List of Arduino APIs:

I2C

#include <Wire.h>

Wire.beginTransmission(addr); // Begin a transmission to the I2C slave device with the given address.
// Subsequently, queue bytes for transmission with the write() function
// and transmit them by calling endTransmission().

Wire.write(val); // Writes data from a slave device in response to a request from a
// master, or queues bytes for transmission from a master to slave
// device (in-between calls to beginTransmission() and
// endTransmission()).

Wire.endTransmission(); // Ends a transmission to a slave device that was begun by
// beginTransmission() and transmits the bytes that were queued by
// write().

Wire.requestFrom(addr, num) //Send read request for 'num' bytes to device with I2C address 'addr'

Wire.available() //Is data we've asked to read available on the I2C bus? Returns
//how many bytes are available.

Wire.read() // Reads a byte that was transmitted from a slave device to a master
// after a call to requestFrom() or was transmitted from a master to a
// slave. **This is a blocking transaction**. If a NACK is received, function returns 0.

Note: *Only the pins **A4 and A5** can be used as I2C pins. It is set automatically by the Wire library. **A4 is SDA and A5 is SCL**.*

I2C example

```
#include <Wire.h>
void setup()
{
  Wire.begin();        // join i2c bus (address optional for master)
  Serial.begin(9600);  // start serial for output
}

void loop()
{
  Wire.requestFrom(2, 6);    // request 6 bytes from slave device #2
  while(Wire.available())    // slave may send less than requested
  {
    char c = Wire.read();    // receive a byte as character
    Serial.print(c);         // print the character
  }

  delay(500);
}
```

Analog Read

Uno: operating voltage: 5V, usable pins: A0-A5, bits 10

analogRead(pin) //input is pin number (A0 to A5 on most boards), output is analog value on pin.

Analog Write

Uno: PWM pins 3, 5, 6, 9, 10, 11. PWM frequency 490 Hz (pins 5 and 6: 980 Hz)

analogWrite(pin, value) // pin to write to. value is the duty cycle: between 0 (always off) and 255 (always on)

Digital I/O

pinMode(pin, mode) //mode is INPUT, OUTPUT or INPUT_PULLUP

digitalWrite(pin, value) //Write value HIGH/LOW at GPIO 'pin'

digitalRead(pin) // Reads the value from a specified digital pin, either HIGH or LOW.

UART/Serial

serial.begin(speed) //initializes the UART to "speed" baud.

serial.read() // returns the first byte of incoming serial data (or -1 if not data is available)

serial.write(buf, len) // buf is an array of characters you wish to send. Len is how many bytes to send

Servo

servo.attach(pin) // Attach the Servo variable to a pin. Note that in Arduino 0016 and earlier,
// the Servo library supports servos on only two pins: 9 and 10.

servo.write(angle) // specifies an angle to write from 0 to 180.

Servo example

```
#include <Servo.h>

Servo myservo;

void setup()
{
  myservo.attach(9);
  myservo.write(90);  // set servo to mid-point
}

void loop() {}
```

3-lead Muscle / Electromyography Sensor for Microcontroller Applications

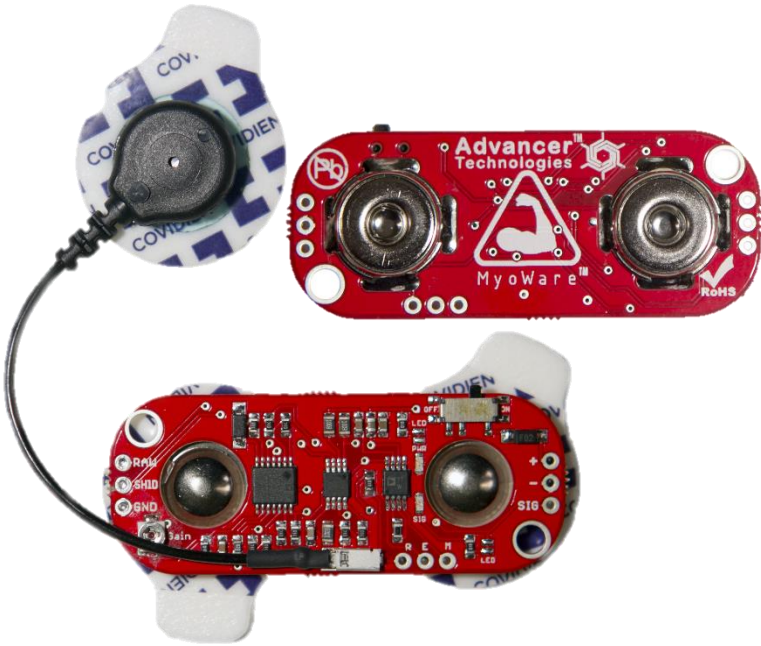
MyoWare™ Muscle Sensor (AT-04-001) DATASHEET

FEATURES

- *NEW* - Wearable Design
- *NEW* - Single Supply
  - +3.1V to +5.9V
  - Polarity reversal protection
- *NEW* - Two Output Modes
  - EMG Envelope
  - Raw EMG
- *NEW* - Expandable via Shields
- *NEW* - LED Indicators
- Specially Designed For Microcontrollers
- Adjustable Gain

APPLICATIONS

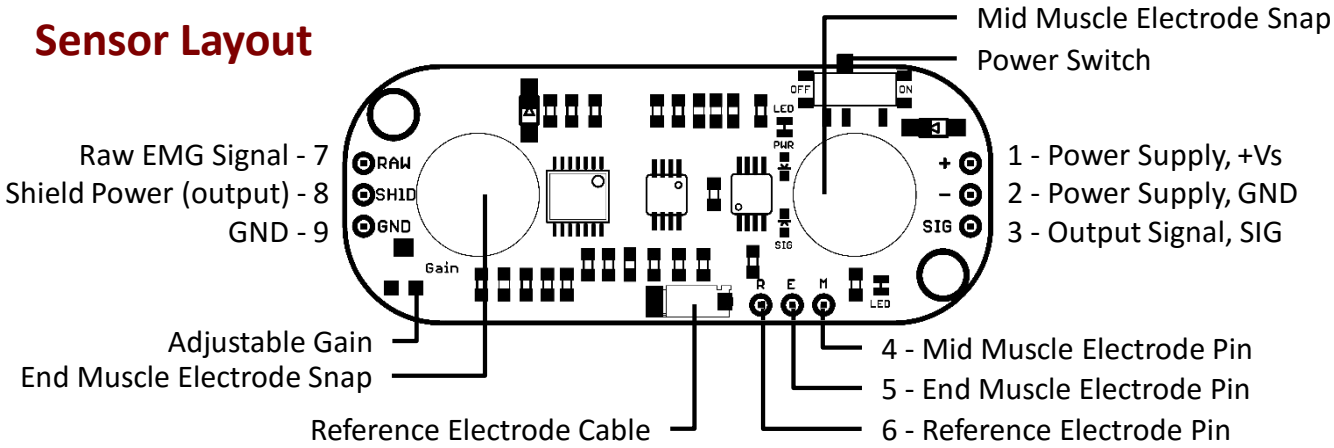
- Video games
- Robotics
- Medical Devices
- Wearable/Mobile Electronics
- Prosthetics/Orthotics



What is electromyography?

Measuring muscle activation via electric potential, referred to as electromyography (EMG), has traditionally been used for medical research and diagnosis of neuromuscular disorders. However, with the advent of ever shrinking yet more powerful microcontrollers and integrated circuits, EMG circuits and sensors have found their way into prosthetics, robotics and other control systems.

Sensor Layout



EMAIL: support@advancer.co

Advancer Technologies™

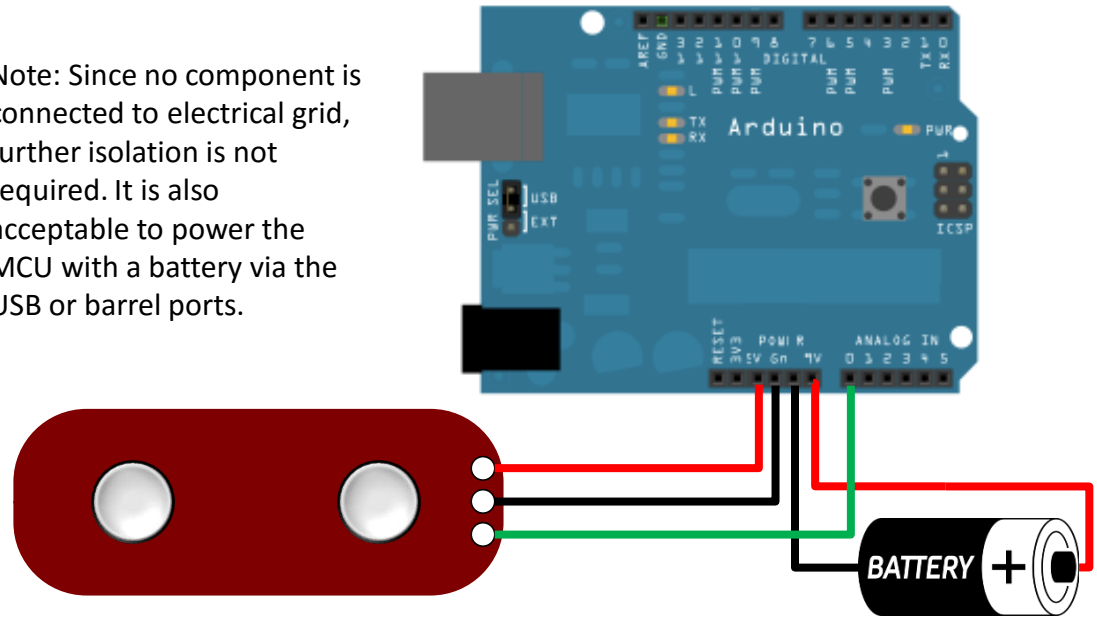
www.AdvancerTechnologies.com

Setup Configurations *(Arduino is shown but MyoWare is compatible with most development boards)*

a) Battery powered with isolation via no direct external connections

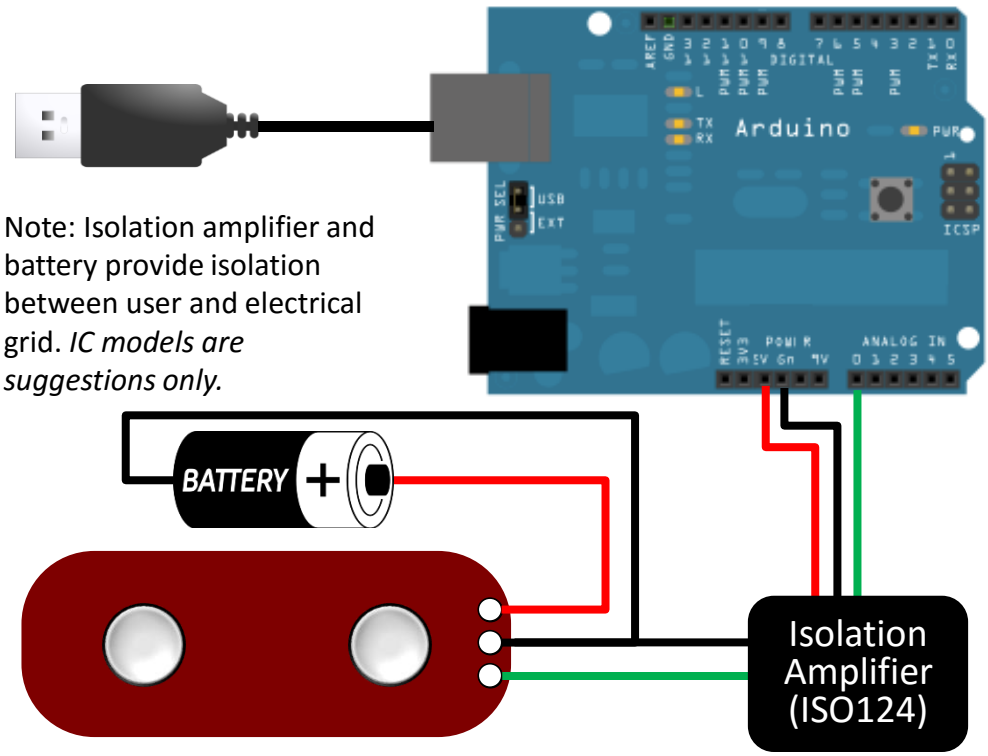
RECOMMENDED

Note: Since no component is connected to electrical grid, further isolation is not required. It is also acceptable to power the MCU with a battery via the USB or barrel ports.



b) Battery powered sensor, Grid powered MCU with output isolation

Note: Isolation amplifier and battery provide isolation between user and electrical grid. *IC models are suggestions only.*



(Note: Arduino and batteries not included. Arduino setup is only an example; sensor will work with numerous other devices.)

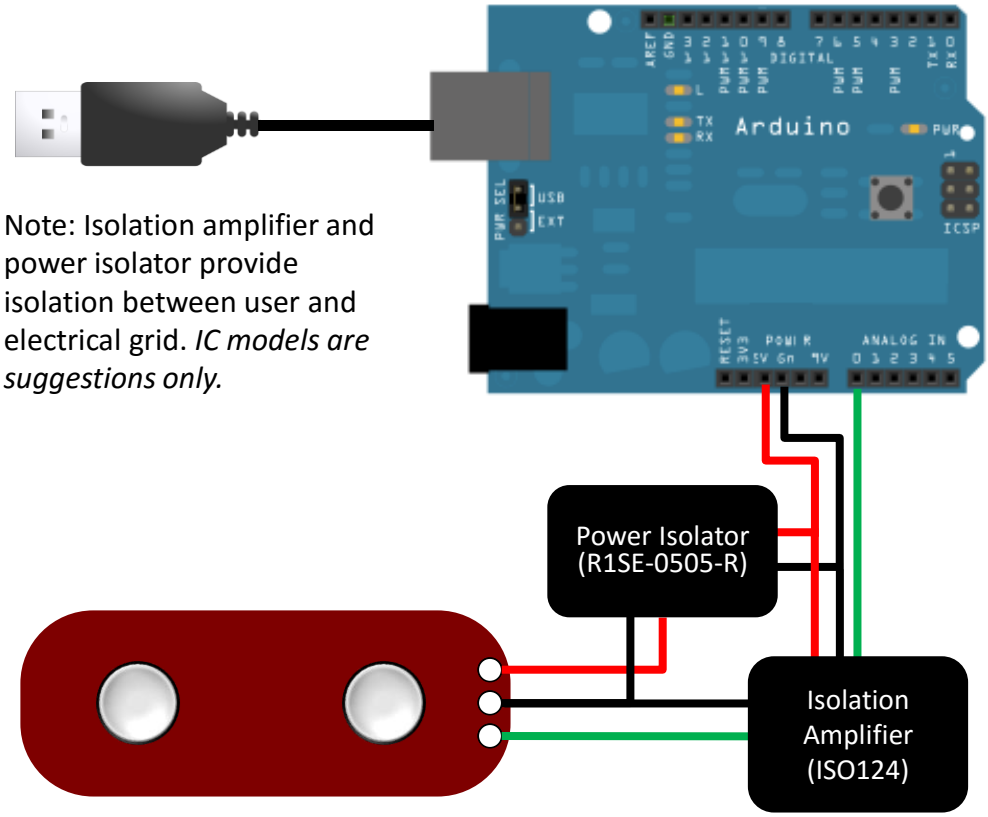
EMAIL: support@advancer.co

Advancer Technologies™

www.AdvancerTechnologies.com

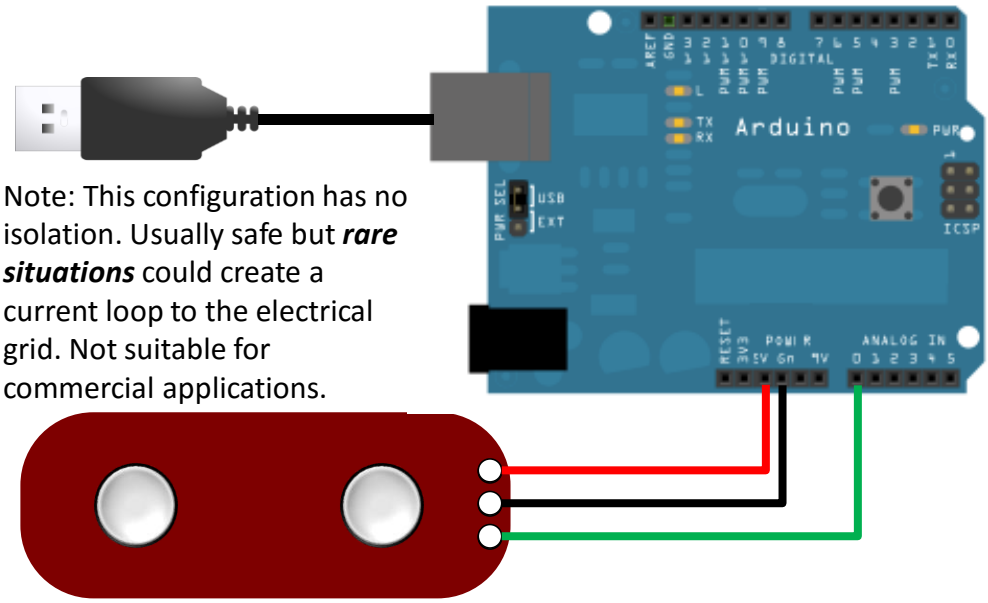
Setup Configurations (cont'd)

c) Grid powered with power and output isolation



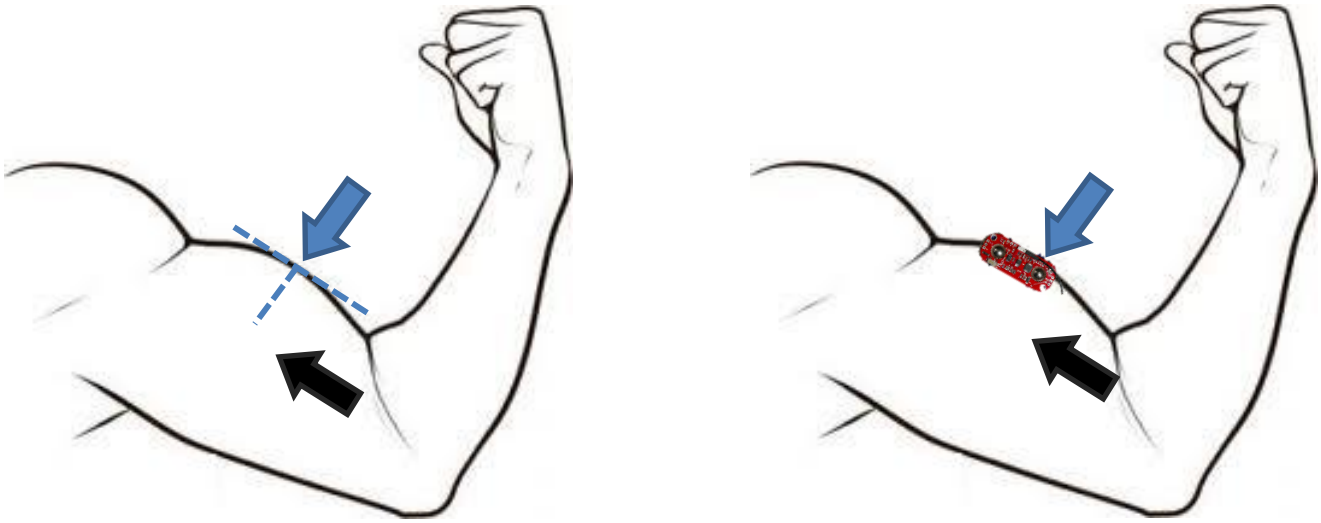
Note: Isolation amplifier and power isolator provide isolation between user and electrical grid. *IC models are suggestions only.*

d) Grid powered. ***Warning: No isolation.***



Note: This configuration has no isolation. Usually safe but ***rare situations*** could create a current loop to the electrical grid. Not suitable for commercial applications.

Setup Instructions



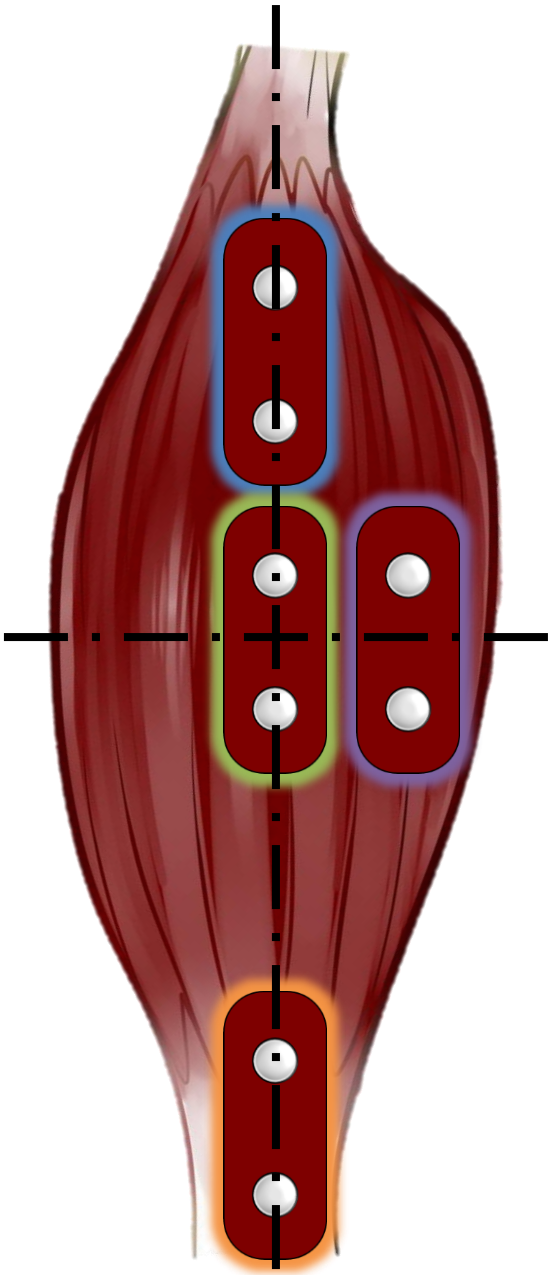
Note: Not To Scale

Example Sensor Location for Bicep

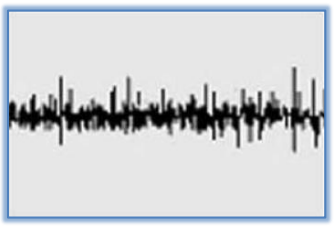
1) Thoroughly clean the intended area with soap to remove dirt and oil
2) Snap electrodes to the sensor's snap connectors

   (Note: While you can snap the sensor to the electrodes after they've been placed on the muscle, we do not recommend doing so due to the possibility of excessive force being applied and bruising the skin.)
3) Place the sensor on the desired muscle
   a. After determining which muscle group you want to target (e.g. bicep, forearm, calf), clean the skin thoroughly
   b. Place the sensor so one of the connected electrodes is in the middle of the muscle body. The other electrode should line up in the direction of the muscle length
   c. Peel off the backs of the electrodes to expose the adhesive and apply them to the skin
   d. Place the reference electrode on a bony or nonadjacent muscular part of your body near the targeted muscle
4) Connect to a development board (e.g. Arduino, RaspberryPi), microcontroller, or ADC
   a. See configurations previously shown

Why is electrode placement important?



Raw EMG output

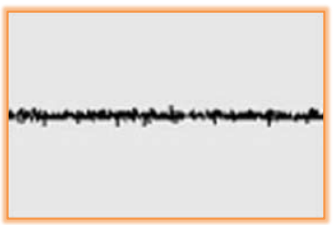


Innervation Zone

Correct Placement

Midline of the muscle belly between an innervation zone and a myotendon junction

Midline Offset



Myotendon Junction

Position and orientation of the muscle sensor electrodes has a vast effect on the strength of the signal. The electrodes should be place in the middle of the muscle body and should be aligned with the orientation of the muscle fibers. Placing the sensor in other locations will reduce the strength and quality of the sensor's signal due to a reduction of the number of motor units measured and interference attributed to crosstalk.

EMAIL: support@advancer.co

Advancer™ Technologies

www.AdvancerTechnologies.com

RAW EMG vs EMG Envelope

Our Muscle Sensors are designed to be used directly with a microcontroller. Therefore, our sensors primary output is not a RAW EMG signal but rather an amplified, rectified, and integrated signal (AKA the EMG's envelope) that will work well with a microcontroller's analog-to-digital converter (ADC). This difference is illustrated below using a representative EMG signal. *Note: Actual sensor output not shown.*

RAW EMG Signal

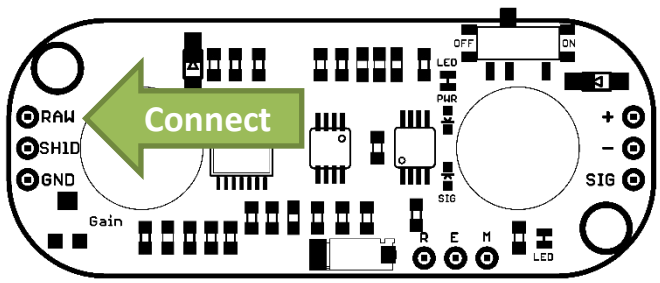
Rectified EMG Signal

Rectified & Integrated EMG Signal

Reconfigure for Raw EMG Output

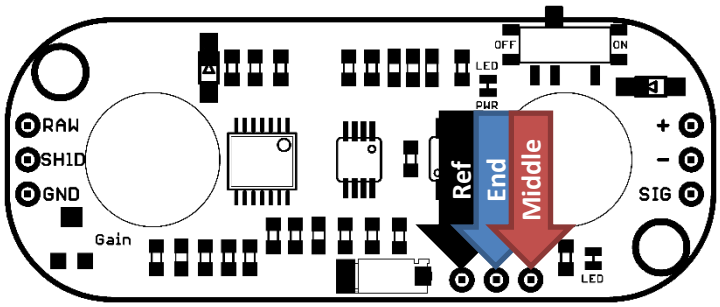
This new version has the ability to output an amplified raw EMG signal.
To output the raw EMG signal, simply connect the raw EMG signal pin to your measuring device instead of the SIG pin.

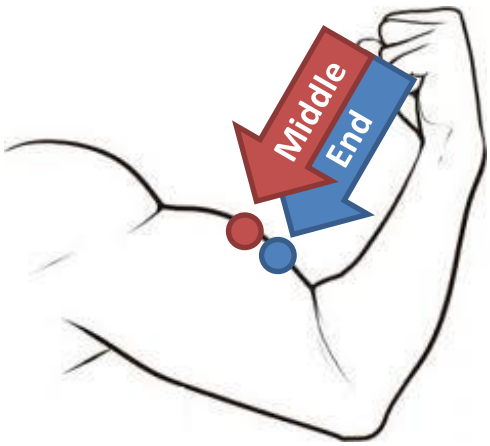
Note: This output is centered about an offset voltage of +Vs/2, see above. It is important to ensure +Vs is the max voltage of the MCU's analog to digital converter. This will assure that you completely see both positive and negative portions of the waveform.



Connecting external electrode cables

This new version has embedded electrode snaps right on the sensor board itself, replacing the need for a cable. However, if the on board snaps do not fit a user's specific application, an external cable can be connected to the board through three through hole pads shown above.





Middle

Connect this pad to the cable leading to an electrode placed in the middle of the muscle body.

End

Connect this to the cable leading to an electrode placed adjacent to the middle electrode towards the end of the muscle body.

Ref

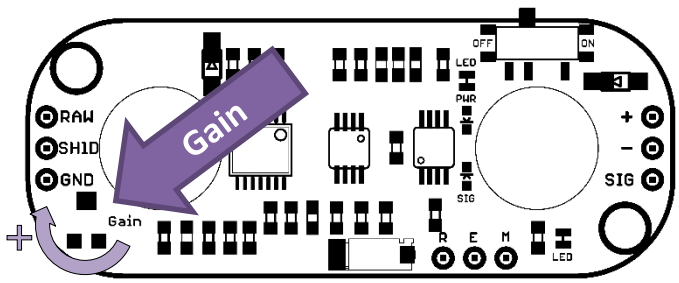
Connect this to the reference electrode. The reference electrode should be placed on an separate section of the body, such as the bony portion of the elbow or a nonadjacent muscle

Adjusting the gain

We recommend for users to get their sensor setup working reliably prior to adjusting the gain. The default gain setting should be appropriate for most applications.

To adjust the gain, locate the gain potentiometer in the lower left corner of the sensor (marked as "GAIN"). Using a Phillips screwdriver, turn the potentiometer counterclockwise to increase the output gain; turn the potentiometer clockwise to reduce the gain.

Note: In order to reduce the required voltage for the sensor, the redesign switch out a JFET amplifier for a CMOS amplifier. However CMOS amplifiers tend to have slower recovery times when saturated. Therefore, we advise users to adjust the gain such that the output signal will not saturate the amplifier.



EMAIL: support@advancer.co

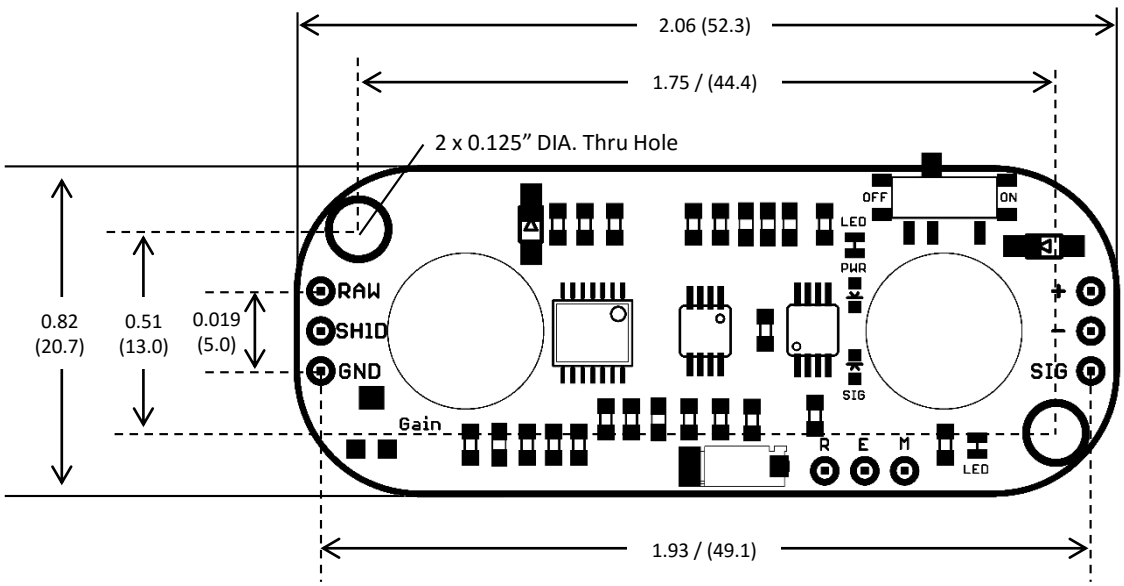
Advancer™ Technologies

www.AdvancerTechnologies.com

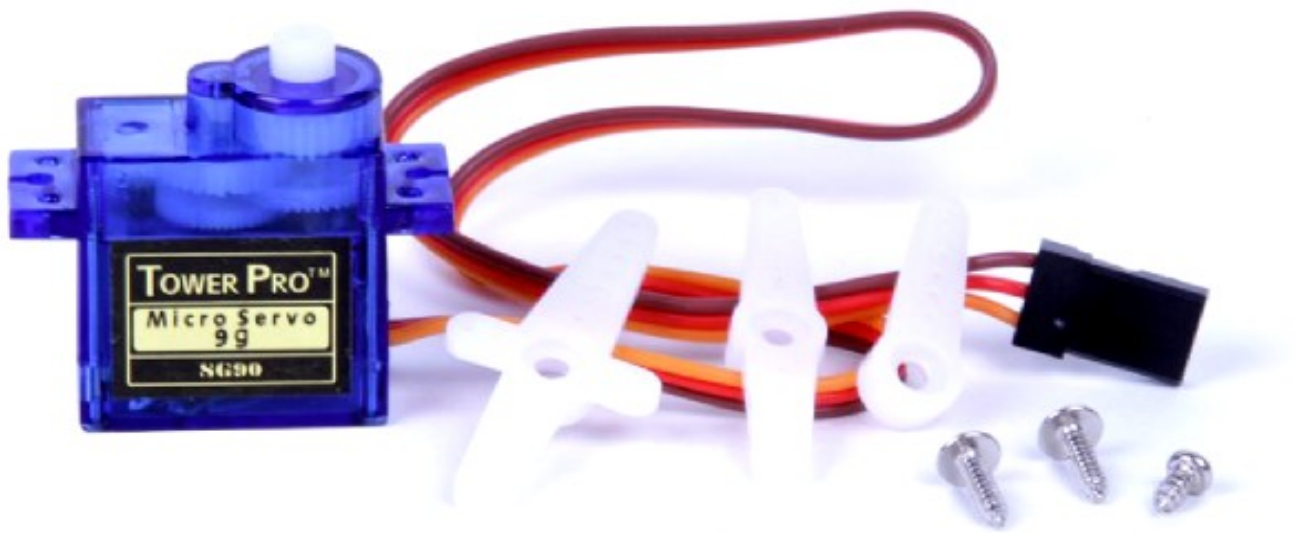
Electrical Specifications

Parameter	Min	TYP	Max
Supply Voltage	+3.1V	+3.3V or +5V	+6.3V
Adjustable Gain Potentiometer	0.01 Ω	50 kΩ	100 kΩ
Output Signal Voltage EMG Envelope Raw EMG (centered about +Vs/2)	 0V 0V	 -- --	 +Vs +Vs
Input Impedance	--	110 GΩ	--
Supply Current	--	9 mA	14 mA
Common Mode Rejection Ratio (CMRR)	--	110	--
Input Bias	--	1 pA	--

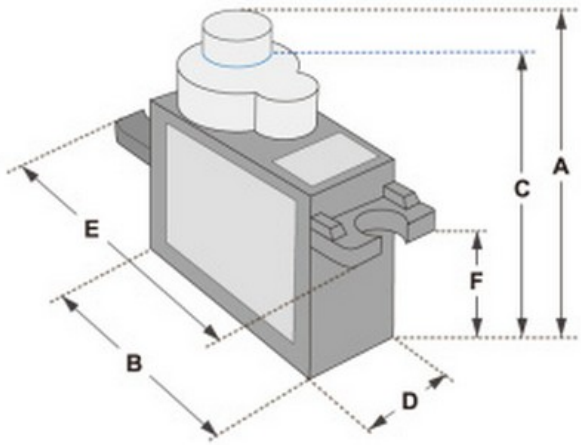
Dimensions



SERVO MOTOR SG90 DATA SHEET

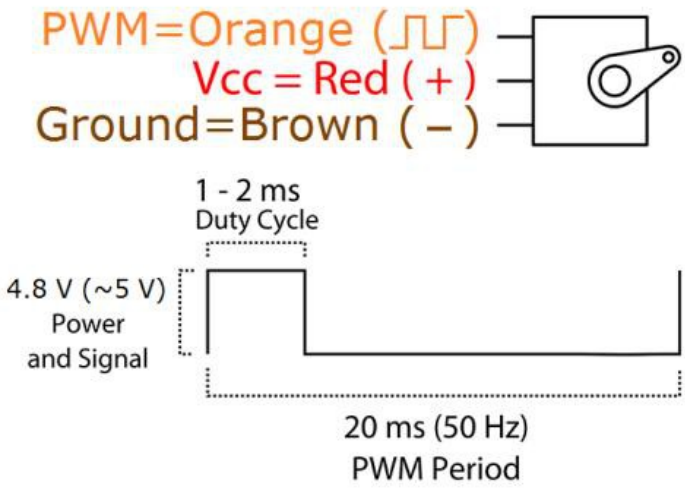


Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.



Dimensions & Specifications
A (mm) : 32
B (mm) : 23
C (mm) : 28.5
D (mm) : 12
E (mm) : 32
F (mm) : 19.5
Speed (sec) : 0.1
Torque (kg-cm) : 2.5
Weight (g) : 14.7
Voltage : 4.8 - 6

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.



LM3940 1-A Low-Dropout Regulator for 5-V to 3.3-V Conversion

1 Features

- Input Voltage Range: 4.5 V to 5.5 V
- Output Voltage Specified over Temperature
- Excellent Load Regulation
- Specified 1-A Output Current
- Requires only One External Component
- Built-in Protection against Excess Temperature
- Short-Circuit Protected

2 Applications

- Laptop and Desktop Computers
- Logic Systems

3 Description

The LM3940 is a 1-A low-dropout regulator designed to provide 3.3 V from a 5-V supply.

The LM3940 is ideally suited for systems which contain both 5-V and 3.3-V logic, with prime power provided from a 5-V bus.

Because the LM3940 is a true low dropout regulator, it can hold its 3.3-V output in regulation with input voltages as low as 4.5 V.

The TO-220 package of the LM3940 means that in most applications the full 1 A of load current can be delivered without using an additional heatsink.

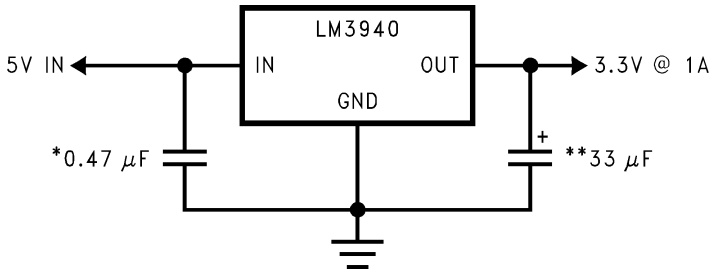
The surface mount DDPAK/TO-263 package uses minimum board space, and gives excellent power dissipation capability when soldered to a copper plane on the PC board.

Device Information[1]

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM3940	SOT-223 (4)	6.50 mm x 3.50 mm
	DDPAK/TO-263 (3)	10.18 mm x 8.41 mm
	TO-220 (3)	14.986 mm x 10.16 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

Simplified Schematic



*Required if regulator is located more than 1 inch from the power supply filter capacitor or if battery power is used.
**See *Application and Implementation*.

Table 36. STATUS_REG description (continued)

ZDA	Z-axis new data available. Default value: 0 (0: new data for the Z-axis is not yet available; 1: new data for the Z-axis is available)
YDA	Y-axis new data available. Default value: 0 (0: new data for the Y-axis is not yet available; 1: new data for the Y-axis is available)
XDA	X-axis new data available. Default value: 0 (0: new data for the X-axis is not yet available; 1: new data for the X-axis is available)

7.10 OUT_X_L (28h), OUT_X_H (29h)

X-axis acceleration data. The value is expressed as two's complement.

7.11 OUT_Y_L (2Ah), OUT_Y_H (2Bh)

Y-axis acceleration data. The value is expressed as two's complement.

7.12 OUT_Z_L (2Ch), OUT_Z_H (2Dh)

Z-axis acceleration data. The value is expressed as two's complement.

7.13 INT1_CFG (30h)

Table 37. INT1_CFG register

AOI	0	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE

Table 38. INT1_CFG description

AOI	AND/OR combination of interrupt events. Default value: 0. (See *Table 39*)
ZHIE	Enable interrupt generation on Z high event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
ZLIE	Enable interrupt generation on Z low event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value lower than preset threshold)
YHIE	Enable interrupt generation on Y high event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
YLIE	Enable interrupt generation on Y low event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value lower than preset threshold)

7.7 HP_FILTER_RESET (25h)

Dummy register. Reading at this address zeroes instantaneously the content of the internal high-pass filter. If the high-pass filter is enabled, all three axes are instantaneously set to 0 *g.* This allows the settling time of the high-pass filter to be overcome.

7.8 REFERENCE (26h)

Table 33. REFERENCE register

Ref7	Ref6	Ref5	Ref4	Ref3	Ref2	Ref1	Ref0

Table 34. REFERENCE description

Ref7 - Ref0	Reference value for high-pass filter. Default value: 00h.

This register sets the acceleration value taken as a reference for the high-pass filter output.

When the filter is turned on (at least one of the FDS, HPen2, or HPen1 bits is equal to ‘1’) and the HPM bits are set to “01”, filter-out is generated, taking this value as a reference.

7.9 STATUS_REG (27h)

Table 35. STATUS_REG register

ZYXOR	ZOR	YOR	XOR	ZYXDA	ZDA	YDA	XDA

Table 36. STATUS_REG description

ZYXOR	X, Y and Z-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: new data has overwritten the previous data before it was read)
ZOR	Z-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: new data for the Z-axis has overwritten the previous data)
YOR	Y-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: new data for the Y-axis has overwritten the previous data)
XOR	X-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: new data for the X-axis has overwritten the previous data)
ZYXDA	X, Y and Z-axis new data available. Default value: 0 (0: a new set of data is not yet available; 1: a new set of data is available)

Table 29. CTRL_REG4 description

BDU	Block data update. Default value: 0 (0: continuous update; 1: output registers not updated between MSB and LSB reading)
BLE	Big/little endian data selection. Default value 0. (0: data LSB @ lower address; 1: data MSB @ lower address)
FS1, FS0	Full scale selection. Default value: 00. (00: ±100 *g*; 01: ±200 *g*; 11: ±400 *g*)
SIM	SPI serial interface mode selection. Default value: 0. (0: 4-wire interface; 1: 3-wire interface)

The **BDU** bit is used to inhibit output register updates between the reading of upper and lower register parts. In default mode (BDU = '0'), the lower and upper register parts are updated continuously. When the BDU is activated (BDU = '1'), the content of the output registers is not updated until both MSB and LSB are read which avoids reading values related to different sample times.

7.6 CTRL_REG5 (24h)

Table 30. CTRL_REG5 register

0	0	0	0	0	0	TurnOn1	TurnOn0

Table 31. CTRL_REG5 description

TurnOn1, TurnOn0	Turn-on mode selection for sleep-to-wake function. Default value: 00.

Turn-on bits are used for turning on the **sleep-to-wake** function.

Table 32. Sleep-to-wake configuration

TurnOn1	TurnOn0	Sleep-to-wake status
0	0	Sleep-to-wake function is disabled
1	1	Turned on: The device is in low-power mode (ODR is defined in CTRL_REG1)

Setting TurnOn[1:0] bits to 11, the "sleep-to-wake" function is enabled. When an interrupt event occurs, the device is turned to normal mode, increasing the ODR to the value defined in CTRL_REG1. Although the device is in normal mode, CTRL_REG1 content is not automatically changed to "normal mode" configuration.

7.4 CTRL_REG3 [interrupt CTRL register] (22h)

Table 25. CTRL_REG3 register

IHL	PP_OD	LIR2	I2_CFG1	I2_CFG0	LIR1	I1_CFG1	I1_CFG0

Table 26. CTRL_REG3 description

IHL	Interrupt active high, low. Default value: 0 (0: active high; 1: active low)
PP_OD	Push-pull/open drain selection on interrupt pad. Default value 0. (0: push-pull; 1: open drain)
LIR2	Latch interrupt request on INT2_SRC register, with INT2_SRC register cleared by reading INT2_SRC itself. Default value: 0. (0: interrupt request not latched; 1: interrupt request latched)
I2_CFG1, I2_CFG0	Data signal on INT 2 pad control bits. Default value: 00. (see *Table 27*)
LIR1	Latch interrupt request on the INT1_SRC register, with the INT1_SRC register cleared by reading the INT1_SRC register. Default value: 0. (0: interrupt request not latched; 1: interrupt request latched)
I1_CFG1, I1_CFG0	Data signal on INT 1 pad control bits. Default value: 00. (see *Table 27*)

Table 27. Data signal on INT 1 and INT 2 pad

I1(2)_CFG1	I1(2)_CFG0	INT 1(2) Pad
0	0	Interrupt 1 (2) source
0	1	Interrupt 1 source OR interrupt 2 source
1	0	Data ready
1	1	Boot running

7.5 CTRL_REG4 (23h)

Table 28. CTRL_REG4 register

BDU	BLE	FS1	FS0	0	0	0	SIM

Table 22. CTRL_REG2 description (continued)

HPen1	High-pass filter enabled for interrupt 1 source. Default value: 0 (0: filter bypassed; 1: filter enabled)
HPCF1, HPCF0	High-pass filter cutoff frequency configuration. Default value: 00 (00: HPc=8; 01: HPc=16; 10: HPc=32; 11: HPc=64)

The **BOOT** bit is used to refresh the content of the internal registers stored in the Flash memory block. At device power-up, the content of the Flash memory block is transferred to the internal registers related to trimming functions in order to permit correct operation of the device itself. If for any reason the content of the trimming registers is changed, it is sufficient to use this bit to restore the correct values. When the BOOT bit is set to '1', the content of the internal Flash is copied inside the corresponding internal registers and it is used to calibrate the device. These values are factory trimmed and they are different for every accelerometer. They permit correct operation of the device and normally they do not have to be changed. At the end of the boot process the BOOT bit is set again to '0'.

Table 23. High-pass filter mode configuration

HPM1	HPM0	High-pass filter mode
0	0	Normal mode (reset reading HP_RESET_FILTER)
0	1	Reference signal for filtering
1	0	Normal mode (reset reading HP_RESET_FILTER)

HPCF[1:0]. These bits are used to configure the high-pass filter cutoff frequency f_t which is given by:

$$f_t = \ln\left(1 - \frac{1}{HPc}\right) \cdot \frac{f_s}{2\pi}$$

The equation can be simplified to the following approximated equation:

$$f_t = \frac{f_s}{6 \cdot HPc}$$

Table 24. High-pass filter cutoff frequency configuration

HPcoeff2,1	f_t [Hz] Data rate = 50 Hz	f_t [Hz] Data rate = 100 Hz	f_t [Hz] Data rate = 400 Hz	f_t [Hz] Data rate = 1000 Hz
00	1	2	8	20
01	0.5	1	4	10
10	0.25	0.5	2	5
11	0.125	0.25	1	2.5

Table 19. Power mode and low-power output data rate configurations

PM2	PM1	PM0	Power mode selection	Output data rate [Hz] ODR_{LP}
0	0	0	Power-down	--
0	0	1	Normal mode	ODR
0	1	0	Low power	0.5
0	1	1	Low power	1
1	0	0	Low power	2
1	0	1	Low power	5
1	1	0	Low power	10

Table 20. Normal mode output data rate configurations and low-pass cutoff frequencies

DR1	DR0	Output data rate [Hz] ODR	Low-pass filter cutoff frequency [Hz]
0	0	50	37
0	1	100	74
1	0	400	292
1	1	1000	780

7.3 CTRL_REG2 (21h)

Table 21. CTRL_REG2 register

BOOT	HPM1	HPM0	FDS	HPen2	HPen1	HPCF1	HPCF0

Table 22. CTRL_REG2 description

BOOT	Reboot memory content. Default value: 0 (0: normal mode; 1: reboot memory content)
HPM1, HPM0	High-pass filter mode selection. Default value: 00 (00: normal mode; Others: refer to *Table 23*)
FDS	Filtered data selection. Default value: 0 (0: internal filter bypassed; 1: data from internal filter sent to output register)
HPen2	High-pass filter enabled for interrupt 2 source. Default value: 0 (0: filter bypassed; 1: filter enabled)

7 Register description

The device contains a set of registers which are used to control its behavior and to retrieve acceleration data. The register address, consisting of 7 bits, is used to identify them and to write the data through the serial interface.

7.1 WHO_AM_I (0Fh)

Table 16. WHO_AM_I register

0	0	1	1	0	0	1	0

Device identification register.

This register contains the device identifier that for the H3LIS331DL is set to 32h.

7.2 CTRL_REG1 (20h)

Table 17. CTRL_REG1 register

PM2	PM1	PM0	DR1	DR0	Zen	Yen	Xen

Table 18. CTRL_REG1 description

PM2 - PM0	Power mode selection. Default value: 000 (000: power-down; Others: refer to *Table 19*)
DR1, DR0	Data rate selection. Default value: 00 (00:50 Hz; Others: refer to *Table 20*)
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)

The **PM** bits allow the user to select between power-down and two operating active modes. The device is in power-down mode when the PD bits are set to “000” (default value after boot). *Table 19* shows all the possible power mode configurations and respective output data rates. Output data in the low-power mode are computed with the low-pass filter cutoff frequency defined by the DR1, DR0 bits.

The **DR** bits, in normal mode operation, select the data rate at which acceleration samples are produced. In low-power modes they define the output data resolution. *Table 20* shows all the possible configurations for the DR1 and DR0 bits.

6 Register mapping

Table 15 provides a listing of the 8-bit registers embedded in the device and the related addresses.

Table 15. Register address map

Name	Type	Register address		Default	Comment
		Hex	Binary		
Reserved (do not modify)		00 - 0E			Reserved
WHO_AM_I	r	0F	000 1111	00110010	Dummy register
Reserved (do not modify)		10 - 1F			Reserved
CTRL_REG1	rw	20	010 0000	00000111	
CTRL_REG2	rw	21	010 0001	00000000	
CTRL_REG3	rw	22	010 0010	00000000	
CTRL_REG4	rw	23	010 0011	00000000	
CTRL_REG5	rw	24	010 0100	00000000	
HP_FILTER_RESET	r	25	010 0101		Dummy register
REFERENCE	rw	26	010 0110	00000000	
STATUS_REG	r	27	010 0111	00000000	
OUT_X_L	r	28	010 1000	Output	
OUT_X_H	r	29	010 1001	Output	
OUT_Y_L	r	2A	010 1010	Output	
OUT_Y_H	r	2B	010 1011	Output	
OUT_Z_L	r	2C	010 1100	Output	
OUT_Z_H	r	2D	010 1101	Output	
Reserved (do not modify)		2E - 2F			Reserved
INT1_CFG	rw	30	011 0000	00000000	
INT1_SRC	r	31	011 0001	00000000	
INT1_THS	rw	32	011 0010	00000000	
INT1_DURATION	rw	33	011 0011	00000000	
INT2_CFG	rw	34	011 0100	00000000	
INT2_SRC	r	35	011 0101	00000000	
INT2_THS	rw	36	011 0110	00000000	
INT2_DURATION	rw	37	011 0111	00000000	
Reserved (do not modify)		38 - 3F			Reserved

Registers marked as *Reserved* must not be changed as they contain the factory calibration values. Their content is automatically restored when the device is powered up. Writing to those registers may change calibration data and thus lead to improper functioning of the device.

Table 12. Transfer when master is writing multiple bytes to slave

Master	ST	SAD + W		SUB		DATA		DATA		SP
Slave			SAK		SAK		SAK		SAK	

Table 13. Transfer when master is receiving (reading) one byte of data from slave

Master	ST	SAD + W		SUB		SR	SAD + R			NMAK	SP
Slave			SAK		SAK			SAK	DATA		

Table 14. Transfer when master is receiving (reading) multiple bytes of data from slave

Master	ST	SAD+W		SUB		SR	SAD+R			MAK		MAK		NMAK	SP
Slave			SAK		SAK			SAK	DATA		DAT A		DAT A		

Data are transmitted in byte format (DATA). Each data transfer contains 8 bits. The number of bytes transferred per transfer is unlimited. Data is transferred with the most significant bit (MSB) first. If a receiver can't receive another complete byte of data until it has performed some other function, it can hold the clock line SCL low to force the transmitter into a wait state. Data transfer only continues when the receiver is ready for another byte and releases the data line. If a slave receiver doesn't acknowledge the slave address (i.e. it is not able to receive because it is performing some real-time function) the data line must be left high by the slave. The master can then abort the transfer. A low-to-high transition on the SDA line while the SCL line is high is defined as a STOP condition. Each data transfer must be terminated by the generation of a STOP (SP) condition.

In order to read multiple bytes, it is necessary to assert the most significant bit of the sub-address field. In other words, SUB(7) must be equal to 1 while SUB(6-0) represents the address of first register to be read.

In the presented communication format MAK is master acknowledge and NMAK is no master acknowledge.

5.2 SPI bus interface

The H3LIS331DL SPI is a bus slave. The SPI allows the writing and reading of the device registers.

The serial interface interacts with the outside world with 4 wires: **CS**, **SPC**, **SDI** and **SDO**.

5.1.1 I²C operation

The transaction on the bus is started through a START (ST) signal. A START condition is defined as a high-to-low transition on the data line while the SCL line is held high. After this has been transmitted by the master, the bus is considered busy. The next byte of data transmitted after the START condition contains the address of the slave in the first 7 bits and the eighth bit tells whether the master is receiving data from the slave or transmitting data to the slave. When an address is sent, each device in the system compares the first seven bits after a START condition with its address. If they match, the device considers itself addressed by the master.

The slave address (SAD) associated to the H3LIS331DL is 001100xb. The **SDO**/**SA0** pad can be used to modify the less significant bit of the device address. If the SA0 pad is connected to the voltage supply, LSB is ‘1’ (address 0011001b) or else, if the SA0 pad is connected to ground, the LSB value is ‘0’ (address 0011000b). This solution allows the connection and addressing of two different accelerometers to the same I²C lines.

Data transfer with acknowledge is mandatory. The transmitter must release the SDA line during the acknowledge pulse. The receiver must then pull the data line low so that it remains stable low during the high period of the acknowledge clock pulse. A receiver which has been addressed is obliged to generate an acknowledge after each byte of data received.

The I²C embedded inside the H3LIS331DL behaves like a slave device and the following protocol must be adhered to. After the START condition (ST) a slave address is sent, once a slave acknowledge (SAK) has been returned, an 8-bit sub-address (SUB) is transmitted: the 7 LSB represent the actual register address while the MSB enables address auto increment. If the MSB of the SUB field is ‘1’, the SUB (register address) is automatically increased to allow multiple data read/write.

The slave address is completed with a Read/Write bit. If the bit is ‘1’ (read), a repeated START (SR) condition must be issued after the two sub-address bytes; if the bit is ‘0’ (write), the master transmits to the slave with the direction unchanged. *Table 10* explains how the SAD+Read/Write bit pattern is composed, listing all the possible configurations.

Table 10. SAD+Read/Write patterns

Command	SAD[6:1]	SAD[0] = SA0	R/W	SAD+R/W
Read	001100	0	1	00110001 (31h)
Write	001100	0	0	00110000 (30h)
Read	001100	1	1	00110011 (33h)
Write	001100	1	0	00110010 (32h)

Table 11. Transfer when master is writing one byte to slave

Master	ST	SAD + W		SUB		DATA		SP
Slave			SAK		SAK		SAK	

5 Digital interfaces

The registers embedded inside the H3LIS331DL 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, the 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)
SA0 SDO	I^2C less significant bit of the device address (SA0) SPI serial data output (SDO)

5.1 I^2C serial interface

The H3LIS331DL I^2C is a bus slave. The I^2C is employed to write data into registers whose content can also be read back.

The relevant I^2C terminology is given in the table below.

Table 9. Serial interface pin description

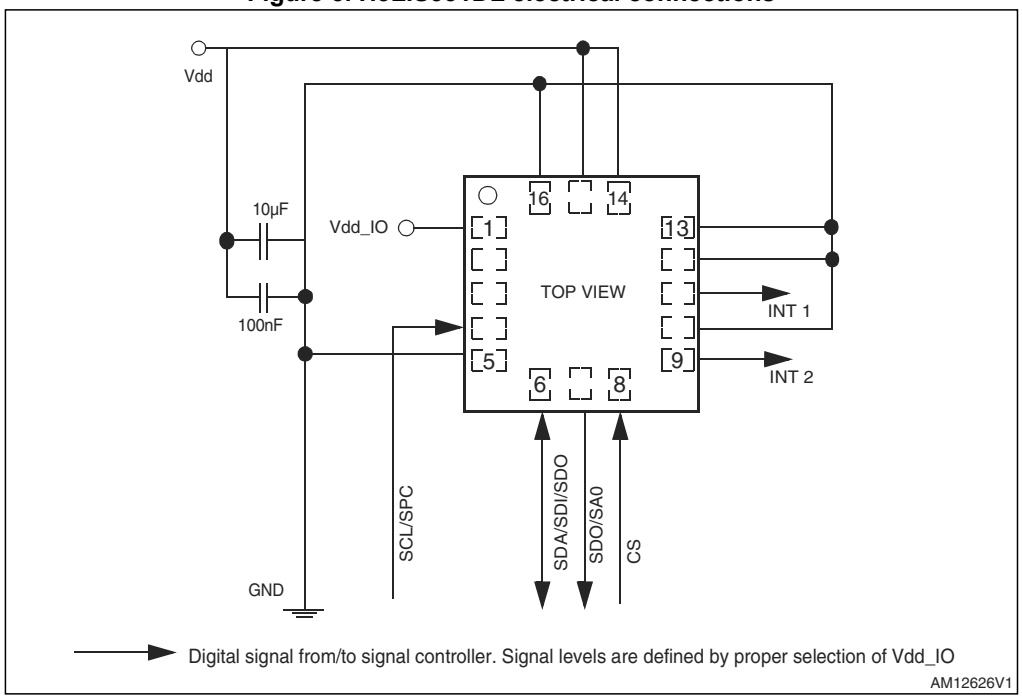
Term	Description
Transmitter	The device which sends data to the bus
Receiver	The device which receives data from the bus
Master	The device which initiates a transfer, generates clock signals and terminates a transfer
Slave	The device addressed by the master

There are two signals associated with the I^2C bus: the serial clock line (SCL) and the serial data line (SDA). The latter is a bi-directional line used for sending and receiving the data to/from the interface. Both the lines are connected to Vdd_IO through a pull-up resistor embedded inside the H3LIS331DL. When the bus is free both lines are high.

The I^2C interface is compliant with fast mode (400 kHz) I^2C standards as well as with normal mode.

4 Application hints

Figure 5. H3LIS331DL electrical connections



The device core is supplied through the Vdd line while the I/O pads are supplied through the Vdd_IO line. Power supply decoupling capacitors (100 nF ceramic, 10 µF aluminum) should be placed as near as possible to pin 14 of the device (common design practice).

All the voltage and ground supplies must be present at the same time to have proper behavior of the IC (refer to *Figure 5*). It is possible to remove Vdd maintaining Vdd_IO without blocking the communication bus, in this condition the measurement chain is powered off.

The functionality of the device and the measured acceleration data are selectable and accessible through the I^2C or SPI interfaces. When using the I^2C, CS must be tied high.

The functions, the threshold and the timing of the two interrupt pins (INT 1 and INT 2) can be completely programmed by the user through the I^2C/SPI interface.

4.1 Soldering information

The LGA package is compliant with the ECOPACK®, RoHS and “Green” standards.
It is qualified for soldering heat resistance according to JEDEC J-STD-020C.

Leave “pin 1 indicator” unconnected during soldering.

Land pattern and soldering recommendations are available at www.st.com.

3 Functionality

The H3LIS331DL is a “nano”, low-power, digital output 3-axis linear accelerometer housed in an LGA package. The complete device includes a sensing element and an IC interface able to take the information from the sensing element and to provide a signal to the external world through an I^2C/SPI serial interface.

3.1 Sensing element

A proprietary process is used to create a surface micromachined accelerometer. The technology allows processing suspended silicon structures, which are attached to the substrate in a few points called anchors and are free to move in the direction of the sensed acceleration. To be compatible with traditional packaging techniques, a cap is placed on top of the sensing element to avoid blocking the moving parts during the molding phase of the plastic encapsulation.

When an acceleration is applied to the sensor, the proof mass displaces from its nominal position, causing an imbalance in the capacitive half bridge. This imbalance is measured using charge integration in response to a voltage pulse applied to the capacitor.

At steady-state the nominal value of the capacitors are a few pF and when an acceleration is applied, the maximum variation of the capacitive load is in the fF range.

3.2 IC interface

The complete measurement chain is composed of a low-noise capacitive amplifier which converts the capacitive unbalancing of the MEMS sensor into an analog voltage that will be available to the user through an analog-to-digital converter.

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

The H3LIS331DL features a data-ready signal (RDY) which indicates when a new set of measured acceleration data is available, therefore simplifying data synchronization in the digital system that uses the device.

3.3 Factory calibration

The IC interface is factory calibrated for sensitivity (So) and zero-*g* level (TyOff).

The trim values are stored inside the device in non-volatile memory. Any time the device is turned on, the trim parameters are downloaded into the registers to be used during active operation. This allows the device to be used without further calibration.

2.5 Terminology

2.5.1 Sensitivity

Sensitivity describes the gain of the sensor and can be determined by applying 1 *g* acceleration to it. As the sensor can measure DC accelerations this can be done easily by pointing the axis of interest towards the center of the Earth, noting the output value, rotating the sensor by 180 degrees (pointing to the sky) and noting the output value again. By doing so, ±1 *g* acceleration is applied to the sensor. Subtracting the larger output value from the smaller one, and dividing the result by 2, leads to the actual sensitivity of the sensor. This value changes very little over temperature and time. The sensitivity tolerance describes the range of sensitivities of a large population of sensors.

2.5.2 Zero-*g* level

The zero-*g* level offset (TyOff) describes the deviation of an actual output signal from the ideal output signal if no acceleration is present. A sensor in a steady-state on a horizontal surface measures 0 *g* for the X-axis and 0 *g* for the Y-axis whereas the Z-axis measures 1 *g*. The output is ideally in the middle of the dynamic range of the sensor (content of OUT registers 00h, data expressed as two's complement number). A deviation from the ideal value in this case is called zero-*g* offset. Offset is, to some extent, a result of stress to the MEMS sensor and therefore can slightly change after mounting the sensor onto a printed circuit board or exposing it to extensive mechanical stress. Offset changes little over temperature, refer to "Zero-*g* level change vs. temperature" (see TCOff in *Table 3*). The zero-*g* level tolerance (TyOff) describes the standard deviation of the range of zero-*g* levels of a population of sensors.

2.5.3 Sleep-to-wakeup

The "sleep-to-wakeup" function, in conjunction with low-power mode, allows to further reduce the system power consumption and develop new smart applications.
The H3LIS331DL may be set in a low-power operating mode, characterized by lower date rate refreshments. In this way the device, even if sleeping, continues to sense acceleration and generate interrupt requests.
When the "sleep-to-wakeup" function is activated, the H3LIS331DL is able to automatically wake up as soon as the interrupt event has been detected, increasing the output data rate and bandwidth.
With this feature the system may be efficiently switched from low-power mode to full performance, depending on user-selectable positioning and acceleration events, therefore ensuring power saving and flexibility.

2.4 Absolute maximum ratings

Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 7. Absolute maximum ratings

Symbol	Ratings	Maximum value	Unit
Vdd	Supply voltage	-0.3 to 4.8	V
Vdd_IO	I/O pins supply voltage	-0.3 to 4.8	V
Vin	Input voltage on any control pin (CS, SCL/SPC, SDA/SDI/SDO, SDO/SA0)	-0.3 to Vdd_IO +0.3	V
A_{POW}	Acceleration (any axis, powered, Vdd = 2.5 V)	3000 *g* for 0.5 ms	
		10000 *g* for 0.1 ms	
A_{UNP}	Acceleration (any axis, unpowered)	3000 *g* for 0.5 ms	
		10000 *g* for 0.1 ms	
T_{OP}	Operating temperature range	-40 to +85	°C
T_{STG}	Storage temperature range	-40 to +125	°C
ESD	Electrostatic discharge protection	4 (HBM)	kV
		1.5 (CDM)	kV
		200 (MM)	V

Note: Supply voltage on any pin should never exceed 4.8 V.

This device is sensitive to mechanical shock, improper handling can cause permanent damage to the part.

This is an electrostatic-sensitive device (ESD), improper handling can cause permanent damage to the part.

2.3.2 I²C - inter-IC control interface

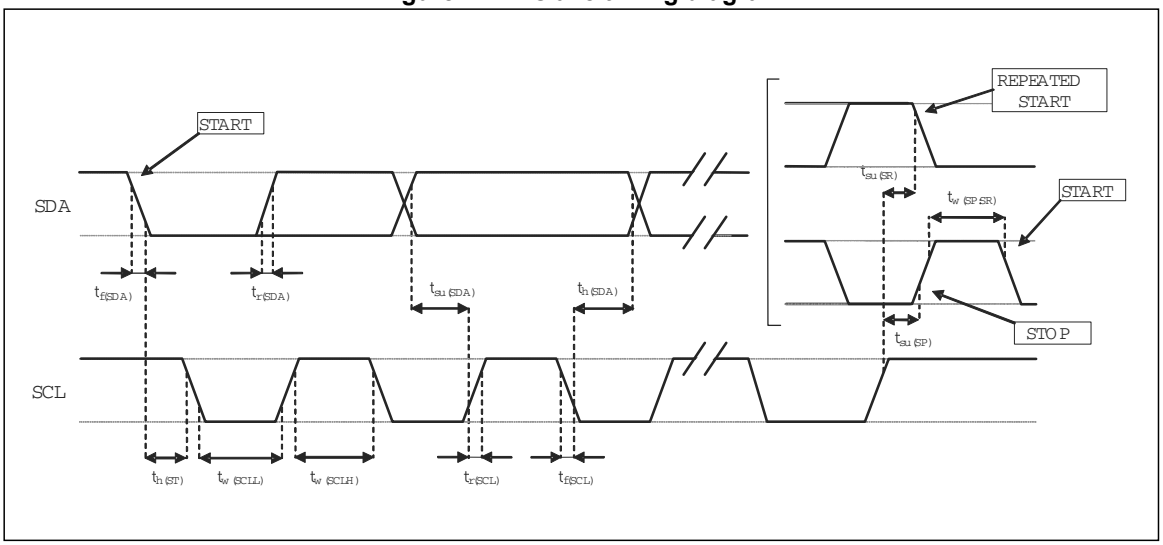
Subject to general operating conditions for Vdd and Top.

Table 6. I²C slave timing values

Symbol	Parameter	I²C standard mode (1)		I²C fast mode (1)		Unit
		Min.	Max.	Min.	Max.	
$f_{(SCL)}$	SCL clock frequency	0	100	0	400	KHz
$t_{w(SCLL)}$	SCL clock low time	4.7		1.3		µs
$t_{w(SCLH)}$	SCL clock high time	4.0		0.6		
$t_{su(SDA)}$	SDA setup time	250		100		ns
$t_{h(SDA)}$	SDA data hold time	0.01	3.45	0.01	0.9	µs
$t_{r(SDA)}$ $t_{r(SCL)}$	SDA and SCL rise time		1000	20 + 0.1C_b (2)	300	ns
$t_{f(SDA)}$ $t_{f(SCL)}$	SDA and SCL fall time		300	20 + 0.1C_b (2)	300	
$t_{h(ST)}$	START condition hold time	4		0.6		µs
$t_{su(SR)}$	Repeated START condition setup time	4.7		0.6		
$t_{su(SP)}$	STOP condition setup time	4		0.6		
$t_{w(SP:SR)}$	Bus free time between STOP and START condition	4.7		1.3		

1. Data based on standard I²C protocol requirement, not tested in production.
2. C_b = total capacitance of one bus line, in pF.

Figure 4. I²C slave timing diagram



Note: Measurement points are done at 0.2·Vdd_IO and 0.8·Vdd_IO, for both ports.

2.2 Electrical characteristics

@ Vdd = 2.5 V, T = 25 °C unless otherwise noted [b].

Table 4. Electrical characteristics

Symbol	Parameter	Test conditions	Min.	Typ.(1)	Max.	Unit
Vdd	Supply voltage		2.16	2.5	3.6	V
Vdd_IO	I/O pins supply voltage(2)		1.71		Vdd+0.1	V
Idd	Current consumption in normal mode			300		µA
IddLP	Current consumption in low-power mode			10		µA
IddPdn	Current consumption in power-down mode			1		µA
VIH	Digital high-level input voltage		0.8*Vdd_IO			V
VIL	Digital low-level input voltage				0.2*Vdd_IO	V
VOH	High-level output voltage		0.9*Vdd_IO			V
VOL	Low-level output voltage				0.1*Vdd_IO	V
ODR	Output data rate in normal mode	DR bit set to 00		50		Hz
		DR bit set to 01		100		
		DR bit set to 10		400		
		DR bit set to 11		1000		
ODR_{LP}	Output data rate in low-power mode	PM bit set to 010		0.5		Hz
		PM bit set to 011		1		
		PM bit set to 100		2		
		PM bit set to 101		5		
		PM bit set to 110		10		
BW	System bandwidth(3)			ODR/2		Hz
Ton	Turn-on time(4)	ODR = 100 Hz		1/ODR+1ms		s
Top	Operating temperature range		-40		+85	°C

1. Typical specifications are not guaranteed.
2. It is possible to remove Vdd maintaining Vdd_IO without blocking the communication busses; in this condition the measurement chain is powered off.
3. Refer to *Table 20* for filter cutoff frequency.
4. Time to obtain valid data after exiting power-down mode.

b. The product is factory calibrated at 2.5 V. The operational power supply range is from 2.16 V to 3.6 V.

2 Mechanical and electrical specifications

2.1 Mechanical characteristics

@ Vdd = 2.5 V, T = 25 °C unless otherwise noted [a].

Table 3. Mechanical characteristics

Symbol	Parameter	Test conditions	Min.	Typ.(1)	Max.	Unit
FS	Measurement range(2)	FS bit set to 00		±100		*g*
		FS bit set to 01		±200		
		FS bit set to 11		±400		
So	Sensitivity(3)	FS bit set to 00 12-bit representation		49		mg/digit
		FS bit set to 01 12-bit representation		98		
		FS bit set to 11 12-bit representation		195		
TCSo	Sensitivity change vs. temperature	FS bit set to 00		±0.01		%/°C
TyOff	Typical zero-*g* level offset accuracy(4)	FS bit set to 00		±1		*g*
TCOff	Zero-*g* level change vs. temperature	Max. delta from 25 °C		±5		mg/°C
An	Acceleration noise density	FS bit set to 00		15		$mg/\sqrt{Hz}$
NL	Non-linearity	FS bit set to 00 Range -70g .. +70g		2		%FS
Top	Operating temperature range		-40		+85	°C
Wh	Product weight			20		mgram

1. Typical specifications are not guaranteed.
2. Verified by wafer level test and measurement of initial offset and sensitivity.
3. Factory calibrated at ±1 *g*
4. Offset can be eliminated by enabling the built-in high-pass filter.

a. The product is factory calibrated at 2.5 V. The operational power supply range is from 2.16 V to 3.6 V. The product calibration is done at ±1 *g*.

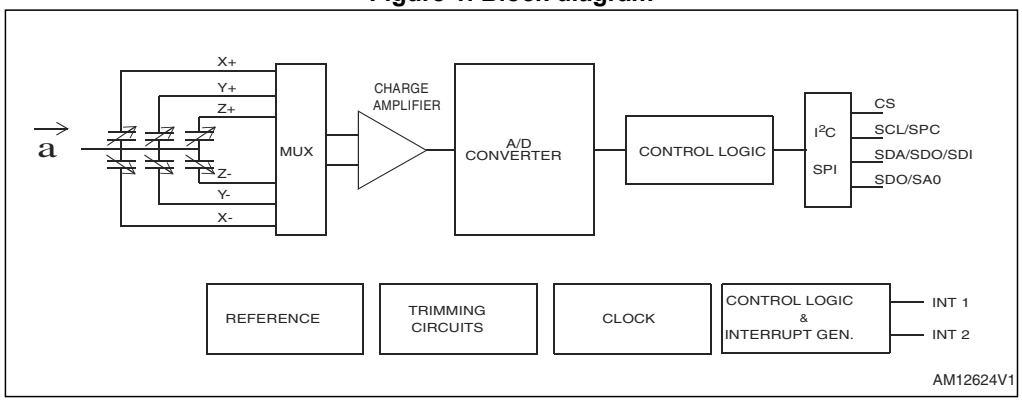
Table 2. Pin description

Pin#	Name	Function
1	Vdd_IO	Power supply for I/O pins
2	NC	Not connected
3	NC	Not connected
4	SCL SPC	I^2C serial clock (SCL) SPI serial port clock (SPC)
5	GND	0 V supply
6	SDA SDI SDO	I^2C serial data (SDA) SPI serial data input (SDI) 3-wire interface serial data output (SDO)
7	SDO SA0	SPI serial data output (SDO) I^2C less significant bit of the device address (SA0)
8	CS	SPI enable I^2C/SPI mode selection (1: I^2C mode; 0: SPI enabled)
9	INT 2	Inertial interrupt 2
10	Reserved	Connect to GND
11	INT 1	Inertial interrupt 1
12	GND	0 V supply
13	GND	0 V supply
14	Vdd	Power supply
15	Reserved	Connect to Vdd
16	GND	0 V supply

1 Block diagram and pin description

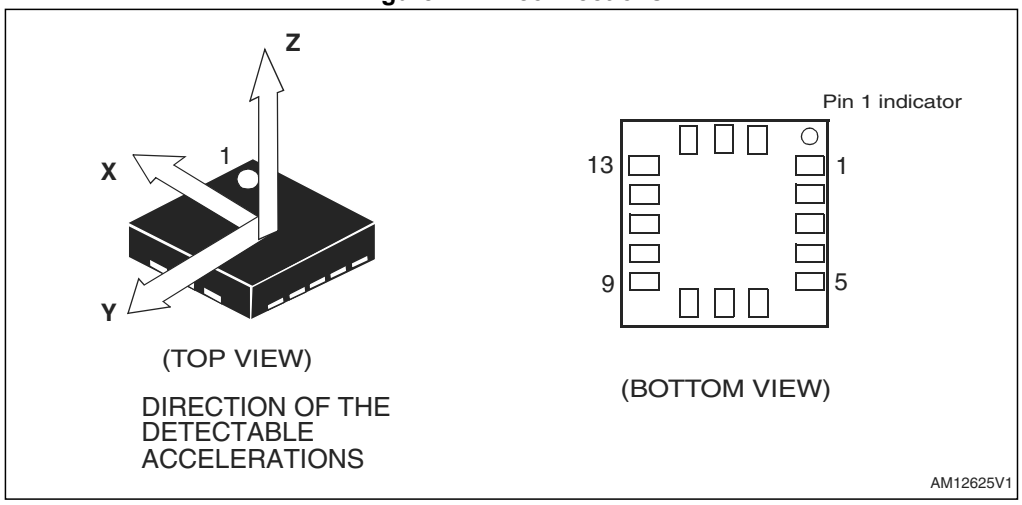
1.1 Block diagram

Figure 1. Block diagram



1.2 Pin description

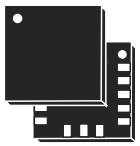
Figure 2. Pin connections



H3LIS331DL

MEMS motion sensor: low-power high-g 3-axis digital accelerometer

Datasheet - production data



TFLGA 3x3x1.0 mm^3 16L

Features

- Wide supply voltage, 2.16 V to 3.6 V
- Low-voltage compatible IOs, 1.8 V
- Ultra-low power consumption down to 10 µA in low-power mode
- ±100*g*/±200*g*/±400*g* dynamically selectable full scales
- I^{2}C/SPI digital output interface
- 16-bit data output
- Sleep-to-wakeup function
- 10000 *g* high-shock survivability
- ECOPACK®, RoHS and “Green” compliant

Applications

- Shock detection
- Impact recognition and logging
- Concussion detection

Description

The H3LIS331DL is a low-power high-performance 3-axis linear accelerometer belonging to the “nano” family, with digital I^{2}C/SPI serial interface standard output.

The device features ultra-low power operational modes that allow advanced power saving and smart sleep-to-wakeup functions.

The H3LIS331DL has dynamically user-selectable full scales of ±100*g*/±200*g*/±400*g* and it is capable of measuring accelerations with output data rates from 0.5 Hz to 1 kHz.

The H3LIS331DL is available in a small thin plastic land grid array package (LGA) and it is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

Table 1. Device summary

Order codes	Temperature range [°C]	Package	Packaging
H3LIS331DL	-40 to +85	TFLGA 3x3x1.0 mm^3 16L	Tray
H3LIS331DLTR	-40 to +85	TFLGA 3x3x1.0 mm^3 16L	Tape and reel