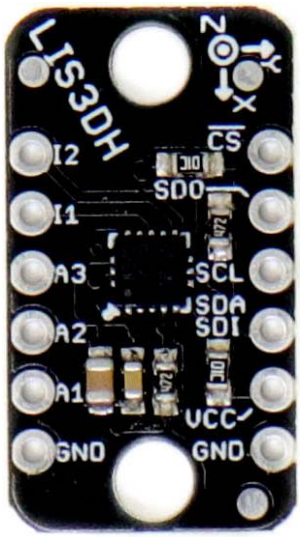
SmartElex Triple Axis Accelerometer Breakout - LIS3DH



The LIS3DH is a triple axis accelerometer you can use to add translation detection to your project. The "3D" in LIS3DH refers to the fact that it is a 3DoF, or 3 Degrees of Freedom. Additionally, it has a few analog inputs to play with, and it has some built in movement detection features to detect things like free-fall, and to indicate if the FIFO buffers are full.

This table gives more information as to each pins functionality. The serial port can be connected as either SPI or I2C, and it uses the same physical pins for both. To get going, just wire up your choice of interface, supply 3.3v, and ground. Note that you will not need to use all the pins no matter which communication method you choose.

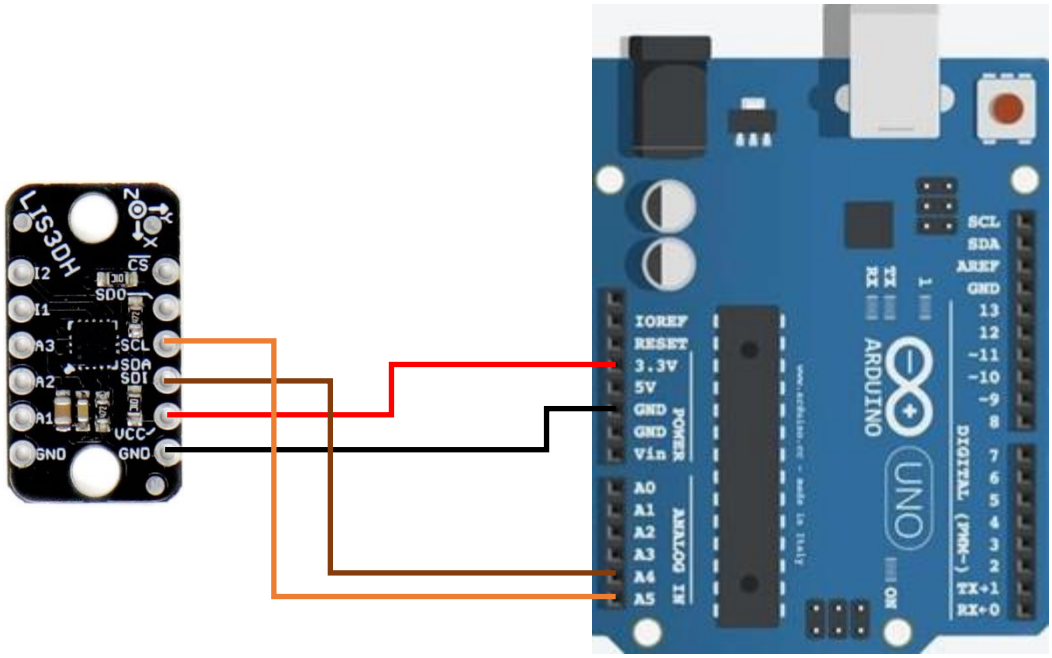
				Connection	
Group	**Name**	**Direction**	**Description**	**I2C**	**SPI**
Serial	!CS	I	Chip select (for SPI)	NC	!CS
	SDO	O	Data output (MISO for SPI)	NC	MISO
	SCL	I	Data clock	SCL	SCK
	SDA/SDI	I/O	Data in (SDA for I2C, MOSI for SPI)	SDA	MOSI
Interrupts	I1	O	Primary int has FIFO + motion	Optional MCU	
	I2	O	Secondary int has motion	Optional MCU	
ADC	A1	I	Analog in	Optional	
	A2	I	Analog in	Optional	
	A3	I	Analog in (unused for temp readings)	Optional	
Power	VCC	I	3.3V input	Supply	
	GND	I	Ground connection (either PTH)	Supply	

On the bottom, there are two jumpers that correspond to the I2C address and pull-up enable.

The following options are available:

- **The I2C Address Jumper** -- Bridge to use alternate address 0x18, otherwise leave open for 0x19. Leave open for SPI use.
- **The I2C Pull-up Enable** -- Closed by default, this connects a pull-up resistor between the I2C lines and VCC. This generally doesn't interfere with SPI operation, but, if less power consumption is required, carefully cut the copper traces.

Wiring:



LIS3DH	Arduino
SCL	A5(SCL)
SDA	A4(SDA)
3V3	VCC
GND	GND

Example: I2C

Basic Accelerometer Data Collection:

Start with just the basic accelerometer sketch, also called "MinimalistExample" from the library. This will periodically samples the sensor and displays data as number of Gs detected. Remember, the vertical axis will read 1G while sitting at rest.

```
#include "SparkFunLIS3DH.h"
#include "Wire.h"
#include "SPI.h"

LIS3DH myIMU; //Default constructor is I2C, addr 0x19.

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  delay(1000); //relax...
  Serial.println("Processor came out of reset.\n");

  //Call .begin() to configure the IMU
  myIMU.begin();

}

void loop()
{
  //Get all parameters
  Serial.print("\nAccelerometer:\n");
  Serial.print(" X = ");
  Serial.println(myIMU.readFloatAccelX(), 4);
  Serial.print(" Y = ");
  Serial.println(myIMU.readFloatAccelY(), 4);
  Serial.print(" Z = ");
  Serial.println(myIMU.readFloatAccelZ(), 4);

  delay(1000);
}
//////////////////////////////////////END//////////////////////////////////////
```

Example output:

```
Processor came out of reset.


Accelerometer:
 X = -0.1481
```

```
 Y = -0.1361
 Z = 0.9768

Accelerometer:
 X = -0.1481
 Y = -0.1361
 Z = 0.9768

Accelerometer:
 X = -0.1481
 Y = -0.1361
 Z = 0.9768

Accelerometer:
 X = -0.1481
 Y = -0.1361
 Z = 0.9768
```

When run, the sketch will display data in Gs to the serial terminal. Every second, the data is collected and printed.