

Advanced Accelerometer Features

Stensat Group LLC, Copyright 2018

Accelerometer Advanced Functions

- The MMA8451 accelerometer has some advanced detection features.
 - Free fall detection
 - The accelerometer can be configured to detect when it is in free fall.
 - Portrait/Landscape orientation detection
 - The accelerometer can be used to detect how it is oriented. This feature targets tablets and cell phones.
 - Tap detection
 - The accelerometer can be configured to detect a single tap and double taps in any axis.
 - Datasheet
 - <https://www.nxp.com/docs/en/data-sheet/MMA8451Q.pdf>
 - Application Notes
 - <https://www.nxp.com/docs/en/application-note/AN4070.pdf>
 - <https://www.nxp.com/docs/en/application-note/AN4072.pdf>

Tap Detection

- The example program is called **mma8451_taps**. Open the program in the Arduino IDE.
- The program configures all the registers to detect a single tap and double tap in the Z axis. The program can be modified to support the other axis.
- Line 7 sets the address of the accelerometer.
- Lines 9-14 is a function that writes to a specified register in the accelerometer. Since multiple registers need to be set, it is easier to create a function that handles the I2C operation for writing to the accelerometer.
- Lines 16-23 is a function to read the registers from the accelerometer. Only one byte is read at a time with this function.
- Lines 25-43 is a function that initializes the accelerometer and sets up the tap detection.

Tap Detection

- The `initACC()` function sets the registers to configure the accelerometer for tap detection.
- The first register (0x2a) is `CTRL_REG1`. It is set to zero which puts the accelerometer into standby mode. Bit 0 sets the active/standby mode.
- `F_READ` is a bit to set the accelerometer in fast read mode. In this mode only 8-bits of the 14 bits of data is read. This is set to zero.
- `LNOISE` bit is set to zero for now.
- `DR1` and `DR0` bits set the sample speed. The bits are set to zero for 800 samples/sec.
- `ASLP_RATE1` and `ASLP_RATE2` are not used and set to zero.

0x2A: CTRL_REG1 register (read/write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ASLP_RATE1	ASLP_RATE0	DR2	DR1	DR0	LNOISE	F_READ	ACTIVE



Tap Detection

- Line 27 set the scale factor of the accelerometer to 2Gs.
- The XYZ_DATA_CFG register is shown below. Bits 1 and 0 configure the scaling of the accelerometer.
 - 00 = 2G
 - 01 = 4G
 - 10 = 8G
- Bit 4 turns on the high pass filter when set to 1. The high pass filter is not used at this time.

0x0E: XYZ_DATA_CFG (read/write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	HPF_OUT	0	0	FS1	FS0



Tap Detection

- Lines 30,31,32 set the threshold detection of the taps for each axis.
- Valid range is zero to 127.
- The resolution is .063 Gs. Multiply the threshold value by 0.063 to get the amount of force required to detect a tap. 127 would be 8Gs.
- Register 0x23 is for the X axis.
- Register 0x24 is for the Y axis.
- Register 0x25 is for the Z axis.
- The program sets the registers to 0x08. A tap of at least 0.504 Gs is required to be detected.

0x23: PULSE_THSX register (read/write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	THSX6	THSX5	THSX4	THSX3	THSX2	THSX1	THSX0



Tap Detection

- Register 0x21 is the register that enables which axis and single or double tap detection.
- DPA bit is an abort register for double tap and is set to zero.
- ELE bit is set to 1 to allow latching of events in the PULSE_SRC register.
- ZDPEFE bit is set to 1 to enable detection of a double tap in the Z axis.
- ZSEFE bit is set to 1 to enable detection of a single tap in the Z axis.
- The other bits perform the same functions for the Y and X axis.
- One or more axis can be enabled at once.

0x21: PULSE_CFG register (read/write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPA	ELE	ZDPEFE	ZSEFE	YDPEFE	YSPEFE	XDPEFE	XSPEFE



Tap Detection

- PULSE_TMLT register specifies the amount of time the accelerometer must detect motion above the set threshold. Refer to page 41 of the datasheet.
- With the sample rate at 800 sample/sec, the time resolution is 0.625 ms. The register value is 0x30 which is 48 decimal. Multiply 0.625 by 48 and the result is 30 ms. The tap must set the accelerometer to detect an acceleration above tbd for 30ms to register as a tap.

0x26: PULSE_TMLT register (read/write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMLT7	TMLT6	TMLT5	TMLT4	TMLT3	TMLT2	TMLT1	TMLT0

Tap Detection

- This PULSE_LTCY register is a time delay setting. During the time period specified, any other taps are ignored. This register needs to be set for single or double tap detection. Page 42 of the datasheet has the tables for the timing dependent on the accelerometer configuration.
- At 800 samples/sec, the resolution is 1.25 ms. The register is set to 0xa0 which is 160. Multiply 160 by 1.25 ms for 200ms delay.

0x27: PULSE_LTCY register (read/write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LTCY7	LTCY6	LTCY5	LTCY4	LTCY3	LTCY2	LTCY1	LTCY0

Tap Detection

- PULSE_WIND register specifies the time after the previous delay for the amount of time for the second tap. If a tap is not detected before the time expires, the tap is not detected as a second tap.

0x28: PULSE_WIND second pulse time window register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WIND7	WIND6	WIND5	WIND4	WIND3	WIND2	WIND1	WIND0

Interrupts

- This program uses an interrupt signal from the sensor to the processor. Digital pin 14 is used to detect the interrupt signal from the sensor. Other digital pins can be used.
- An interrupt is a method to force the processor to execute a function when a specific event occurs. For this example, the sensor will generate a low signal on the **INT1** pin which is connected to **pin 14** on the SLATE. When ever the **INT1** signal goes from high to low, that indicates the sensor detected something and has new data. The processor will detect the interrupt signal and stop executing what ever code was being executed and execute the function assigned to the interrupt. Once the function is done, the processor goes back to the code it was executing.

Interrupts

- Line 41 sets up the interrupt with the **attachInterrupt()** function. The first argument specifies the pin to be used as the interrupt.
- The argument contains the function **digitalPinToInterrupt()** that assigns pin 14 to be an interrupt signal.
- The second argument is the name of the function to be executed.
- The third argument sets how the interrupt is triggered. **FALLING** indicates the trigger is the signal transitioning from high to low. **RISING** would be for the interrupt to occur when the signal transitions from low to high.

Interrupts

- The accelerometer sensor needs to be configured to generate the interrupt. Two registers are set in lines 39 and 40.
- CTRL_REG4 register is used to select what event generates the interrupt. Setting bit 3 to 1 enables interrupts to be generated when taps are detected.
- CTRL_REG5 register is used to route which interrupt pin to use for each interrupt source. Bit 3 is for the tap detection and setting it to 1 will route the interrupt signal to INT1 signal. Having INT1 and INT2 allows for two interrupts to be used simultaneously.

0x2D: CTRL_REG4 register (read/write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INT_EN_ASLP	INT_EN_FIFO	INT_EN_TRANS	INT_EN_LNDPRT	INT_EN_PULSE	INT_EN_FF_MT	—	INT_EN_DRDY

0x2E: CTRL_REG5 interrupt configuration register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INT_CFG_ASLP	INT_CFG_FIFO	INT_CFG_TRANS	INT_CFG_LNDPRT	INT_CFG_PULSE	INT_CFG_FF_MT	—	INT_CFG_DRDY



Interrupts

- Lines 50 – 54 is the interrupt function.
- It reads the status register for the tap detection.
- The status register bit 7 indicates an interrupt occurred. The other bits indicate which tap event occurred.
- Bits 4 – 6 indicate which axis the tap occurred. More than one axis can be active.
- Bit 3 indicates if the tap was a double tap.
- Bits 0 – 2 indicate the direction of the tap.

0x22: PULSE_SRC register (read only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EA	AxZ	AxY	AxX	DPE	PolZ	PolY	PolX

loop() Function

- Nothing is in the loop() function. The function needs to be included even if nothing is executed. The processor will continuously execute the empty function until an interrupt occurs.
- You can add code to do other things.

Tap Detection

- Connect the accelerometer to the I2C bus. Connect INT1 to pin 14.
- Run the program. Every time a tap or double tap is detected, the interrupt will be called and the register indicating what was detected will be updated. The register is cleared after reading the register automatically.
- The interrupt displays the interrupt in hexadecimal

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EA	AxZ	AxY	AxX	DPE	PolZ	PolY	PolX

Portrait/Landscape Orientation Detection

- Open the next program called mm8451_orientation.
- This program is a modification of the m8451_taps program. All configurations related to tap detection are commented out.
- Lines 28 and 29 are now uncommented. The registers configure the sensor for detecting orientation.
- Lines 39 and 40 control the interrupt operation and the register values are modified to generate an interrupt when orientation change is detected.
- The interrupt function is modified to indicate the orientation detected.

Register Configuration

- Two registers are needed to configure the sensor to detect orientation.
- Bit 6 in the PL_CFG register is set to 1. This enables the logic in the sensor to detect orientation changes.
- PL_COUNT register is set to 0x50 which is 80 decimal. At 800 samples/second, the delay is 100 ms. The accelerometer has some noise so when it is rotated, the orientation detection logic can detect the change multiple times called bouncing between two states. Waiting 100ms will let the sensor settle and the detection to stop changing.

0x11: PL_CFG register (read/write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBCNTM	PL_EN	—	—	—	—	—	—

0x12: PL_COUNT register (read/write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBNCE[7]	DBNCE[6]	DBNCE[5]	DBNCE[4]	DBNCE[3]	DBNCE[2]	DBNCE[1]	DBNCE[0]



Testing Orientation Change

- Load the program and run it.
- Hold the SLATE board vertically so the z-axis of the sensor is horizontal. Rotate the SLATE board and observe the serial monitor. The register value and the orientation will be displayed as the board is rotated.

Free Fall Detection

- Check out the application note on configuring the program to detect free fall. Modify the current program to support free fall detection.
- <https://www.nxp.com/docs/en/application-note/AN4070.pdf>