ

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x1E, Register 0x1F, Register 0x20—OFSX, OFSY, OFSZ (Read/Write) in the ADXL345 reference manual for additional information. Array passed in should be of size PMODACL_NUM_OFFSET_BYTES listed in Table 1: PmodACL Register/Configuration Definitions.

PmodACLSetThreshTap(CHN,THRESH_TAP)

Parameters

SpiChannel CHN - Spi channel
uint8_t THRESH_TAP - tap threshold

Return Value

none

Sets threshold value for interrupts.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x1D—THRESH_TAP (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetThreshTap(CHN)

Parameters

SpiChannel CHN - Spi channel

Return Value

uint8_t - PMODACL_REG_THRESH_TAP register contents

Returns the contents of the THRESH_TAP register (defined as PMODACL_REG_THRESH_TAP in Table 1: PmodACL Register/Configuration Definitions)

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x1D—THRESH_TAP (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLSetIntEnable(CHN,INT_ENABLE)

Parameters

SpiChannel CHN - Spi channel
uint8_t INT_ENABLE - Combination of ORed values prefixed with PMODACL_BIT_INT_ENABLE defined in Table 1: PmodACL Register/Configuration Definitions

Return Value

none

Sets the contents of the PMODACL_REG_INT_ENABLE register.

Setting bits in this register enable their respective functions to generate interrupts, clearing bits disables them.

INT_ENABLE Register *

D7	D6	D5	D4	D3	D2	D1	D0
DATA_READY	SINGLE_TAP	DOUBLE_TAP	ACTIVITY	INACTIVITY	FREE_FALL	WATERMARK	OVERRUN

*Values prefixed with PMODACL_BIT_INT_ENABLE_ in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x2E—INT_ENABLE (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetIntEnable(CHN)

Parameters

SpiChannel CHN - Spi channel

Return Value

uint8_t - PMODACL_REG_INT_ENABLE register contents

Returns the INT_ENABLE register (defined in PMODACL_REG_INT_ENABLE)

INT_ENABLE Register *

D7	D6	D5	D4	D3	D2	D1	D0
DATA_READY	SINGLE_TAP	DOUBLE_TAP	ACTIVITY	INACTIVITY	FREE_FALL	WATERMARK	OVERRUN

*Values prefixed with PMODACL_BIT_INT_ENABLE_in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x2E—INT_ENABLE (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLSetIntMap(CHN,INT_MAP)

Parameters

SpiChannel CHN - Spi channel
uint8_t INT_MAP - Combination of ORed values prefixed with PMODACL_BIT_INT_MAP defined in Table 1: PmodACL Register/Configuration Definitions.

Return Value

None

INT_MAP Register*

D7	D6	D5	D4	D3	D2	D1	D0
DATA_READY	SINGLE_TAP	DOUBLE_TAP	ACTIVITY	INACTIVITY	FREE_FALL	WATERMARK	OVERRUN

*Values prefixed with PMODACL_BIT_INT_MAP_ in Table 1: PmodACL Register/Configuration Definitions.

Bits set to 0 map their interrupt events to the INT1 pin, bits set to 1 map their interrupt events to the INT2 pin. Interrupts set for any given pin are OR'ed.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x2F—INT_MAP (R/W) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetIntMap(CHN)

Parameters

SpiChannel CHN - Spi channel

Return Value

uint8_t - PMODACL_REG_INT_MAP register contents

Returns the INT_MAP register (defined in PMODACL_REG_INT_MAP)

INT_MAP Register *

D7	D6	D5	D4	D3	D2	D1	D0
DATA_READY	SINGLE_TAP	DOUBLE_TAP	ACTIVITY	INACTIVITY	FREE_FALL	WATERMARK	OVERRUN

*Values prefixed with PMODACL_BIT_INT_MAP_ in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x2F—INT_MAP (R/W) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetIntSource(CHN)

Parameters

SpiChannel CHN - Spi channel

Return Value

uint8_t - contents of PMODACL_REG_INT_SOURCE register, which is a combination of ORed values prefixed with PMODACL_BIT_INT_SOURCE defined in Table 1: PmodACL Register/Configuration Definitions.

INT_SOURCE Register *

D7	D6	D5	D4	D3	D2	D1	D0
DATA_READY	SINGLE_TAP	DOUBLE_TAP	ACTIVITY	INACTIVITY	FREE_FALL	WATERMARK	OVERRUN

*Values prefixed with PMODACL_BIT_INT_SOURCE in Table 1: PmodACL Register/Configuration Definitions.

Bits set to 1 indicate an event of this type occurred. Reading this register will clear the interrupt bits.

NOTE: A double tap will set the SINGLE_TAP and DOUBLE_TAP bits, this value is provided as PMODACL_BIT_INT_SOURCE_DOUBLE_TAP defined in Table 1: PmodACL Register/Configuration Definitions.

See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x30—INT_SOURCE (Read Only) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLSetTapAxes(CHN,TAP_AXES)

Parameters

SpiChannel CHN - Spi channel
uint8_t TAP_AXES - Combination of ORed values prefixed with PMODACL_BIT_TAP_AXES defined in Table 1: PmodACL Register/Configuration Definitions.

Return Value

none

Enables/Disable Axes for tap detection

TAP_AXES Register *

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	SUPPRESS	TAP_X_ENABLE	TAP_Y_ENABLE	TAP_Z_ENABLE

*Values prefixed with PMODACL_BIT_TAP_AXES in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x2A—TAP_AXES (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetTapAxes(CHN)

Parameters

SpiChannel CHN - Spi channel

Return Value

uint8_t - PMODACL_REG_TAP_AXES register contents

Returns the TAP_AXES register (defined in PMODACL_REG_TAP_AXES)

TAP_AXES Register (VALUES PREFIXED WITH PMODACL_BIT_TAP_AXES)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	SUPPRESS	TAP_X_ENABLE	TAP_Y_ENABLE	TAP_Z_ENABLE

*Values prefixed with PMODACL_BIT_TAP_AXES in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x2A—TAP_AXES (Read/Write) in the ADXL345 reference manual for additional information.

PmodACLSetTapDuration(CHN,DUR)

Parameters

SpiChannel CHN - Spi Channel
uint8_t DUR - tap duration (625 microseconds/LSB)

Return Value

none

Set the amount of time the an event must be above THRESH_TAP to trigger a tap event.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x21—DUR (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetTapDuration(CHN)

Parameters

SpiChannel CHN - Spi Channel

Return Value

uint8_t - tap duration (625 microseconds/LSB)

Returns the DUR register (defined as PMODACL_REG_DUR Table 1: PmodACL Register/Configuration Definitions.)

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x21—DUR (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLSetTapLatency(CHN,TAP_LATENCY)

Parameters

SpiChannel CHN - Spi channel
uint8_t TAP_LATENCY - time between tap detection and start of the time window (1.25ms/LSB)

Return Value

none

Sets the amount of time to wait from a tap detection to the start of the time window as defined in the PMODACL_REG_WINDOW register (Table 1: PmodACL Register/Configuration Definitions).

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x22—Latent (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetTapLatency(CHN)

Parameters

SpiChannel CHN - Spi channel

Return Value

uint8_t - time between tap detection and start of the time window (1.25ms/LSB)

Defines the amount of time to wait from a tap detection to the start of the time window as defined in the PMODACL_REG_WINDOW register (Table 1: PmodACL Register/Configuration Definitions).

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x22—Latent (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLSetTapWindow(CHN,TAP_WINDOW)

Parameters

SpiChannel CHN - Spi channel
uint8_t TAP_WINDOW - tap window (1.25ms/LSB)

Return Value

none

Represents the amount of time after the latency time (PMODACL_REG_LATENT) expires that a second tap can occur. Setting this register to 0 disables double taps. Register defined in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x23—Window (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetTapWindow(CHN)

Parameters

SpiChannel CHN - Spi channel

Return Value

uint8_t - tap window (1.25ms/LSB)

Represents the amount of time after the latency time (PMODACL_REG_LATENT) expires that a second tap can occur. Return value of 0 indicates that double taps are disabled.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x23—Window (Read/Write) in the ADXL345 reference manual for additional information. Register defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetActTapStatus(CHN)

Parameters

SpiChannel CHN - Spi channel

Return Value

uint8_t - contents of PMODACL_REG_ACT_TAP_STATUS register, which is a combination of ORed values prefixed with PMODACL_BIT_ACT_TAP_STATUS defined in Table 1: PmodACL Register/Configuration Definitions.

Bits set to 1 indicate the tap or activity source of an event, this register should be read before interrupts are cleared.

ACT_TAP_STATUS Register *

D7	D6	D5	D4	D3	D2	D1	D0
0	ACT_X_SOURCE	ACT_Y_SOURCE	ACT_Z_SOURCE	SUPPRESS	TAP_X_ENABLE	TAP_Y_ENABLE	TAP_Z_ENABLE

*Values prefixed with PMODACL_BIT_ACT_TAP_STATUS in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x2B—ACT_TAP_STATUS (Read Only) in the ADXL345 reference manual for additional information.

PmodACLSetThreshFF(CHN,THRESH_FF)

Parameters

SpiChannel CHN - Spi Channel
uint8_t THRESH_FF- free fall threshold (62.5mg/LSB)

Return Value

none

Sets the threshold for free fall detection, acceleration on all axes is compared to this register, value range should be between 300mg and 600mg.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x28—THRESH_FF (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetThreshFF(CHN)

Parameters

SpiChannel CHN - Spi Channel

Return Value

uint8_t - free fall threshold (62.5mg/LSB)

Returns threshold for free fall detection, acceleration on all axes is compared to this register, value range should be between 300mg and 600mg.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x28—THRESH_FF (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLSetBwRate(CHN,BW_RATE)

Parameters

SpiChannel CHN - Spi Channel

Return Value

uint8_t - contents of PMODACL_REG_BW_RATE register, which is a combination of ORed values prefixed with PMODACL_BIT_BW_RATE defined in Table 1: PmodACL Register/Configuration Definitions.

Register defines low/normal power modes, device bandwidth and data output rate.

BW_RATE Register *

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	LOW_POWER	RATE			

*Values prefixed with PMODACL_BIT_BW_RATE in Table 1: PmodACL Register/Configuration Definitions.

NOTE: Values for tables 7 and 8 are defined in "Local Type Definitions" and following the following format PMODACL_BIT_BW_RATE_<OUTPUT_BIT_RATE>_HZ, the bandwidth is half of the value defined.

See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x2C—BW_RATE (Read/Write) in the ADXL345 reference manual for additional information.

PmodACLGetBwRate(CHN)

Parameters

SpiChannel CHN - Spi Channel

Return Value

uint8_t - PMODACL_REG_BW_RATE register contents

Register defines low/normal power modes, device bandwidth and data output rate.

BW_RATE Register*

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	LOW_POWER	RATE			

*Values prefixed with PMODACL_BIT_BW_RATE in Table 1: PmodACL Register/Configuration Definitions.

NOTE: Values for tables 7 and 8 are defined in "Local Type Definitions" and following the following format PMODACL_BIT_BW_RATE_<OUTPUT_BIT_RATE>_HZ, the bandwidth is half of the value defined.

See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x2C—BW_RATE (Read/Write) in the ADXL345 reference manual for additional information.

PmodACLSetTimeInact(CHN,TIME_INACT)

Parameters

SpiChannel CHN - Spi Channel
uint8_t TIME_INACT- amount of time acceleration is less than THRESH_INACT (1sec/LSB)

Return Value

none

Sets the amount of time acceleration is less than THRESH_INACT (1sec/LSB) before a state of inactivity is detected.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x26—TIME_INACT (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetTimeInact(CHN)

Parameters

SpiChannel CHN - Spi Channel

Return Value

uint8_t - mount of time acceleration is less than THRESH_INACT (1sec/LSB)

Returns the amount of time acceleration is less than THRESH_INACT (1sec/LSB) before a state of inactivity is detected.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x26—TIME_INACT (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetFIFOStatus(CHN)

Parameters

SpiChannel CHN - Spi Channel

Return Value

uint8_t - PMODACL_REG_FIFO_STATUS register contents

Status of FIFO trigger events and number of data values available for collection from the FIFO.

FIFO_STATUS register*

D7	D6	D5	D4	D3	D2	D1	D0
FIFO_TRIG	0	Entries					

*Values prefixed with PMODACL_BIT_TAP_AXES in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> 0x39—FIFO_STATUS (Read Only) in the ADXL345 reference manual for additional information.

PmodACLSetThreshAct(CHN,THRESH_ACT)

Parameters

SpiChannel CHN - Spi Channel
uint8_t THRESH_ACT - Threshold for detecting activity (62.5 mg/LSB)

Return Value

none

Sets the threshold for detecting activity.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> THRESH_ACT (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetThreshAct(CHN)

Parameters

SpiChannel CHN - Spi Channel

Return Value

uint8_t - Threshold for detecting activity (62.5 mg/LSB)

Gets the threshold for detecting activity.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> THRESH_ACT (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLSetActInactCtl(CHN,ACT_INACT_CTL)

Parameters

SpiChannel CHN - spiChannel

uint8_t - contents of PMODACL_REG_ACT_INACT_CTL register, which is a combination of ORed values prefixed with PMODACL_BIT_ACT_INACT_CTL defined in Defined in Table 1: PmodACL Register/Configuration Definitions.

Return Value

none

Enables or disables activity or inactivity detection for individual axes, set ac/dc coupled operations.

ACT_INACT_CTL Register*

D7	D6	D5	D4	D3	D2	D1	D0
ACT ac/dc	ACT_X enable	ACT_Y enable	ACT_Z enable	INACT ac/dc	INACT_X enable	INACT_Y enable	INACT_Z enable

*Values prefixed with PMODACL_BIT_ACT_INACT_CTL in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x27—ACT_INACT_CTL (Read/Write) in the ADXL345 reference manual for additional information.

PmodACLGetActInactCtl(CHN)

Parameters

SpiChannel CHN - spiChannel

Return Value

uint8_t -PMODACL_REG_ACT_INACT_CTL register contents

Restores state of activity or inactivity detection for individual axes, ac/dc coupled operations.

ACT_INACT_CTL Register*

D7	D6	D5	D4	D3	D2	D1	D0
ACT ac/dc	ACT_X enable	ACT_Y enable	ACT_Z enable	INACT ac/dc	INACT_X enable	INACT_Y enable	INACT_Z enable

*Values prefixed with PMODACL_BIT_ACT_INACT_CTL in Table 1: PmodACL Register/Configuration Definitions.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x27—ACT_INACT_CTL (Read/Write) in the ADXL345 reference manual for additional information.

PmodACLSetThreshInact(CHN,THRESH_INACT)

Parameters

SpiChannel CHN - Spi channel
uint8_t THRESH_INACT - Threshold for detecting inactivity (62.5 mg/LSB)

Return Value

none

Threshold for detecting inactivity.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x25—THRESH_INACT (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACLGetThreshInact(CHN)

Parameters

SpiChannel CHN - Spi channel

Return Value

uint8_t - Threshold for detecting inactivity (62.5 mg/LSB)

Threshold for detecting inactivity.

NOTE: See REGISTER MAP->REGISTER DEFINITIONS-> Register 0x25—THRESH_INACT (Read/Write) in the ADXL345 reference manual for additional information. Defined in Table 1: PmodACL Register/Configuration Definitions.

PmodACL Additional Information

PmodACL Structs

Struct PMODACL_AXIS

Fields

int16_t xAxis - xAxis register values
int16_t yAxis - yAxis register values
int16_t zAxis - zAxis register values

Table 1: PmodACL Register/Configuration Definitions

Macro Definition	Value	Type	Description
PMODACL_REG_DEVID	0x00	Register	Device ID
PMODACL_REG_THRESH_TAP	0x1D	Register	Tap threshold
PMODACL_REG_OFSX	0x1E	Register	X-axis offset
PMODACL_REG_OFSY	0x1F	Register	Y-axis offset
PMODACL_REG_OFSZ	0x20	Register	Z-axis offset
PMODACL_NUM_OFFSET_BYTES	0x03		
PMODACL_REG_DUR	0x21	Register	Tap duration
PMODACL_REG_LATENT	0x22	Register	Tap latency
PMODACL_REG_WINDOW	0x23	Register	Tap window
PMODACL_REG_THRESH_ACT	0x24	Register	Activity threshold
PMODACL_REG_THRESH_INACT	0x25	Register	Inactivity threshold
PMODACL_REG_TIME_INACT	0x26	Register	Inactivity time
PMODACL_REG_ACT_INACT_CTL	0x27	Register	Axis enable control for activity and inactivity detection
PMODACL_BITS_ACT_INACT_CTL_ACT_ACDC	0x80	Register Bits	See ADXL345 Ref Manual
PMODACL_BITS_ACT_INACT_CTL_ACT_X	0x40	Register Bits	See ADXL345 Ref Manual
PMODACL_BITS_ACT_INACT_CTL_ACT_Y	0x20	Register Bits	See ADXL345 Ref Manual
PMODACL_BITS_ACT_INACT_CTL_ACT_Z	0x10	Register Bits	See ADXL345 Ref Manual
PMODACL_BITS_ACT_INACT_CTL_INACT_ACDC	0x08	Register Bits	See ADXL345 Ref Manual
PMODACL_BITS_ACT_INACT_CTL_INACT_X	0x04	Register Bits	See ADXL345 Ref Manual
PMODACL_BITS_ACT_INACT_CTL_INACT_Y	0x02	Register Bits	See ADXL345 Ref Manual
PMODACL_BITS_ACT_INACT_CTL_INACT_Z	0x01	Register Bits	See ADXL345 Ref Manual

Macro Definition	Value	Type	Description
PMODACL_REG_THRESH_FF	0x28	Register	Free-fall threshold
PMODACL_REG_TIME_FF	0x29	Register	Free-fall time
PMODACL_REG_TAP_AXES	0x2A	Register	Axis control for single tap/double tap
PMODACL_BIT_TAP_AXES_SUPRESS	0x08	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_TAP_AXES_TAP_X	0x04	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_TAP_AXES_TAP_Y	0x02	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_TAP_AXES_TAP_Z	0x01	Register Bits	See ADXL345 Ref Manual
PMODACL_REG_ACT_TAP_STATUS	0x2B	Register	Source of single tap/double tap
PMODACL_BIT_ACT_TAP_STATUS_ACT_X	0x40	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_ACT_TAP_STATUS_ACT_Y	0x20	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_ACT_TAP_STATUS_ACT_Z	0x10	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_ACT_TAP_STATUS_ASLEEP	0x08	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_ACT_TAP_STATUS_TAP_X	0x04	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_ACT_TAP_STATUS_TAP_Y	0x02	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_ACT_TAP_STATUS_TAP_Z	0x01	Register Bits	See ADXL345 Ref Manual
PMODACL_REG_BW_RATE	0x2C	Register	Data rate and power mode control
PMODACL_BIT_BW_RATE_LOW_POWER	0x10	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_BW_RATE_3200HZ	0x0F	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_BW_RATE_1600HZ	0x0E	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_BW_RATE_800HZ	0x0D	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_BW_RATE_400HZ	0x0C	Register Bits	See ADXL345 Ref Manual (Low Power)
PMODACL_BIT_BW_RATE_200HZ	0X0B	Register Bits	See ADXL345 Ref Manual (Low Power)
PMODACL_BIT_BW_RATE_100HZ	0x0A	Register Bits	See ADXL345 Ref Manual (Low Power)
PMODACL_BIT_BW_RATE_50HZ	0x09	Register Bits	See ADXL345 Ref Manual (Low Power)
PMODACL_BIT_BW_RATE_25HZ	0x08	Register Bits	See ADXL345 Ref Manual (Low Power)
PMODACL_BIT_BW_RATE_12_5HZ	0x07	Register Bits	See ADXL345 Ref Manual (Low Power)
PMODACL_BIT_BW_RATE_6_25HZ	0x06	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_BW_RATE_3_13HZ	0x05	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_BW_RATE_1_56HZ	0x04	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_BW_RATE_0_78HZ	0x04	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_BW_RATE_0_39HZ	0x02	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_BW_RATE_0_20HZ	0x01	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_BW_RATE_0_10HZ	0x00	Register Bits	See ADXL345 Ref Manual

Macro Definition	Value	Type	Description
PMODACL_REG_POWER_CTL	0x2D	Register	Power-saving features control
PMODACL_BIT_POWER_CTL_LINK	0x20	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_POWER_CTL_AUTO_SLEEP	0x10	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_POWER_CTL_MEASURE	0x08	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_POWER_CTL_SLEEP	0x04	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_POWER_CTL_WAKEUP_8HZ	0x00	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_POWER_CTL_WAKEUP_4HZ	0x01	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_POWER_CTL_WAKEUP_2HZ	0x02	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_POWER_CTL_WAKEUP_1HZ	0x03	Register Bits	See ADXL345 Ref Manual
PMODACL_REG_INT_ENABLE	0x2E	Register	Interrupt enable control
PMODACL_BIT_INT_ENABLE_DATA_READY	0x80	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_ENABLE_SINGLE_TAP	0x40	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_ENABLE_DOUBLE_TAP	0x20	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_ENABLE_ACTIVITY	0x10	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_ENABLE_INACTIVITY	0x08	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_ENABLE_FREE_FALL	0x04	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_ENABLE_WATERMARK	0x02	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_ENABLE_OVERRUN	0x01	Register Bits	See ADXL345 Ref Manual
PMODACL_REG_INT_MAP	0x2F	Register	Interrupt mapping control
PMODACL_BIT_INT_MAP_DATA_READY	0x80	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_MAP_SINGLE_TAP	0x40	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_MAP_DOUBLE_TAP	0x20	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_MAP_ACTIVITY	0x10	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_MAP_INACTIVITY	0x08	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_MAP_FREE_FALL	0x04	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_MAP_WATERMARK	0x02	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_MAP_OVERRUN	0x01	Register Bits	See ADXL345 Ref Manual
PMODACL_REG_INT_SOURCE	0x30	Register	Source of interrupts
PMODACL_BIT_INT_SOURCE_DATA_READY	0x80	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_SOURCE_DOUBLE_TAP	0x60	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_SOURCE_SINGLE_TAP	0x40	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_SOURCE_ACTIVITY	0x10	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_SOURCE_INACTIVITY	0x08	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_SOURCE_FREE_FALL	0x04	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_SOURCE_WATERMARK	0x02	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_INT_SOURCE_OVERRUN	0x01	Register Bits	See ADXL345 Ref Manual

Macro Definition	Value	Type	Description
PMODACL_REG_DATA_FORMAT	0x31	Register	Data format control
PMODACL_BIT_DATA_FORMAT_SELF_TEST	0x80	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_DATA_FORMAT_SPI	0x40	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_DATA_FORMAT_INT_INVERT	0x20	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_DATA_FORMAT_FULL_RES	0x08	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_DATA_FORMAT_JUSTIFY	0x04	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_DATA_FORMAT_RANGE_16G	0x03	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_DATA_FORMAT_RANGE_8G	0x02	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_DATA_FORMAT_RANGE_4G	0x01	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_DATA_FORMAT_RANGE_2G	0x00	Register Bits	See ADXL345 Ref Manual
PMODACL_MASK_DATA_FORMAT_RANGE	0x07	Register Mask	See ADXL345 Ref Manual
PMODACL_REG_DATAX0	0x32	Register	X-Axis Data 0
PMODACL_REG_DATAX1	0x33	Register	X-Axis Data 1
PMODACL_REG_DATAY0	0x34	Register	Y-Axis Data 0
PMODACL_REG_DATAY1	0x35	Register	Y-Axis Data 1
PMODACL_REG_DATAZ0	0x36	Register	Z-Axis Data 0
PMODACL_REG_DATAZ1	0x37	Register	Z-Axis Data 1
PMODACL_NUM_AXIS_REGISTERS	0x06		Axis register count
PMODACL_REG_FIFO_CTL	0x38	Register	FIFO Control
PMODACL_BIT_FIFO_CTL_BYPASS	0x00	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_FIFO_CTL_FIFO	0x40	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_FIFO_CTL_STREAM	0x80	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_FIFO_CTL_TRIGGER	0xC0	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_FIFO_CTL_TRIGGER_INT2	0x10	Register Bits	See ADXL345 Ref Manual
PMODACL_REG_FIFO_STATUS	0x39	Register	FIFO Control
PMODACL_BIT_FIFO_STATUS_FIFO_TRIG	0x80	Register Bits	See ADXL345 Ref Manual
PMODACL_BIT_MASK_FIFO_STATUS_ENTRIES	0x3F	Register Bits	See ADXL345 Ref Manual
PMODACL_READ_BIT	0x80	Misc	READ/WRITE Command
PMODACL_MB_BIT	0x40	Misc	MULTBYTE Command
PMODACL_DEVICE_ID	0xE5	Misc	ADXL345 Device ID
PMODACL_CALIBRATE_X_AXIS	0x00	Misc	Calibration Orientation X Axis
PMODACL_CALIBRATE_Y_AXIS	0x01	Misc	Calibration Orientation Y Axis
PMODACL_CALIBRATE_Z_AXIS	0x02	Misc	Calibration Orientation Z Axis

PmodACL Example Use

The following example demonstrates data acquisition utilizing the PmodACL with interrupt INT1 and Cerebot MX4/MX7 INT3. Calibration is performed on the Z-axis, X and Y axis positions are reported using LD1 through LD4. {X+ = LD1 ON, X- = LD2 ON, Y+ = LD3 ON, Y- = LD4 ON} LD1 and LD2 are mutually exclusive, as are LD3 and LD4. A rolling average is used on the X and Y axis values to provide consistent LED illumination.

```
/* ------------------------------------------------------------ */
/*              PIC32 Configuration Settings           */
/* ------------------------------------------------------------ */
#pragma config FPLLMUL = MUL_20, FPLLIDIV = DIV_2, FPLLODIV = DIV_1
#pragma config FWDTEN = OFF
#pragma config POSCMOD = HS, FNOSC = PRIPLL
#pragma config FPBDIV = DIV_2

#include <stdint.h>
#include <plib.h>
#include <pmodlib.h>

#define SYSTEM_CLOCK (80000000L)                    //system clock speed (8 MHz Crystal/ FPLLIDIV * FPLLMUL / FPLLODIV)
#define PB_DIV 2                                    //Peripheral bus divisor
#define PB_CLOCK (SYSTEM_CLOCK/PB_DIV)              //Peripheral bus clock
#define SPI_BITRATE 625000                          //Bit rate for SPI port
#define ROLLING_AVG_NUM_POINTS 20                   //Number of data points to collect during rolling average

//Used for data point rolling average
typedef struct
{
  int16_t avgX;
  int16_t avgY;
}ROLL_AVG;

//flag indicating the PmodACL has Data
uint8_t aclDataReady = 0;

void configInt()
{

            INTDisableInterrupts();

            //make sure interrupt flags are cleared
            INTClearFlag(INT_INT3);

            //configure for multi vector interrupts
            INTConfigureSystem(INT_SYSTEM_CONFIG_MULT_VECTOR);

            //configure External INT3 to trigger on rising edge transition with priority 7
            //fires when PmodACL has data
            ConfigINT3(EXT_INT_PRI_7 | FALLING_EDGE_INT | EXT_INT_ENABLE);

            //get interrupt source, clears interrupt
            PmodACLGetIntSource(SPI_CHANNEL2);

            //map all interrupts to PmodACL INT1
            PmodACLSetIntMap(SPI_CHANNEL2,0);

            //Enable PmodACL interrupts for DataReady, Activity
            PmodACLSetIntEnable(SPI_CHANNEL2,PMODACL_BIT_INT_MAP_DATA_READY);

            INTEnableInterrupts();
}
```

```
//Rolling average of data points smoothes out LED transitions,
//without data point filtering the LED positions will jump around.
ROLL_AVG getDataPointRollingAverage(int16_t dataX, int16_t dataY)
{
        uint8_t pointIndex = 0;
        //index of current data point in rollAverageDataPoints
        static uint8_t currentDataPoint = 0;
        static int16_t rollAverageDataPoints[2][ROLLING_AVG_NUM_POINTS] = {{0},{0}};
        ROLL_AVG rollingAvg;

        rollingAvg.avgX = 0;
        rollingAvg.avgY = 0;

        rollAverageDataPoints[0][currentDataPoint] = dataX;
        rollAverageDataPoints[1][currentDataPoint] = dataY;

        currentDataPoint++;

        currentDataPoint %= ROLLING_AVG_NUM_POINTS;

        for(pointIndex = 0;pointIndex <  ROLLING_AVG_NUM_POINTS;pointIndex++)
        {
                rollingAvg.avgX += rollAverageDataPoints[0][pointIndex];
                rollingAvg.avgY += rollAverageDataPoints[1][pointIndex];
        }

        rollingAvg.avgX /= ROLLING_AVG_NUM_POINTS;
        rollingAvg.avgY /= ROLLING_AVG_NUM_POINTS;

        return rollingAvg;
}

int main(void)
{
        //Rolling average returned by getDataPointRollingAverage
        ROLL_AVG rollAvg;
        //PmodACL axis measurements
        PMODACL_AXIS pmodACLAxis;
        //Set digital IO for LD1 - LD4 on Cerebot32MX4
        PORTSetPinsDigitalOut(IOPORT_B,BIT_10|BIT_11|BIT_12|BIT_13);
        //Set digital IO for INT1 on PmodACL, External INT3 on Cerebot MX4/MX7
        PORTSetPinsDigitalIn(IOPORT_A,BIT_14);
        //Init PmodACL
        PmodACLInitSpi(SPI_CHANNEL2,PB_CLOCK,SPI_BITRATE);
        //Sensitivity is set at 4G, Invert interrupt signal to idle high, active low
        PmodACLSetDataFormat(SPI_CHANNEL2,PMODACL_BIT_DATA_FORMAT_RANGE_4G |
                                        PMODACL_BIT_DATA_FORMAT_INT_INVERT);
        //Place the PmodACL in measure mode
        PmodACLSetPowerCtl(SPI_CHANNEL2,PMODACL_BIT_POWER_CTL_MEASURE);
        //Bypass the FIFO
        PmodACLSetFIFOCtl(SPI_CHANNEL2,PMODACL_BIT_FIFO_CTL_BYPASS);
        //Set the data output rate to 100Hz
        PmodACLSetBwRate(SPI_CHANNEL2,PMODACL_BIT_BW_RATE_100HZ);
        //Perform a calibration based on the Z axis using 100 samples.
        PmodACLCalibrate(SPI_CHANNEL2,100,PMODACL_CALIBRATE_Z_AXIS);

        configInt();

        while(1)
        {
                //Only poll when data is actually available, INT1 on ACL fires trigging the Cerebot INT3
                //which sets this flag.
                if(aclDataReady)
                {
                        //Clear data ready flag from external in
```

```
            aclDataReady = 0;

            //Get axis measurements, reading axis measurements also clears PmodACL interrupt
            PmodACLGetAxisData(SPI_CHANNEL2,&pmodACLAxis);
            //turn LEDs on or off based on rolling average of data points
            rollAvg = getDataPointRollingAverage(pmodACLAxis.xAxis, pmodACLAxis.yAxis);
            //PmodACL measures X+ turn on LD1
            if(rollAvg.avgX > 0)
            {
                PORTClearBits(IOPORT_B,BIT_11);
                PORTSetBits(IOPORT_B,BIT_10);
            }
            //PmodACL measures X- turn on LD2
            else
            {
                PORTSetBits(IOPORT_B,BIT_11);
                PORTClearBits(IOPORT_B,BIT_10);
            }

            //PmodACL measures Y+ turn on LD3
            if(rollAvg.avgY > 0)
            {
                PORTClearBits(IOPORT_B,BIT_13);
                PORTSetBits(IOPORT_B,BIT_12);
            }
            //PmodACL measures Y- turn on LD4
            else
            {
                PORTSetBits(IOPORT_B,BIT_13);
                PORTClearBits(IOPORT_B,BIT_12);
            }
        }
    }
}

void __ISR(_EXTERNAL_3_VECTOR, ipl7) Ext3Handler_PmodACLInt1(void)
{
    //we have data, signal main loop to collect
    aclDataReady = 1;
    INTClearFlag(INT_INT3);
}
```