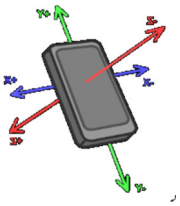
IC interface on STM32F4 Discovery board:



LIS302DL: 3-axis accelerometer

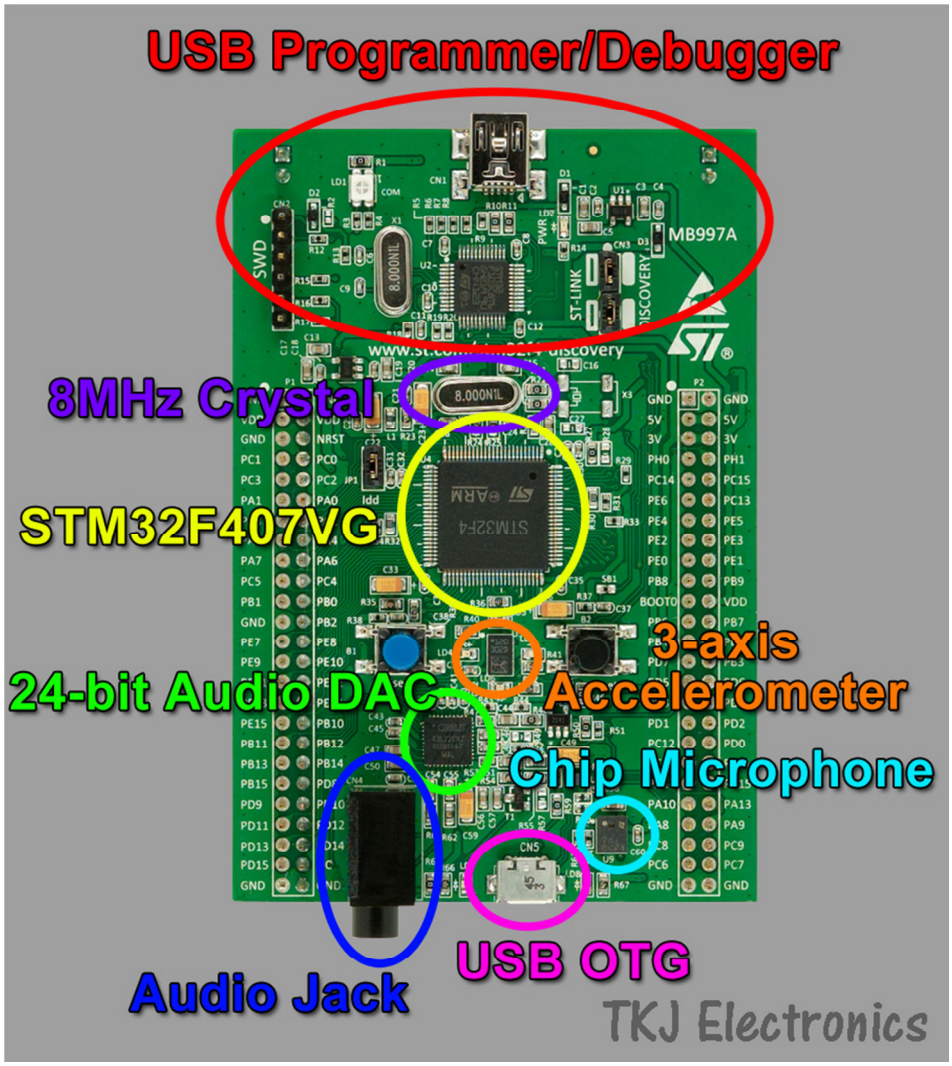
MP45DT02: digital microphone

CS43L22: audio DAC, speaker driver

By

Tharinee Munkwamdee 5731055721

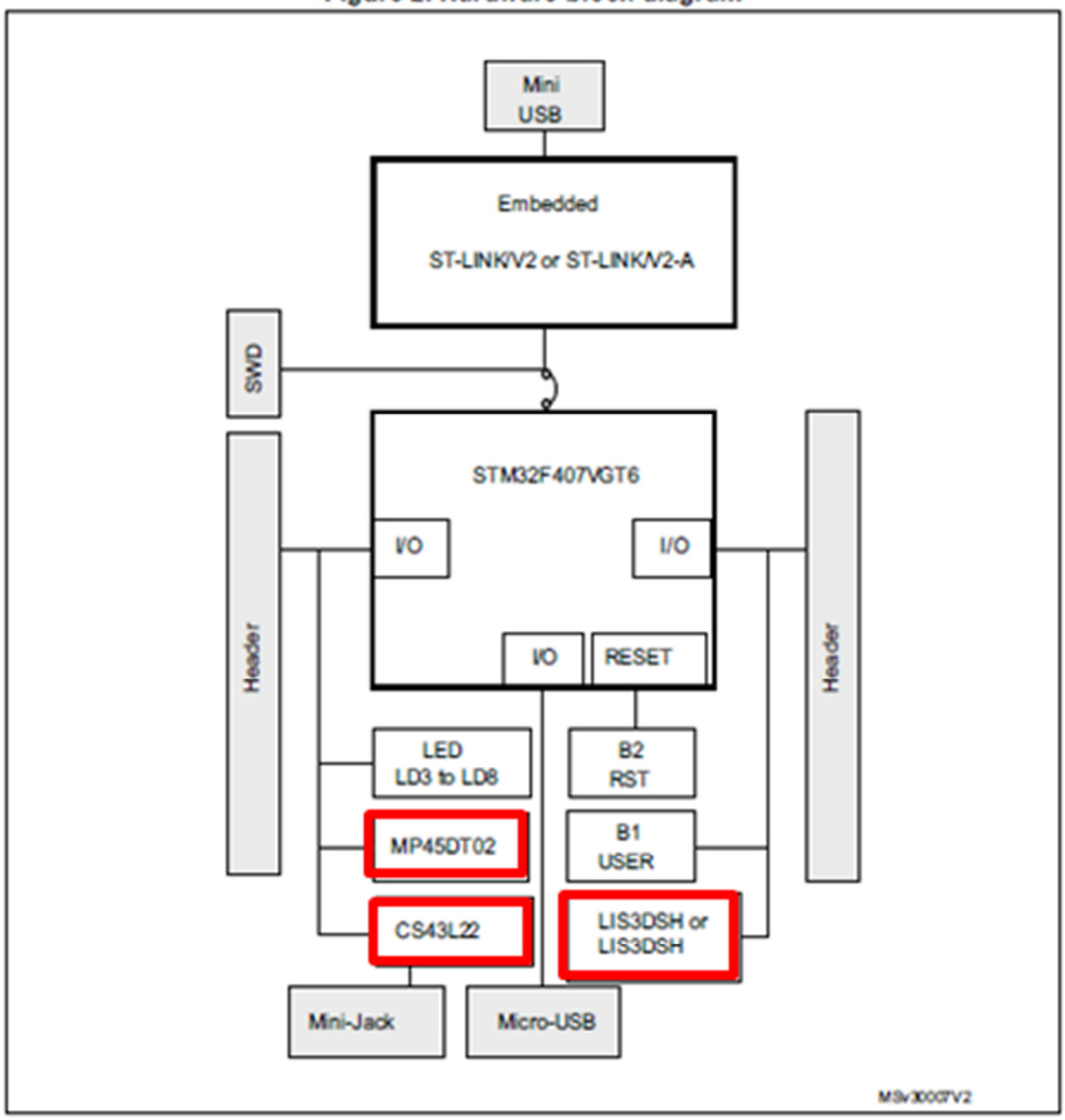
Overview of STM32F4 Discovery board



Credit: http://blog.tkjelectronics.dk/2012/04/review-stm32f4-discovery/

The STM32F4 Discovery board contains on-board peripherals as shown in the image above.

Figure 2. Hardware block diagram



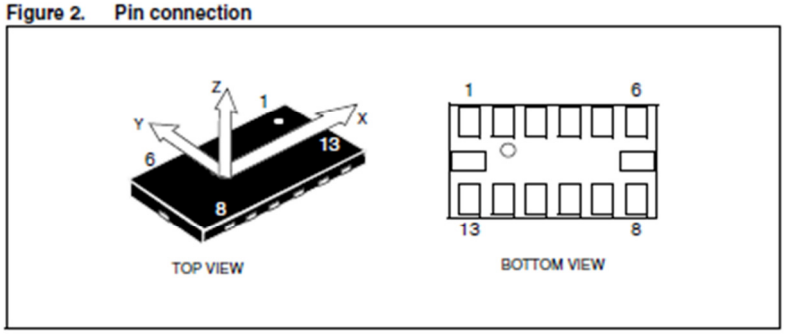
Credit: **UM1472 User manual**

Discovery kit with STM32F407VG MCU (pg.9/39)

http://www.st.com/content/ccc/resource/technical/document/user_manual/70/fe/4a/3f/e7/e1/4f/7d/DM00039084.pdf/files/DM00039084.pdf/jcr:content/translations/en.DM00039084.pdf

LIS302DL: 3-axis accelerometer

Figure 2. Pin connection



Credit: **LIS302DL**

MEMS motion sensor

3-axis - ± 2g/± 8g smart digital output "piccolo" accelerometer (pg.8/42)

http://www.st.com/content/ccc/resource/technical/document/datasheet/63/67/d2/6d/88/e0/4e/39/CD00135460.pdf/files/CD00135460.pdf/jcr:content/translations/en.CD00135460.pdf

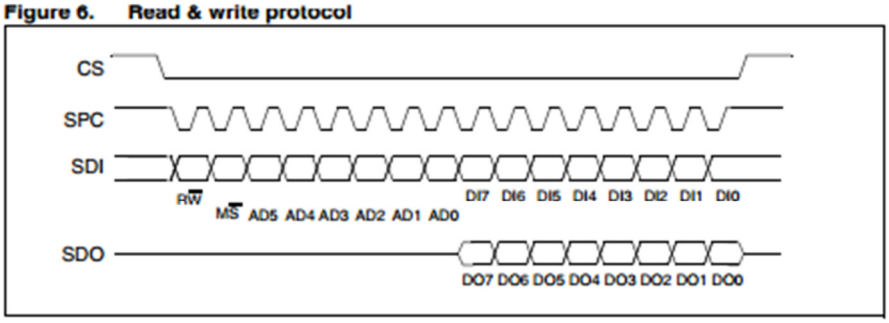
- In STM32F4 Discovery board, there is motion sensor (ST MEMS LIS302DL or LIS3DSH) in the middle of 4 LEDs.
- 2 different versions of motion sensors are available on the board -LIS302DL--PCB revision B, and LIS3DSH--PCB revision C.
- As for this report, LIS302DL will be focused.
- The LIS302D is an ultra-compact low-power 3-axis linear accelerometer.
- The motion sensor includes a sensing element and an IC interface.
- The LIS302DL can be configured in full scales of +-2g/+-8g and it is capable of measuring acceleration with an output rate of 100Hz to 400Hz.
- The registers embedded inside the LIS302DL may be accessed through both the I2C and SPI serial interfaces.

- For SPI,

Table 8. Serial interface pin description

PIN name	PIN description
CS	SPI enable I²C/SPI mode selection (1: I²C mode; 0: SPI enabled)
SCL/SPC	I²C Serial Clock (SCL) SPI Serial Port Clock (SPC)
SDA/SDI/SDO	I²C Serial Data (SDA) SPI Serial Data Input (SDI) 3-wire Interface Serial Data Output (SDO)
SDO	SPI Serial Data Output (SDO)

Figure 6. Read & write protocol



CS is the Serial Port Enable and it is controlled by the SPI master. It goes low at the start of the transmission and goes back high at the end. **SPC** is the Serial Port Clock and it is controlled by the SPI master. It is stopped high when **CS** is high (no transmission). **SDI** and **SDO** are respectively the Serial Port Data Input and Output. Those lines are driven at the falling edge of **SPC** and should be captured at the rising edge of **SPC**.

Both the Read Register and Write Register commands are completed in 16 clock pulses or in multiple of 8 in case of multiple byte read/write. Bit duration is the time between two falling edges of **SPC**. The first bit (bit 0) starts at the first falling edge of **SPC** after the falling edge of **CS** while the last bit (bit 15, bit 23, ...) starts at the last falling edge of SPC just before the rising edge of **CS**.

- For I2C,

Table 9. Serial interface pin description

Term	Description
Transmitter	The device which sends data to the bus
Receiver	The device which receives data from the bus
Master	The device which initiates a transfer, generates clock signals and terminates a transfer
Slave	The device addressed by the master

- In my project, I use SPI protocol.
- Registers used to configure the accelerometer includes:

- CTRL_REG1 (20h) is used to initially set up the sensor.

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)

- OUT_X (29h) is used to store x axis output data.

Table 28. OUT_X (29h) register

XD7	XD6	XD5	XD4	XD3	XD2	XD1	XD0

- OUT_Y (2Bh) is used to store y axis output data.

Table 29. OUT_Y (2Bh) register description

YD7	YD6	YD5	YD4	YD3	YD2	YD1	YD0

- OUT_Z (2Dh) is used to store z axis output data.

Table 30. OUT_Z (2Dh) register

ZD7	ZD6	ZD5	ZD4	ZD3	ZD2	ZD1	ZD0

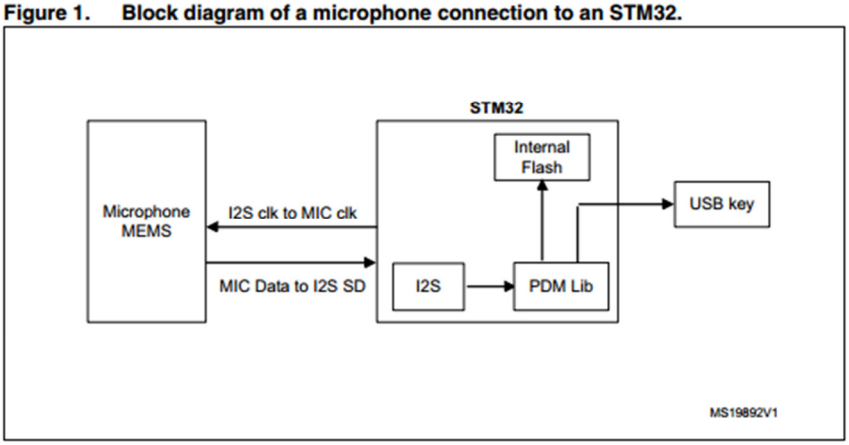
MP45DT02: digital microphone

http://pdf1.alldatasheet.com/datasheet-pdf/view/519456/STMICROELECTRONICS/MP45DT02.html

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

- The sound can come independently from different inputs:
  - digital using PDM protocol or
    - PDM : Pulse Density Modulation = represent analog signal in digital domain
    - In order to convert these PDM signal, performing sampling by choosing 16-bit data and shift 1 bit to next data and find accumulated sum of such data. And to find the amplitude of the sine wave, we have to take absolute of such sum. Such amplitude can be considered volume of the sound wave.

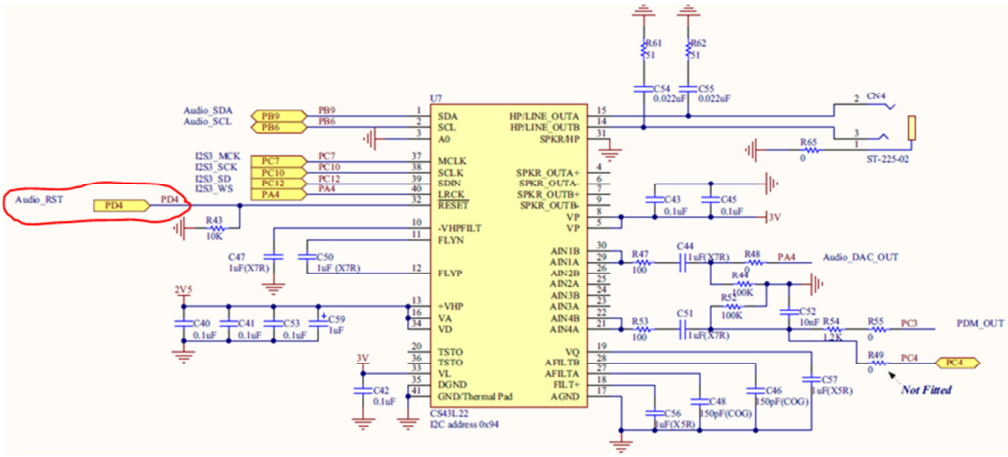
Figure 1. Block diagram of a microphone connection to an STM32.



  - analog when using the low pass filter
  - USB connector: from external mass storage such as a USB key, USB HDD, and so on
  - Internal memory of the STM32F4

CS43L22: audio DAC, speaker driver

http://pdf1.alldatasheet.com/datasheet-pdf/view/255533/CIRRUS/CS43L22.html



- The STM32F4 uses an audio DAC (CS43L22) to output sounds through the audio mini jack connector.
- The STM32F4 controls the audio DAC through the I2C interface

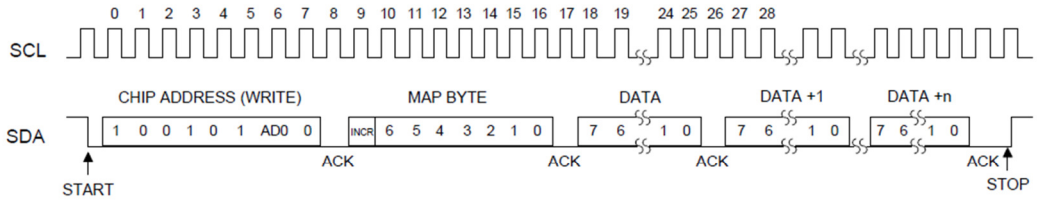


Figure 16. Control Port Timing, I²C Write

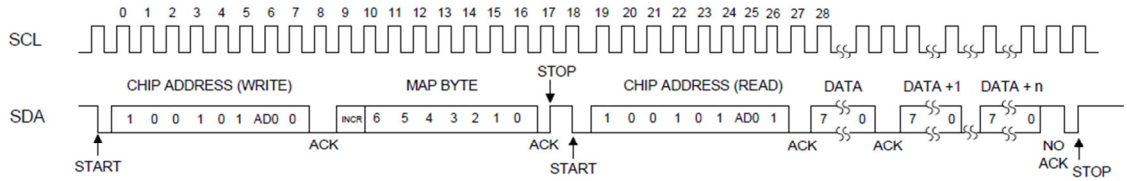


Figure 17. Control Port Timing, I²C Read

- And processes digital signals through an I2S connection or an analog input signal.
- R/W bit -> R=1 W=0, AD0 = '0' -> connected to DGND, '1' -> connected to VL
- So, in this case, the address for speaker should be 0x94 (1001_0100)
- If the operation is a write, the next byte is the Memory Address Pointer (MAP), which selects the register to be read or written.
- Setting the auto-increment bit in MAP allows successive reads or writes of consecutive registers.

- Initialization setting
  - Write 0x99 to register 0x00.
  - Write 0x80 to register 0x47.
  - Write ‘1’b to bit 7 in register 0x32.
  - Write ‘0’b to bit 7 in register 0x32.
  - Write 0x00 to register 0x00.
- In conclusion, instruction to communicate with speaker is:

```
uint8_t data_to_send[2];
data_to_send[0] = [addr of reg]; data_to_send[1] = [data to be written];
HAL_I2C_Master_Transmit(&hi2c1, 0x94, data_to_send, 2, 50);
```

- Registers used:

7.15 Beep Frequency & On Time (Address 1Ch)

7	6	5	4	3	2	1	0
FREQ3	FREQ2	FREQ1	FREQ0	ONTIME3	ONTIME2	ONTIME1	ONTIME0

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	

Notes:

1. This setting must not change when BEEP is enabled.
2. Beep frequency will scale directly with sample rate, Fs, but is fixed at the nominal Fs within each speed mode.

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

Notes:

1. This setting must not change when BEEP is enabled.
2. Beep on time will scale inversely with sample rate, Fs, but is fixed at the nominal Fs within each speed mode.

7.17 Beep & Tone Configuration (Address 1Eh)

7	6	5	4	3	2	1	0
BEEP1	BEEP0	BEEPMIXDIS	TREBCF1	TREBCF0	BASSCF1	BASSCF0	TCEN

7.17.1 Beep Configuration

Configures a beep mixed with the HP/Line and SPK output.

BEEP[1:0]	Beep Occurrence
00	Off
01	Single
10	Multiple
11	Continuous
Application:	"Beep Generator" on page 22

Notes:

1. When used in analog pass through mode, the output alternates between the signal from the Passthrough Amplifier and the beep signal. The beep signal does not mix with the analog signal from the Passthrough Amplifier.
2. Re-engaging the beep before it has completed its initial cycle will cause the beep signal to remain ON for the maximum ONTIME duration.

7.17.2 Beep Mix Disable

Configures how the beep mixes with the serial data input.

BEEPMIXDIS	Beep Output to HP/Line and Speaker
0	Mix Enabled; The beep signal mixes with the digital signal from the serial data input.
1	Mix Disabled; The output alternates between the signal from the serial data input and the beep signal. The beep signal does not mix with the digital signal from the serial data input.
Application:	"Beep Generator" on page 22

Note: This setting must not change when BEEP is enabled.

7.17.3 Treble Corner Frequency

Sets the corner frequency (-3 dB point) for the treble shelving filter.

TREBCF[1:0]	Treble Corner Frequency Setting
00	5 kHz
01	7 kHz
10	10 kHz
11	15 kHz

7.17.4 Bass Corner Frequency

Sets the corner frequency (-3 dB point) for the bass shelving filter.

BASSCF[1:0]	Bass Corner Frequency Setting
00	50 Hz
01	100 Hz
10	200 Hz
11	250 Hz

7.17.5 Tone Control Enable

Configures the treble and bass activation.

TCEN	Bass and Treble Control
0	Disabled
1	Enabled
Application:	"Beep Generator" on page 22

- The sound can be output in different ways through audio DAC:
  - Using I2S protocol
  - Using the STM32F4 DAC to analog input AIN1x of the CS43L22
  - Using the microphone output directly via a low pass filter to analog input AIN4x of the CS43L22