IC Datasheet Study

LIS302DL : 3-axis accelerometer

Page 16 : IC interface

3.2 IC interface

The complete measurement chain is composed by a low-noise capacitive amplifier which converts into an analog voltage the capacitive unbalancing of the MEMS sensor and by analog-to-digital converters.

The acceleration data may be accessed through an I^2C/SPI interface thus making the device particularly suitable for direct interfacing with a microcontroller.

The LIS302DL features a Data-Ready signal (RDY) which indicates when a new set of measured acceleration data is available thus simplifying data synchronization in the digital system that uses the device.

The LIS302DL may also be configured to generate an inertial Wake-Up and Free-Fall interrupt signal accordingly to a programmed acceleration event along the enabled axes. Both Free-Fall and Wake-Up can be available simultaneously on two different pins.

Information : how to access data?
Conclusion : by using I2C or SPI interface.

Page 18 : Digital interfaces

5 Digital interfaces

The registers embedded inside the LIS302DL may be accessed through both the I^2C and SPI serial interfaces. The latter may be SW configured to operate either in 3-wire or 4-wire interface mode.

The serial interfaces are mapped onto the same pads. To select/exploit the I^2C interface, CS line must be tied high (i.e connected to Vdd_IO).

Table 8. Serial interface pin description

PIN name	PIN description
CS	SPI enable I^2C/SPI mode selection (1: I^2C mode; 0: SPI enabled)
SCL/SPC	I^2C Serial Clock (SCL) SPI Serial Port Clock (SPC)
SDA/SDI/SDO	I^2C Serial Data (SDA) SPI Serial Data Input (SDI) 3-wire Interface Serial Data Output (SDO)
SDO	SPI Serial Data Output (SDO)

Information : how I2C or SPI serial interfaces look like?
Conclusion : the description is in the above table.

Page 22 : SPI Read

The SPI Read command is performed with 16 clock pulses. Multiple byte read command is performed adding blocks of 8 clock pulses at the previous one.

bit 0: READ bit. The value is 1.

bit 1: MS̄ bit. When 0 do not increment address, when 1 increment address in multiple reading.

bit 2-7: address AD(5:0). This is the address field of the indexed register.

bit 8-15: data DO(7:0) (read mode). This is the data that will be read from the device (MSb first).

bit 16-... : data DO(...-8). Further data in multiple byte reading.

Information : how to read data by using SPI?
Conclusion : can be done by this 4 steps.

1. Bring the CS pin to low (reset) to activate IC.
2. Declare transmit address (0x20) and stored data (0x67).
3. Write declared address in step 3 to register.
4. Bring the CS pin back to high (set) to finish initializing.

Reference : www.youtube.com/watch?v=OBuuWzyPMPg

Page 24 : Register Mapping

OutX	r	29	010 1001	output	
--	r	2A	010 1010		Not Used
OutY	r	2B	010 1011	output	
--	r	2C	010 1100		Not Used
OutZ	r	2D	010 1101	output	

Information : what address that is an output value in each axis?
Conclusion : X-axis = 0x29 , Y-axis = 0x2B , Z-axis = 0x2D.

Page 26 : CTRL_REG1

7.2 CTRL_REG1 (20h)

Table 18. CTRL_REG1 (20h) register

DR	PD	FS	STP	STM	Zen	Yen	Xen

Table 19. CTRL_REG1 (20h) register description

DR	Data rate selection. Default value: 0 (0: 100 Hz output data rate; 1: 400 Hz output data rate)
PD	Power Down Control. Default value: 0 (0: power down mode; 1: active mode)
FS	Full Scale selection. Default value: 0 (refer to *Table 3* for typical full scale value)
STP, STM	Self Test Enable. Default value: 0 (0: normal mode; 1: self test P, M enabled)
Zen	Z axis enable. Default value: 1 (0: Z axis disabled; 1: Z axis enabled)
Yen	Y axis enable. Default value: 1 (0: Y axis disabled; 1: Y axis enabled)
Xen	X axis enable. Default value: 1 (0: X axis disabled; 1: X axis enabled)

Information : giving more deeper information about control register.
Conclusion : the description is in the above table.

Page 29-30 : OUT_X , OUT_Y , OUT_Z

7.7 OUT_X (29h)

Table 28. OUT_X (29h) register

XD7	XD6	XD5	XD4	XD3	XD2	XD1	XD0

X axis output data.

7.8 OUT_Y (2Bh)

Table 29. OUT_Y (2Bh) register description

YD7	YD6	YD5	YD4	YD3	YD2	YD1	YD0

Y axis output data.

7.9 OUT_Z (2Dh)

Table 30. OUT_Z (2Dh) register

ZD7	ZD6	ZD5	ZD4	ZD3	ZD2	ZD1	ZD0

Z axis output data.

Information : giving more deeper information about output register.
Conclusion : the description is in the above table.

MP45DT02 : digital microphone

Unfortunately, MP45DT02 Datasheet doesn't give any useful information.

So, I will use reference data from AN3998 Application note instead.

reference :
http://www.st.com/content/ccc/resource/technical/document/application_note/ca/18/be/bb/f8/53/47/a5/DM00040808.pdf/files/DM00040808.pdf/jcr:content/translations/en.DM00040808.pdf

Page 4 : PDM signal introduction

2 PDM signal introduction

Pulse density modulation, or PDM, is a form of modulation used to represent an analog signal in the digital domain.

In a PDM signal, specific amplitude values are not encoded into pulses as they would be in PCM. Instead it is the relative density of the pulses that corresponds to the analog signal's amplitude.

To get the framed data from the PDM bit stream, decimation filters are usually used. The first stage of decimation is used to reduce the sampling frequency, followed by a high pass filter to remove the signal DC offset.

Information : what is PDM signal?
Conclusion : used to represent an analog signal in digital domain.

Page 5 : Hardware interface

3 Hardware interface: microphone connection and acquisition

The MP45DT02 MEMS microphone outputs a PDM signal, which is a high frequency (1 to 3.25 MHz) stream of 1-bit digital samples.

This output is acquired in blocks of 8 samples by using a synchronous serial port (SPI or I2S) of the STM32 microcontroller. The microphone's PDM output is synchronous with its input clock; therefore an STM32 SPI/ I2S peripheral generates a clock signal for the microphone.

Information : how to access data?
Conclusion : by using I2C or SPI interface.

Page 6 : Software interface

4 Software interface: digital signal processing

The data coming from the microphone is sent to the decimation process, which consists of two parts: a decimation filter converting 1-bit PDM data to PCM data, followed by two individually configurable IIR filters (low pass and high pass). The reconstructed audio is in 16-bit pulse-code modulation (PCM) format. After the conversion, it produces raw data that can be handled depending on the application implementation (stored as wave/compressed data in a mass storage media, transferred to an external audio codec DAC through I2S peripheral...).

Information : how to use the data?
Conclusion : by converting PDM data to PCM data.

Page 7 : Library Description

5 PDM audio software decoding library description

The PDM library is composed of a structure and the implementation of four PDM filter functions. The library uses two buffers, the PDM Input buffer and the PCM Output buffer; the application must define these buffers in the main program.

- Input buffer (data) is a uint8 variable with a length equal to (Output frequency / 1000 * decimation factor * Input Microphone Channels / 8) at least.
- Output buffer (dataOut) is a uint16 variable with a length equal to (Output frequency / 1000 * Output Microphone Channels) at least.

Information : how to use build in library?
Conclusion : we need PDM,PCM buffer.

CS43L22 : audio DAC, speaker driver

Page 22 : Beep Generator

4.2.1 Beep Generator

The Beep Generator generates audio frequencies across approximately two octave major scales. It offers three modes of operation: Continuous, multiple and single (one-shot) beeps. Sixteen on and eight off times are available.

Note: The Beep is generated before the limiter and may affect desired limiting performance. If the limiter function is used, it may be required to set the beep volume sufficiently below the threshold to prevent the peak detect from triggering. Since the master volume control, MSTxVOL[7:0], will affect the beep volume, DAC volume may alternatively be controlled using the PCMxVOL[6:0] bits.

Information : how to implement this speaker?
Conclusion : by using beep generator.

Page 32 : Required Initialization Settings

4.11 Required Initialization Settings

Various sections in the device must be adjusted by implementing the initialization settings shown below after power-up sequence step 3. All performance and power consumption measurements were taken with the following settings:

1. Write 0x99 to register 0x00.
2. Write 0x80 to register 0x47.
3. Write '1'b to bit 7 in register 0x32.
4. Write '0'b to bit 7 in register 0x32.
5. Write 0x00 to register 0x00.

Information : how to initialize device?
Conclusion : by doing 5 steps guideline above.

Page 33 : Control Port Operation

5. CONTROL PORT OPERATION

The control port is used to access the registers allowing the CS43L22 to be configured for the desired operational modes and formats. The operation of the control port may be completely asynchronous with respect to the audio sample rates. However, to avoid potential interference problems, the control port pins should remain static if no operation is required.

The control port operates using an I²C interface with the CS43L22 acting as a slave device.

Information : how to control port operation?
Conclusion : by using I2C interface.

Page 47 : Beep Frequency

7.15.1 Beep Frequency

Sets the frequency of the beep signal.

FREQ[3:0]	**Frequency** *(Fs = 12, 24, 48 or 96 kHz)*	**Pitch**
0000	260.87 Hz	C4
0001	521.74 Hz	C5
0010	585.37 Hz	D5
0011	666.67 Hz	E5
0100	705.88 Hz	F5
0101	774.19 Hz	G5
0110	888.89 Hz	A5
0111	1000.00 Hz	B5
1000	1043.48 Hz	C6
1001	1200.00 Hz	D6
1010	1333.33 Hz	E6
1011	1411.76 Hz	F6
1100	1600.00 Hz	G6
1101	1714.29 Hz	A6
1110	2000.00 Hz	B6
1111	2181.82 Hz	C7
Application:	"Beep Generator" on page 22	

Information : how to tell the device to play correct note sound?
Conclusion : the description is in the above table.

Page 48 : Beep On Time

7.15.2 Beep On Time

Sets the on duration of the beep signal.

ONTIME[3:0]	**On Time** *(Fs = 12, 24, 48 or 96 kHz)*
0000	~86 ms
0001	~430 ms
0010	~780 ms
0011	~1.20 s
0100	~1.50 s
0101	~1.80 s
0110	~2.20 s
0111	~2.50 s
1000	~2.80 s
1001	~3.20 s
1010	~3.50 s
1011	~3.80 s
1100	~4.20 s
1101	~4.50 s
1110	~4.80 s
1111	~5.20 s
Application:	"Beep Generator" on page 22

Information : how to tell the device to play sound in a specific time?
Conclusion : need as long as it provide, using 0x0F.