



## SmartElex Micro 6DoF IMU - ISM330DHCX

The SmartElex Micro 6DoF IMU - ISM330DHCX breakout features STMicroelectronics' ISM330DHCX; a high-performance 3D digital accelerometer and 3D digital gyroscope tailored for Industry 4.0 applications such as platform, optical image, and lens stabilization, robotics and industrial automation, navigations systems, and vibration monitoring and compensation.



The accelerometer has a full-scale acceleration range of  $\pm 2/\pm 4/\pm 8/\pm 16$  g and a wide angular rate range of  $\pm 125/\pm 250/\pm 500/\pm 1000/\pm 2000/\pm 4000$  dps that enable its usage in a broad range of applications. An unmatched set of embedded features (Machine Learning Core, programmable FSM, FIFO, sensor hub, event decoding and interrupts) are enablers for implementing smart and complex sensor nodes which deliver high performance at very low power.

The ISM330DHCX is a small, system-in-package from STMicroelectronics featuring a high-performance 3D digital accelerometer and 3D digital gyroscope capable of wide bandwidth, ultra-low noise and a selectable full-scale range of  $\pm 2/\pm 4/\pm 8/\pm 16$  g. The 3D gyroscope has an angular rate range of  $\pm 125/\pm 250/\pm 500/\pm 1000/\pm 2000/\pm 4000$  dps and offers superior stability over temperature and time along with ultra-low noise.

The ISM330DHCX can run in four different modes:

## Mode 1

This is the default "peripheral only" mode. This mode allows you to use either I<sup>2</sup>C or SPI. By default, I<sup>2</sup>C is enabled with an address of 0x6B. By manipulating the associated jumper, you can change the I<sup>2</sup>C address to 0x6A (cut the power side and close the ground side) or switch to SPI mode (both jumpers open).

## Power

Ideally, power will be supplied via the connectors, but we've also broken out plated through hole pins to supply 3.3V and Ground. Make sure to pay attention to logic levels - supply voltage to this board should range from **1.71 V to 3.6 V**.

## I<sup>2</sup>C

For flexibility, we've broken out the I<sup>2</sup>C functionality as seen below. Primary I<sup>2</sup>C pins are broken out to SDA and SCL

Secondary I<sup>2</sup>C pins are broken out to SDX/PICOX and SCX. These pins are used solely for Mode 2- Sensor Hub Mode - where the 6DoF reads other sensors. You must cut the PICOX and SCX jumpers on the back of the board in order to use this mode.

## Interrupt Pins: 6DoF - ISM330DHCX

Interrupt generation can be sourced by the gyroscope or accelerometer; for interrupt-generation purposes, the accelerometer/gyroscope sensor has to be set in an active operating mode (not in Power-Down).

The interrupt generator can be configured to detect:

- Free-fall;
- Wake-up;
- 6D/4D orientation detection;
- Single-tap and double-tap sensing;
- Activity/Inactivity and Motion/Stationary recognition

## Interrupt Pin: Micro 6DoF IMU Breakout - ISM330DHCX

We've broken out a single PTH pin for Interrupt functionality on the 6DoF - ISM330DHCX Micro Breakout.

## Jumpers

## I<sup>2</sup>C

The 6DoF - ISM330DHCX comes equipped with pull-up resistors on the clock and data pins. If you are daisy-chaining multiple devices, you will want to cut this jumper; if multiple sensors are connected to the bus with the pull-up resistors enabled, the parallel equivalent resistance will create too strong of a pull-up for the bus to operate correctly. As a general rule of thumb, disable all but one pair of pull-up resistors if multiple devices are connected to the bus. To disable the pull up resistors, use an X-acto knife to cut the joint between the two jumper pads highlighted below.

## LED

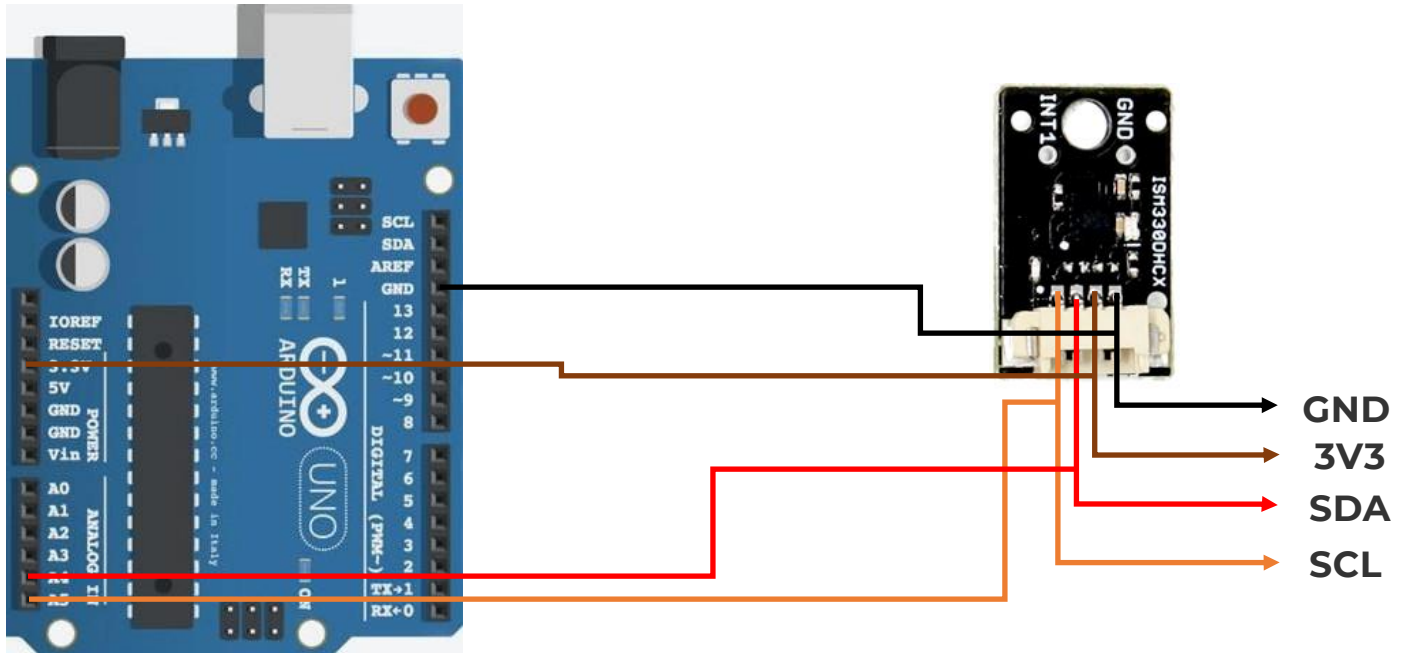
If power consumption is an issue, cutting this jumper will disable the Power LED on the front of the board.

## I<sup>2</sup>C Address - Micro 6DoF

SPI is not available on the Micro Breakout, but you can select the I<sup>2</sup>C address on the back of the board by closing either side of this trace.

| Address |      |                   |
|---------|------|-------------------|
| 1       | VDD  | 0x6B              |
| 0       | GND  | 0x6A              |
|         | OPEN | SPI Not Supported |

## Wiring



| Arduino | ISM330DHCX |
|---------|------------|
| A5(SCL) | SCL        |
| A4(SDA) | SDA        |
| 3.3V    | 3V3        |
| GND     | GND        |

## Arduino library

Sparkfun has written a simple Arduino library to quickly get started reading data from the ISM330DHCX. Install the library through the Arduino Library Manager tool by searching for "**SparkFun Qwiic 6DoF - ISM330DHCX**".

## Example Code - Basic Reading

The first example initializes the 6DoF to communicate over I<sup>2</sup>C with default settings. Open the example by navigating to **File Examples > SparkFun 6DOF ISM330DHCX > example1\_basic**. Select your Board and Port and click upload.

```
#include <Wire.h>
#include "SparkFun_ISM330DHCX.h"

SparkFun_ISM330DHCX myISM;

// Structs for X,Y,Z data
sfe_ism_data_t accelData;
sfe_ism_data_t gyroData;

void setup(){

  Wire.begin();

  Serial.begin(115200);

  if( !myISM.begin() ){
    Serial.println("Did not begin.");
    while(1);
  }

  // Reset the device to default settings. This if helpful is you're doing multiple
  // uploads testing different settings.
  myISM.deviceReset();

  // Wait for it to finish resetting
  while( !myISM.getDeviceReset() ){
    delay(1);
  }

  Serial.println("Reset.");
  Serial.println("Applying settings.");
  delay(100);

  myISM.setDeviceConfig();
  myISM.setBlockDataUpdate();

  // Set the output data rate and precision of the accelerometer
  myISM.setAccelDataRate(ISM_XL_ODR_104Hz);
  myISM.setAccelFullScale(ISM_4g);

  // Set the output data rate and precision of the gyroscope
  myISM.setGyroDataRate(ISM_GY_ODR_104Hz);
  myISM.setGyroFullScale(ISM_500dps);
```

```

// Turn on the accelerometer's filter and apply settings.
myISM.setAccelFilterLP2();
myISM.setAccelSlopeFilter(ISM_LP_ODR_DIV_100);

// Turn on the gyroscope's filter and apply settings.
myISM.setGyroFilterLP1();
myISM.setGyroLP1Bandwidth(ISM_MEDIUM);

}

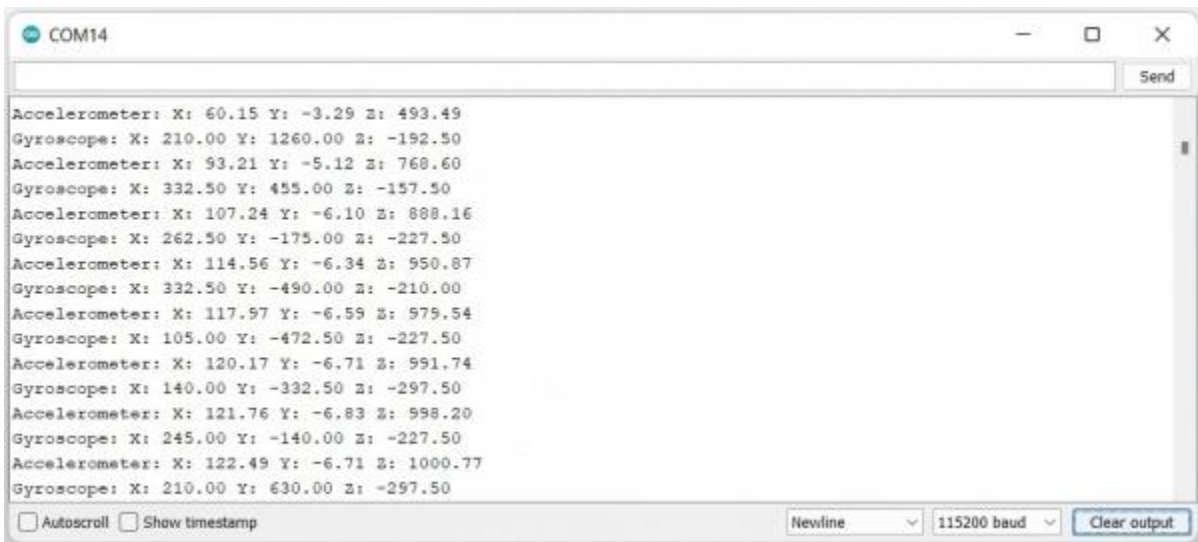
void loop(){

// Check if both gyroscope and accelerometer data is available.
if( myISM.checkStatus() ){
  myISM.getAccel(&accelData);
  myISM.getGyro(&gyroData);
  Serial.print("Accelerometer: ");
  Serial.print("X: ");
  Serial.print(accelData.xData);
  Serial.print(" ");
  Serial.print("Y: ");
  Serial.print(accelData.yData);
  Serial.print(" ");
  Serial.print("Z: ");
  Serial.print(accelData.zData);
  Serial.println(" ");
  Serial.print("Gyroscope: ");
  Serial.print("X: ");
  Serial.print(gyroData.xData);
  Serial.print(" ");
  Serial.print("Y: ");
  Serial.print(gyroData.yData);
  Serial.print(" ");
  Serial.print("Z: ");
  Serial.print(gyroData.zData);
  Serial.println(" ");
}

  delay(100);
}
//////////////////////////////////////////////////////////////////END//////////////////////////////////////////////////////////////////

```

Open the serial monitor after the upload completes with the baud set to **115200** to watch data print out.



The image shows a terminal window titled "COM14" with a "Send" button in the top right corner. The window contains a list of sensor readings for an accelerometer and a gyroscope. Each reading consists of a label (Accelerometer or Gyroscope) followed by X, Y, and Z coordinates. At the bottom of the window, there are checkboxes for "Autoscroll" and "Show timestamp", a dropdown menu for "Newline", a dropdown menu for "115200 baud", and a "Clear output" button.

```
Accelerometer: X: 60.15 Y: -3.29 Z: 493.49
Gyroscope: X: 210.00 Y: 1260.00 Z: -192.50
Accelerometer: X: 93.21 Y: -5.12 Z: 768.60
Gyroscope: X: 332.50 Y: 455.00 Z: -157.50
Accelerometer: X: 107.24 Y: -6.10 Z: 888.16
Gyroscope: X: 262.50 Y: -175.00 Z: -227.50
Accelerometer: X: 114.56 Y: -6.34 Z: 950.87
Gyroscope: X: 332.50 Y: -490.00 Z: -210.00
Accelerometer: X: 117.97 Y: -6.59 Z: 979.54
Gyroscope: X: 105.00 Y: -472.50 Z: -227.50
Accelerometer: X: 120.17 Y: -6.71 Z: 991.74
Gyroscope: X: 140.00 Y: -332.50 Z: -297.50
Accelerometer: X: 121.76 Y: -6.83 Z: 998.20
Gyroscope: X: 245.00 Y: -140.00 Z: -227.50
Accelerometer: X: 122.49 Y: -6.71 Z: 1000.77
Gyroscope: X: 210.00 Y: 630.00 Z: -297.50
```

Autoscroll  Show timestamp    Newline    115200 baud    Clear output