chipKIT™ GYRO Library Reference Manual

Revision: April 2, 2012

www.digilentinc.com

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

Overview

The GYRO library is a programming interface for an STMicroelectronics® L3G4200D Mems motion sensor. The L3G4200D is a three-axis digital-output gyroscope with built-in temperature sensor.

The gyroscope can communicate via I2C or SPI, but the GYRO library uses the latter. Regardless of the communication protocol, the gyroscope is configured by writing to its configuration registers. The temperature and angular speed can be read from its data registers.

Library Operation

Library Interface

The header file GYRO.h defines the interfaces to the GYRO library. The library is accessed via the methods and constants defined for the GYRO object class. To instantiate a GYRO object, simply include the library and instantiate a GYRO object (e.g., myGYRO, or whatever name you want).

GYRO Library Functions

Display Management Functions

bool begin(void)

Parameters:
None

Return Value:
“False” for failure and “True” for success

Initializes the SPI port and readies gyro for use.

void end(void)

Parameters:
none

Turns off the SPI port.

Int_16_t getX()

Parameters:
None

Return Value:
16-bit signed value representing X-axis angular rate

Reads the XL and XH registers then merges them into a 16-bit value.

Int_16_t getY()

Parameters:
None

Return Value:
16-bit signed value representing Y-axis angular rate

Reads the YL and YH registers then merges them into a 16-bit value.

Int_16_t getZ()

Parameters:
None

Return Value:
16-bit signed value representing Z-axis angular rate

Reads the ZL and ZH registers then merges them into a 16-bit value.

Int_8_t getTemp()

Parameters:
none

Return Value:
8-bit signed value representing temperature in degrees C

Reads the Temp register.

bool enableInt1(uint8_t mode)

Parameters:
Mode This parameter can be filled with any of the following parameters:

```
INT1_ANDOR : AND/OR combination of Interrupt events
INT1_LIR   : Latch Interrupt Request
INT1_ZHIE  : Enable interrupt generation on Z high event
INT1_ZLIE  : Enable interrupt generation on Z low event
INT1_YHIE  : Enable interrupt generation on Y high event
INT1_YLIE  : Enable interrupt generation on Y low event
INT1_XHIE  : Enable interrupt generation on X high event
INT1_XLIE  : Enable interrupt generation on X low event
```

For more information on these parameters, please refer to your product's data sheet.

Return Value:
"False" for failure and "True" for success

Enables interrupt one and sets the In1_CFG register with the given values.

Void disableInt1()

Parameters:
none

Disables interrupt 1.

bool enableInt2(uint8_t mode)

Parameters:
Mode This parameter can be filled with any of the following parameters:

```
REG3_I2_DRDY : Interrupt generated on data ready
REG3_I2_WTM  : Interrupt generated on watermark
REG3_I2_ORUN : Interrupt generated on buffer overun
REG3_I2_EMPTY: Interrupt generated on buffer empty
```

For more information on these parameters, please refer to your product's data sheet.

Return Value:
"False" for failure and "True" for success

Enables interrupt one and sets the CTLR_REG3 register with the given values.

Void disableInt2()

Parameters:
none

Disables interrupt 2.

Void setThsXL(uint8_t ths)

Parameters:

Ths	Sets the minimum X-axis value before the device triggers interrupt 1.

Sets the XL THS register.

Void setThsXH(uint8_t ths)

Parameters:

Ths	Sets the maximum X-axis value before the device triggers interrupt 1.

Sets the XH THS register.

Void setThsYH(uint8_t ths)

Parameters:

Ths	Sets the maximum Y-axis value before the device triggers interrupt 1.

Sets the YH THS register.

Void setThsZH(uint8_t ths)

Parameters:

Ths	Sets the maximum Z-axis value before the device triggers interrupt 1.

Sets the ZH THS register.

Void setThsZL(uint8_t ths)

Parameters:

Ths	Sets the maximum Z-axis value before the device triggers interrupt 1.

Sets the ZL THS register.