Intel® Integrated Sensor Solution Calibration Tool

User Guide

Version 0.94

Document Release Date: August 2020

Legal Notice and Disclaimer

Contents

Figures

Revision History

Version	Details
0.3	First release
0.4	CLI and GUI changes; new screenshots
0.5	Added documentation of GUI tool dependencies; Updated log format
0.6	Added ALS and magnetometer pre-test sections; Updated screenshots; Select sensors option; Added System requirements
0.7	Add system requirement
0.9	Add Calibration Data Format and Location and Raw Data File sections. More elaboration to the CLI interface
0.91	Add Linux version support
0.92	Corrected some typos and format errors; Updated screenshot figures; Added –SecondaryOnly option usage
0.93	Clarify DUT placement requirement for devices with hinge junction (like 2-in-1 etc.)
0.94	Add AGS calibration requirement.

Terms & Abbreviations

Abbreviation	Definition
DUT	Device Under Test
ADB	Android® Debug Bridge
OEM	Original Equipment Manufacturer
ODM	Original Design Manufacturer

1 Overview

The Intel® Integrated Sensor Solution Calibration Tool is a stand-alone tool for use by OEMs and ODMs in calibrating sensors connected to the Intel Integrated Sensor Solution - The Intel Integrated Sensor Solution will use the errors calculated by the tool to compensate for errors introduced by system design or manufacturing variations.

Sensor error can occur for several reasons, including the following:

- Design influences:
  - Sensors may be tilted within a system design.
  - Other system components may affect the system, e.g. magnetic flux.
- Manufacturing variations:
  - Sensors provided by a third-party vendor may have varying sensitivities or errors.
  - Soldering processes within a manufacturing line may cause components to tilt or warp within their package leading to sensor error.
- Changes in system environment:
  - Changes in temperature may cause a pressure sensor to drift.
  - External magnetic fields, e.g. power lines, may affect the system magnetometer.

The tool can write the sensor calibration directly onto the firmware or output an XML calibration file. This enables:

- Per Model Calibration: measures sensor error and determines sensor orientation based on one or a small sub-set of systems. The per-model calibration assumes that variations in systems on the manufacturing line are small and negligible.
- Per System Calibration: typically performed on the manufacturing line for all systems if needed. Per-system calibration should be avoided as it increases the overall labor cost for a system. It is typically required when the difference in calibration error on system individuals under the same model is great enough that the per-model calibration is insufficient.

Intel recommends that a per-model calibration is achieved by analyzing the calibration XML created from multiple systems. The Intel Calibration Analysis tool will generate a "best fit" per-model calibration that the tool generates from the calibration XML of a minimum of 20 or more systems.

The Intel Integrated Sensor Solution Calibration Tool is installed locally on the device under test; generates an .xml-based calibration file and includes debug logging prints.

The Intel Integrated Sensor Solution Calibration Tool can be run through command line interface or graphic user interface. The tool can be activated locally or remotely on Windows and Android operating systems via a wireless connection. In all instances it must be run with administrative privileges, unless running in offline mode.

The Intel Integrated Sensor Solution Calibration Tool is included in the Intel Integrated Sensor Solution kit and located in:

```
SW\Calibration\rotationCalibration
```

This folder includes the following:

- Intel Integrated Sensor Solution Calibration Tool User Guide
- Windows Calibration Tool
- Android Calibration Tool
- Linux Calibration Tool (Command Line Interface Only)
- GUI Calibration Tool

The Intel Integrated Sensor Solution Calibration Tool is designed to run on the device under test (DUT) after all Intel Integrated Sensor Solution drivers are installed and functioning. Before running the Intel Integrated Sensor Solution Calibration Tool, please make sure all sensors to be calibrated are connected and responsive.

When doing A/G/M/ALS calibration. Please close the not related sensors or service. Because they will cause the calibration fail.

2 Set-up & Requirements

2.1 Physical Testing Environment

The DUT (device under test) must follow these guidelines.

The DUT must contain Intel® Sensor Solution Hardware.

The calibration needs to be conducted in the following environment:

1. Environment with a fixed magnetic field. Hard iron, i.e. constant magnetic field, will not impact the calibration. Soft iron. I.e. fluctuating magnetic field, will impact the calibration.

2. The DUT must be placed on a leveled table immune to vibrations

3. An equipment designed to perform accurate DUT positional manipulations to achieve high calibration accuracy should be used.

4. For calibration of Accelerometer/Gyrometer/Magnetometer, if the DUT contains hinge junction like 2-in-1 convertible, the DUT must be kept as fully opened, i.e. tablet mode (or fully closed if the DUT does not support tablet mode) during the calibration process.

The Intel Integrated Sensor Solution Calibration Tool will run with u-drivers and algorithms created by Intel. It can provide calibration data to 3rd party u-drivers and dynamic calibration algorithms that follow the same format used by Intel.

A functioning regular Accelerometer is required in order to calibrate either the Magnetometer or Gyrometer sensors.

2.2 Supported Operating Systems

DUT supported operating systems:

- Windows 8.1 and above
- Windows PE 10
- Android Lollipop and above
- Yocto Linux 64 bit

Remote activation supported operating systems:

- Windows 7 and above

2.3 Remote Platform

The DUT calibration can be activated remotely over a wireless connection to the DUT. The remote platform must be running Windows® 7 or above. A remote connection from a native Android platform is not supported.

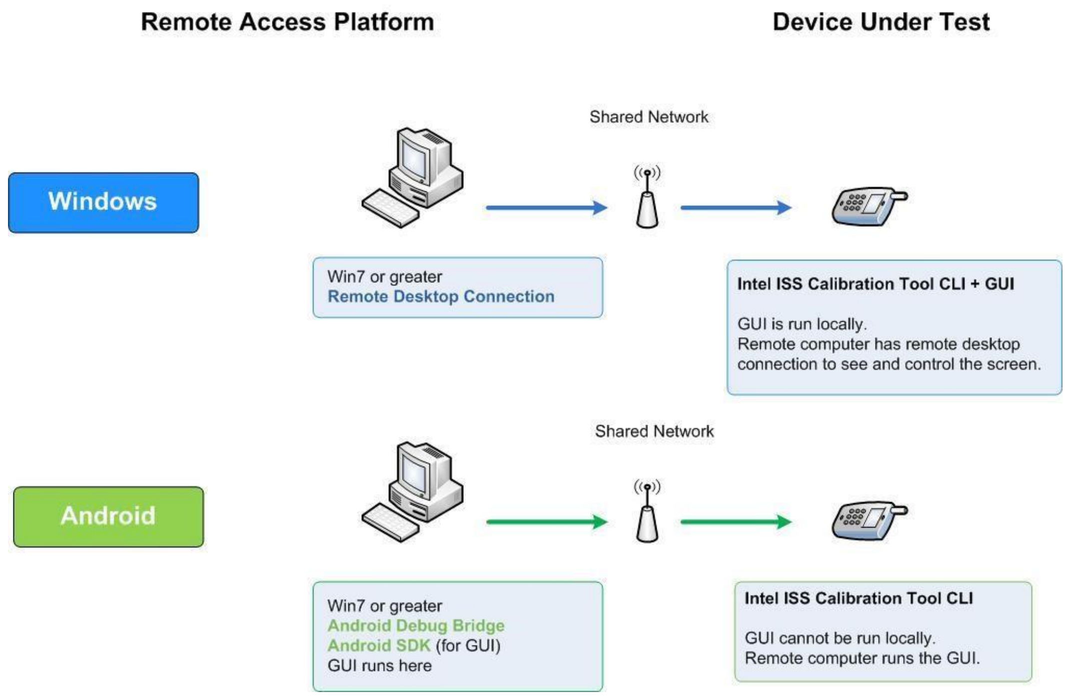


Figure 1 Remote Access Options

2.3.1 Windows Remote

The Intel Integrated Sensor Solution Calibration Tool runs locally on the DUT. Either the command line or graphic user interface can run. Remote control can be activated using Remote Desktop Connection or any other remote control program.

2.3.2 Android Remote

The Intel Integrated Sensor Solution Calibration Tool runs locally on the DUT as the command line interface. Remote control can be activated using the Android Debug Bridge (ADB) or any other Android remote control option the user desires.

To use the Intel Integrated Sensor Solution Calibration Tool GUI, the remote platform must include:

- Android SDK installed.
- ANDROID_HOME environment variable must point to the Android SDK installation directory.
- Android Debug Bridge TCP mode activated.

- Remote platform and DUT on the same network.

2.3.3.1 Activating Android Debug Bridge TCP Mode

The Android Debug Bridge TCP mode can be activated automatically by an application or manually by commands.

Automation Applications

There are several apps on Google Play that automate this process, including ADB Wireless, WIFI ADB and ADB WIFI.

Manual Process

If your device is rooted, you can enable Android Debug Bridge over WIFI. If you already have a USB connection between your remote platform and the DUT it is even easier to activate the TCP.

The following table lists the commands to issue to activate TCP for remote Android control and return the Android Debug Bridge to listening on USB.

Connectivity	Activate TCP	Disable & Return to listening on USB
USB	adb tcpip 5555	adb usb
WIFI	su setprop service.adb.tcp.port 5555 stop adbd start adbd	setprop service.adb.tcp.port -1 stop adbd start adbd

3 Calibration Data Format and Location

The calibration data is part of the PDT. The PDT is stored on the flash. When ISS starts, it reads the calibration data to the ISS RAM.

The Calibration Tool has the capability of reading currently stored data in flash and writing to the flash new calibration data. By using the export command (see section 8) the tool creates an XML file which contains current calibration. By using import command, the tool reads an XML file and write its content to ISS flash. By using the calibrate command the tool calculates the calibration data and can write it to ISS flash or export it to XML file. A sample of calibration data file can be found at section 11.1.

Each sensor has a LUID – a unique ID. The ID is combined from sensor type, vendor, model, instance and flags. The LUID identifies to what sensor the calibration data is associated.

The actual calibration data has a format and version.

The format is used to specify how to parse the data. Values 0-127 are Intel® reserved, 128 to 255 are vendor defined.

The calibration tool can parse Intel® formats and treats vendor formats as a binary blob.

4 Magnetic Environment Pre-Calibration Test

The calibration tool provides a test that can be run prior to calibration in order to determine if the environment is magnetically clean.

The test consists of the following steps:

1. Rotating the device 360 degrees around X axis.
2. Rotating the device 360 degrees around Y axis.
3. Rotating the device 360 degrees around Z axis.
4. Translating (moving without rotating) the device in a rectangle in the area used for calibration.

Run the test as follows:

- **Windows CLI**: Run **WindowsCalibrationTool.exe –MagTest [–ExportAllFiles <folder>]**
- **Android CLI**: Run **AndroidCalibrationTool –MagTest [–ExportAllFiles <folder>]**
- **Linux CLI**: Run **LinuxCalibrationTool –MagTest [–ExportAllFiles <folder>]**
- **GUI**: Go to the **Settings** tab and select **Run Test Flow**. Select the **Run Magnetic Environment Pre-Calibration Test** radio button. Click **Start**.

Follow the instructions displayed on the screen. When the test is concluded, you will be informed as to whether it passed or failed.

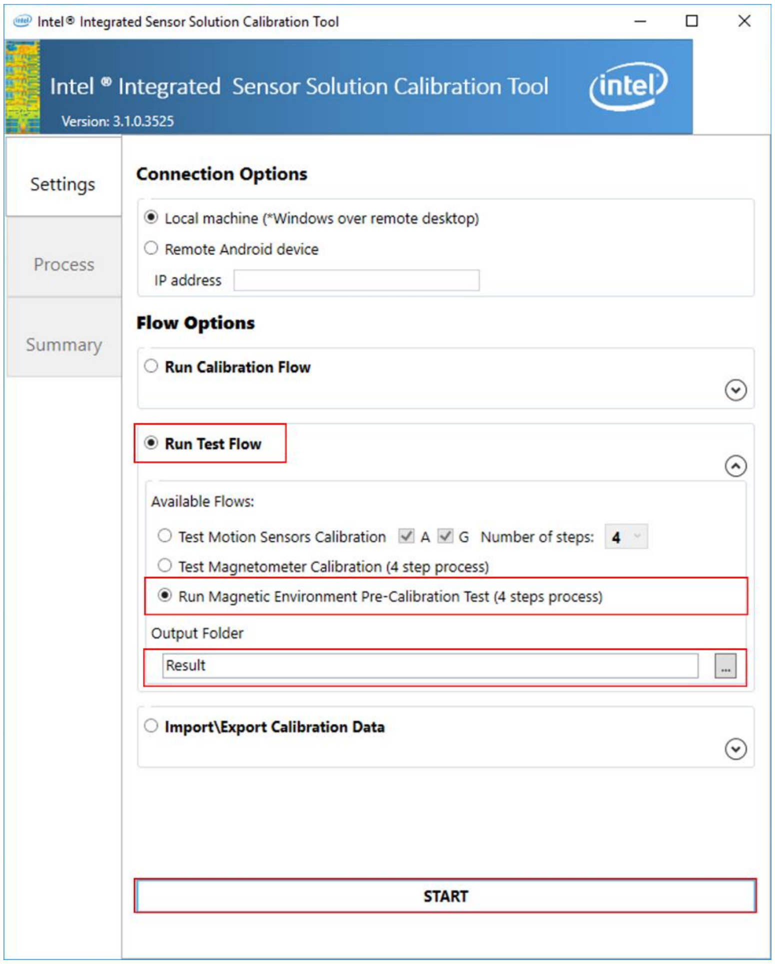


Figure 2 Magnetic Environment Pre-Calibration

5 Motion Sensor Calibration Sequences

This section presents the different device movement sequences that can be used to calibrate your motion sensors (accelerometer, gyrometer, and magnetometer, also known as AGM).

These tests involve orienting and/or rotating the device around three axes:

- The Z axis is perpendicular to the plane of the device. The positive direction is straight up when the device is laid on a flat surface. This axis is represented in diagrams by a blue arrow.
- The X axis is the horizontal axis. When the device is laid on a flat surface, the positive direction of this axis is to the right. This axis is represented in diagrams by a red arrow.
- The Y axis is the vertical axis. When the device is laid on a flat surface, the positive direction of this axis is towards the top of the device, directly away from the user. This axis is represented in diagrams by a green arrow.

In the diagrams in this section, the large axes in the lower left corner of the image are the axes from your perspective; the smaller axes are from the perspective of the device. Although the diagrams display the axes as coming out of the center of the device, you can always turn the device around a parallel axis located along the device's edge.

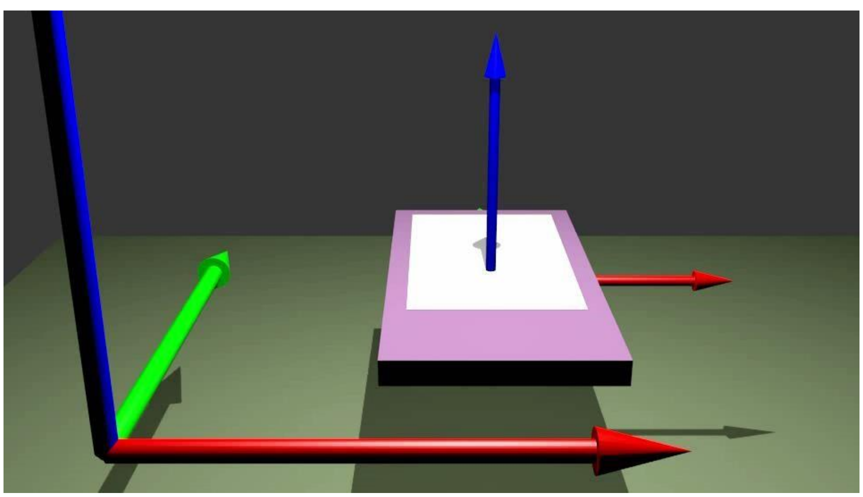


Figure 3 A sample orientation diagram

A diagram always displays the orientation of the device immediately *after* any rotations described in the step have been performed.

A rotation can be performed at any speed, but we recommend that you complete a rotation within 2 seconds of starting it. The rotation must be made smoothly. If you pause in the middle of rotating the device, the calibration tool may incorrectly assume that you have finished the rotation.

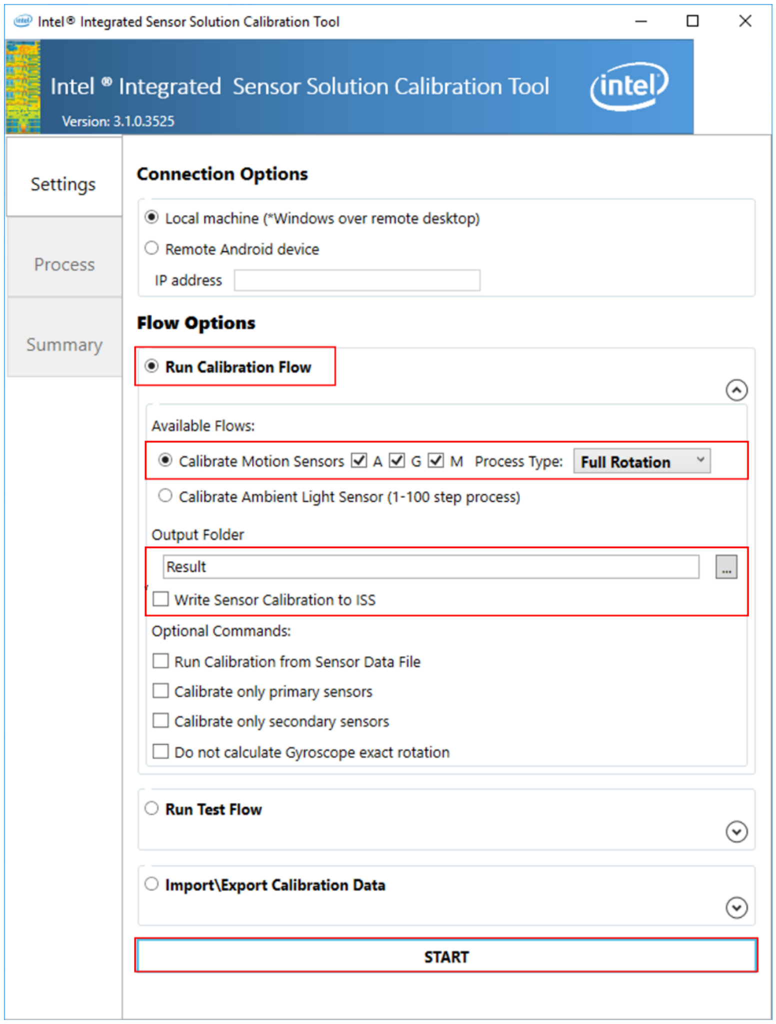
5.1 The Full-Process Motion Sensor Calibration Sequence

The full-process motion sensor calibration sequence is the most thorough, designed to provide accelerometer, gyrometer, and magnetometer calibration on a per-model basis. To successfully calibrate the magnetometer, the calibration must be conducted in a clean magnetic environment.

After completing the full-process calibration sequence, you can supplement it by conducting the four-step calibration sequence or one-step calibration (see below) on a per-system basis. These will refine the calibration of the accelerometer and gyrometer.

Perform the full-process motion sensor calibration sequence as follows:

- **Windows CLI**: Run **WindowsCalibrationTool.exe -Calibrate <sensors> [-ExportAllFiles <folder>] [-UpdateISS] [-PrimaryOnly | -SecondaryOnly] [-UseRawData <dump-folder-path> [-DumpPrefix <dump-file-prefix>]]**
- **Android CLI**: Run **AndroidCalibrationTool -Calibrate <sensors> [-ExportAllFiles <folder>] [-UpdateISS] [-PrimaryOnly | -SecondaryOnly] [-UseRawData <dump-folder-path> [-DumpPrefix <dump-file-prefix>]]**
- **Linux CLI**: Run **LinuxCalibrationTool -Calibrate <sensors> [-ExportAllFiles <folder>] [-UpdateISS] [-PrimaryOnly | -SecondaryOnly] [-UseRawData <dump-folder-path> [-DumpPrefix <dump-file-prefix>]]**
- **GUI**: Go to the **Settings** tab and select **Run Calibration Flow**. Select **Calibrate Motion Sensors** and select which sensors (accelerometer, gyrometer, magnetometer) you want to calibrate. Select **Full Rotation** from the **Process Type** drop-down menu. Choose an output folder and select the **Write Sensor Calibration to ISS** checkbox.



Scroll down to the bottom of the screen and click **Start** to begin the calibration. The Calibration Tool will provide instructions on the screen. The steps of this sequence are as follows; after each step, hold the device in place for two seconds until the calibration tool marks the movement as successful and displays the next instruction:

1. Lay the device face up on the table so that it is oriented correctly for you. In this position, the Z axis points straight up, and the Y axis points

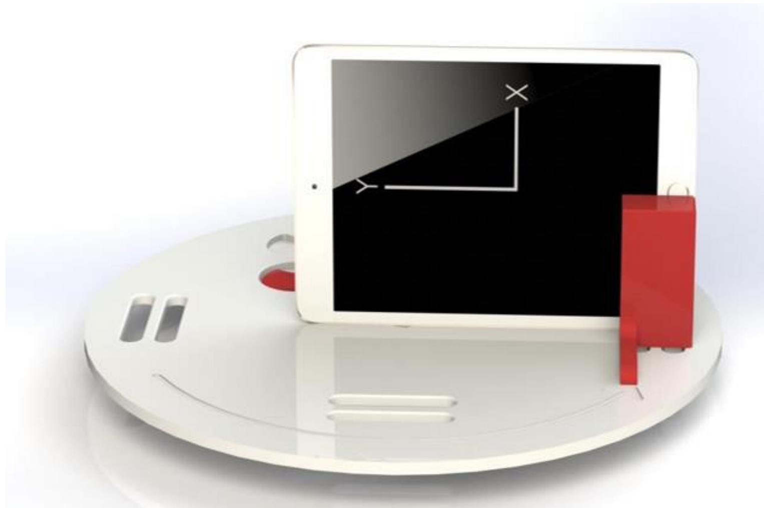
directly away from you. **If you are calibrating the magnetometer, it is crucial that the Y axis be pointing to the north.**

2. Rotate the device 360° around the Z axis in a counterclockwise direction. (In this rotation, no part of the device will be lifted off of the table.)

3. Rotate the device 90° around the X axis. In this position, the Y axis points straight up, and the Z axis points towards you (to the south):

4. Rotate the device 360° around the Y axis in a counterclockwise direction. (In this rotation, no part of the device will be lifted off of the table.)

5. Rotate the device 90° around the Z axis in a counterclockwise direction. In this position, the X axis points straight up, and the Z axis points towards you (to the south):



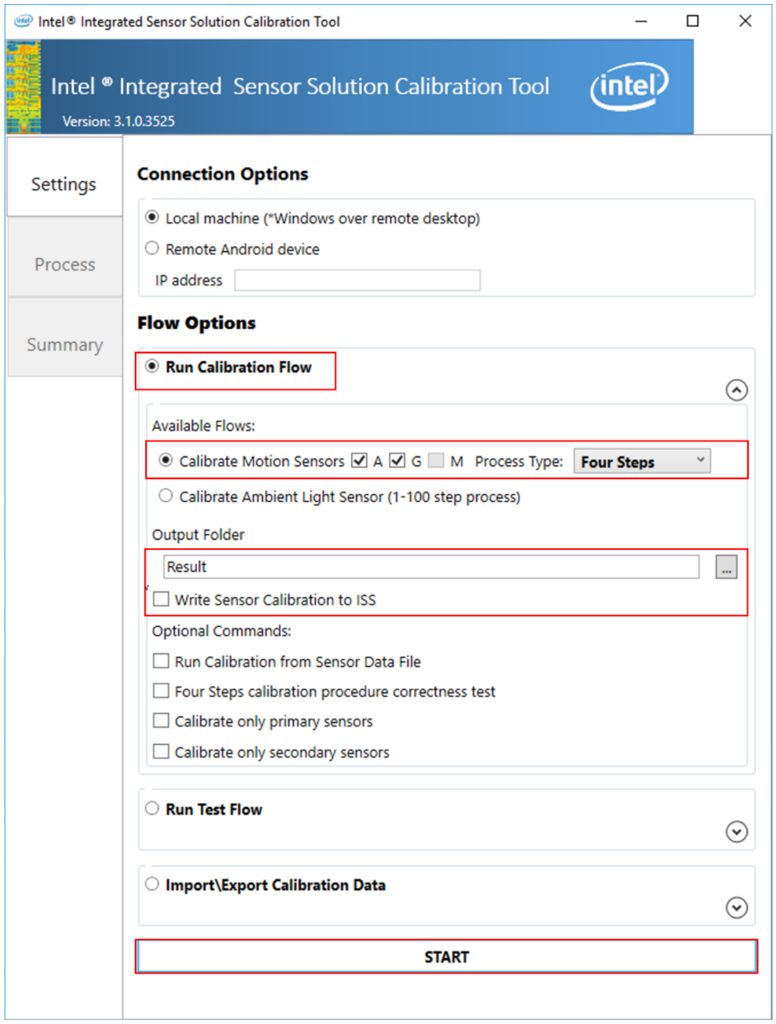
6. Rotate the device 360° around the X axis in a counterclockwise direction. (In this rotation, no part of the device will be lifted off of the table.)
7. Rotate the device 90° around the Z axis in a counterclockwise direction. In this position, the Y axis points straight down, while the Z axis points towards you (to the south).

5.2 The Four-step Motion Sensor Calibration Sequence

The four-step motion sensor calibration sequence is designed to increase the accuracy of the accelerometer and gyrometer calibration that was previously set using the Full-Process sequence above. It does not provide additional calibration for the magnetometer and therefore does not require a clean magnetic environment.

Perform the four steps motion sensor calibration sequence as follows:

- **Windows CLI**: Run **WindowsCalibrationTool.exe -Calibrate <sensors> [-ExportAllFiles <folder>] -FourSteps [-UpdateISS] [-PrimaryOnly | -SecondaryOnly] [-UseRawData <dump-folder-path> [-DumpPrefix <dump-file-prefix>]]**
- **Android CLI**: Run **AndroidCalibrationTool -Calibrate <sensors> [-ExportAllFiles <folder>] -FourSteps [-UpdateISS] [-PrimaryOnly | -SecondaryOnly] [-UseRawData <dump-folder-path> [-DumpPrefix <dump-file-prefix>]]**
- **Linux CLI**: Run **LinuxCalibrationTool -Calibrate <sensors> [-ExportAllFiles <folder>] -FourSteps [-UpdateISS] [-PrimaryOnly | -SecondaryOnly] [-UseRawData <dump-folder-path> [-DumpPrefix <dump-file-prefix>]]**
- **GUI**: Go to the **Settings** tab and select **Run Calibration Flow**. Select **Calibrate Motion Sensors** and select **Four Steps** from the **Process Type** drop-down menu. Finally, choose an output folder, and select the **Write Sensor Calibration to ISS** checkbox.

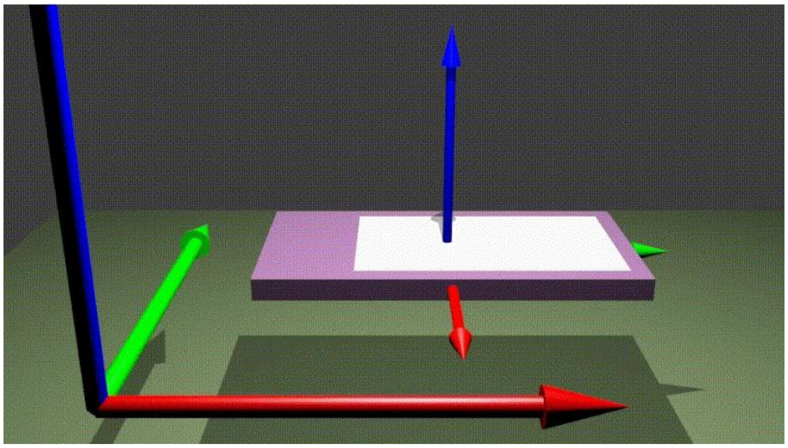


Note that the M checkbox, indicating the magnetometer, has been disabled.

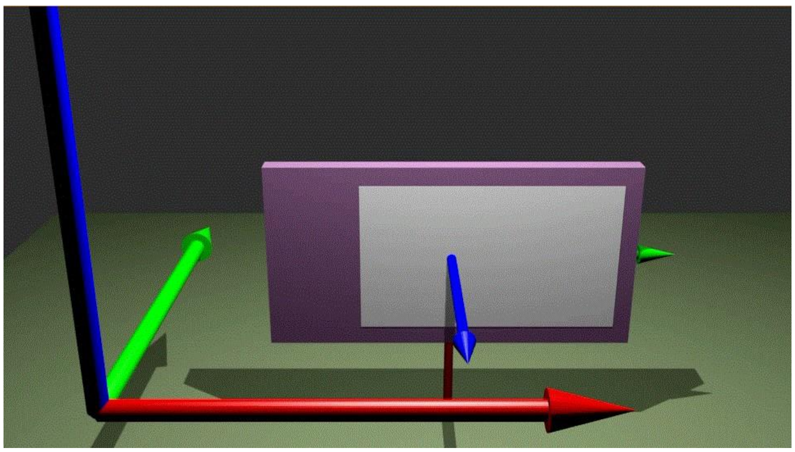
Scroll down to the bottom of the screen and click **Start** to begin the calibration. The Calibration Tool will provide instructions on the screen. The four steps of this sequence are as follows; after each step, hold the device in place for two seconds until the calibration tool acknowledges the success of the movement and provides the next instruction:

1. Place the device on a flat surface, such that the device's X axis points directly to you, the Y axis points to your right, and the Z axis points straight up. In this

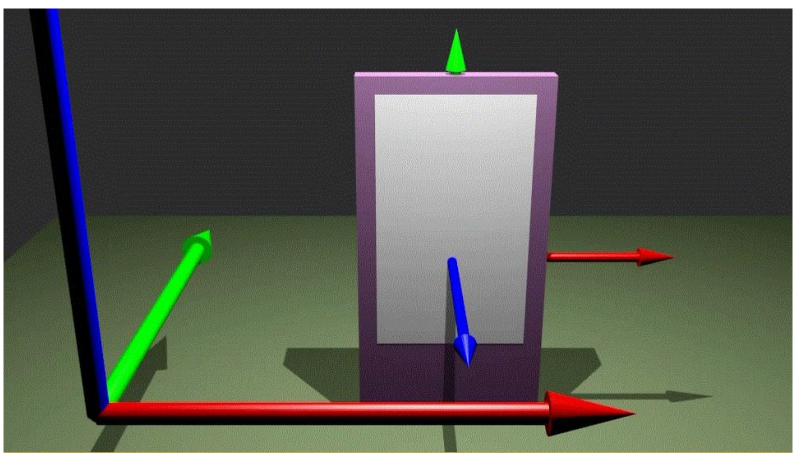
orientation, the device is, in effect, positioned appropriately for somebody sitting 90 degrees clockwise of your position:



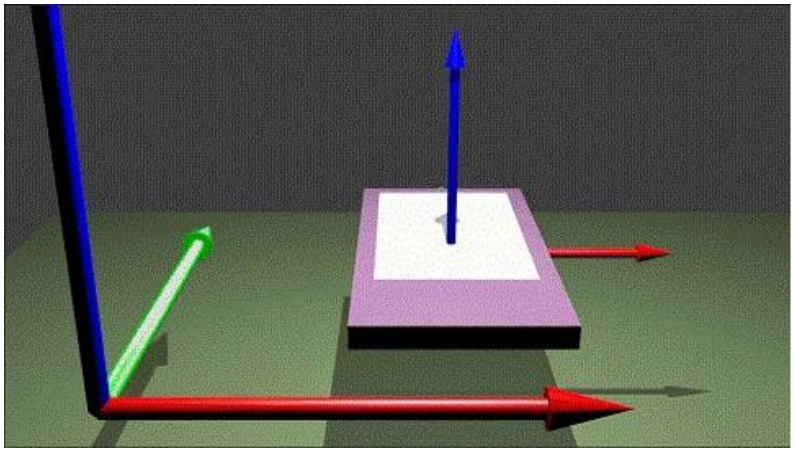
2. Turn the device 90 degrees around its Y axis by lifting the farthest end of the device upwards, until it is standing on its edge:



3. Turn the device 90 degrees around its Z axis by tilting it to the left (counterclockwise) until it is, from your perspective, standing straight up:



4. Turn the device 90 degrees around its X axis by tilting it back until it is flat on the table, correctly oriented for your use:



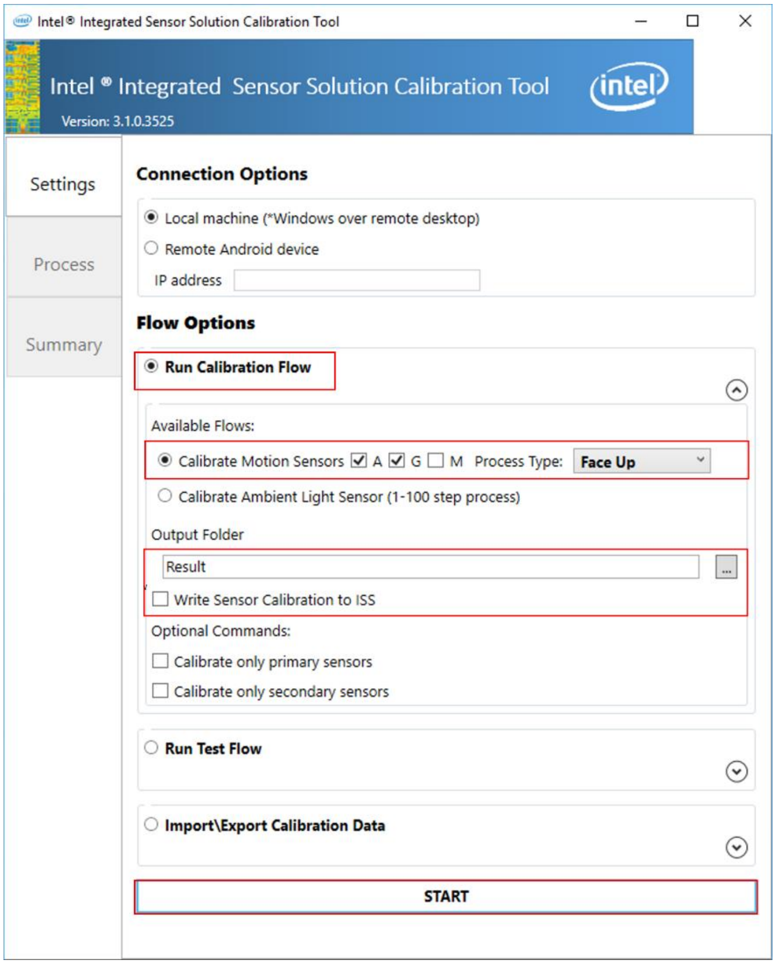
5.3 One-step Motion Sensor Calibration

The one-step calibration option, also known as face-up calibration, can be used on a per-system basis to supplement a Full-Process calibration sequence that was previously performed on a per-model basis. It represents a simpler though less effective alternative to the four-step calibration sequence offered above.

It supports the sensors of accelerometer and gyrometer only.

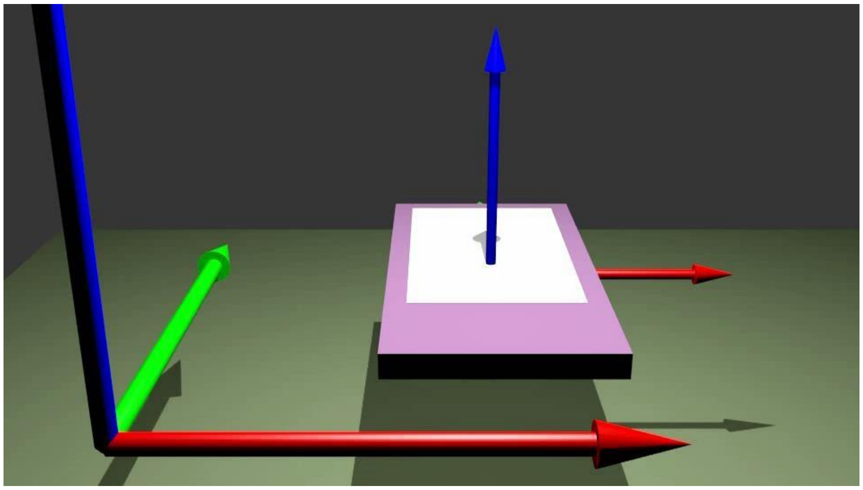
Perform the face up motion sensor calibration sequence as follows:

- **Windows CLI**: Run **WindowsCalibrationTool.exe -Calibrate <sensors> [-ExportAllFiles <folder>] -Minimal [-UpdateISS] [-PrimaryOnly | -SecondaryOnly] [-UseRawData <dump-folder-path> [-DumpPrefix <dump-file-prefix>]]**
- **Android CLI**: Run **AndroidCalibrationTool -Calibrate <sensors> [-ExportAllFiles <folder>] -Minimal [-UpdateISS] [-PrimaryOnly | -SecondaryOnly] [-UseRawData <dump-folder-path> [-DumpPrefix <dump-file-prefix>]]**
- **Linux CLI**: Run **LinuxCalibrationTool -Calibrate <sensors> [-ExportAllFiles <folder>] -Minimal [-UpdateISS] [-PrimaryOnly | -SecondaryOnly] [-UseRawData <dump-folder-path> [-DumpPrefix <dump-file-prefix>]]**
- **GUI**: Go to the **Settings** tab and select **Run Calibration Flow**. Select **Calibrate Motion Sensors** and select **Face Up** from the **Process Type** drop-down menu. Finally, choose an output folder, and select the **Write Sensor Calibration to ISS** checkbox.



Scroll down to the bottom of the window and click **Start** to begin the calibration.

The Calibration Tool will provide instructions, of which there is only one: place the device on a flat surface, with the screen facing up and the lid (if any) closed. (Note that, if you decided to calibrate the magnetometer in addition to the accelerometer and gyrometer, you will need to ensure a clean magnetic environment and to orient the device such that the Y axis points to magnetic north.)



The one-step calibration verifies the following values:

- The accelerometer should show no acceleration along both the X and Y axes, and 1g (or the equivalent in other units) of acceleration along the Z axis. Note that in Windows the acceleration due to gravity is considered a negative number, while on an Android OS the acceleration due to gravity is considered a positive number.

- The gyrometer should show no movement along each axis.

- The magnetometer should show a 0 value along the X axis. The expected values of the magnetometer along the Y and Z axes vary widely depending on your location and will not be verified.

If the expected and actual values do not match each other precisely, the sensors' offsets will be adjusted until they do. However, if the calibration tool judges the values to be too distant from each other, it will assume that something is wrong with the sensors or with the device's orientation and the calibration will not be performed.

6 Motion Sensor Testing Sequences

Once a device’s sensors are calibrated, you can use the Calibration Tool to test the accuracy of the calibration. There are currently two calibration tests that can be performed, each verifying the accuracy of both the accelerometer and gyrometer.

As with the calibration sequences, the testing sequences involve orienting and/or rotating the device around three axes:

- The Z axis is perpendicular to the plane of the device. The positive direction is straight up when the device is laid on a flat surface. This axis is represented in diagrams by a blue arrow.
- The X axis is the horizontal axis. When the device is laid on a flat surface, the positive direction of this axis is to the right. This axis is represented in diagrams by a red arrow.
- The Y axis is the vertical axis. When the device is laid on a flat surface, the positive direction of this axis is towards the top of the device, directly away from the user. This axis is represented in diagrams by a green arrow.

In the diagrams in this section, the large axes in the lower left corner of the image are the axes from your perspective; the smaller axes are from the perspective of the device. Although the diagrams display the axes as coming out of the center of the device, you can always instead turn the device around a parallel axis located along the device's edge.

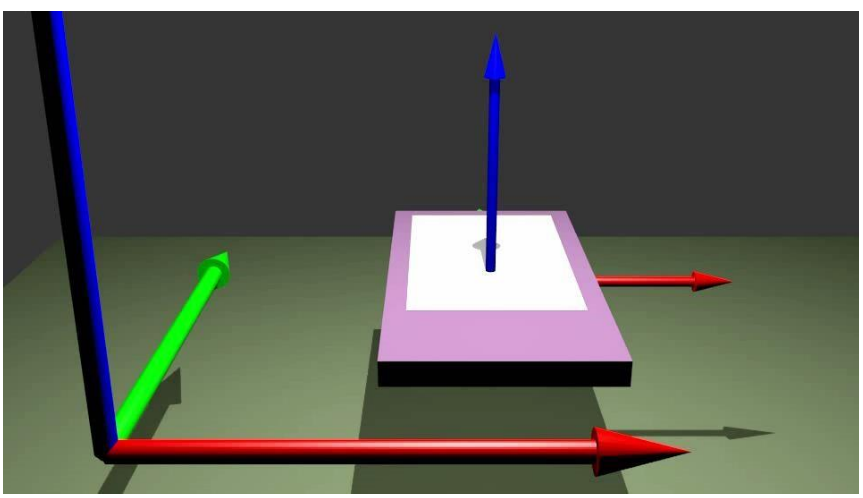


Figure 4 A sample orientation diagram

A diagram always displays the orientation of the device immediately *after* any rotations described in the step have been performed.

A rotation can be performed at any speed, but we recommend that you complete a rotation within 2 seconds of starting it.

The rotation must be made smoothly. If you pause in the middle of rotating the device, the calibration tool may incorrectly assume that you have finished the rotation.

The parameters of the test are highly customizable; you can edit the values in TestingConfig.xml, located in the same directory as the calibration tool executable, to define the results that will pass and fail.

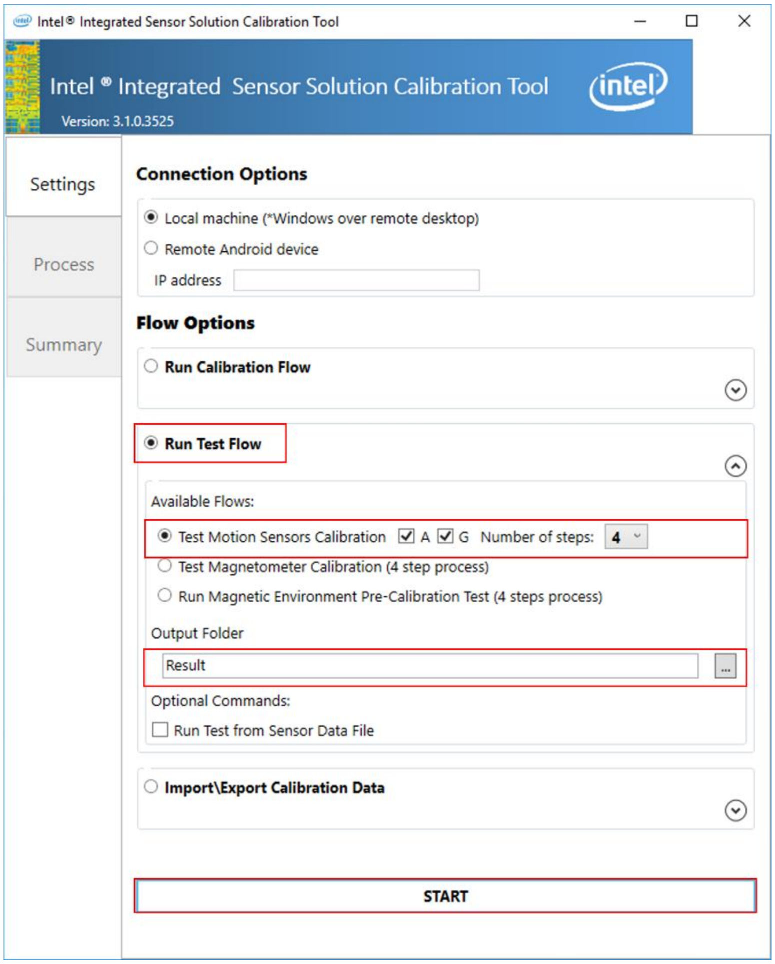
6.1 The Four-step Motion Sensor Testing Sequence

The four-step motion sensor testing sequence is designed to verify the accuracy of the magnetometer alone, or of the accelerometer and gyrometer, after calibration has been applied.

6.1.1 Accelerometer and Gyrometer

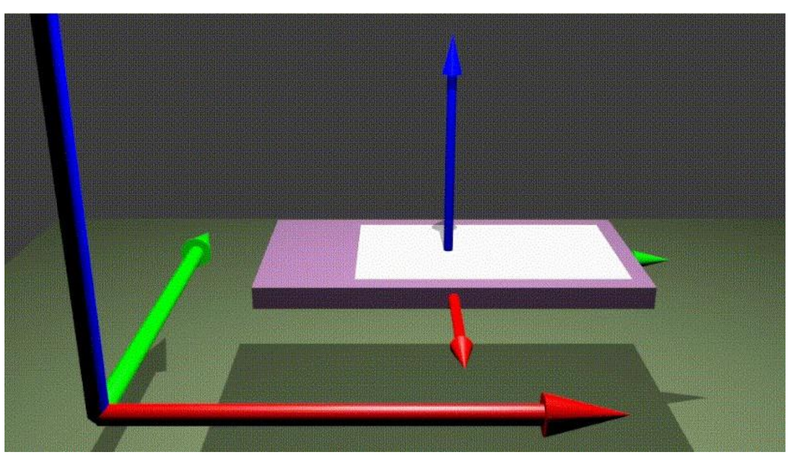
To test the accuracy of the accelerometer and gyrometer, use the following command:

- **Windows CLI**: Run **WindowsCalibrationTool.exe -TestCalibration AG [-ExportAllFiles <folder>] -FourSteps [-PrimaryOnly | -SecondaryOnly]**
- **Android CLI**: Run **AndroidCalibrationTool -TestCalibration AG [-ExportAllFiles <folder>] -FourSteps [-PrimaryOnly | -SecondaryOnly]**
- **Linux CLI**: Run **LinuxCalibrationTool -TestCalibration AG [-ExportAllFiles <folder>] -FourSteps [-PrimaryOnly | -SecondaryOnly]**
- **GUI**: Go to the **Settings** tab and select **Run Test Flow**. Select **Test Motion Sensors Calibration** and select **4** from the **Number of steps** drop-down menu. Choose an output folder for the log file.

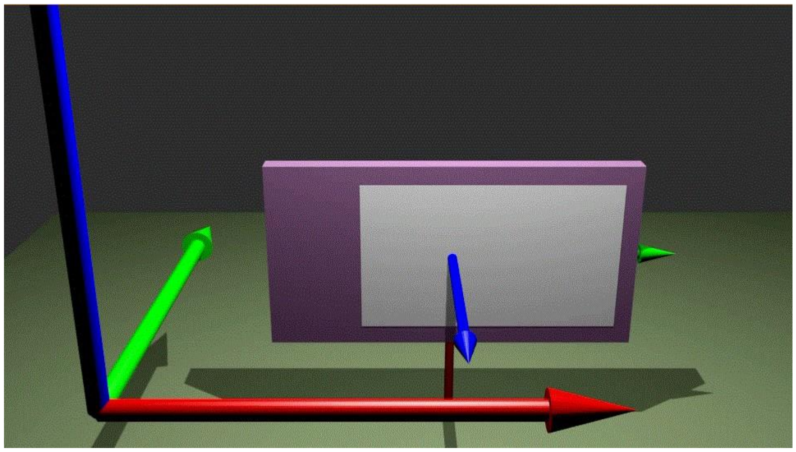


The actions to be performed in this test are identical to those in the four-step calibration sequence. You will perform the following actions in response to prompts on the screen:

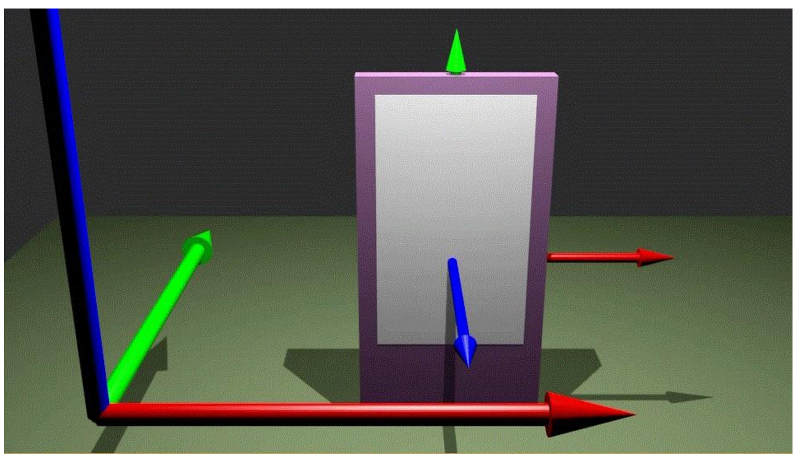
1. Place the device on a flat surface, such that the device's X axis points directly to you, the Y axis points to your right, and the Z axis points straight up. In this orientation, the device is, in effect, positioned appropriately for somebody sitting 90 degrees clockwise of your position:



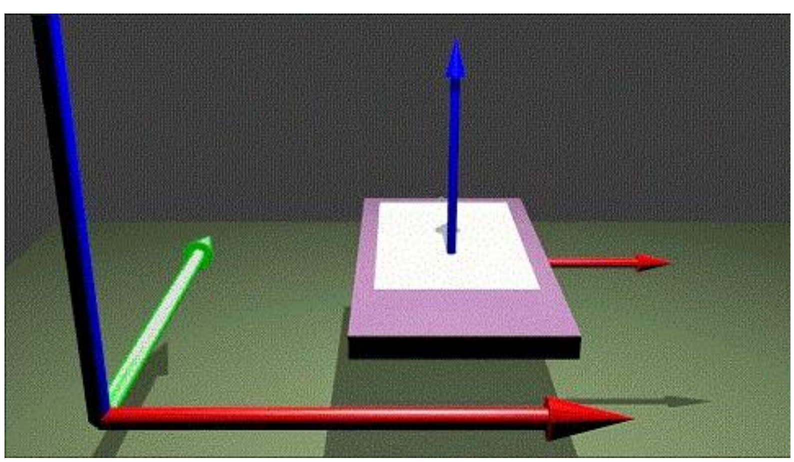
2. Turn the device 90 degrees around its Y axis by lifting the farthest end of the device upwards, until it is standing on its edge:



3. Turn the device 90 degrees around its Z axis by tilting it to the left (counterclockwise) until it is, from your perspective, standing straight up:



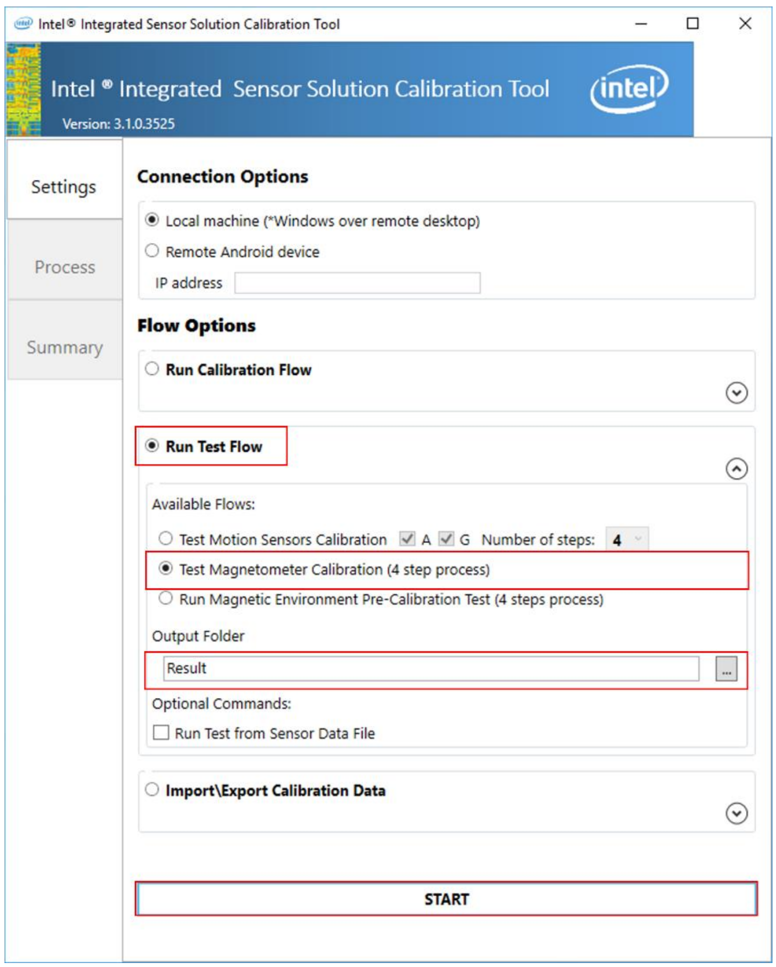
4. Turn the device 90 degrees around its X axis by tilting it back until it is flat on the table, in the position in which it would normally be used:



6.1.2 Magnetometer

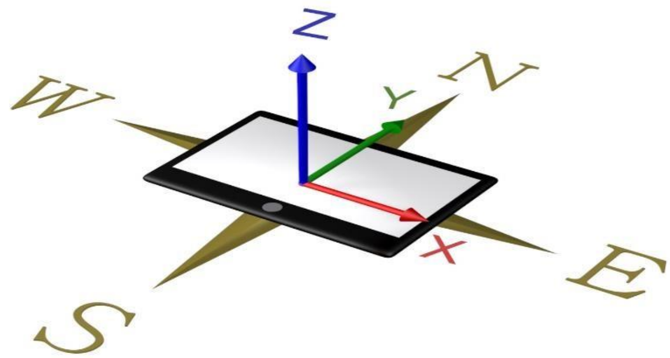
To test the accuracy of the magnetometer, use the following command:

- **Windows CLI**: Run **WindowsCalibrationTool.exe -TestCalibration M [-ExportAllFiles <folder>] -MagAccuracyTest**
- **Android CLI**: Run **AndroidCalibrationTool -TestCalibration M [-ExportAllFiles <folder>] -MagAccuracyTest**
- **Linux CLI**: Run **LinuxCalibrationTool -TestCalibration M [-ExportAllFiles <folder>] -MagAccuracyTest**
- **GUI**: Go to the **Settings** tab and select **Run Test Flow**. Select **Test Magnetometer Calibration**. Choose an output folder for the log file.

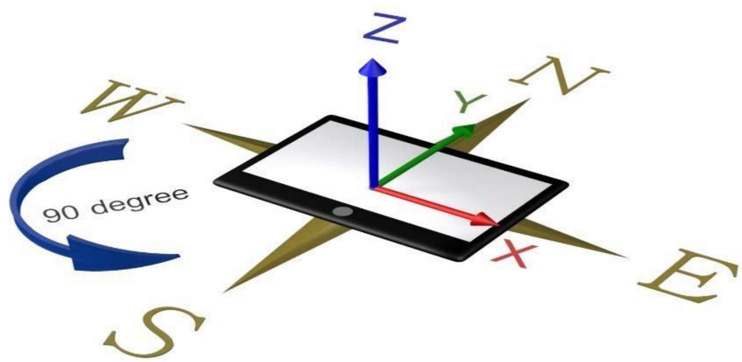


You will perform the following actions in response to prompts on the screen:

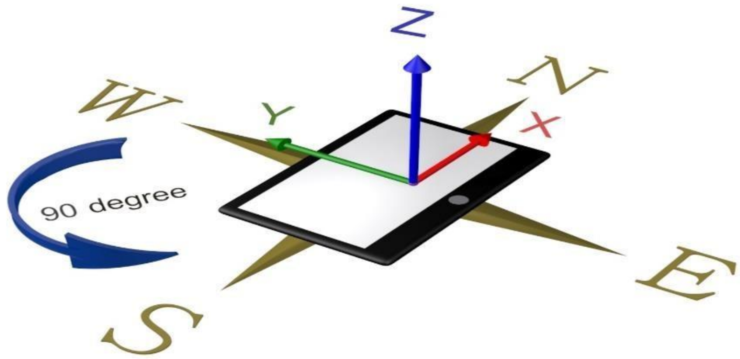
1. Place the device on a flat surface, such that the device's X axis points to your right, the Y axis points directly away from you, and the Z axis points straight up. In this orientation, the device is positioned appropriately for you to work with it:



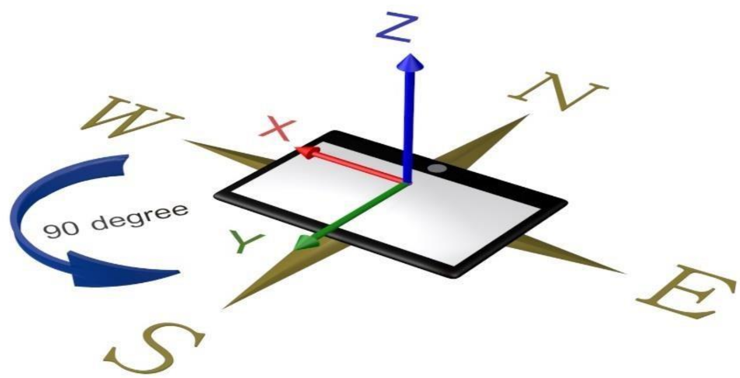
2. Turn the device 90 degrees counterclockwise around the Z axis, such that the X axis is now pointing directly away from you and the Y axis is pointing to your left. In this orientation, the device is, in effect, positioned appropriately for somebody sitting 90 degrees counterclockwise of your position:



3. Turn the device an additional 90 degrees around the Z axis, such that the Y axis is now pointing directly towards you and the X axis is pointing to your left. In this orientation, the device is, in effect, positioned appropriately for somebody sitting directly across from you:



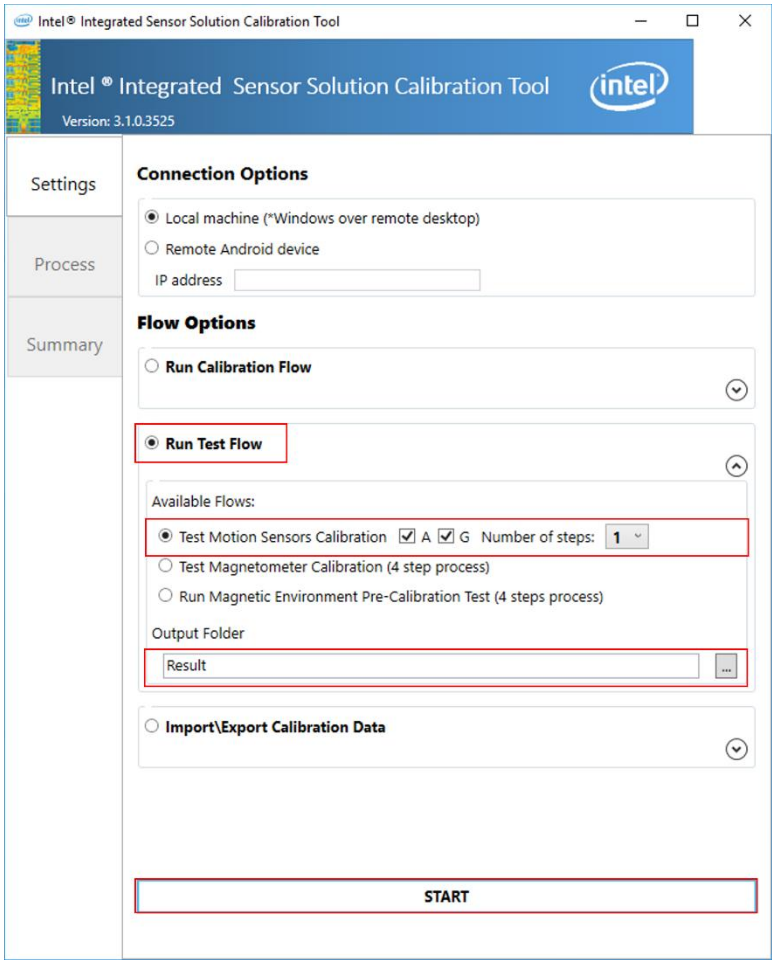
4. Turn the device an additional 90 degrees around the Z axis, such that the X axis is pointing directly towards you, and the Y axis is pointing to your right. In this orientation, the device is, in effect, positioned appropriately for somebody sitting 90 degrees clockwise of your position:



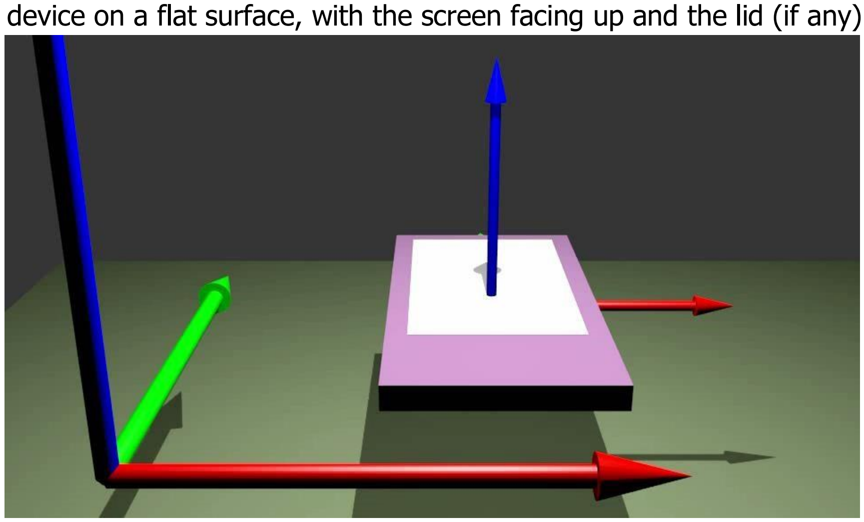
6.2 One-step Motion Sensor Testing

The one-step test, also known as the face-up test, is a simpler though less effective alternative to the four-step testing sequence offered above.

- **Windows CLI**: Run **WindowsCalibrationTool.exe -TestCalibration AG [-ExportAllFiles <folder>] -Minimal [-PrimaryOnly | -SecondaryOnly]**
- **Android CLI**: Run **AndroidCalibrationTool -TestCalibration AG [-ExportAllFiles <folder>] -Minimal [-PrimaryOnly | -SecondaryOnly]**
- **Linux CLI**: Run **LinuxCalibrationTool -TestCalibration AG [-ExportAllFiles <folder>] -Minimal [-PrimaryOnly | -SecondaryOnly]**
- **GUI**: go to the **Settings** tab and select **Run Test Flow**. Select **Test Motion Sensors Calibration** and select **1** from the **Number of steps** drop-down menu. Choose an output folder for the log file.



The Calibration Tool will provide instructions, of which there is only one: place the device on a flat surface, with the screen facing up and the lid (if any) closed.



The one-step test verifies the following values:

- The accelerometer should show no acceleration along both the X and Y axes, and 1g (or the equivalent in other units) of acceleration along the Z axis. Note that in Windows the acceleration due to gravity is considered a negative number, while on an Android OS the acceleration due to gravity is considered a positive number.

- The gyrometer should show no movement along each axis.
- Noise and spikes are at a minimum.

7 Ambient Light Sensor Calibration

ALS calibration is required to correct for the "light error" caused by the ALS module (i.e. cover glass, light pipe) and the sensor itself. The Ambient Light Sensor calibration process is used to determine the error. The result of the calibration is a scale number. The calibration data also includes ALR curve, which is not calculated by the tool. The range for the ALR curve is a curve that is added by the customer and it defines how the customer wants the system to adjust the screen brightness relative to the ALS data.

The output of the process should be data entered into the Calibration XML. This XML would be written directly to the Intel® Integrated Sensor Solution (over HECI) or consumed by other tools.

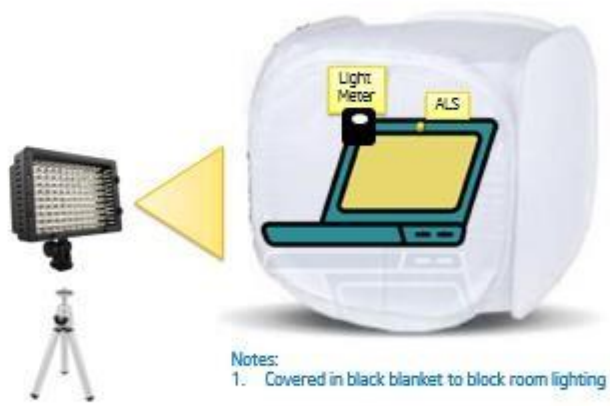


Figure 5 ALS Sensor Calibration

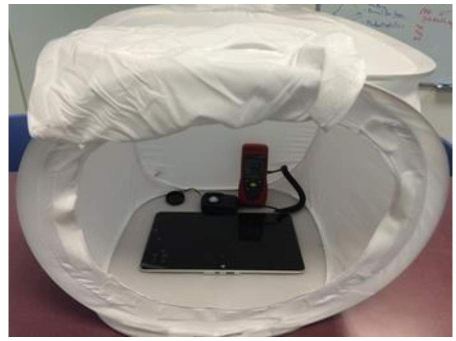
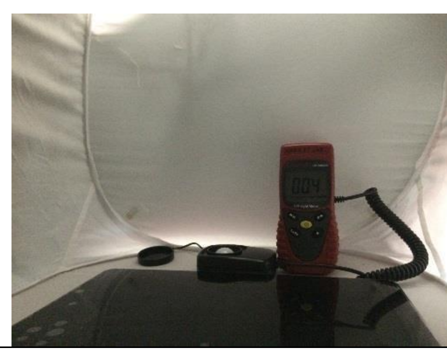
The equipment used to calibration the ALS will include:

- DUT with an ambient light sensor
- Adjustable Halogen Light source
- Calibrated Light meter (optional - if the light values reaching the ALS are not known).
- Light box and cover (optional – to shield the system from stray light)

Note: Calibrated light meters generally have better accuracy compared to uncalibrated (cheaper) light meters. Because the Microsoft Windows Hardware Certification pass/fail criteria for ambient light-testing is +/-10% an uncalibrated light meter may be sufficient – however – this document recommends using a calibrated light meter.

7.1 Light Test Setup

Follow the recommended light-test setup below:

Step 1: Open the light tent and place it on a flat surface. Place the system inside the tent. It should lie face-up with the ALS away from the edges. Setup the light meter “light sensor” next to the ALS of the system.	
Step 2: Place the light meter on top of the tent facing downwards.	
Step 3: Cover the light tent in black material to limit any external light that can affect the measurement.	
Step 4: Check that the light meter within the tent reads close to “zero lux” when the light source is turned off.	

7.2 Calibration Process Flow

The calibration process will run as the user moves the light source from either "**max to min brightness**" or the opposite "**min to max brightness**". The user stops the light a number of times for the program to record the "light level". Minimum delta between different light levels is 300 LUX. The output of the calibration process is the multiplier coefficient.

Manual calibration: The user manually enters in the correct "light" value read from the light meter.

- **Windows CLI**: Run **WindowsCalibrationTool.exe -Calibrate ALS [-ExportAllFiles <folder>] [-PrimaryOnly | -SecondaryOnly]**
- **Android CLI**: Run **AndroidCalibrationTool -Calibrate ALS [-ExportAllFiles <folder>] [-PrimaryOnly | -SecondaryOnly]**
- **Linux CLI**: Run **LinuxCalibrationTool -Calibrate ALS [-ExportAllFiles <folder>] [-PrimaryOnly | -SecondaryOnly]**
- **GUI**: Go to the **Settings** tab and select **Run Calibration Flow**. Select **Calibrate Ambient Light Sensor (1-100 step process)** and select **Manual** from the process types. Specify the **Number of Steps**. Choose an **Output Folder** for the log file.

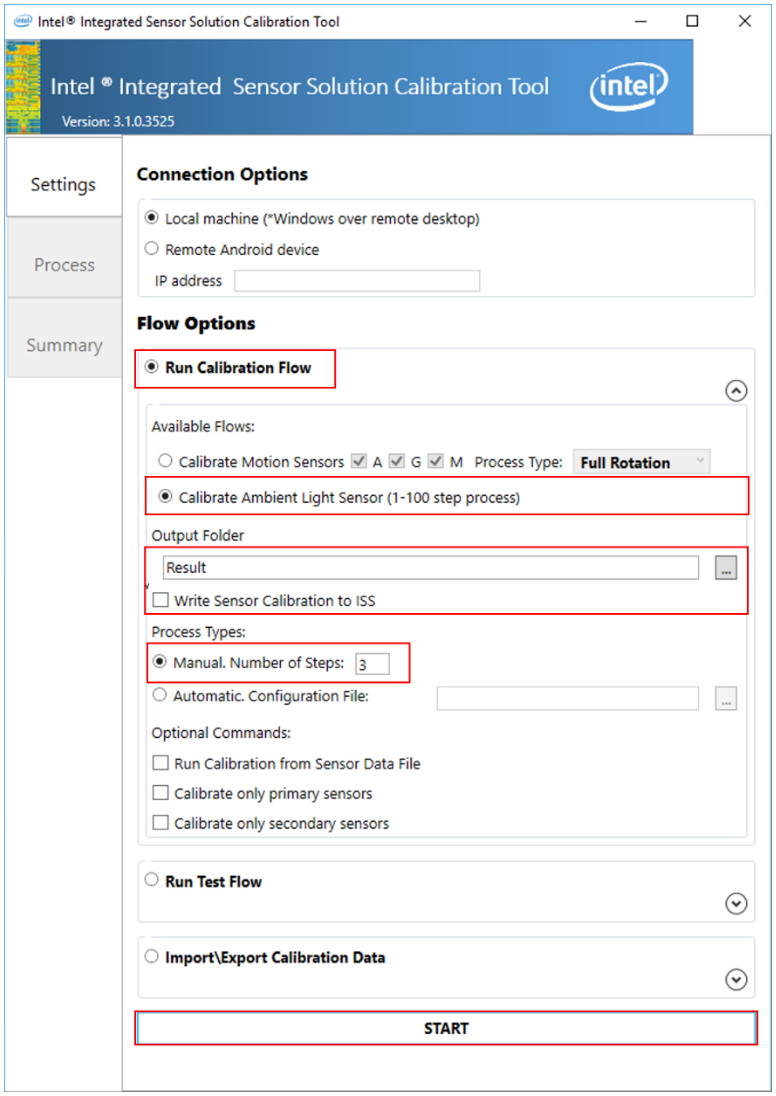


Figure 6 ALS Manual Calibration Screen

Automated calibration: The light values to reach the ALS are known already and entered into a separate configuration file. The SW waits for the light to have stayed at each level for between 2 and 5 seconds before recording each entry.

- **Windows CLI**: Run **WindowsCalibrationTool.exe -Calibrate ALS [-ExportAllFiles <folder>] [-PrimaryOnly | -SecondaryOnly] -Auto <config-file-path>**
- **Android CLI**: Run **AndroidCalibrationTool -Calibrate ALS [-ExportAllFiles <folder>] [-PrimaryOnly | -SecondaryOnly] -Auto <config-file-path>**
- **Linux CLI**: Run **LinuxCalibrationTool -Calibrate ALS [-ExportAllFiles <folder>] [-PrimaryOnly | -SecondaryOnly] -Auto <config-file-path>**
- **GUI**: Go to the **Settings** tab and select **Run Calibration Flow**. Select **Calibrate Ambient Light Sensor (1-100 step process)** and select

Automatic from the process types. Specify the **Configuration File**. Choose an **Output Folder** for the log file.

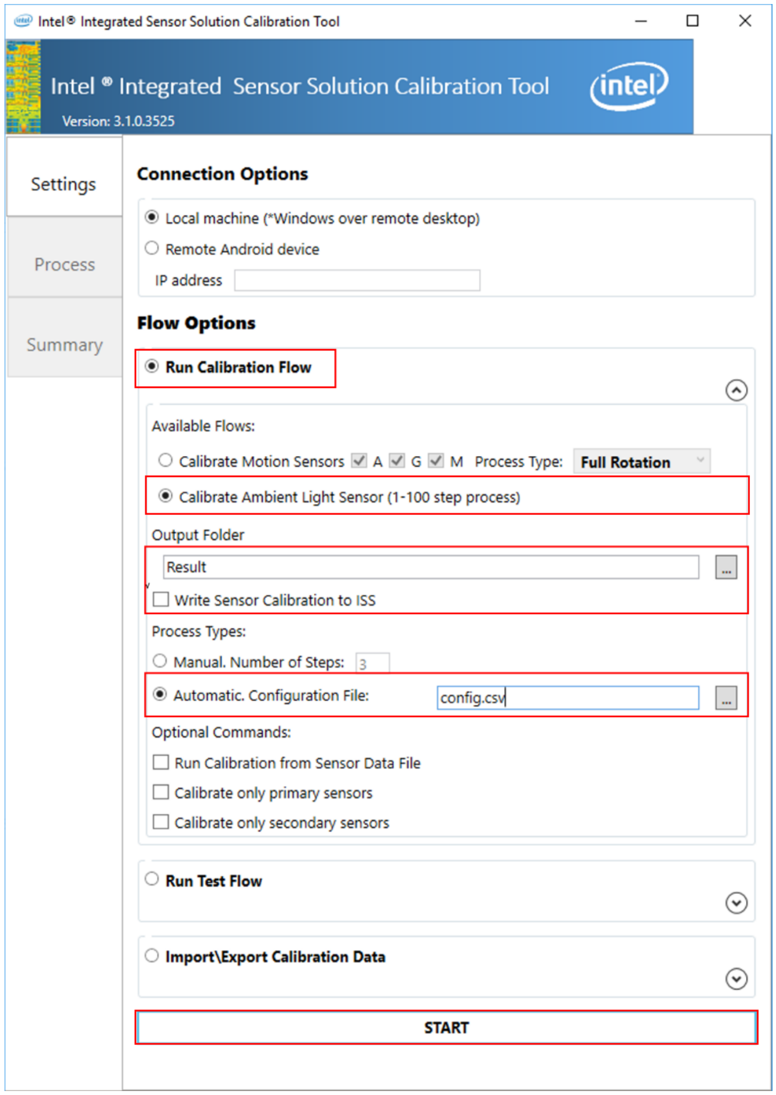


Figure 7 ALS Automated Calibration Screen

7.3 Configuration File Format

The configuration file will consist of the following lines:

delay_time=maximum waiting time (in seconds) to the next light level
values=comma separated list of 1 or more LUX values

Note: the configuration of delay_time is optional; if it's not set in the configuration file, the default value (5 seconds) will be applied. The minimum allowed delay_time is 3 seconds and maximum allowed delay_time is 50 seconds.

8 Calibration Tool CLI

The CLI version of the Calibration Tool must be run with administrative privileges locally on the SUT.

See Typical Workflows sections for more data.

Parameter	Description	Notes
-ExportXml [xml-file-path]	Exports calibration data to XML file	Calibration data is exported from the flash. If calibration is activated, the relevant sensors data will be changed accordingly. This option is usually used for **per-model calibration**.
-ImportXml [xml-file-path]	Imports calibration data from XML file to ISS flash. An ISS reset is performed and the new data is loaded to the RAM as well.	XML format should match the PDT Editor Tool XML format
-Calibrate [sensors]	Activates the Full-Process calibration flow on selected sensors	Applicable arguments: • AGM, any combination of those three (AG, AM, M, etc.), calibrates the Accelerometer (A), Gyrometer(G) and/or Magnetometer (M). • ALS, activates calibration on the Ambient Light Sensor

- CalibrateOffline [sensors]	Activates the Full-Process offline calibration flow on selected sensors. This option is available only when using the –UseRawData argument. –UpdateISS applicable is not applicable when offline.	Same as above. The offline is applicable only when re-running a full calibration flow. Usage of offline is in the following flow: • Regular calibration is run on DUT1 with –ExportAllFiles command. The raw data is saved. • The tool has fixes, new algorithm etc. and a new version is released. • The calibration can be now run offline on any machine with the recorded data. • The output is a calibration XML that can be loaded to DUT1
-FourSteps	Performs the four-step calibration instead of the default Full-Process calibration.	This option is only available when using the –Calibrate AG, -Calibrate A, or –Calibrate G arguments. It is not available when using the –Minimal argument.
-Minimal	Instead of the default Full-Process calibration or four-step test, performs the one-step calibration or test.	This option is only available when using any of the -Calibrate AGM or –TestCalibration AG arguments. It is not available when using the –FourSteps argument.

-TestCalibration [sensors]	Runs the four-step (or, with –Minimal, the one-step) calibration verification test on the selected sensors.	Applicable arguments: • A: tests the calibration of the accelerometer • G: tests the calibration of the gyrometer • AG: tests the calibration of both • M: tests the calibration of the magnetometer (with the argument –MagAccuracyTest)
-NoEnter [ms]	Does not wait for the user to press Enter before beginning the test or calibration, but rather begins the test or the calibration after the indicated number of milliseconds.	This option is not recommended! Use it only if you sure that the device is stable in the correct position. It is applicable in calibration flow and test flow.
-MagAccuracyTest	Additional argument to –TestCalibration, for four steps test calibration for magnetometer	This option is only available for test calibration of magnetometer.
-ExportAllFiles [folderpath]	Create output files. The output files include: – A data dump file for each sensors. The output format is a .csv file. See example in section 13. -Current calibration, see format at section 11.1. -Result calibration, see format at section 11.1. - Log file, see format at section 12.1. All those files are created in the selected folder.	

-UseRawData [dump-filepath]	Read the sensors' data from a file instead of sampling sensors. The output calibration can be either compatible with current machine hardware settings – if –Calibrate option is activated, or with the machine the data was recorder on - if –CalibrateOffline option is activated. See section 13 for more details.	Valid only when calibration flow is activated. The sensor data input file should be generated by this Calibration Tool. When using this option - provide the name of the sensor data file without the suffix of sensor name. E.g. If you have the files dataDumpAccelerometer.csv and dataDumpMagnetomer.csv, provide dataDump.csv as input. **Note** – If you load data recorded by other tools then accelerometer data is expected to follow Android notation where Z up is +1G.
-UpdateISS	Write the calibration data to the Intel Integrated Sensor Solution flash. An ISS reset is performed, and the new data is loaded to the RAM as well.	This option is usually for **per-system calibration**.
-MagTest	Tests the level of magnetic noise in the environment. Should be run prior to running calibration.	
-PrimaryOnly	Calibrate only the first instance, if there are multiple sensors of the same type.	This argument can be used for both calibration flow and test flow.
-SecondaryOnly	Calibrate only the second instance, if there are multiple sensors of the same type.	This argument can be used for both calibration flow and test flow.
-CheckVsOriginalAxis	Cause a failure in the calibration if the axis mapping of the accelerometer differs from the original one.	This argument is applicable only for –FourSteps option.

8.1 Examples

Following are some examples of the Calibration Tool CLI usages.

Flow 1:

Perform full calibration flow and update the calibration data in flash:

```
WindowsCalibrationTool.exe -Calibrate AGM -UpdateIss –
ExportAllFiles result
```

```
The result is a calibrated DUT and output files in the result
folder. The data is stored on the flash, and ISS resets and reads
the data to the RAM.
```

Flow 2:

Update the calibration data in flash from file:

```
WindowsCalibrationTool.exe -ImportXml filename.xml
```

```
The result is a calibrated DUT with the data that was on the file.
The data is stored on the flash, and ISS resets and reads the data
to the RAM.
```

Flow 3:

Extract current calibration data on flash to file:

```
WindowsCalibrationTool.exe -ExportXml filename.xml
```

Flow 4:

Rerun previous calibration:

```
WindowsCalibrationTool.exe -Calibrate AGM -UpdateIss –
ExportAllFiles rerunFolder –UseRawData result\dataDump.csv
```

```
The result is a calibrated DUT based on the recorded data and
output files in the result folder. The data is stored on the flash,
and ISS resets and reads the data to the RAM.
```

9. Calibration Tool GUI

The Calibration Tool GUI can be run either locally on the SUT or on a remote Windows machine in order to connect remotely to an Android SUT.

When running locally on Windows, the tool must be run as an administrator.

NOTE: The GUI tool must be in the same directory of the Windows\Android CLI tool.

The tool includes three screens:

- Settings: equivalent to CLI parameters.
- Process: graphic calibration instructions.
- Summary: summary of all calibration actions.

9.1 GUI Calibration Flow

1. Set your SUT and calibration environment.

   A, G, M are respectively Accelerometer, Gyrometer and Magnetometer sensors.

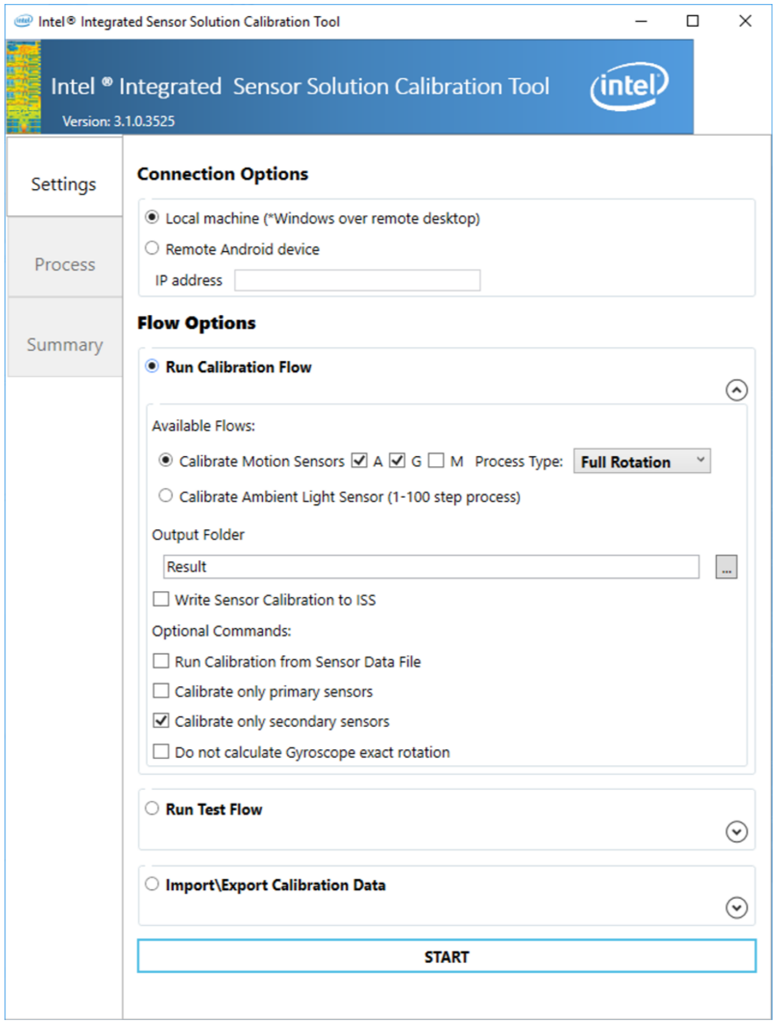


Figure 8 Calibration Settings Screen

2. Follow the calibration instructions if activate calibration flow is selected without data dump.

   If the calibration flow was not activated or a data dump was selected, you will be moved directly to the Summary screen.

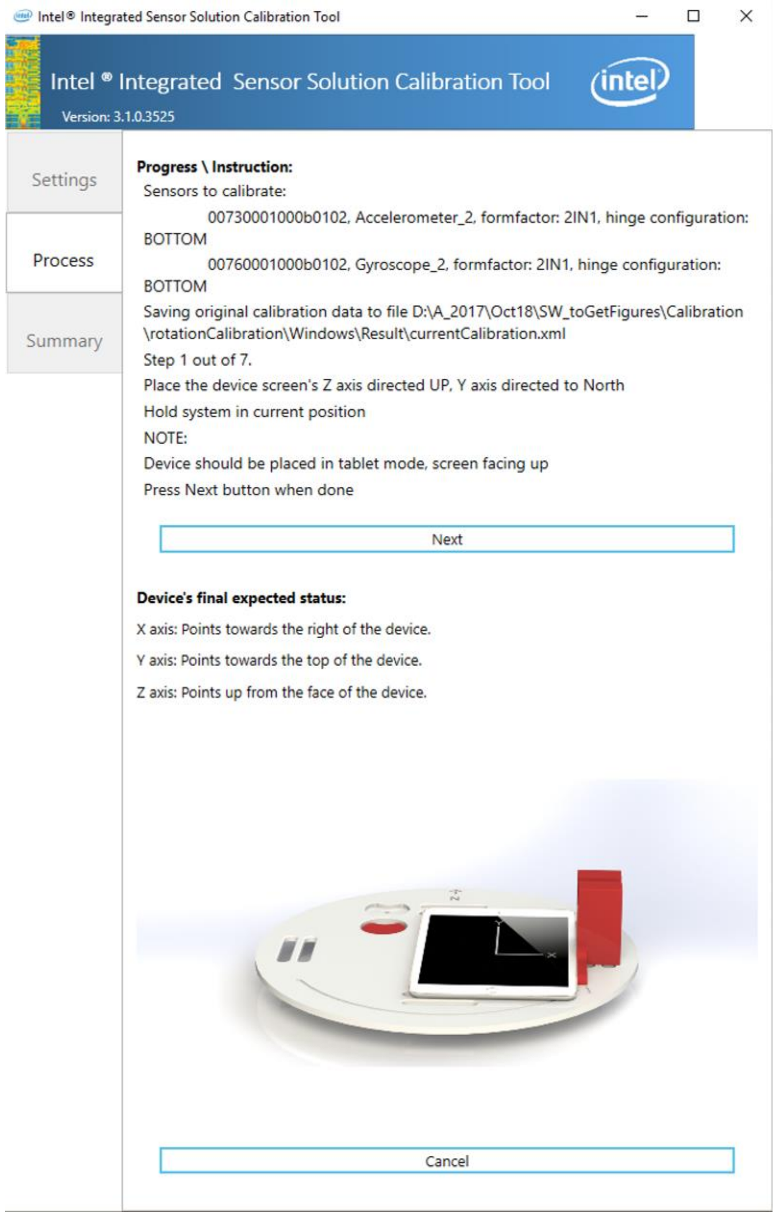


Figure 9 Graphic Calibration Instructions Screen

3. View calibration summary.

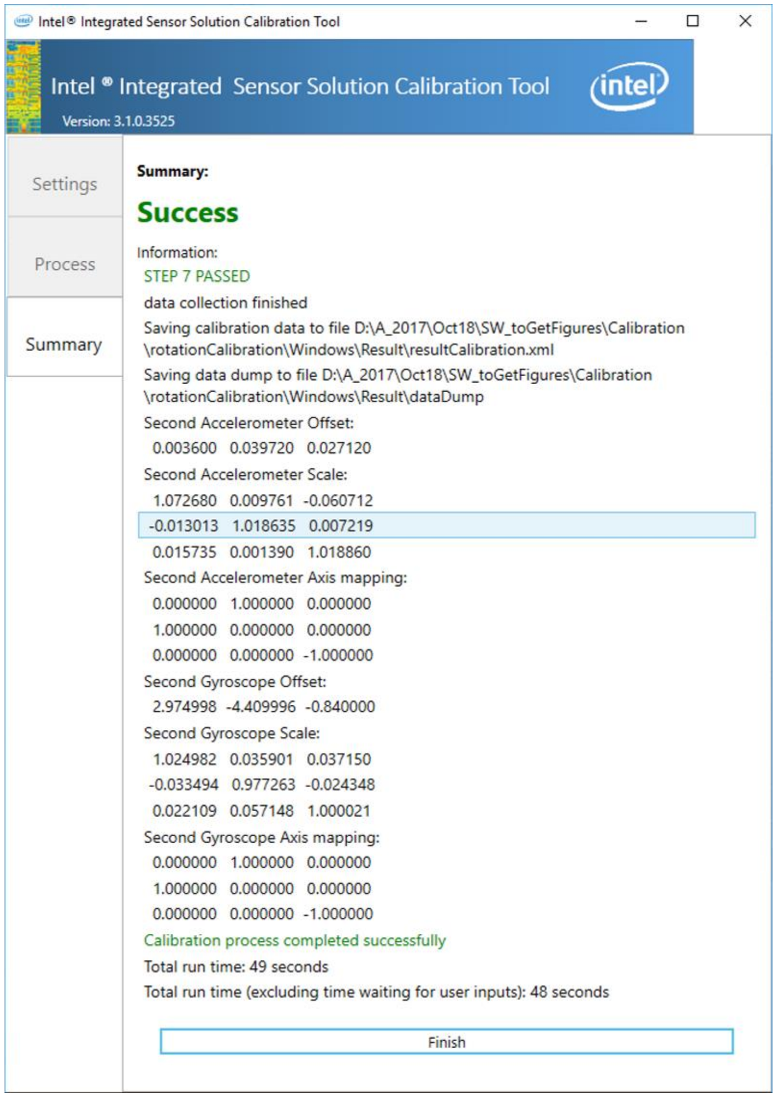


Figure 10 Calibration Summary Screen

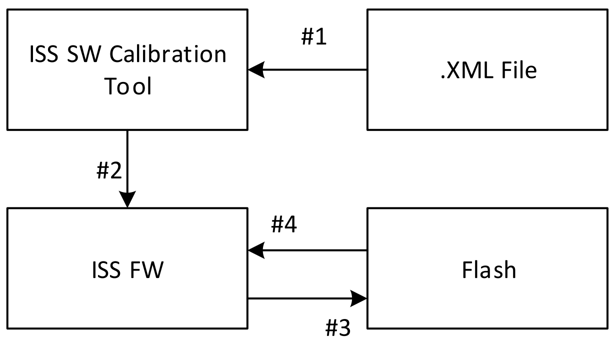
10. Typical Workflows

Both the CLI and GUI versions of the Calibration tool support the same functions. Following are typical workflows:

Usage 1

Import the calibration data file to flash.

CLI: `WindowsCalibrationTool.exe [-ImportXml <filename>]`



Calibration Import Flow:

#1 ISS SW Calibration Tool reads calibration from .XML File.

#2 ISS SW Calibration Tool writes calibration to ISS FW.

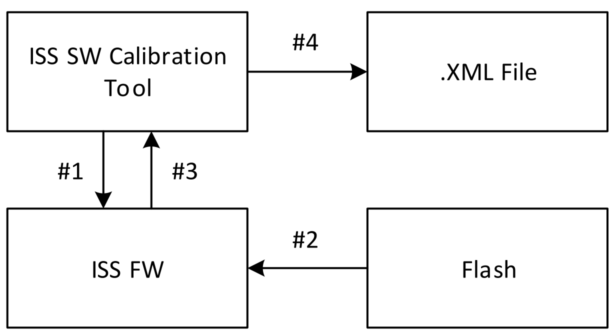
#3 After checking data structure – ISS FW writes calibration data to the flash.

#4 ISS FW is reset by ISS SW Calibration Tool – then the ISS FW reloads the new calibration data from flash.

Usage 2

Export the current calibration data from flash to XML file.

CLI: `WindowsCalibrationTool.exe [-ExportXml <filename>]`



Calibration Export Flow:

#1 ISS SW Calibration Tool requests calibration data from ISS FW

#2 ISS FW reads calibration data from flash

#3 The calibration data is transferred to the ISS SW Calibration Tool.

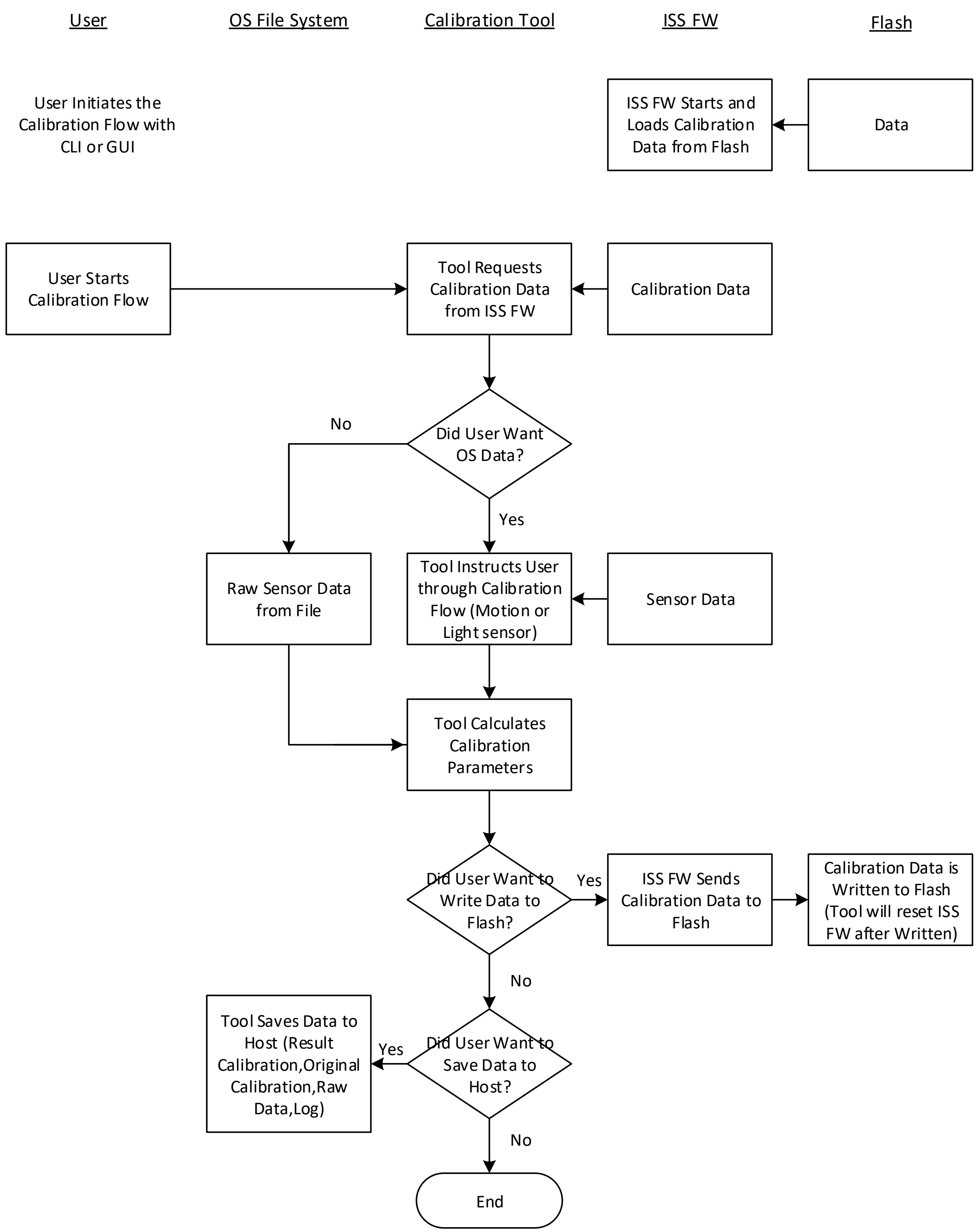
#4 The ISS SW Calibration Tool writes the data to XML file

Usage 3

Activate calibration flow on the current machine online.

Optional variation: Load a data dump instead of sampling data. If saving the results to a file, the file settings will match the hardware definitions of the current machine, and not of the one the data was sampled on.

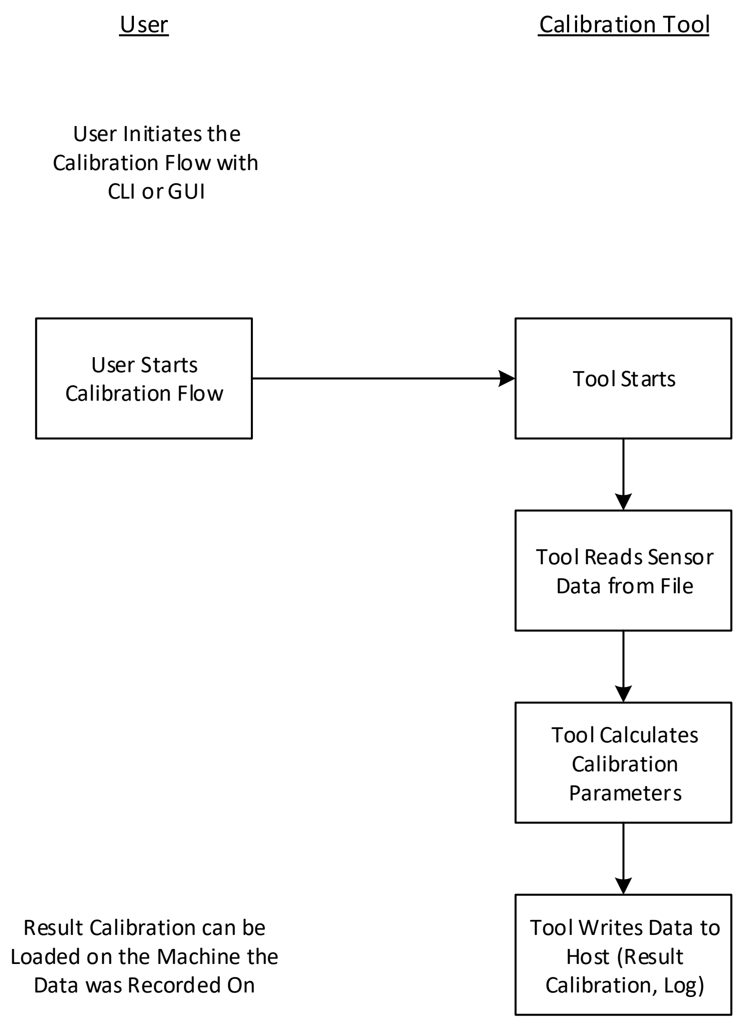
CLI:
```
WindowsCalibrationTool.exe -Calibrate <sensors> [-ExportAllFiles <folder>] [-UpdateISS] [-UseRawData <dump-file-path>
```


User
OS File System
Calibration Tool
ISS FW
Flash
User Initiates the Calibration Flow with CLI or GUI
ISS FW Starts and Loads Calibration Data from Flash
Data
User Starts Calibration Flow
Tool Requests Calibration Data from ISS FW
Calibration Data
Did User Want OS Data?
No
Yes
Raw Sensor Data from File
Tool Instructs User through Calibration Flow (Motion or Light sensor)
Sensor Data
Tool Calculates Calibration Parameters
Did User Want to Write Data to Flash?
Yes
ISS FW Sends Calibration Data to Flash
Calibration Data is Written to Flash (Tool will reset ISS FW after Written)
No
Tool Saves Data to Host (Result Calibration,Original Calibration,Raw Data,Log)
Yes
Did User Want to Save Data to Host?
No
End

Usage 4

Activate calibration flow offline, i.e. with no administrative privileges or any other machine dependencies, by loading a data dump and saving the results to a calibration file. The file settings will match the hardware definitions of the machine the one the data was sampled on.

CLI: WindowsCalibrationTool.exe -CalibrateOffline <sensors> -ExportAllFiles <folder> -UseRawData <dump-file-path>



Usage 5

Run magnetic test prior to calibration in order to make sure the environment is clean.

CLI: WindowsCalibrationTool.exe -MagTest -ExportAllFiles <folder>

11 Calibration File

The Calibration Tool's resultant calibration file can be exported and used for offline calibration, i.e. without the need for real firmware, or imported onto another firmware platform to form the starting point for its calibration.

11.1 Sample Calibration File


```
<?xml version="1.0" encoding="utf-8" ?>

<ISS>

    <Calibration>

        <Sensor SensorTypeId="115" SubTypeId="1"
VendorId="1" InstanceId="0" Flags="130">

            <CalibrationData Version="0" FormatId="0">

                <AxisMapping>

                    <XX>1</XX>

                    <XY>0</XY>

                    <XZ>0</XZ>

                    <YX>0</YX>

                    <YY>1</YY>

                    <YZ>0</YZ>

                    <ZX>0</ZX>

                    <ZY>0</ZY>

                    <ZZ>1</ZZ>

                </AxisMapping>

                <Matrix>

                    <Mat11>1</Mat11>

                    <Mat12>0</Mat12>
```

```
                    <Mat13>0</Mat13>
                    <Mat21>0</Mat21>
                    <Mat22>1</Mat22>
                    <Mat23>0</Mat23>
                    <Mat31>0</Mat31>
                    <Mat32>0</Mat32>
                    <Mat33>1</Mat33>
                    <ScaleFactor>0</ScaleFactor>
                </Matrix>
                <Offset>
                    <OffsetX>0</OffsetX>
                    <OffsetY>0</OffsetY>
<OffsetZ>0</OffsetZ>
                </Offset>
            </CalibrationData>
        </Sensor>
        <Sensor SensorTypeId="65" SubTypeId="1" VendorId="2"
InstanceId="0" Flags="2">
            <CalibrationData Version="0" FormatId="1">
                <Steps NumOfSteps="3">
                    <Step>
                        <Lux value="30" />
                        <Scale value="10" />
                    </Step>
```

```
                    <Step>
                        <Lux value="60" />
                        <Scale value="100" />
                    </Step>
                    <Step>
                        <Lux value="100" />
                        <Scale value="1000" />
                    </Step>
                </Steps>
                <Multiplier>1</Multiplier>
            </CalibrationData>
        </Sensor>

    </Calibration>
</ISS>
```

12. Log File

The Calibration Tool provides a log file for debugging the calibration: calibration.log. With each run of the Calibration Tool the log file is overwritten. If the log file is wanted for further future investigation it must be saved with a different name to prevent being overwritten.

12.1 Sample Log File

```
<?xml version="1.0" encoding="utf-8" ?>

<Log Date="2014-12-4. 8:1">

    <Sensors LUID="7300010001000082" Type="Accelerometer"
SamplingUnit="g0" Success="true" isPrimary="true">

        <!--Original scale matrix as extracted from the ISS->

        <OriginalScaling Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </OriginalScaling>

        <!--The inverse matrix of the scale matrix extracted
from the ISS-->

        <InvertedScaling Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </InvertedScaling>

        <!--Original axis mapping as extracted for the ISS->

        <OriginalAxisMapping Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </OriginalAxisMapping>
```

```
        <!--The inverse matrix of the axis mapping extracted
from the ISS-->

        <InvertedAxisMapping Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </InvertedAxisMapping>

        <!--The matrix used in order to reverse static
calibration done by ISS-->

        <ReverseMatrix Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </ReverseMatrix>

        <!--Original offset as extracted from the ISS-->

<OriginalOffset Rows="1" Cols="3">

            <Values>0.000000,0.000000,0.000000;</Values>

        </OriginalOffset>

        <!--The offset used in order to reverse static
calibration done by ISS-->

        <ReverseOffset Rows="1" Cols="3">

            <Values>-0.000000,-0.000000,-0.000000;</Values>

        </ReverseOffset>

        <!--The measured averages of the calibration process-
->

        <Averages Rows="7" Cols="3">
<Values>0.026592,0.010100,1.076946;0.020876,-
0.995437,0.062717;0.037470,1.020280,0.058306;0.029510,0.0131
14,1.076910;0.988670,-0.003404,0.047598;0.014099,0.006719,-
0.955084;-0.941465,0.022055,0.076387;</Values>
```

```
        </Averages>

        <!--The averages that are expected at a perfectly
calibrated system-->

        <ExpectedAverages Rows="7" Cols="3">

            <Values>0.000000,0.000000,1.000000;0.000000,-
1.000000,0.000000;0.000000,1.000000,0.000000;0.000000,0.0000
00,1.000000;1.000000,0.000000,0.000000;0.000000,0.000000,-
1.000000;-1.000000,0.000000,0.000000;</Values>

        </ExpectedAverages>

        <!--The calculated offset of the sensor-->

        <Offset Rows="1" Cols="3">

            <Values>0.024374,0.010052,0.061145;</Values>

        </Offset>

        <!--The calculated scale matrix of the sensor-->

        <Scale Rows="3" Cols="3">

            <Values>1.036806,-0.013447,-
0.002263;0.013810,0.989697,-
0.000077;0.013866,0.002587,0.984415;</Values>

        </Scale>

        <!--The calculated axis mapping of the sensor-->

        <AxisMapping Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </AxisMapping>

        <!--Measured averages after applying the calculated
calibration-->

        <CalibratedAverages Rows="7" Cols="3">
```

```
            <Values>0.000000,0.000000,1.000000;0.009891,-
0.995178,-
0.001102;0.000000,1.000000,0.000000;0.002984,0.003022,1.0000
13;1.000000,0.000000,-0.000000;-0.008308,-0.003362,-
1.000542;-1.001584,-0.001460,0.001643;</Values>

        </CalibratedAverages>

        <!--The calculated white noise of the sensor during
the calibration-->

        <WhiteNoise Rows="1" Cols="3">

            <Values>0.004806,0.003646,0.006074;</Values>

        </WhiteNoise>

    </Sensors>

    <Sensors LUID="7600010004000042" Type="Gyrometer"
SamplingUnit="deg__sec" Success="true" isPrimary="true">>

        <!--Original scale matrix as extracted from the ISS-
->

        <OriginalScaling Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </OriginalScaling>

        <!--The inverse matrix of the scale matrix extracted
from the ISS-->

        <InvertedScaling Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </InvertedScaling>

        <!--Original axis mapping as extracted for the ISS--
>
```

```
        <OriginalAxisMapping Rows="3" Cols="3">

<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.000000;0.000000,0.000000,1.000000;</Values>

        </OriginalAxisMapping>

        <!--The inverse matrix of the axis mapping extracted from the ISS-->

        <InvertedAxisMapping Rows="3" Cols="3">

<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.000000;0.000000,0.000000,1.000000;</Values>

        </InvertedAxisMapping>

        <!--The matrix used in order to reverse static calibration done by ISS-->

        <ReverseMatrix Rows="3" Cols="3">

<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.000000;0.000000,0.000000,1.000000;</Values>

        </ReverseMatrix>

        <!--Original offset as extracted from the ISS-->

        <OriginalOffset Rows="1" Cols="3">

            <Values>0.000000,0.000000,0.000000;</Values>

        </OriginalOffset>

        <!--The offset used in order to reverse static calibration done by ISS-->

        <ReverseOffset Rows="1" Cols="3">

            <Values>-0.000000,-0.000000,-0.000000;</Values>

        </ReverseOffset>
```

```
        <!--The measured averages of the calibration process-
->

        <Averages Rows="6" Cols="3">

            <Values>-91.407408,-3.261829,4.169206;-
8.533615,-14.145728,-177.028891;-92.353372,-5.649280,-
7.855205;-1.412809,-86.770624,2.093379;5.404201,-
87.216211,4.141751;3.291423,-86.626783,-4.437661;</Values>

        </Averages>

        <!--The averages that are expected at a perfectly
calibrated system-->

        <ExpectedAverages Rows="6" Cols="3">

            <Values>-
90.000000,0.000000,0.000000;0.000000,0.000000,-180.000000;-
90.000000,0.000000,0.000000;0.000000,-
90.000000,0.000000;0.000000,-90.000000,0.000000;0.000000,-
90.000000,0.000000;</Values>

        </ExpectedAverages>

        <!--The calculated offset of the sensor-->

        <Offset Rows="1" Cols="3">

            <Values>-1.116430,-4.128769,-0.917550;</Values>

        </Offset>

        <!--The calculated scale matrix of the sensor-->

        <Scale Rows="3" Cols="3">

            <Values>0.979214,0.027024,-0.049362;-
0.048536,1.034106,-0.080292;-
0.020717,0.006437,1.017267;</Values>

        </Scale>

        <!--The calculated axis mapping of the sensor-->

        <AxisMapping Rows="3" Cols="3">
```

```
<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </AxisMapping>

        <!--Measured averages after applying the calculated
calibration-->

        <CalibratedAverages Rows="6" Cols="3">

            <Values>-
88.641859,4.870477,7.050749;1.159506,4.141727,-179.063158;-
89.039129,3.412977,-5.177065;-2.672121,-
85.687835,2.537077;3.890038,-86.643960,4.476721;2.260601,-
85.243024,-4.203271;</Values>

        </CalibratedAverages>

        <!--The calculated white noise of the sensor during
the calibration-->

        <WhiteNoise Rows="1" Cols="3">

            <Values>0.546209,1.091133,0.990205;</Values>

        </WhiteNoise>

    </Sensors>

    <Sensors LUID="7d00010001000042" Type="Magnetometer"
SamplingUnit="milliGauss" Success="true" isPrimary="true">>

        <!--Original scale matrix as extracted from the ISS-
->

        <OriginalScaling Rows="3" Cols="3">
<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </OriginalScaling>

        <!--The inverse matrix of the scale matrix extracted
from the ISS-->

        <InvertedScaling Rows="3" Cols="3">
```

```
<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </InvertedScaling>

        <!--Original axis mapping as extracted for the ISS--
>

        <OriginalAxisMapping Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </OriginalAxisMapping>

        <!--The inverse matrix of the axis mapping extracted
from the ISS-->

        <InvertedAxisMapping Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </InvertedAxisMapping>

        <!--The matrix used in order to reverse static
calibration done by ISS-->

        <ReverseMatrix Rows="3" Cols="3">
<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </ReverseMatrix>

        <!--Original offset as extracted from the ISS-->

<OriginalOffset Rows="1" Cols="3">

            <Values>0.000000,0.000000,0.000000;</Values>

        </OriginalOffset>

        <!--The offset used in order to reverse static
calibration done by ISS-->
```

```
        <ReverseOffset Rows="1" Cols="3">

            <Values>-0.000000,-0.000000,-0.000000;</Values>

        </ReverseOffset>

        <!--The measured averages of the calibration process-
->

        <Averages Rows="7" Cols="3">

            <Values>-19.644194,324.930000,-
104.981290;5.695593,256.068136,444.337967;8.094102,-
220.692068,455.892881;4.291188,-290.362299,-90.208123;-
259.369084,-287.136419,151.864454;-30.905180,-
288.528447,390.431872;227.973870,-
291.802613,164.641206;</Values>

        </Averages>

        <!--The averages that are expected at a perfectly
calibrated system-->

        <ExpectedAverages Rows="7" Cols="3">
<Values>0.000000,309.847074,-
240.687361;0.000000,240.687361,309.847074;-0.000000,-
240.687361,309.847074;-0.000000,-309.847074,-240.687361;-
240.687361,-309.847074,0.000000;0.000000,-
309.847074,240.687361;240.687361,-309.847074,-
0.000000;</Values>

        </ExpectedAverages>

        <!--The calculated offset of the sensor-->

        <Offset Rows="1" Cols="3">

            <Values>-3.238624,18.200777,142.725291;</Values>

        </Offset>

        <!--The calculated scale matrix of the sensor-->

        <Scale Rows="3" Cols="3">

            <Values>1.002390,0.001908,0.001267;-
0.001883,1.002725,-0.018739;-
```

```
0.001294,0.018632,1.008351;</Values>

        </Scale>

        <!--The calculated axis mapping of the sensor-->

        <AxisMapping Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </AxisMapping>

        <!--Measured averages after applying the calculated
calibration-->

        <CalibratedAverages Rows="7" Cols="3"> <Values>-
16.173353,312.237628,-244.039009;-
1.627069,232.868218,308.566592;11.300705,-
245.433528,311.317179;6.664152,-305.053090,-240.637546;-
257.313439,-305.858126,3.857890;-28.004089,-
312.154640,244.096039;231.201384,-
311.694124,16.023772;</Values>

        </CalibratedAverages>

        <!--The calculated white noise of the sensor during
the calibration-->

        <WhiteNoise Rows="1" Cols="3">

            <Values>1.438047,1.077251,0.974736;</Values>

        </WhiteNoise>

    </Sensors>

    <Sensors LUID="7300010002000102" Type="Accelerometer"
SamplingUnit="g0" Success="true" isPrimary="false">>

        <!--Original scale matrix as extracted from the ISS-
->

        <OriginalScaling Rows="3" Cols="3">
```

```
<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </OriginalScaling>

        <!--The inverse matrix of the scale matrix extracted
from the ISS-->

        <InvertedScaling Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </InvertedScaling>

        <!--Original axis mapping as extracted for the ISS--
>

        <OriginalAxisMapping Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </OriginalAxisMapping>

        <!--The inverse matrix of the axis mapping extracted
from the ISS-->

        <InvertedAxisMapping Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>

        </InvertedAxisMapping>

        <!--The matrix used in order to reverse static
calibration done by ISS-->

        <ReverseMatrix Rows="3" Cols="3">


<Values>1.000000,0.000000,0.000000;0.000000,1.000000,0.00000
0;0.000000,0.000000,1.000000;</Values>
```

```
        </ReverseMatrix>

        <!--Original offset as extracted from the ISS-->

<OriginalOffset Rows="1" Cols="3">

            <Values>0.000000,0.000000,0.000000;</Values>

        </OriginalOffset>

        <!--The offset used in order to reverse static
calibration done by ISS-->

        <ReverseOffset Rows="1" Cols="3">
<Values>-0.000000,-0.000000,-0.000000;</Values>

        </ReverseOffset>

        <!--The measured averages of the calibration process-
->

        <Averages Rows="7" Cols="3">

            <Values>-0.008303,-0.011826,1.013119;-0.033306,-
0.994319,0.026066;0.010153,0.966142,0.015970;-0.005499,-
0.007890,1.015375;0.973059,-0.012398,0.004769;-0.026495,-
0.025505,-0.969181;-1.005333,-0.022848,0.033721;</Values>

</Averages>

        <!--The averages that are expected at a perfectly
calibrated system-->

        <ExpectedAverages Rows="7" Cols="3">

            <Values>0.000000,0.000000,-
1.000000;0.000000,1.000000,0.000000;0.000000,-
1.000000,0.000000;0.000000,0.000000,-
1.000000;1.000000,0.000000,0.000000;0.000000,0.000000,1.0000
00;-1.000000,0.000000,0.000000;</Values>

        </ExpectedAverages>

        <!--The calculated offset of the sensor-->

        <Offset Rows="1" Cols="3">
```

```
            <Values>-0.015038,-0.016792,0.020744;</Values>

        </Offset>

        <!--The calculated scale matrix of the sensor-->

        <Scale Rows="3" Cols="3">

<Values>1.011945,0.018975,0.011545;0.004694,1.023029,-
0.008949;-0.016306,0.005194,1.009943;</Values>

        </Scale>

        <!--The calculated axis mapping of the sensor-->

<AxisMapping Rows="3" Cols="3">

            <Values>1.000000,0.000000,0.000000;0.000000,-
1.000000,0.000000;0.000000,0.000000,-1.000000;</Values>

        </AxisMapping>

        <!--Measured averages after applying the calculated
calibration-->

        <CalibratedAverages Rows="7" Cols="3">

            <Values>-0.004737,0.003832,-1.002378;-
0.000000,1.000000,-0.000000;0.006896,-1.005495,-0.000694;-
0.001999,-0.000161,-
1.004722;1.000000,0.000000,0.000000;0.000000,0.000000,1.0000
00;-1.002159,0.001663,0.003073;</Values>

        </CalibratedAverages>

        <!--The calculated white noise of the sensor during
the calibration-->

        <WhiteNoise Rows="1" Cols="3">

            <Values>0.015073,0.013236,0.016347;</Values>

        </WhiteNoise>

    </Sensors>

</Log>
```

13 Raw Data File

When running the calibration, one of the outputs is a file to each sensor that contains the data the Calibration Tool received from the sensors. This data can be used for analysis and for future testing of the tool. When the tool has a new version – the calibration can be recalculated with the data.

The data is a .csv file with the following format:

First line specifies the unique ID of the sensor.

For second Accelerometer, there is an optional line to specify the hinge connection side.

The next line specifies the header. First column specifies the timestamp in milliseconds "time[ms]". The next columns specify the sensor, axis and unit.

The units used by the Calibration Tool are the Operating System sensor API default units.

A sample file:

```
LUID=7300010000000002

[HINGE = BOTTOM]

time[ms],Accelerometer_X[g0],Accelerometer_Y[g0],Acceleromet
er_Z[g0]

11,-0.025996,-0.016000,-1.064006

20,-0.041998,-0.020001,-1.060007

29,-0.037998,-0.020001,-1.056008

39,-0.039997,-0.020001,-1.062008

49,-0.029998,-0.030002,-1.070008

59,-0.031997,-0.020001,-1.064009

69,-0.043998,-0.018001,-1.070008

79,-0.035998,-0.016002,-1.070007

89,-0.025996,-0.010000,-1.064006
```

```
99,-0.027998,-0.016001,-1.066007

109,-0.021997,-0.012000,-1.056006
```