1. Overview

The *Kionix IoT Evaluation Kit* provides a powerful and easy to use environment to begin evaluation of Rohm and Kionix products. Multiple hardware options are supported (*Kionix Evaluation Kit HW*), as well as common, hardware independent SW tools (*Kionix Evaluation Kit SW*). *Kionix Windows Sensor Evaluation Software* consists of an easy to use graphical user interface for displaying / recording sensor data and a graphical register map editor which allows the user to see and change the content of control registers. *Kionix Multi-OS Evaluation Software,* in turn, provides reference implementation for driver SW and also a convenient way to access sensor low level features and versatile data logging functionality.

For a quick set up with Windows OS, refer to chapter 5.3.

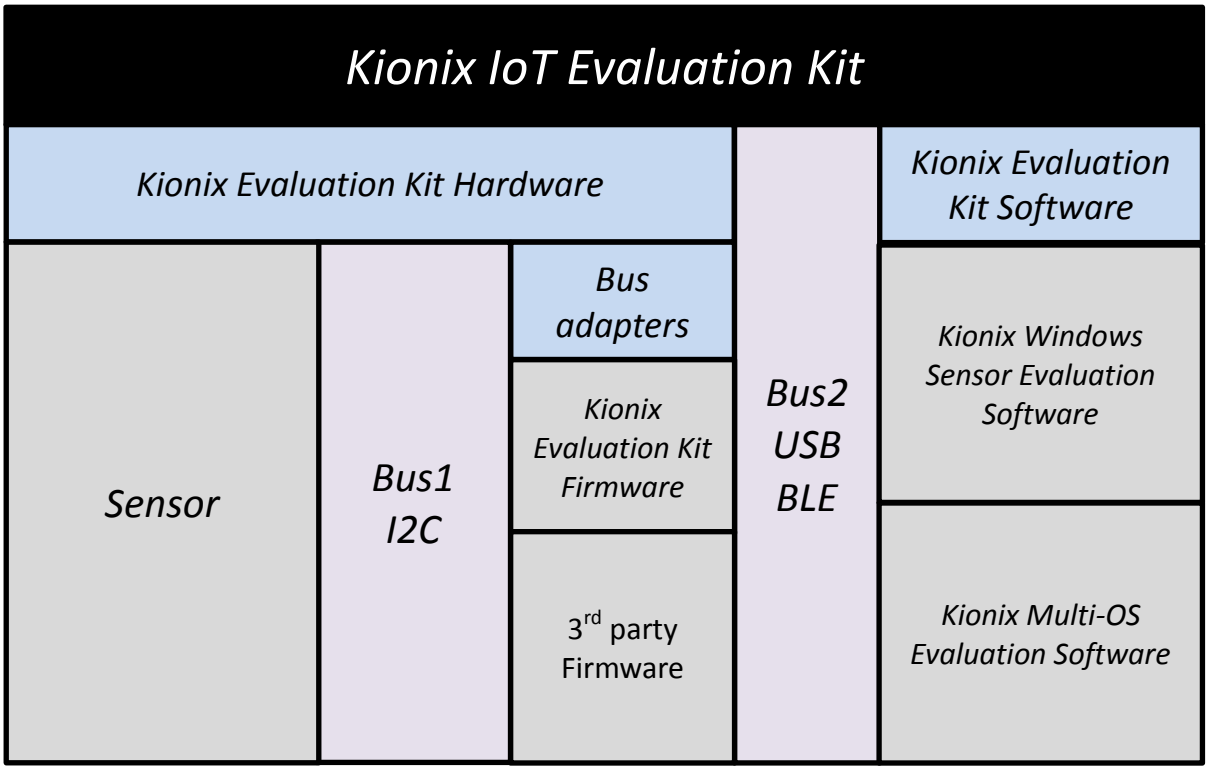


Figure 1: Structure of Kionix IoT Evaluation Kit

36 Thornwood Dr. — Ithaca, NY 14850 USA
Tel: 607-257-1080 — Fax: 607-257-1146
www.kionix.com — info@kionix.com

1.1. Definitions

Kionix IoT Evaluation Kit	*Complete offering of HW and SW for sensor evaluation purposes*
Kionix Evaluation Kit HW	PWB containing sensor component and/or bus connection between sensor and bus adapter.
Kionix IoT Sensor Node	MCU + BLE + sensors + battery running *Kionix Evaluation Kit Firmware*
Kionix IoT Sensor Node Mounting Accessory	Wrist band for *Kionix IoT Sensor Node*
Kionix Multi-Platform Adapter Board	Board specifically designed to easily interface with Kionix sensors and numerous development platforms. Example: A1 Adapter Board
Kionix Evaluation Board	Kionix sensor evaluation board with 14 pin header http://www.kionix.com/developer-tools
Kionix Evaluation Kit SW	*Kionix Windows Sensor Evaluation Software* or *Kionix Multi-OS Evaluation Software*
Kionix Windows Sensor Evaluation Software	Kionix Sensor evaluation software with graphical user interface (GUI) running in Windows OS
Kionix Multi-OS Evaluation Software	Kionix sensor evaluation software with command line interface for quickly testing sensor low level features
Kionix Evaluation Kit Firmware	Proprietary firmware running on microcontroller based bus adapters
Kionix BLE Router	Android application which passes messages between *Kionix IoT Sensor Node* over BLE connection and PC over USB connection.

Table of Contents

2. Hardware

Kionix IoT Evaluation Kit is designed to support multiple Bus adapters for accessing sensors from *Kionix Evaluation Kit SW*.

2.1. Hardware architecture

Usually evaluation kit software runs on a client machine where there is no direct connectivity to the sensor. Because of this, a separate bus adapter which transfers messages from the client machine to the sensor is needed. For the connectivity, two separate bus connections are needed.

- Bus 1 between the sensor and the Bus adapter. Currently I2C and SPI connectivity options are supported with dedicated adapter boards
- Bus 2 between the Bus adapter and the client machine. Currently standard USB and BLE connections are supported

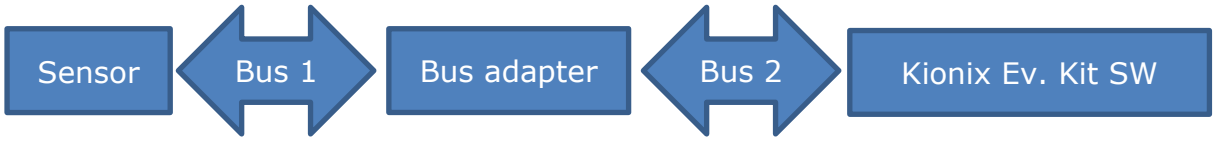


Figure 2: Hardware architecture of *Kionix IoT Evaluation Kit*

2.1.1. Supported hardware configurations

Both "off the shelf" adapters (For example Aardvark) and microcontroller based adapters (For example *Kionix IoT Sensor Node*) are supported. Microcontroller based adapters use proprietary *Kionix Evaluation Kit Firmware* for communication.

Adapter boards are offered for "Bus 1" connection between the bus adapter and sensor (2.3, 2.4 and 2.5).

For Bus 2 connection both wired USB and wireless BLE connections are supported.

Currently supported hardware configurations are listed in the following table:

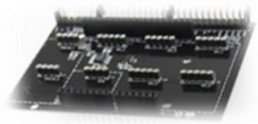
<table>
<tr><th>Sensor</th><th>Bus 1</th><th>Bus Adapter</th><th>Bus 2</th><th>Client</th></tr>
<tr><td colspan="2"> 
Kionix IoT node</td><td rowspan="2"></td><td rowspan="2"> 
USB cable
or
 
Bluetooth</td><td rowspan="5">Kionix Windows Sensor Evaluation Software
or
Kionix Multi-OS Evaluation Software</td></tr>
<tr><td> 
IoT Node Add on board</td><td colspan="2"> 
Kionix IoT Node</td></tr>
<tr><td colspan="2"> 
ROHM multi sensor shield</td><td rowspan="3"> 
nRF51-DK</td><td rowspan="3"> 
USB cable
or
 
Bluetooth</td></tr>
<tr><td> 
Kionix Ev.board</td><td> 
Kionix Multi-Platform adapter board</td></tr>
<tr><td> 
Rohm sensor module</td><td> 
Rohm Sensor Evaluation Kit</td></tr>
<tr><td> 
Kionix Ev. Board</td><td> 
Kionix Multi-Platform adapter board</td><td colspan="2"> 
Raspberry Pi3</td><td rowspan="2">Kionix Multi-OS Evaluation Software</td></tr>
<tr><td>Kionix Ev. Board, Rohm sensor module or IoT node add-on board.</td><td> 
Level shifter Board</td><td> 
Aardvark I2C/SPI host adapter</td><td> 
USB cable</td></tr>
</table>

Table 1: Supported HW configurations

2.2. Kionix IoT Sensor Node

Figure 3: *Kionix IoT Sensor Node* board

The *Kionix IoT Sensor Node* is a small sensor node based on the Nordic Semiconductor nRF51822 Series system-on-chip (SoC), and it supports both Bluetooth Low Energy (BLE 4.1) and USB interfaces. It consists of four (4) sensors: Gyroscope, Accelerometer, Magnetometer and Barometer.

The *Kionix IoT Sensor Node* has *Kionix Evaluation Kit Firmware* preinstalled. Firmware update instructions are described in chapter 4.3.1.

The hardware configuration variants of the *Kionix IoT Sensor Node* are as follows:

serial number	#0-99 White sticker	#100-899 White sticker	#900- Green sticker
Board name in *Kionix Windows Sensor Evaluation Software*	kionix_iot_node_with_kx122	kionix_iot_node_with_kx122	kionix_iot_node_with_kx126
Gyroscope	KXG03	KXG03	KXG03
Accelerometer	KX122	KX122	KX126
Magnetometer	KMX62	KMX62	KMX62
Barometer	BM1383GLV	BM1383AGLV	BM1383AGLV

Table 2: Sensors in *Kionix IoT Sensor Node*

Serial number can be found from the label located on the sticker of the *Kionix IoT Sensor Node*.

Figure 4: *Kionix IoT Sensor Node* serial number

Detailed information about the Kionix IoT board can be found in the directory `Kionix-IoT-Evaluation-Kit\Kionix-Evaluation-Kit-HW\Kionix-IoT-Sensor-Node\`

Notes:

- Before using the *Kionix IoT Sensor Node* with a battery, please make sure the battery is fully charged. The battery is charged via the USB connector. The battery is fully charged when the battery charging LED turns off.
- If a battery is connected, the *Kionix IoT Sensor Node* does not automatically turn on the power when the USB is connected to the PC. The power switch must be pressed to power on the device before connecting the USB cable.

2.2.1. Kionix IoT Sensor Node Layout

The main parts of layout and component placement are as follows:

A. Power switch
B. USB connector for communication and battery charging
C. Red battery charging indication LED. When battery is charging this LED is lit.
D. Red + Green status LEDs. Red status LED is lit when device is powered on. Green status LED indicates BLE connection.
E. Blue or Yellow status LED.
F. Test pad area
G. Battery connection pads (recommended to use PRT-13112 battery)

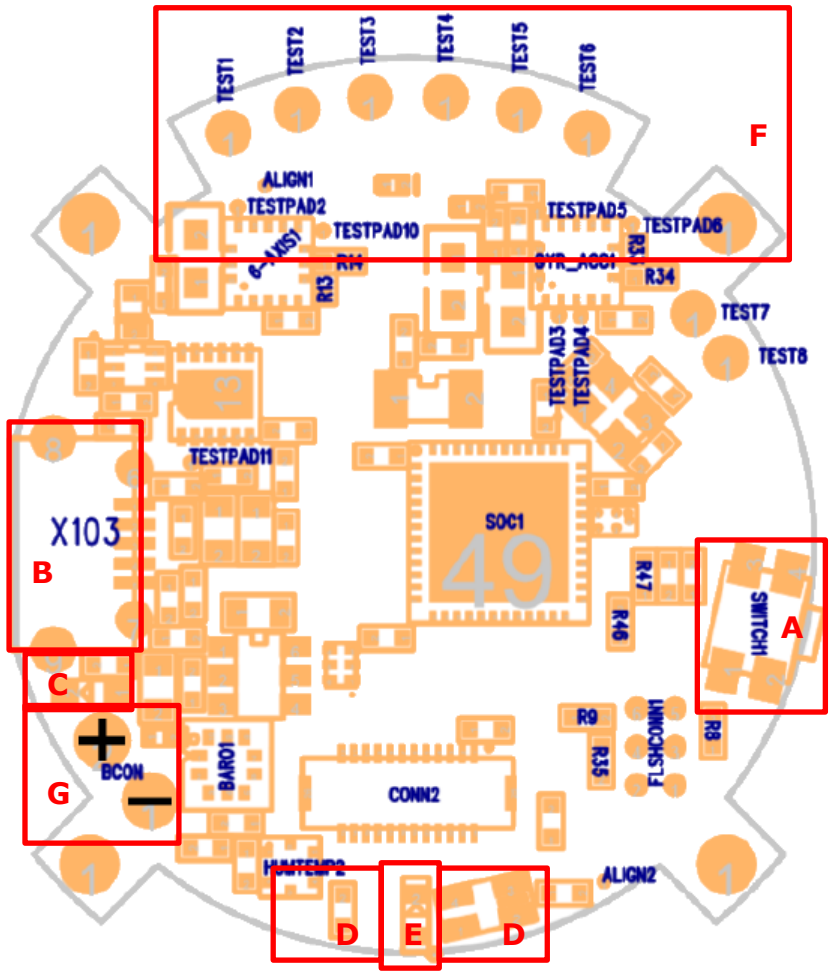


Figure 5: *Kionix IoT Sensor Node* component placement

2.2.2. Adding new I2C slave devices

More sensors can be connected to the *Kionix IoT Sensor Node* in two ways:

- Kionix IoT Node add-on board
- Test points

TEST1	GENIO1/INT1 KMX62-1031 GPIO1 INPUT/OUTPUT
TEST2	I2C_SDA I2C_SDA LINE (SENSORS)
TEST3	I2C_SCL I2C_SCL LINE (SENSORS)
TEST4	SENSOR INTERRUPT KXG03/INT2, KX122/INT2, BM1383/INT1
TEST5	SENSOR INTERRUPT KXG03/INT1, KX122/INT1
TEST6	VSEN SENSOR VOLTAGE SUPPLY (VCC SWITCHED ON BY SEN)
TEST7	VCC REGULATED VOLTAGE SUPPLY (2.8V)
TEST8	GND COMMON GROUND

Table 3: *Kionix IoT Sensor Node* test points

2.3. *Kionix Multi-Platform Adapter Board*

The *Kionix Multi-Platform Adapter Board* is an inexpensive board specifically designed to easily interface with Kionix sensors and numerous development platforms.

It is a great platform for someone who needs an easy way to interface with *Kionix Evaluation Boards* featuring a 14-pin male header or additional sensors using ready to use development platforms such as Cypress, Arduino, and Raspberry Pi.

The board utilizes 5V power from the host platform and converts to jumper-selectable (1.8V/2.5V/3.3V) VDD voltage using an on-board ROHM adjustable LDO. IO_VDD is selected with a jumper to be either 3.3V (provide from another dedicated on-board ROHM adjustable LDO) or tied directly to VDD. Current measurement is possible via removable jumpers on IO_VDD, and VDD rails. A LED is provided on the power rail to indicate the board is powered.

2.3.1. Supported development platforms

- Cypress Low-Cost Prototype Kits: CY8CKIT-059 (PSoC 5LP / Cortex M3), CY8CKIT-043 (PSoC 4M / Cortex M0), CY8CKIT-049 (PSoC 4 / Cortex M0). These are low cost, yet very powerful Mixed Digital & Analog platforms with free development tools. The PSoC 5LP Cortex M3 platform also comes with built in USB 2.0.
- Arduino Uno: The adapter board acts like a shield that can be plugged on top of compatible Arduino PCBs.
- Raspberry Pi: Pi 1 ModelA+, Pi 1 Model B+, Pi 2 Model B, Pi 3 Model B. The Raspberry Pi is a low cost, small form factor computer with endless capabilities.

At the moment *Kionix Evaluation Kit SW* supports Raspberry Pi (using flat cable, Figure 7) and nRF51-DK (using Arduino Uno R3 compatible connector).

2.4. Rohm Multi-Sensor Shield

ROHM Semiconductor's Multi-Sensor Shield, the SENSORSHLD1-EVK-101, is a shield evaluation platform that integrates multiple ROHM sensor products on a single board. The shield uses standard Arduino UNO R3 shield interface pins, making it possible to connect to any evaluation kit with a shield interface header.

The following sensors are included:

- Analog Temperature Sensor (BDE0600G)
- Digital Barometric Pressure Sensor (BM1383AGLV)
- Hall Switch Sensor (Omni polar with Polarity Discrimination) (BU52014HFV)
- Geomagnetic Sensor (BM1422GMV)
- Digital Color Sensors (BH1745)
- Optical Proximity Sensors and Ambient Light Sensors (RPR-0521)

- Analog UV Sensor (ML8511)
- Digital Accelerometer (KX122-1037/KX122-1048)
- Digital Magnetometer and Accelerometer (KMX62)
- Digital Gyroscope and Accelerometer (KXG03)

NOTE: Before using this shield with Arduino, please refer to Usage Manual document for needed HW rework.

2.4.1. Usage with Kionix Evaluation Kit SW

Rohm Multi-Sensor Shield can be used with *Kionix IoT Evaluation Kit* by using Nordic Semiconductor's nRF51-DK development board and *Kionix Evaluation Kit Firmware* ref. chapter 3.1.1

2.4.2. Standalone example applications

Standalone example applications and documentation are also available for the following platforms:

- Arduino UNO, Firmware Example and Documentation
- LPCXpresso54102, Firmware Example Only
- Nordic Semiconductor nRF51-DK, Firmware Example and Documentation.

2.5. Rohm Sensor Evaluation Kit

ROHM's SensorShield-EVK-001 sensor evaluation kit is compatible with open platforms such as Arduino Uno and mbed. The sensor shield enables easy evaluation of the 8 onboard high-performance sensors, making it ideal for verifying sensor operation, initial set development, and as a learning/training tool.

2.5.1. Usage with Kionix Evaluation Kit SW

Rohm Multi-Sensor Shield can be used with Kionix IoT Evaluation Kit by using Nordic Semiconductor's nRF51-DK development board and Kionix Evaluation Kit Firmware ref. chapter 3.1.1.

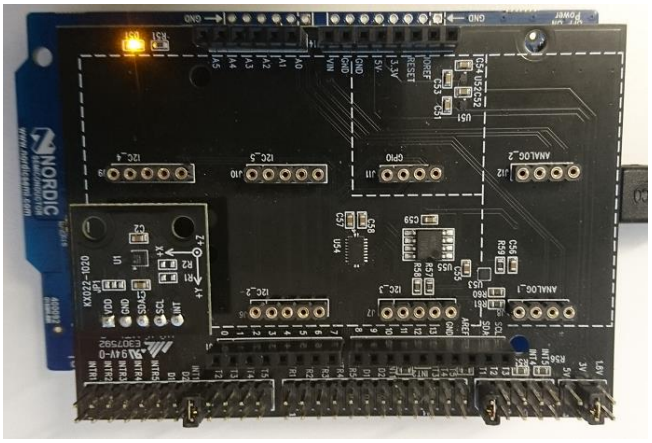


Figure 6: Rohm EVK-001 with proper jumper configuration

2.6. Aardvark I2C/SPI Host Adapter

Aardvark is a product of Total Phase Inc. *Kionix Multi-OS Evaluation Software* running on a Windows PC supports both I2C and SPI connections on Aardvark.

Since Aardvark uses 5 volts for logic voltage, level shifting may be required. Exemplary documentation on how to connect the KXG03 *Kionix Evaluation Board* to the Aardvark I2C bus can be found in Appendix 8.1.

2.7. Embedded Linux

Kionix Multi-OS Evaluation Software can also be run on an embedded Linux system.

The recommended configuration setup is to use Raspberry Pi3 with the *Kionix Multi-Platform Adapter Board* and *Kionix Evaluation Board*.

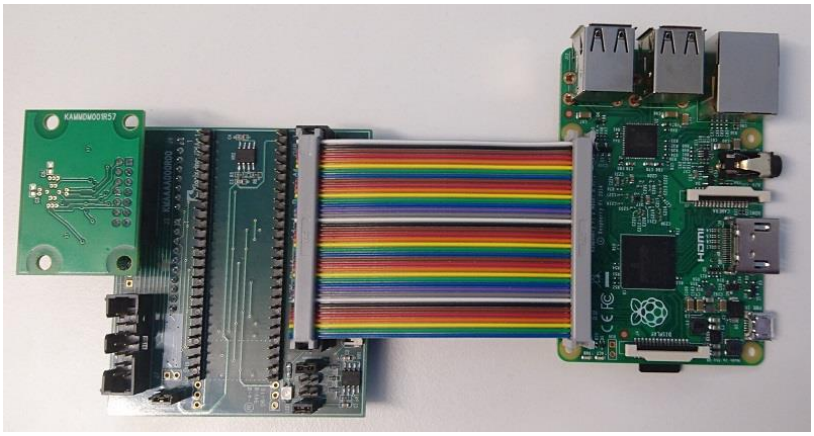


Figure 7: *Kionix Evaluation Board* with Raspberry Pi3

Other Linux boards may require jump wires for sensor connection. An example connection of the Beagle Bone Black and *Kionix Evaluation Board* is illustrated in Appendix 8.2. Please note the possible need of a level shifter. The Beagle Bone Black is a 3.3V system so the evaluation board can be connected to it directly.

3. Software

Kionix IoT Evaluation Kit consists of both *Kionix Evaluation Kit Firmware* for supported evaluation kit hardware, as well as, *Kionix Evaluation Kit SW* for evaluating the array of sensors Kionix/ROHM has to offer.

3.1. Overview

3.1.1. *Kionix Evaluation Kit Firmware*

Firmware for supported microcontroller based *Kionix IoT Evaluation Kits* is provided by Kionix. Programming instructions and updating instructions are described in chapter 4.3. *Kionix Evaluation Kit Firmware* supports communication with *Kionix Evaluation Kit SW*. Kionix currently offers firmware for:

- Kionix IoT Sensor Node
- Nordic Semiconductor nRF51-DK

3.1.2. *Kionix Evaluation Kit SW*

The provided *Kionix Evaluation Kit SW* is divided into two parts: *Kionix Windows Sensor Evaluation Software* and *Kionix Multi-OS Evaluation Software*.

3.1.2.1. Windows Sensor Evaluation Kit Software

Windows Sensor Evaluation Kit Software provides an intuitive graphical user interface demonstrating high level sensor offerings and features. Some of the features included are:

- Demonstration of Kionix Software offering like the Sensor Fusion Algorithm
- Visual display of real time sensor data
- Ability to record sensor data to a file
- Sensor registry editor
- Air mouse

Please see chapter 0 for usage instructions for *Kionix Windows Sensor Evaluation Software.*

3.1.2.2. *Kionix Multi-OS Evaluation Software*

Kionix Multi-OS Evaluation Software provides an advanced interface for demonstrating low level sensor features. Some of the features included are:

- Independent of Operating System
- Reference implementation for creating sensor driver software
- Flexible tool for recording sensor data
- Reference implementation for usage of sensor ASIC level features
- Framework for quick modification and testing of sensor functionality

Usage instructions for *Kionix Multi-OS Evaluation Software* are in chapter 6.

3.1.3. Kionix IoT Logger

Kionix IoT Logger application's purpose is to collect sensor data over Bluetooth low energy connection. Currently this application is available for Android devices in Play store or GitHub. The application is also bundled with the *Kionix IoT Evaluation Kit* and located in the folder `Documents\Kionix\Kionix-IoT-Logger\Android`.

3.1.3.1. QR code

3.1.4. *Kionix BLE Router* Android Software

Kionix BLE Router application is provided for the Android Operating System. The application's purpose is to provide a Bluetooth Low Energy (BLE) connection which is required for using the offered software where no other BLE connection is present. An example of this is if a desktop machine does not support Bluetooth or there are performance issues related to the built-in BLE functionality. Installation instructions for the *Kionix BLE Router* application are in chapter 4.4. Currently this application is available for Android devices in GitHub.

3.1.4.1. QR code

4. Dependencies

This chapter lists dependencies needed by *Kionix Evaluation Kit SW*.

4.1. Installing required 3rd party software

4.1.1. Python

Python (version 2.7.x) interpreter is needed to run the *Kionix Multi-OS Evaluation Software*.

The following Python modules are needed and can be installed with the Python package manager after python is installed.

For USB serial communication the following packages are needed:
`python –m pip install pyserial==3.0.1`

If `plot.py` log viewer tool is used then Matplotlib is required http://matplotlib.org/. Installing instructions are located in https://matplotlib.org/users/installing.html. Attention is needed to also fulfill the required dependencies https://matplotlib.org/users/installing.html#required-dependencies .

Windows install

Download the python interpreter (32 bit version) install package.
https://www.python.org/downloads/ or
ActivePython-2.7 http://www.activestate.com/activepython/downloads

With Windows OS, if automatic USB serial port detection is used in the *Kionix Multi-OS Evaluation Software* (by default this feature is active in `settings.cfg`. Ref chapter 6.2.2.1), the following modules are also needed:

```
python –m pip install wmi
python –m pip install pypiwin32
```

If *Kionix IoT Sensor Node* is used with USB connection and FTDI drivers are not already installed, refer to section 4.1.3.

Embedded Linux install

With Linux package manager (for example apt-get) install the following packages:

- Python
- i2c-dev
- i2c-tools
- python-smbus

In this case the port settings to the `settings.cfg` file is used as an example:

```
com_port = /dev/ttyUSB0
```

Ports of the connected devices can be obtained with the following command

```
ls –la /dev/serial/by-id
```

4.1.2. Aardvark

The Aardvark I2C/SPI host adapter is supported by *Kionix Multi-OS Evaluation Software*.

The Total Phase USB driver for Aardvark is bundled with the *Kionix IoT Evaluation Kit* and located in the folder `Kionix-Evaluation-Kit-Firmware\dependencies\Aardvark`
The latest drivers are also available at http://www.totalphase.com/products/usb-drivers-windows

4.1.3. FTDI USB Serial driver

Kionix IoT Sensor Node uses FTDI for USB serial communications. If Windows does not install the needed drivers automatically, they can be loaded from: http://www.ftdichip.com/Drivers/VCP.htm
Interoperability is tested with the driver version
http://www.ftdichip.com/Drivers/CDM/CDM21216_Setup.exe

If drivers are not installed correctly, the *Kionix IoT Sensor Node* would be shown in the Device Manager the following way:

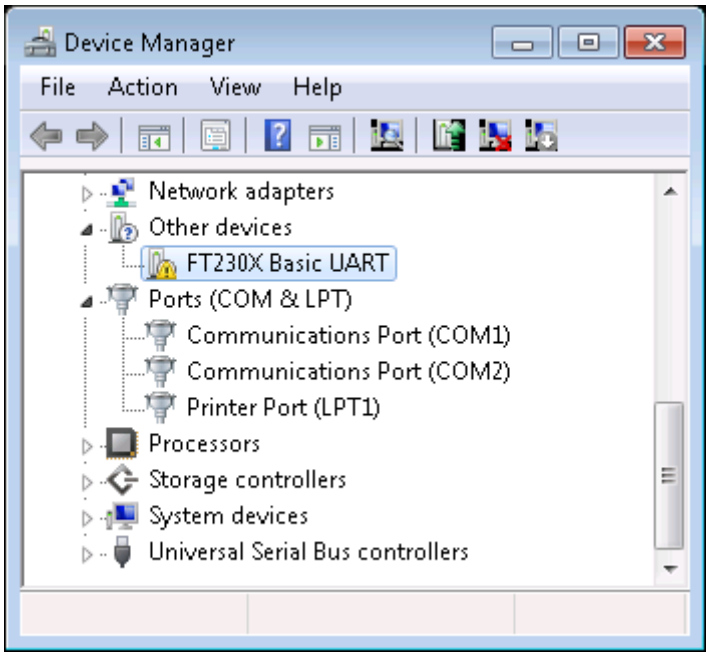


Figure 8: *Kionix IoT Sensor Node* USB driver not properly installed

After drivers are installed, *Kionix IoT Sensor Node* is listed as "USB Serial Port (COMx)" where 'x' is the COM port allocated for *Kionix IoT Sensor Node*.

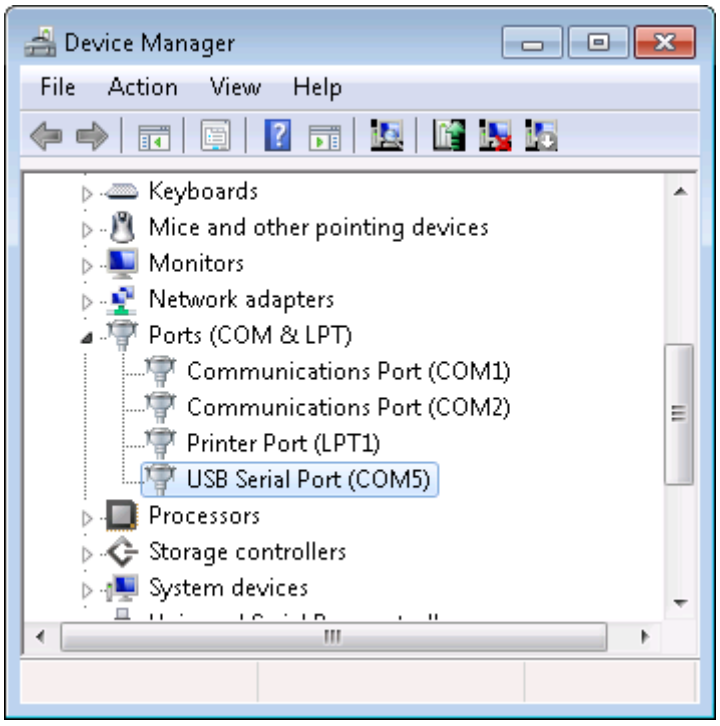


Figure 9: *Kionix IoT Sensor Node* USB driver properly installed

When using FTDI USB Serial connections with high ODR speeds (≥400Hz), the COM port setting should be updated to ensure data is transferred as soon as it is

available. Otherwise the data may come in bursts and time stamping values may not be correct.

The following configuration of a COM port was found to be suitable:

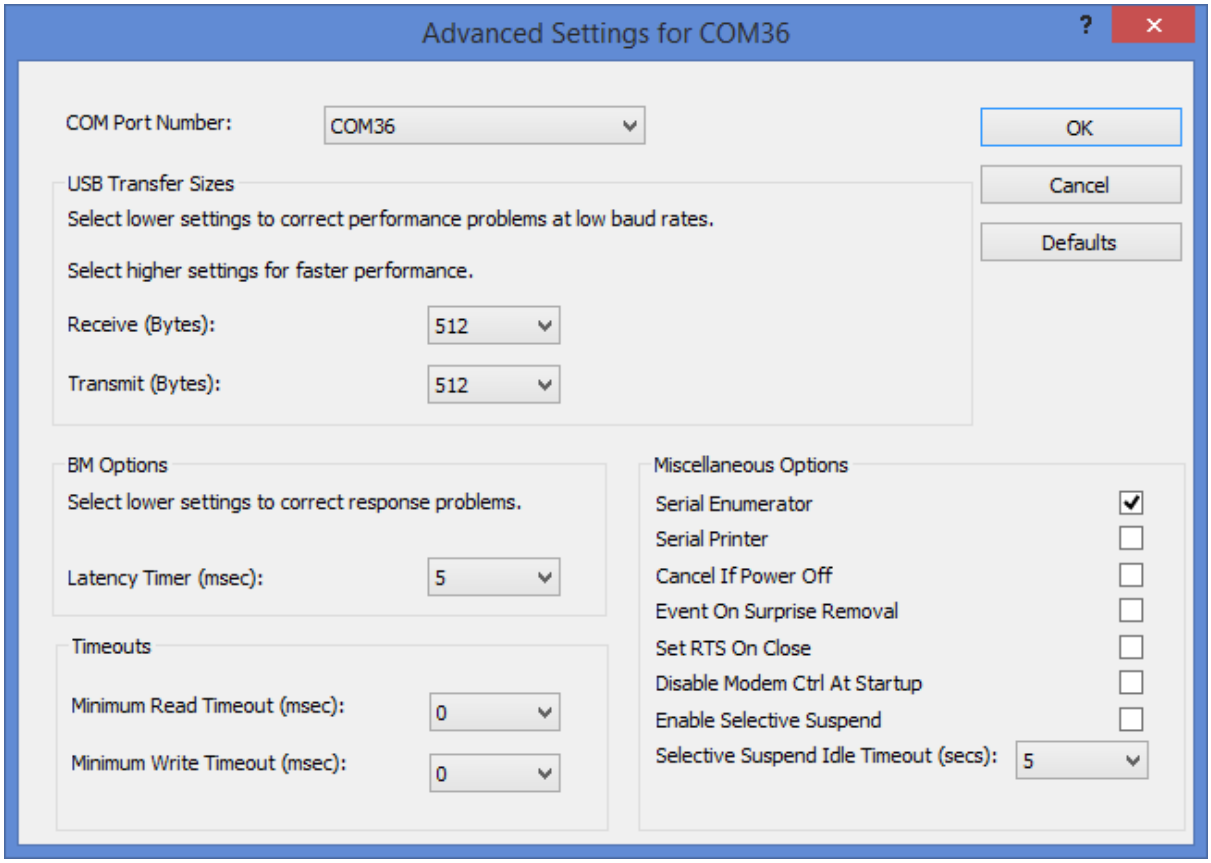


Figure 10: *Kionix IoT Sensor Node* USB settings

4.1.4. Android USB driver and ADB

The *Kionix Evaluation Kit SW* can use an Android device as BLE-USB router. This method is recommended to be used only when recommended BLE connection (ref chapter 4.2) is not available.

For this purpose both Android USB drivers and Android Debug Bridge (ADB) must be installed. When an Android device is connected to a Windows PC via USB connection and Windows does not install Android USB drivers automatically, they can be loaded from: http://developer.android.com/sdk/win-usb.html#download

For example the Android Debug Bridge can be found in the following locations:

- Android Studio https://developer.android.com/studio/index.html
- Android platform-tools http://stackoverflow.com/tags/adb/info

Android Debug Bridge related files; `ADB.EXE` and DLLs must be found in the directory listed in the PATH environment variable or it must be copied to:

`Documents\Kionix\Kionix-Multi-OS-Evaluation-Software\lib\`
and
`Documents\Kionix\Kionix-Windows-Sensor-Evaluation-Software\`

Kionix BLE Router set-up is described in chapter 4.4.

4.2. Pairing Bluetooth enabled *Kionix Evaluation Kit HW* with Windows

If the established *Kionix Evaluation Kit HW* supports Bluetooth connectivity, the pairing with Windows is done with the following steps:

1. Go to Control Panel\All Control Panel Items\Devices and Printers
2. Add a device
3. Choose "`KX_IoTv1.3`"
4. Next
   ≫ *Kionix IoT Sensor Node* led light should turn green during the device installation

If the pairing succeeded, the "`KX_IoTv1.3`" device is visible in the "`Unspecified`" - section of "Devices and Printers"

When the connection is changed back to USB, the *Kionix IoT Sensor Node* will also start to use USB automatically. The color of the *Kionix IoT Sensor Node* status LEDs will change from green to red.

4.3. *Kionix Evaluation Kit Firmware* programming / updating

This chapter describes the programming process of *Kionix Evaluation Kit Firmware* to the supported *Kionix Evaluation Kit HW*.

4.3.1. *Kionix IoT Sensor Node* firmware

This chapter provides instructions on how to use the DFU (Device Firmware Update) to update the boot loader and *Kionix Evaluation Kit Firmware* to IOT Sensor Node with OTA (Over The Air) using an Android or IOS phone.

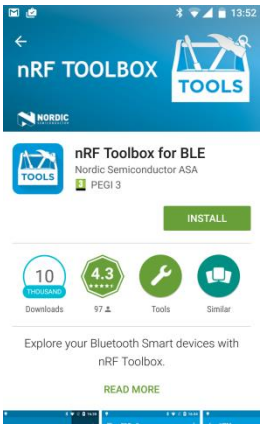
The *Kionix IoT Sensor Node* is shipped with a pre-installed bootloader and *Kionix Evaluation Kit Firmware*. Both can be updated with an iPhone or Android using the "DFU" functionality of the "nRF Toolbox", which is available in the application store (for example in Google Play)

4.3.1.1. QR code for Kionix Evaluation Kit Firmware

4.3.1.2. QR code for Kionix bootloader

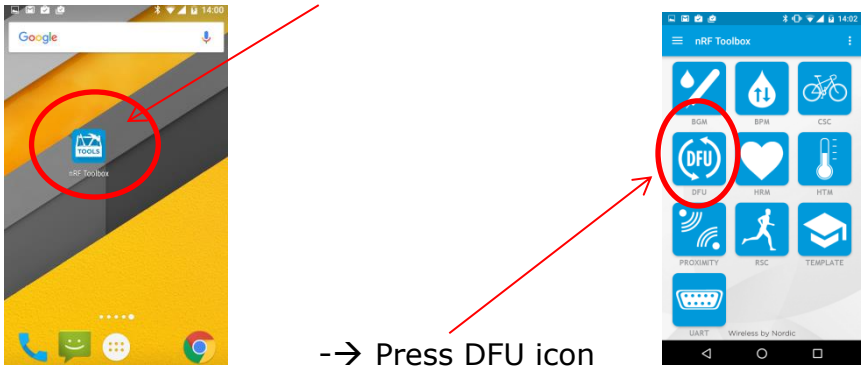
Update the *Kionix Evaluation Kit Firmware* to the *Kionix IoT Sensor Node* with the following procedure:

1) Load the Nordic Toolbox App (nRF Toolbox App) for your Android –phone (Google Play store) or Apple Store for IPhone.
   Here is an example of the Android phone installation via the Google Play store;



Press Install button

2) Start application from main icon

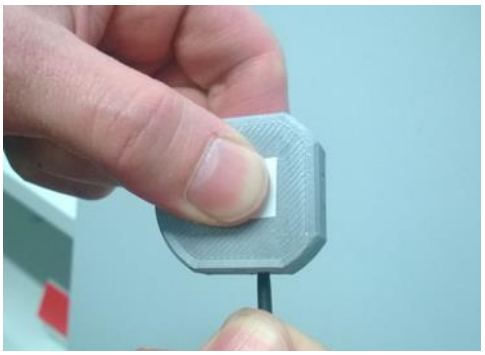


-→ Press DFU icon

3) DFU menu will open

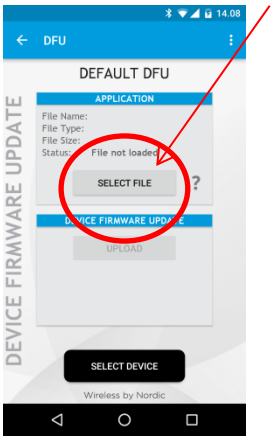
4) Set the *Kionix IoT Sensor Node* to DFU mode.
   a. If the *Kionix IoT Sensor Node* is on, power it off by pressing the power button
   b. Press the *Kionix IoT Sensor Node* power button for 3 seconds. The status LEDs will blink and finally turn off when the device has entered the DFU mode.
   c. Check the device name using the nRF Toolbox. Press the 'select device' button and see available devices. **Note! If the device name in the list is different than 'KX_Bv1' (or same prefix with greater number) then the bootloader must be updated before updating firmware. Otherwise the DFU will stop**

working and the Bootloader and firmware update need to be done using the ARM JTAG interface.

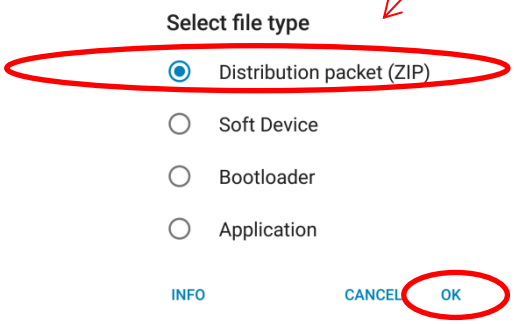


5) The *Kionix Evaluation Kit Firmware* is bundled with the *Kionix IoT Evaluation Kit* and located in the directory `Documents\Kionix\Kionix-Evaluation-Kit-Firmware\Kionix-IoT-Sensor-Node\`
   a. Load the firmware update file to the Android phone into the directory of your choice. Example `kionix-iot-firmware-latest-release.zip` to `/sdcard` –directory. Alternatively
   b. `copy_latest_firmware_to_phone.bat` can be used for copying (`adb.exe` [ref 4.4.1] must be found from PATH)
   c. The *Kionix Evaluation Kit Firmware* and bootloader can be downloaded to the device from GitHub:
      - Kionix Evaluation Kit Firmware
      - Kionix bootloader

6) Press "Select File"-button

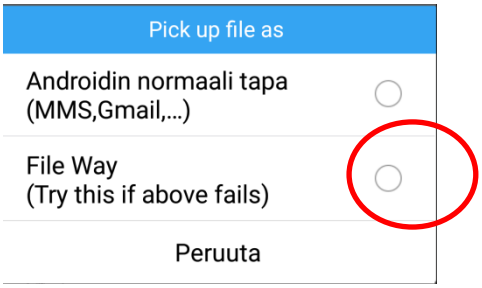


Select "Distribution packet (ZIP)

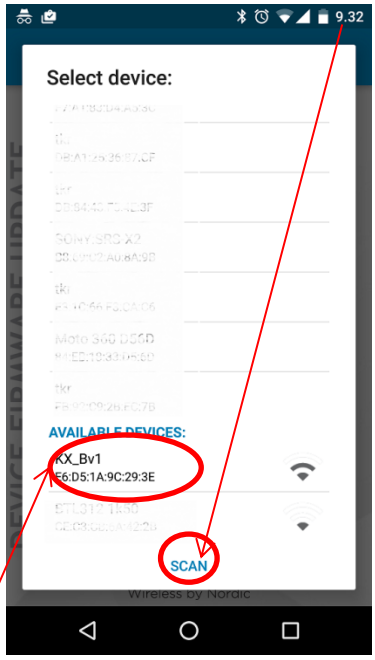


a. Press OK button

7) Select your favorite file manager program
   a. Select the file
   b. Select the pick up method (File Way) button if requested to do so



8) Press "Select Device"-button
   a. Scan to update MAC number info and choose your device



b. Select your *Kionix IoT Sensor Node*'s MAC number from the list

9) Press "Upload"-button
   a. Wait until the uploading starts (may take up to 2 minutes)
      i. The upload status bar informs the percentage level of the current upload
      ii. If uploading does not start, please check "Troubleshooting hints" below.
   b. The DFU application should notify the user that the update was successful

Troubleshooting hints:
   a. Make sure that the battery of the *Kionix IoT Sensor Node* is fully charged or plug the USB cable to the *Kionix IoT Sensor Node* during the DFU operation.
   b. Reboot the device in which the nRF toolbox is running
   c. Disable WLAN from the device in which the nRF toolbox is running
   d. Check the package size setting parameters in the DFU Settings

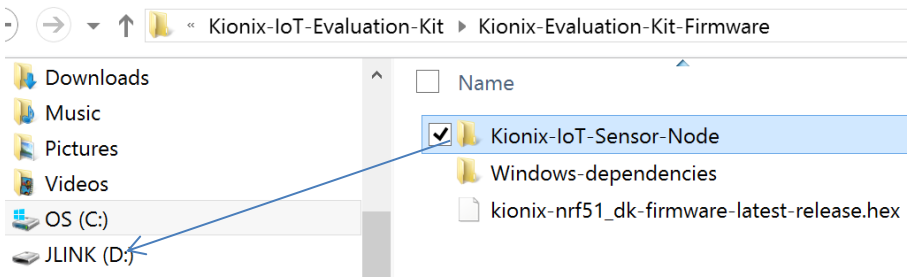
References:

- nRF Toolbox manual
- Nordic Semiconductor Infocenter
- Nordic Semiconductor Developer Forum

4.3.2. *Nordic Semiconductor nRF51-DK* firmware

The nRF51-DK development kit has a pre-installed mbed boot loader. The *Kionix Evaluation Kit Firmware* can be programmed to the development board with the following steps:

- Connect the nRF51-DK to the PC with a USB cable. The nRF51-DK is listed as a hard drive in the Windows File Explorer
- Copy & Paste the firmware to nRF51-DK. The firmware is located at `Documents\Kionix\Kionix-Evaluation-Kit-Firmware`



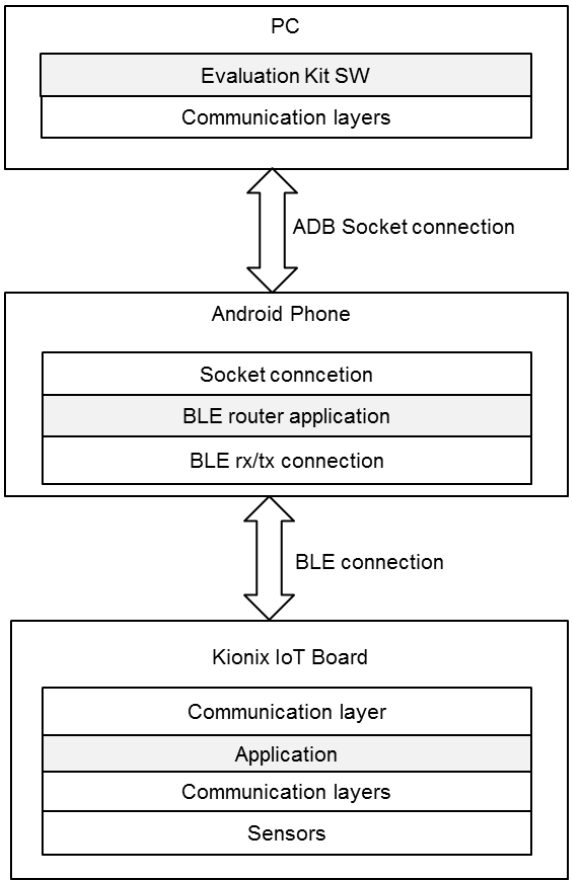
4.3.2.1. QR code

4.4. *Kionix BLE Router* setup to Android

This chapter provides instructions on how to install the *Kionix BLE Router* for Android devices. The router application passes the BLE message between the phone and the *Kionix Evaluation Kit SW* in PC.

NOTE: Please refer to chapter 4.2 for recommended way to use wireless connection in Windows.

The figure below illustrates how the connection is implemented.



4.4.1. Installation to PC

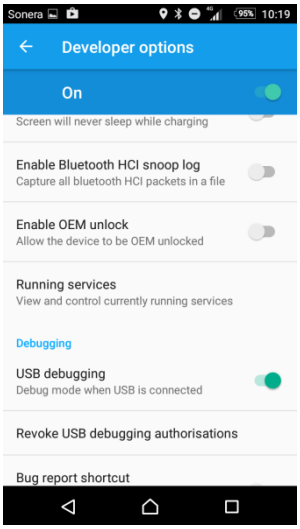
Windows requires the Google USB driver and the `ADB.EXE` tool (ref. chapter 4.1.4).

4.4.2. Installation to Android devices

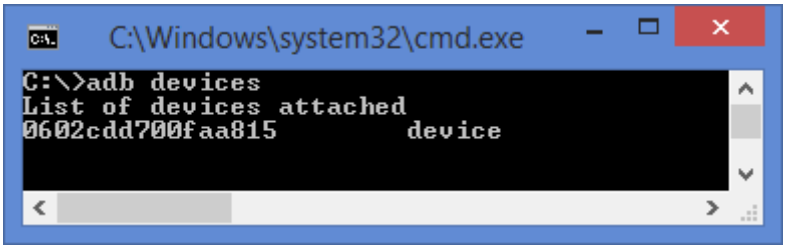
Enable developer mode from the Android device

On Android 5.0 and higher, the Developer options screen is hidden by default. To make it visible, go to **Settings > About phone** and tap **Build number** seven times. Return to the previous screen to find **Developer options** at the bottom. On some devices, the Developer options screen may be located or named differently.

The *Kionix BLE Router* uses a USB cable connection to the PC. On top of the USB connection the ADB socket connection is used. To enable ADB socket connection, **Developer mode** and **USB debugging** must be enabled from within the device system settings, under **Developer options**.



Connect the Android device to the PC via USB and test connection. Test that the Android device is found:



The Kionix BLE Router is bundled with *Kionix IoT Evaluation Kit* and located in folder `Documents\Kionix\Kionix-BLE-Router\Android\`. It can be also downloaded directly from GitHub.

Install the *Kionix BLE Router* to the Android device with the following command:
`install_latest_ble_router_to_phone.bat`
or:
`adb install evkit-ble-router-latest-release.apk`

NOTE: For Android application installation, the directory in which adb is installed should contain the application files (.bat file from the Kionix BLE Router directory). A way around this is to add the adb folder into your PATH by changing the environment variables.

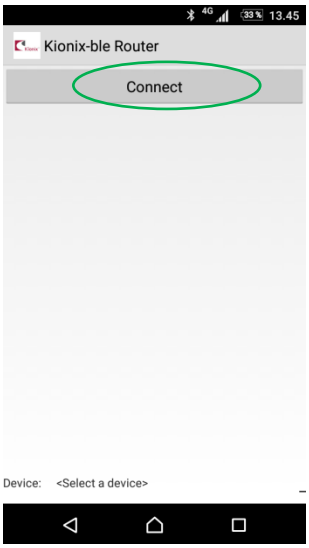
4.4.3. Connecting Android to IoT Board

On the *Kionix IoT Sensor Node* do the following steps:

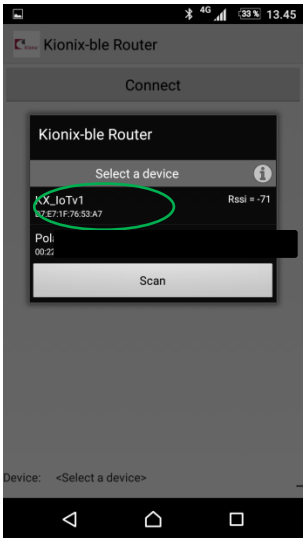
- Connect USB cable to power the device or use the battery.
- Turn on the *Kionix IoT Sensor Node* on by short pressing the power key. Red status LED should be alight (ref 2.2.1), when the device is powered on.

On the Android device do the following steps:

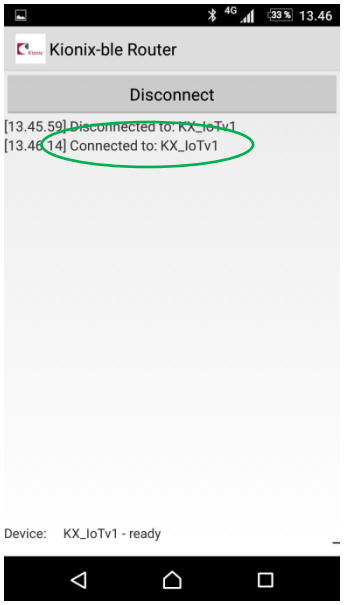
- Start *Kionix BLE Router* application.
- Press the "connect" button.



- *The Kionix IoT Sensor Node* should be automatically scanned and will be seen on the devices list.
- Select the device to connect.



If connection was successful, you should see the connected message in the list view. Green status LED will be lit on *Kionix IoT Sensor Node.*



5. Kionix Windows Sensor Evaluation Software

5.1. Introduction

The *Kionix Windows Sensor Evaluation Software* provides an easy to use graphical user interface demonstrating high level sensor offerings and features. Some of the features include:

- Demonstration of Kionix software offering which includes the Sensor Fusion Algorithm
- A Visual display of real time sensor data
- Ability to record sensor data onto a file
- Sensor registry editor

5.2. Setup

5.2.1. Installation

The *Kionix Windows Sensor Evaluation Software* is located at `Documents\Kionix\Kionix-Windows-Sensor-Evaluation-Software\KionixEvaluationKit.exe`

Before running the software, the following installation steps may be necessary:

1. When using *Kionix IoT Sensor Node* with *Kionix BLE Router*, Android USB device drivers are needed (4.1.4)
2. When using *Kionix IoT Sensor Node* with USB connection, necessary USB serial drivers (4.1.3) (if Windows does not install automatically)
3. When using *Kionix IoT Sensor Node* with Windows Bluetooth connection, pairing is needed (4.2)

Download link to latest release of *Kionix Windows Sensor Evaluation Software*: Latest release

5.2.2. Configuration

5.2.2.1. Kionix Evaluation Kit board configuration and connection type

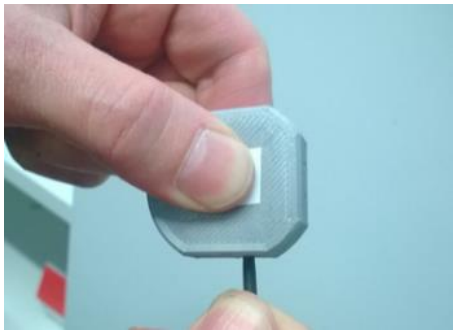
The following settings must be done from the application menu to get the sensor connected:

- Board configuration from Board-menu 5.4.7
- Connection type from Connection-menu 5.4.3

5.3. Getting Started

Follow these simple steps to get started with the *Kionix Windows Sensor Evaluation Software* using USB connection and the *Kionix IoT Sensor Node*:

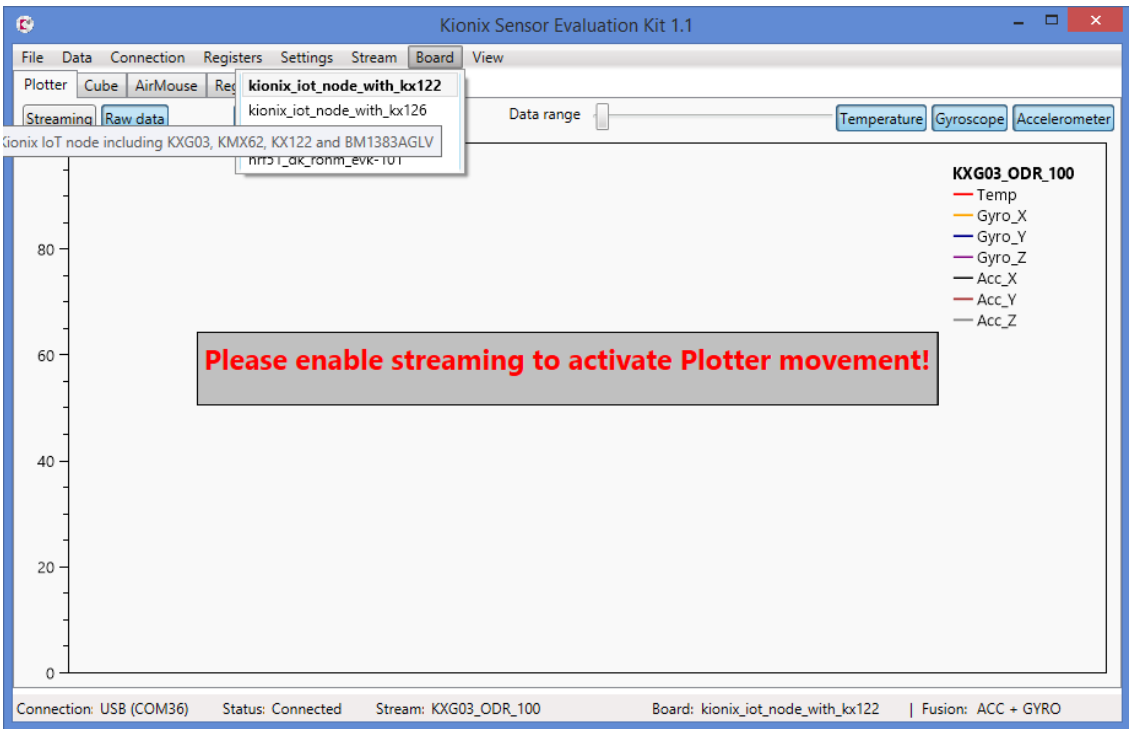
1. If a battery powered board is used, power on the *Kionix IoT Sensor Node* by pressing the power switch (ref 2.2.1)



2. Plug in a USB cable between the PC and the *Kionix IoT Sensor Node*

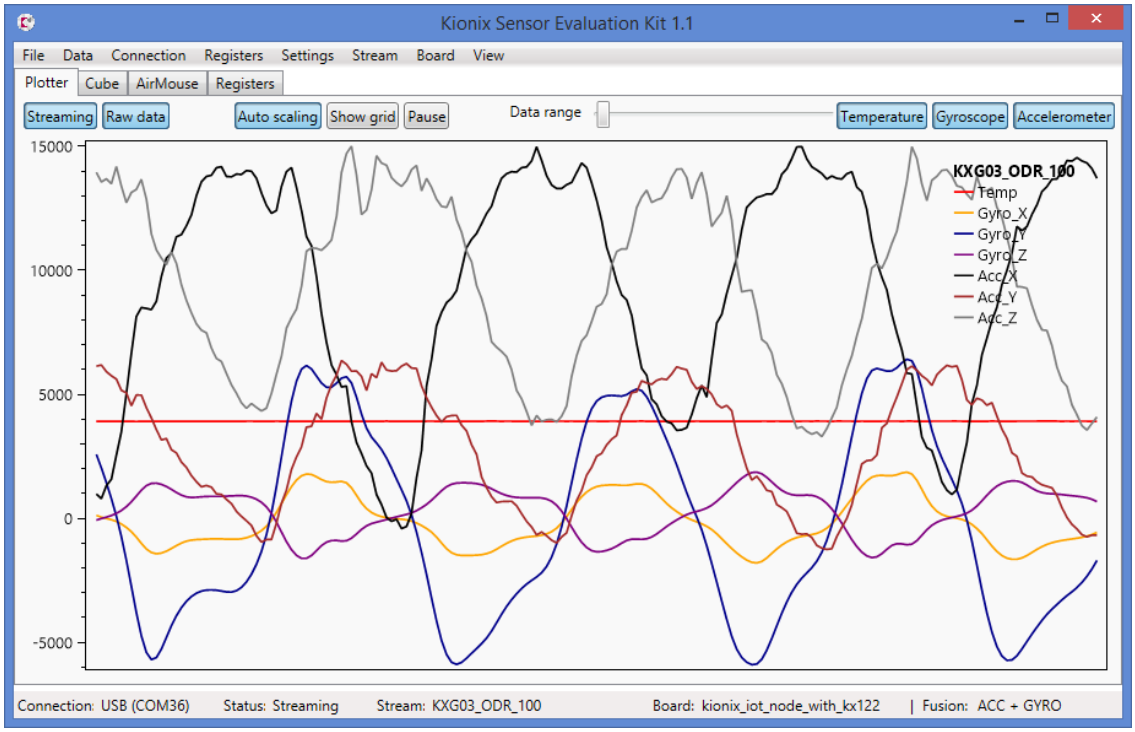
NOTE: When using the *Kionix IoT Sensor Node* on your PC for the first time, Windows will install a USB serial driver for it. It may take a while, please be patient. If the installation does not occur automatically, please refer to chapter 4.1.3 for installation instructions.

3. Start *Kionix Windows Sensor Evaluation Software* with `KionixEvaluationKit.exe`
4. Select which board is connected. Refer to chapter 2.2 to determine the right board.



5. When the "Please enable streaming to activate Plotter movement!" – Pop-up window appears on the screen, enable data streaming with "Streaming" button.

6. The Plotter now shows graphs according to your *Kionix IoT Sensor Node* movements



5.4. User Interface – Menu bar

5.4.1. File – Menu

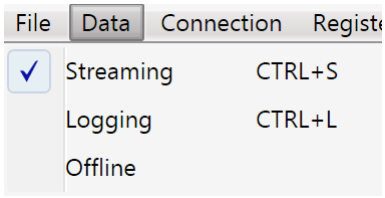
5.4.1.1. About

Shows the *Kionix Windows Sensor Evaluation Software*'s version and which Git commit it is from and when it has been built.

5.4.1.2. Exit

Exits from application.

5.4.2. Data – Menu



The data menu contains all main options for sensor data usage

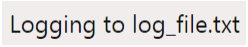
5.4.2.1. Streaming

The streaming menu item is used for enabling / disabling sensor data streaming.
Shortcut: CTRL + S

NOTE: Data stream enabling / disabling may take a while please be patient.

5.4.2.2. Logging

The logging menu item is used for enabling / disabling sensor data logging. The status bar will show the log file name.



More logging settings defined in 5.4.5.3.

Shortcut: CTRL + L

NOTE: Offline data is not supported in logging.

5.4.2.3. Offline

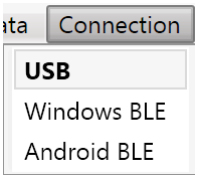
When "Offline" is selected a file dialog will open for choosing the offline sensor data file to be used. Offline data is mainly used in sensor fusion development and demo purposes. The selected offline data file must have accelerometer + magnetometer + gyro data to be used by sensor fusion algorithm.

NOTE: Offline mode is not supported in Plotter and logging.

5.4.3. Connection – Menu

The *Kionix Windows Sensor Evaluation Software* connects to the *Kionix IoT Sensor Node* either via USB COM port or via BLE.

The connection menu is used to select the desired connection type. *Kionix Windows Sensor Evaluation Software* uses USB COM connection by default. When auto connect is enabled, USB connection is established automatically when the *Kionix Evaluation Kit HW* is connected.



A BLE connection can be established with direct Windows BLE connection (`Windows BLE`) or by using an Android device (`Android BLE`).

NOTE: "`Windows BLE`" is recommended to be used when the system is running on Windows OS version 8.1 or later.

NOTE: Changing connection may take a while, please be patient.

NOTE: If you are having connection problems, "CTRL + R" can be used to refresh current connection.

5.4.3.1. Windows BLE connection

The Windows BLE connection works only with Windows 8.1 or later.
Please note that when selecting Windows BLE, ODR can sometimes be a bit slow. The FAQ section has information about this.

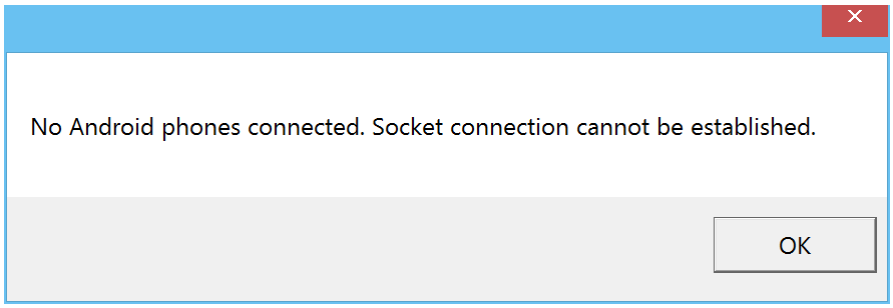
The *Kionix IoT Sensor Node* must be paired with Windows before BLE connection can be created. This procedure is described in chapter 4.2.

NOTE: In order to disconnect the Windows BLE connection manually, the *Kionix IoT Sensor Node* device must be turned off.

5.4.3.2. Android BLE connection

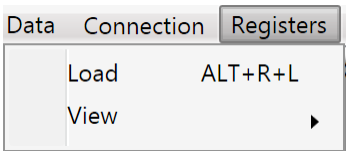
With an Android BLE connection an Android phone is needed to receive the BLE data from the *Kionix IoT Sensor Node*. The *Kionix Windows Sensor Evaluation Software* reads the data via TCP socket connected to the Android phone. The application will start and use the `adb.exe` which is used for communicating with the Android device via socket.

Before selecting the BLE connection, please make sure that your Android device's USB drivers are installed and working per chapter 4.4. If you see this dialog, there is a problem with the Android phone connection:



NOTE: When you are having connection problems, please restart both the Android application *Kionix BLE Router* and *Kionix Windows Sensor Evaluation Software*. This will then clean the whole socket data transfer pipe.

5.4.4. Registers – Menu



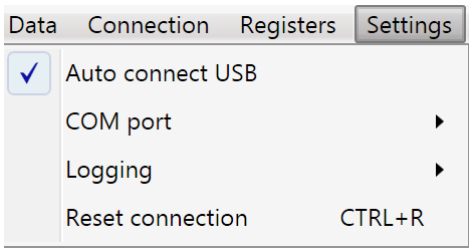
5.4.4.1. Load

Load sensor's register definition file from the "Registers" – tab. Please refer to chapter 5.5.4.

5.4.4.2. View

Change "Registers" – tab view type. "Simple" is a descriptive register view and "Advanced" is a line per register type of an approach.
"Advanced" – view is used by default.

5.4.5. Settings – Menu



5.4.5.1. Auto Connect

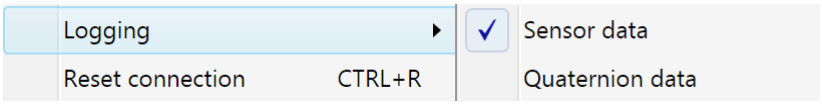
When "Auto connect" is enabled, the *Kionix Windows Sensor Evaluation Software* will automatically select the USB COM port for the connected device and connect to it.

5.4.5.2. COM port

When there are multiple devices connected or there is some problem with the USB COM port selection, the COM port can be selected from the dropdown list. Before doing this the "Auto connect" feature must be disabled.

5.4.5.3. Logging

Under logging menu you can define what data is stored to file when logging sensor data.



NOTE: Logging will always use raw sensor data instead of SI units.

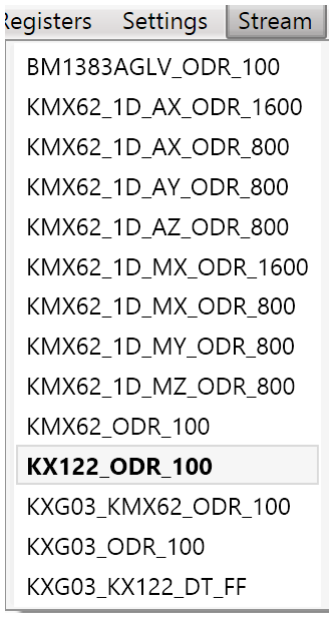
5.4.5.4. Reset connection

If you are having connection problems, "Reset connection" can be used for refreshing the current connection. It also initializes the current data stream again.

Shortcut: CTRL + L

The *Kionix Windows Sensor Evaluation Software* will store the selected settings to the user settings on the computer and will restore them the next time the application is started.

5.4.6. Stream – Menu



The stream menu is used for selecting the sensor data stream which the application uses to receive sensor data.

The list of streams is dynamic and will change according to the chosen board configurations (5.4.7 Board - Menu).

NOTE: Both the KXG03/KXG07, KX122/KX126 and the BM1383GLV/BM1383AGLV have the same I2C slave addresses if the corresponding part is assembled on the *Kionix IoT Sensor Node*.
For example: this means that the application allows the use of KXG07 stream even if KXG03 is present in the device. Please make sure that the correct board configuration (ref chapter 2.2) and stream are selected to avoid misleading sensor data.

5.4.6.1. Sensor Fusion

The Sensor Fusion algorithm to be used is selected automatically according to the used sensor data stream:

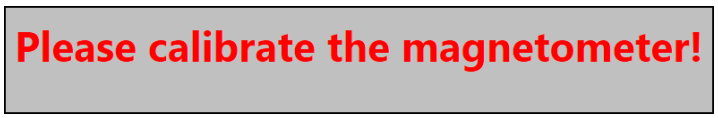
- Stream contains accelerometer and gyroscope data, but is missing the magnetometer data
  - Accelerometer + Gyro fusion is selected
- Stream contains accelerometer, magnetometer and gyroscope data
  - Accelerometer + Magnetometer + Gyro fusion is selected
- Stream contains accelerometer and magnetometer data, but is missing the gyroscope data
  - Accelerometer + Magnetometer fusion is selected

This selection affects the operation of Cube 5.5.2, Air mouse 5.5.3, and logging 5.4.5.3.

You can also use key shortcuts to override sensor fusion mode:
CTRL + F1: Accelerometer + Gyro
CTRL + F2: Accelerometer + Magnetometer + Gyro
CTRL + F3: Accelerometer + Magnetometer

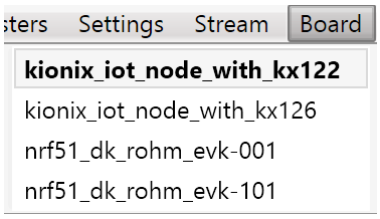
5.4.6.2. Streams with magnetometer

When magnetometer is in use and the dialog box shown below appears, it is time to calibrate the magnetometer:



Calibration is done by performing "8" – like movement with the sensor.

5.4.7. Board – Menu



The *Kionix Windows Sensor Evaluation Software* supports multiple board configurations:

Board configurations with *Kionix IoT Sensor Node*:
- "kionix_iot_node_with_kx122"
  - *Kionix IoT Sensor Node* including KXG03, KMX62, KX122 and BM1383AGLV
- "kionix_iot_node_with_kx126"
  - *Kionix IoT Sensor Node* including KXG03, KMX62, KX126 and BM1383AGLV

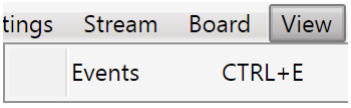
Board configurations with *Nordic nRF51-DK:*
- "nrf51_dk_rohm_evk-001"
  - Nordic nRF51-DK board with Rohm SensorShield-EVK-001 or with Kionix multi-Platform Adapter Board (ref chapters 2.5 and 2.3)
- "nrf51_dk_rohm_evk-101"
  - Nordic nRF51-DK board with Rohm EVK-101 including KMX62, KX122, KXG03, BM1383AGLV, BM1522GMV, RPR0521 and BH1745 (ref chapter 2.4)

The type of board that is used can be selected from this menu. The default board configuration is "kionix_iot_node_with_kx122".

When the board is changed, the connection will be changed to USB automatically.

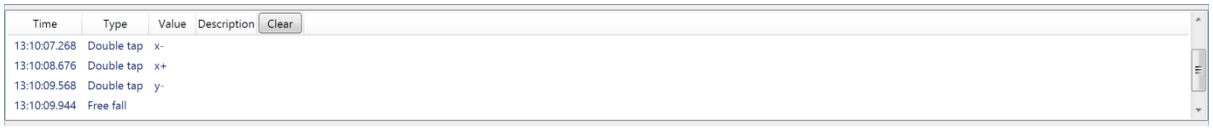
5.4.8. View – Menu

From View – Menu the "Events" – view can be enabled. It will appear on the bottom of the *Kionix Windows Sensor Evaluation Software* window.

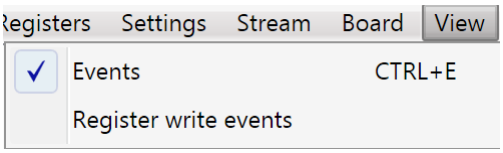


The "Events" – view will show "Double tap" and "Free fall" events.

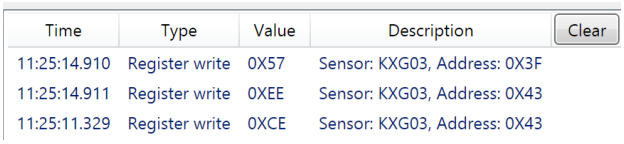
NOTE: This functionality currently supports only "KXG03_KX122_DT_FF" and "KXG03_KX126_DT_FF" – sensor streams.



From View – Menu you can enable "Register write events" if "Events" has also been enabled.



When the "Register write events" is enabled, you will see all register write actions which the *Kionix Windows Sensor Evaluation Software* is executing in the background.



Time	Type	Value	Description	Clear
11:25:14.910	Register write	0X57	Sensor: KXG03, Address: 0X3F	
11:25:14.911	Register write	0XEE	Sensor: KXG03, Address: 0X43	
11:25:11.329	Register write	0XCE	Sensor: KXG03, Address: 0X43	

5.5. User Interface - Tabs

The functionalities of the *Kionix Windows Sensor Evaluation Software* are divided between separate tabs.

5.5.1. Plotter – Tab

The Plotter shows sensor data from current stream.

The Plotter has its own `Streaming` and `Raw data` buttons in order to change them quickly.

When `Auto scaling` is enabled, plotter will auto scale the minimum and maximum values in y-axis according to the sensor data.

`Show grid` – enables data grid lines.
NOTE: This may slow down the plotter performance.

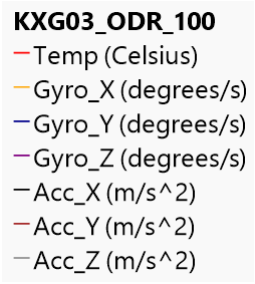
`Pause` – pauses plotter. There is also shortcut "P" for this.

`Data range` – adjusts the amount of data points shown in the plotter.

The Plotter has dynamic buttons in order to show/hide data channels within the used sensor stream. For example in the KXG03_ODR_100 stream there are "Temperature", "Gyroscope" and "Accelerometer" buttons that can be toggled.

5.5.1.1. Raw data

When `Raw data` is disabled, SI units for data streams are visible:



5.5.1.2. Zooming

You can zoom in and out using the mouse scroll button or right mouse button+CTRL.

NOTE: When zooming and "Auto scaling" is enabled, Plotter will no longer perform auto scaling. In order to re-enable "Auto scaling" after zooming, it must be disabled and enabled again.

5.5.1.3. Pausing

You can pause with "Pause" – button or with shortcut "P". Plotter can also be paused to certain a position using the left mouse button.

5.5.1.4. Moving

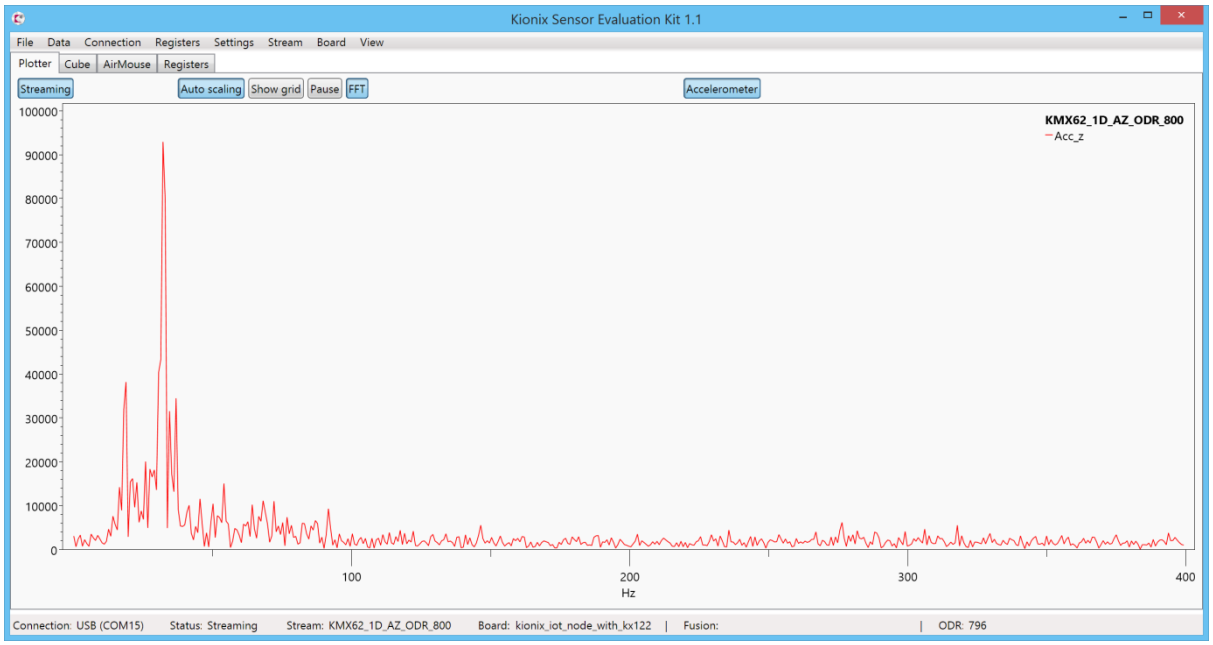
Data axis (y-axis) position can be moved up and down using the right mouse button.

5.5.1.5. Frequency analysis

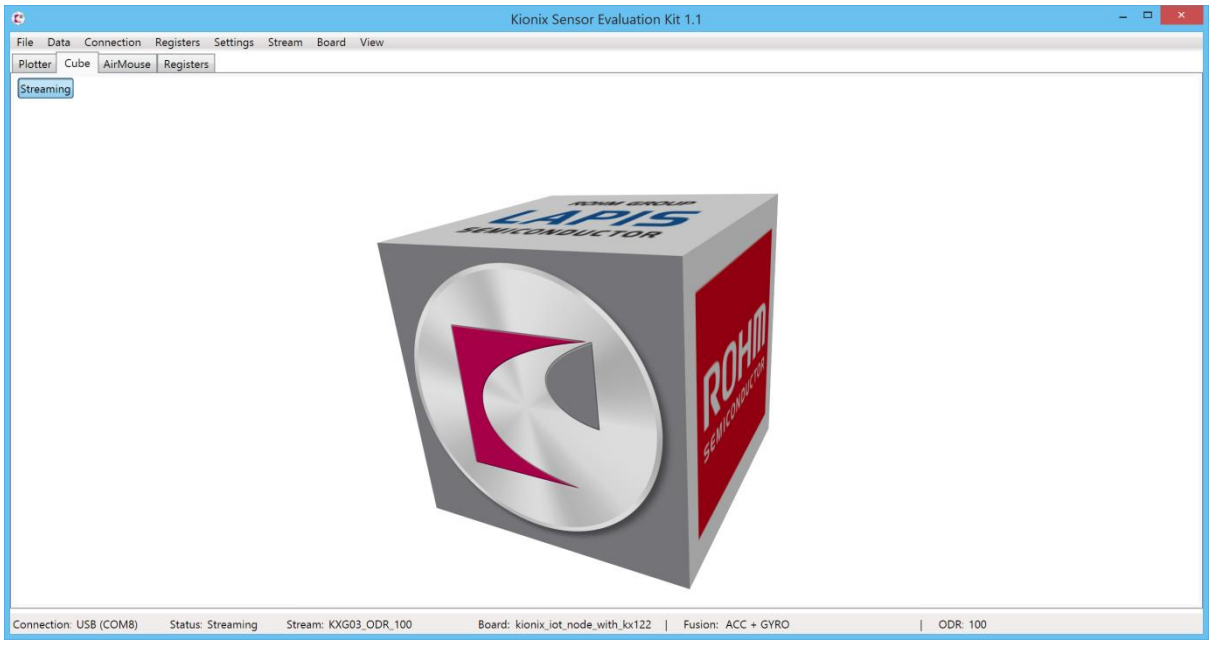
The Plotter has also an FFT (Fast Fourier Transform) functionality to show frequency data. The "FFT" – button is visible only with sensor streams which support FFT.

NOTE: FFT is supported only for KMX62 1-axis streams. (= sensor stream starts with "KMX62_1D")

NOTE: KMX62 1-axis streams with 1600 ODR require USB connection.



5.5.2. Cube – Tab



The 3D rotating cube is used for demonstrating the sensor fusion algorithm's performance. Sensor fusion can be started by pressing the "`Streaming`" button.

Cube rotation works only when the used data stream supports sensor fusion. It must include one of these channel combinations:

- Accelerometer + Gyroscope
- Accelerometer + Magnetometer + Gyroscope
- Accelerometer + Magnetometer

If sensor stream does not support fusion, "Fusion:" will show empty value in the status bar and the Cube is not rotating correctly.

Please refer to chapter 5.8.1 to get the rotation operating correctly.

5.5.2.1. Double Tap and Free Fall demo

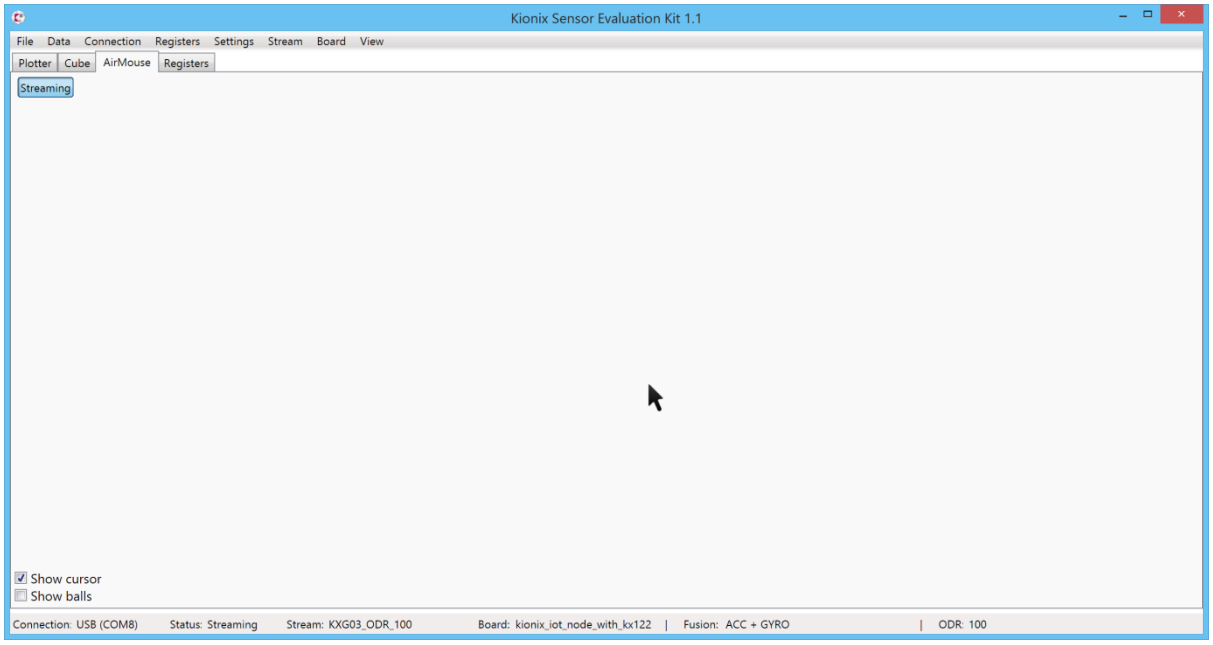
When "KXG03_KX122_DT_FF" or "KXG03_KX126_DT_FF" is selected sensor stream and "Streaming" is enabled and the Kionix sensor's Double Tap and Free Fall functionalities can be tested with the Cube. "KXG03_KX122_DT_FF" – stream uses the KX122 sensor for receiving Double Tap and Free Fall interrupts. "KXG03_KX126_DT_FF" is a similar kind of stream for the KX126 sensor.

When double tapping the *Kionix IoT Sensor Node,* the corresponding side of the Cube will be shortly highlighted with a red color.

When performing free fall (e.g. drop *Kionix IoT Sensor Node* to your hand), the Cube will also drop.

NOTE: This functionality is currently tied to the "KXG03_KX122_DT_FF" and "KXG03_KX126_DT_FF" – sensor streams.

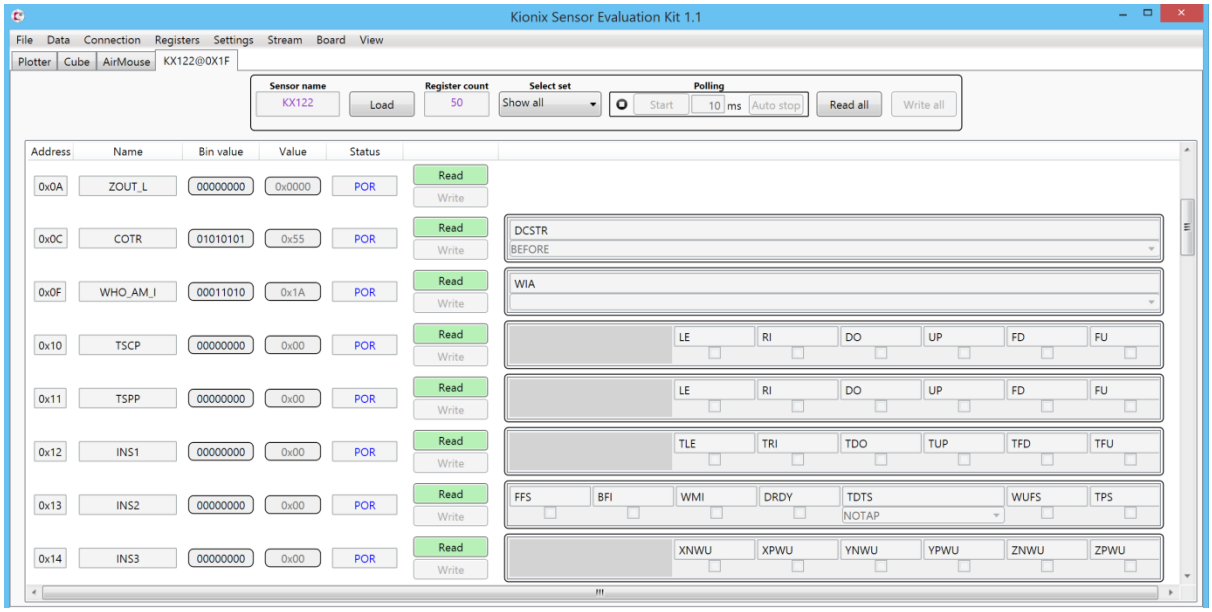
5.5.3. Air Mouse – Tab



Air mouse uses sensor fusion for moving the cursor. "Show balls" can be activated to show movement of x, y and z axis. Air Mouse can be started by pressing "`Streaming`" button.

Please refer to chapter 5.8.2 to get the movement operating correctly.

5.5.4. Registers – Tab



The register editor tab can be used for reading and writing sensor register values. When you have loaded the sensor register XML file, the tab name will change to <sensor name>@<SAD>. If the selected sensor is not supported in the currently connected board, then "NOT FOUND" – text will be displayed instead of SAD hex address.

When a new sensor in the register tab is opened, the register content is not updated automatically. Instead, register values can be updated with the "Read All" button or "Read" of individual register.

NOTE: Streaming and logging is automatically paused once Register editor tab is selected.

NOTE: When register values are changed and you return back to the Cube or Air Mouse, it is possible that changes have been made to the register values that had an effect on data streaming. Please change the data channel to any other channel and then change back to the original channel from the Stream-menu (5.4.6) in order to reset the data streaming register values.

5.5.4.1. Register sets

It is also possible to create application specific register sets. Register set will view only a subset of sensor registers. This helps in using a particular feature of the sensor. These are example settings related to:

- Changing data streaming related parameters: g-range, ODR, etc.
- Testing of ASIC level functionality: Motion wake-up, double tap detection, etc.

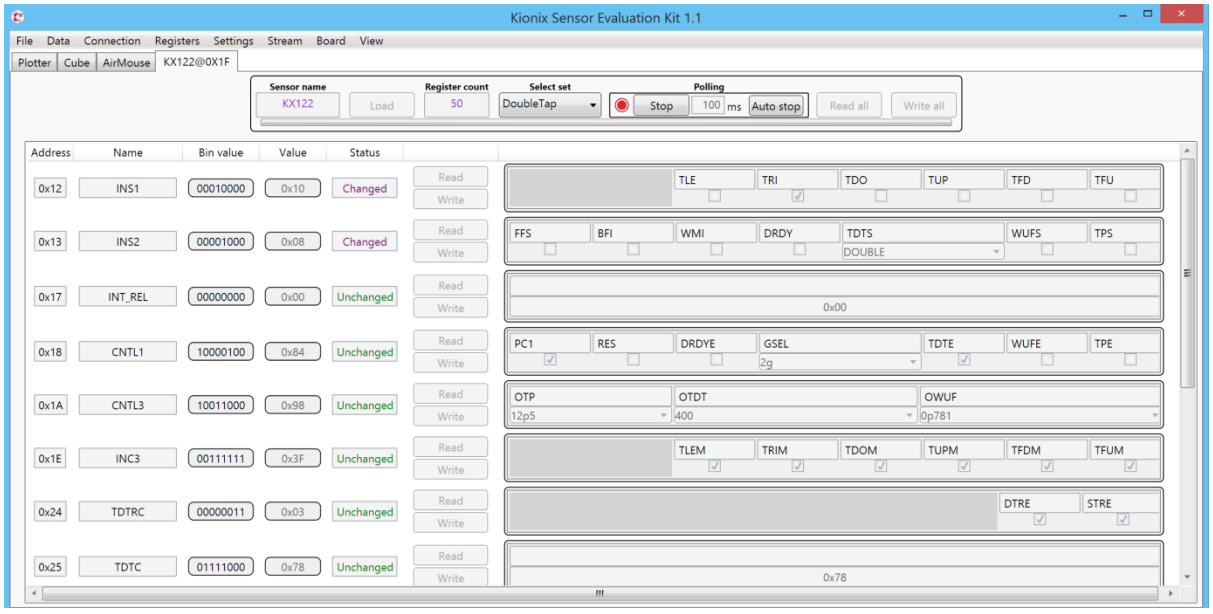
5.5.4.2. Register polling function

Register polling is a simple way to monitor register values when register editor-tab is active. Polling can be activated with a specific interval (default interval is 10ms). The polling feature has an "auto stop" – checkbox, which stops the polling to the first register value change. The below example shows how to monitor double tap detection:

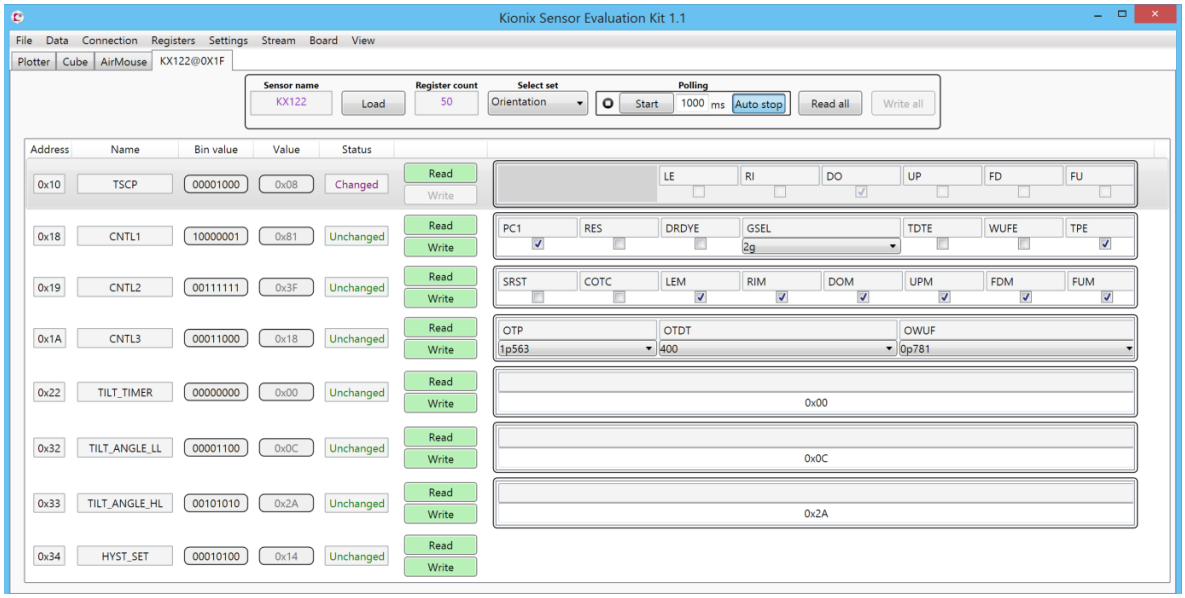
- Select sensor KX122 by pressing "Load" button.
- Select set "Double Tap" from pull down menu
- Check Double tap detection enabled (TDTE) and Power control (PC1) from register CNTL1 (0x18) and press write.
- Activate "Auto stop" from "Polling"
- Press "Start" button from "polling" (optionally change polling interval to 0)

Once double tap has been detected, the updated values are highlighted in red.

NOTE: Streaming and logging are automatically paused when register polling is started.



Double tap detection register set



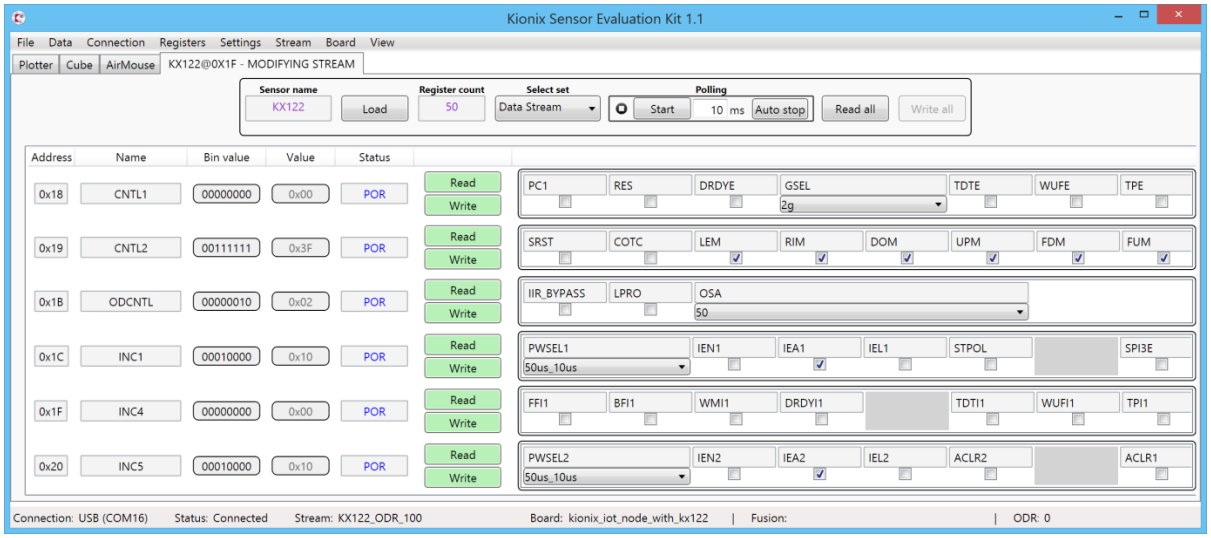
Orientation detection register set

5.5.4.3. Stream modify mode

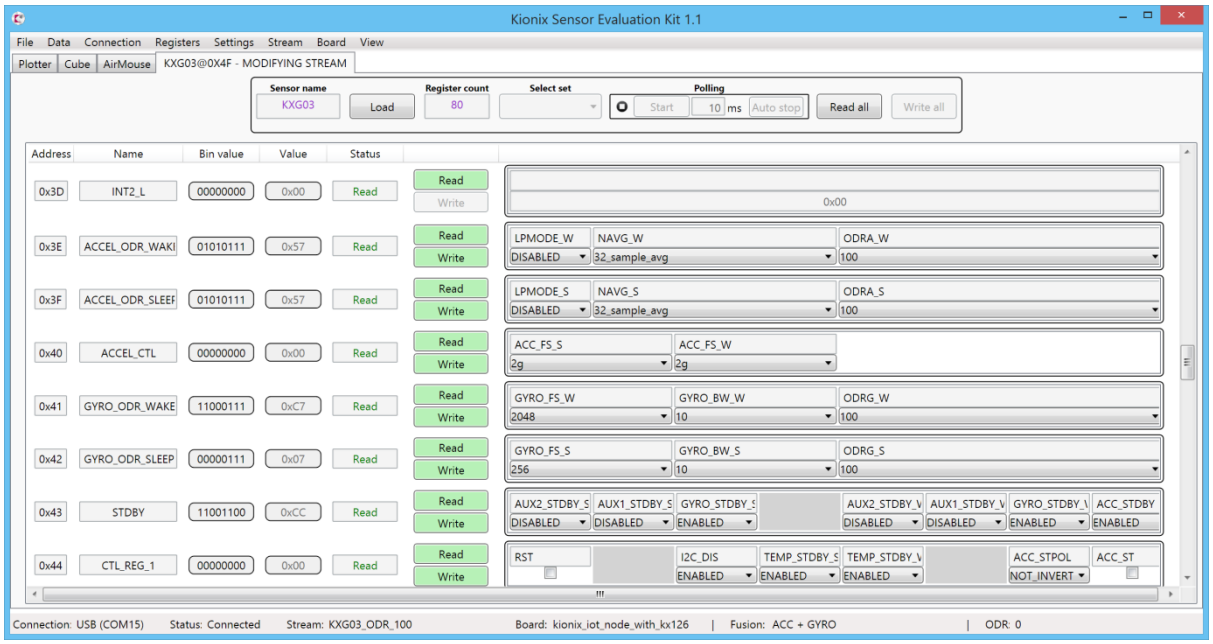
When the Register editor tab is selected while Streaming is enabled, the register editor will enter into a data stream modify mode. This mode offers the user a way to change the registers which will have an effect on the data stream itself. When wanted register changes have been made and the user selects the Plotter, Cube or Air Mouse tab, Streaming will be re-enabled with the changed register values. For example this makes it easy to change the ODR or data range.

NOTE: Some sensors require that the power control bit (PC) is set to 0 before changing the register values. Otherwise register value changes are not applied. After registers are edited, the PC bit must be set back to 1 to enable the sensor again.

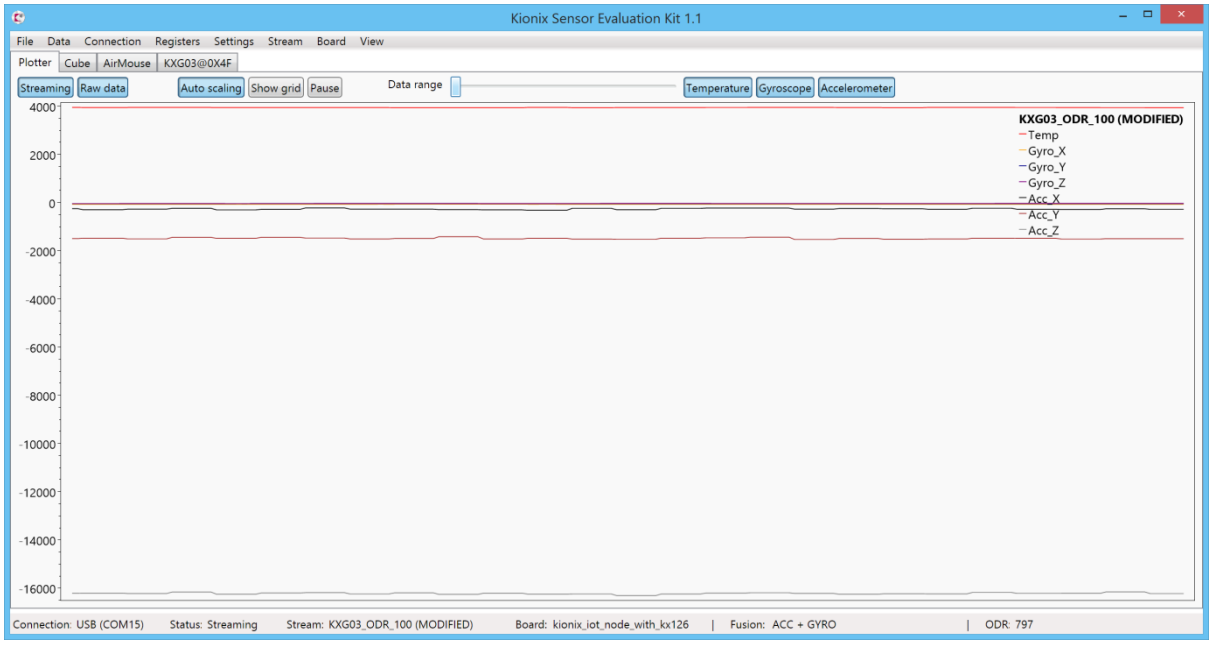
NOTE: In order to restore default register values of the data stream, Streaming must be disabled and enabled again.



NOTE: Many data streams have their own register sets called “Data stream”. They contain only registers which are directly linked to the used data stream. Thus, this makes it easier to modify data registers related to data reading.



Stream modify mode in register editor



When registers have been changed, (MODIFIED) text will be shown after stream name

5.6. User Interface - Status bar

The status bar shows the current connection and its status. It also keeps the user updated on the selected stream and board configuration, sensor fusion and magnetometer calibration status.

ODR is visible in the right corner. It can be hidden with "O" – key shortcut if needed.

NOTE: It is normal to see slight variation in the ODR value. Data is received at varying intervals and ODR is calculated when *Kionix Windows Sensor Evaluation Software* receives the data from the used connection layer.

5.7. User Interface - Pop-up windows

The application makes use of pop-up windows to notify the user about important actions. This section provides detailed information about pop-up windows.

5.7.1. Streaming pop-up window

Streaming pop-up window is showed in Cube, Plotter, and Air Mouse to notify the user about data stream enabling.
Streaming can be enabled with specific "Streaming" – button, from Data/Streaming – menu or with shortcut "CTRL + S"

5.7.2. Orientation reset pop-up window

Orientation reset pop-up is showed once per created device connection. It reminds the user to perform device orientation reset in Cube and Air Mouse.

5.7.3. Magnetometer calibration pop-up window

Magnetometer calibration pop-up is showed in Cube and Air Mouse when a channel including magnetometer is used and the calibration status is 0.
The calibration is done by performing "8" – like movement with the sensor.

5.7.4. No data pop-up window

No data pop-up is showed when streaming has been started, but no data is received. The problem could be invalid board configuration selection or some connection problem.

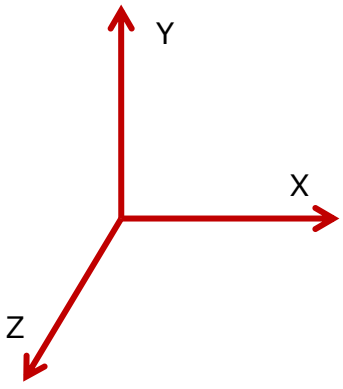
No data received! Please check your board configuration and device functionality.

5.8. Orientation reset

5.8.1. Cube reset

In order to reset rotating cube position and calibrate it with the position of the *Kionix IoT Sensor Node*, please follow these steps:

1. Place the *Kionix IoT Sensor Node* in the position where USB cable connector is pointing downward and status LEDs point to the right
2. Press "Space" – button
3. Cube is now reset to the *Kionix IoT Sensor Nodes* position and the Kionix logo is on the front side



5.8.2. Air Mouse reset

In order to reset Air Mouse position and calibrate it with the position of the *Kionix IoT Sensor Node*, please follow these steps:

1. Place the *Kionix IoT Sensor Node* in a position where the USB cable connector points away from the screen, the power button points towards the screen, and status LEDs point to the right
2. Press "Space" – button
3. Air Mouse is now reset to the *Kionix IoT Sensor Nodes* position

5.9. Shortcuts

The *Kionix Windows Sensor Evaluation Software* has many keyboard shortcuts:

CTRL + F1	=	Activate Accelerometer + Gyro sensor fusion mode
CTRL + F2	=	Activate Accelerometer + Magnetometer + Gyro sensor fusion mode
CTRL + F3	=	Activate Accelerometer + Magnetometer sensor fusion mode
CTRL + L	=	Enable/disable logging
CTRL + S	=	Enable/disable streaming
CTRL + R	=	Reset used connection and data streaming (disconnect & connect when having connection problem)
CTRL + E	=	Show events view
O	=	Hide/show ODR in status bar
P	=	Pause Plotter

6. Kionix Multi-OS Evaluation Software

6.1. Introduction

The *Kionix Multi-OS Evaluation Software* offers versatile access to sensor register level functionality. Simple and minimalistic demonstrator applications are provided for testing variety of sensor features. These will help to test and evaluate sensor functionality and provide needed information on how to implement sensor driver SW.

6.2. Set Up

6.2.1. Installation

The *Kionix Evaluation Kit SW* Windows installer includes the *Kionix Multi-OS Evaluation Software* and it will be installed to the following directory: `Documents\Kionix\Kionix-Multi-OS-Evaluation-Software\`. Additionally the installer creates start menu items for the *Kionix Multi-OS Evaluation Software*.

Kionix Multi-OS Evaluation Software requires:

- Python interpreter installation
- Drivers for utilized bus adapters, e.g.:
  - Aardvark I2C/SPI adapter
  - FTDI – USB serial
  - BLE
  - Embedded Linux

Chapter 4 describes the install processes of the necessary 3rd party software.
The following URL will always point to the latest version of the *Kionix Multi-OS Evaluation Software*.

- https://github.com/RohmSemiconductor/Kionix-IoT-Evaluation-Kit/archive/master.zip

On the UNIX system, the latest version of the *Kionix Multi-OS Evaluation Software* can be downloaded with the following command:

```
wget https://github.com/RohmSemiconductor/Kionix-IoT-Evaluation-Kit/archive/master.zip
```

6.2.2. Configuration

The *Kionix Multi-OS Evaluation Software* settings are in the `settings.cfg` file located in the root directory of *Kionix Multi-OS Evaluation Software*. The settings file uses windows .ini file syntax. Semicolon ";" at the beginning of the line indicates a comment.

There are settings for both:

- *Kionix Multi-OS Evaluation Software* framework and
- Test applications included in *Kionix Multi-OS Evaluation Software*

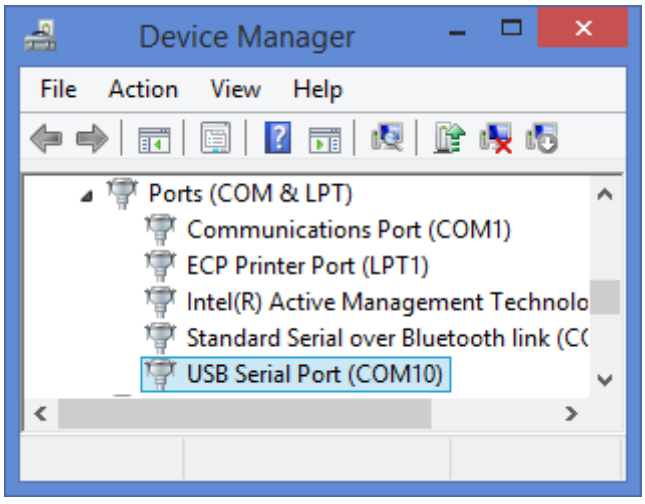
6.2.2.1. Connection to *Kionix Evaluation Kit HW*

The most important framework level setting is connection. It defines the bus adapter which is used. The definition is in section `[connection]`.

```
[connection]
; Bus index selection
; 1             Aardvark with I2C
; 2             Aardvark with SPI
; 3             BLE connection with windows 8.1/10 inbox bluetooth
connection
; 4             direct i2c connection through Embedded Linux GPIO
; 5             BLE connection using "Kionix BLE Router" Android
application
; serial_com_*  various COM port connections

;bus_index          = 3
bus_index          = serial_com_kx_iot
;bus_index          = serial_com_nrf51dk_a1
;bus_index          = serial_com_nrf51dk_rohm_evk_101
;bus_index          = serial_com_nrf51dk_rohm_evk_001
```

Key `bus_index` defines the bus adapter to be used. The default connection is `serial_com_kx_iot` e.g. *Kionix IoT Sensor Node* connected to a PC with a USB cable.

The bus adapters with names starting with "`serial_com_`" connect to the sensor through a windows USB serial port. For example *Kionix IoT Sensor Node* is listed as "USB Serial Port".



Corresponding COM port can be written in `settings.cfg` file:

```
[serial_com_kx_iot]
com_port              = COM10
```

The default connection is set to automatic port search:

```
[serial_com_kx_iot]
com_port = auto ; use "auto" for automatic search
```

This method works in Windows when the necessary dependencies are installed (ref 4.1.1). Otherwise the port must be configured manually.

On Linux the latest connected USB device can also be seen with the command `dmesg`.

6.2.2.2. Generic settings

The `[generic]` section contains connection independent settings. These include debug logging settings and interrupt related settings.

The debug logging level is defined with the `logging_level` and the default level is `INFO`:

```
logging_level = INFO
```

Other alternatives are: `DEBUG`, `INFO`, `WARNING`, `ERROR` and `CRITICAL`.
Interrupt related configurations are explained in following chapters.

6.2.2.3. Data Ready operation settings

There are four options for data ready operations.

```
;drdy_operation         = ADAPTER_GPIO1_INT
;drdy_operation         = ADAPTER_GPIO2_INT
;drdy_operation         = DRDY_REG_POLL
drdy_operation          = INTERVAL_READ
```

The default setting used for `drdy_operation` is `INTERVAL_READ`. The interval time is defined in `drdy_poll_interval` and the default value is set to 40ms. If `INTERVAL_READ` or `DRDY_REG_POLL`is used it is not mandatory to connect physical interrupt lines between the sensor and Bus adapter. Also, there is no need to configure the GPIO numbers for interrupt lines as described in chapter 0.

When the `DRDY_REG_POLL` option is used the data ready bit is monitored from the status register to trigger sensor data reading. This will introduce more traffic on the bus and thus reduce ODR. This will also verify that all data is read from the sensor. Physical interrupt lines are not used in this option either.

6.2.2.4. Interrupt polarity

The interrupts can be configured either active LOW or active HIGH. Active LOW is used by default since *Kionix IoT Sensor Node* requires active low interrupts.

```
;use active low as default
int1_active_high    = FALSE                         ; TRUE / FALSE
int2_active_high    = FALSE                         ; TRUE / FALSE
```

6.2.2.5. Interrupt lines

If the test application uses sensor's interrupt lines then the interrupt line mapping to HW board GPIO lines must also be configured properly. The default options are listed in the `settings.cfg`.For example:

```
pin1_index = 16 ; PAD TEST 5, int1 of kxg03, int1 of kx122
pin2_index = 13 ; PAD TEST 4, int2 of kxg03 and int2 of kx122
```

Example: If the INT2 signal of KX122 is used in the test application for signaling data ready interrupt then the corresponding data ready operation setting is:

```
drdy_operation          = ADAPTER_GPIO2_INT
```

Sometimes the pin index must be changed depending on how the sensor interrupt lines are connected to the microcontroller. Please refer to chapter 2 for details.

6.3. Getting Started

The quickest way to test *Kionix Multi-OS Evaluation Software* is to run the `hello_sensor.py` test application located at `Documents\Kionix\Kionix-Multi-OS-Evaluation-Software\`

The default connection is serial_com_kx_iot e.g. The *Kionix IoT Sensor Node* is connected to the PC with a USB cable (USB driver installation described in 4.1.3).

Executing `hello_sensor.py` the *Kionix Multi-OS Evaluation Software* will find the right USB serial port looks for all supported sensors from bus I2C and prints 10 data samples from each. An example output is shown below. NOTE: if running on Linux please refer to chapter 6.2.2.1.

```
>python hello_sensor.py
INFO :  Probe with kx022_driver
INFO :  KX112/KX122/KX123/KX124 found
INFO :  Sensor found from slave address 0x1f
Reading 10 data samples with kx122_driver.
0 (1496, 2420, 16214)
1 (1489, 2405, 16216)
2 (1480, 2407, 16214)
3 (1479, 2407, 16220)
4 (1481, 2417, 16226)
5 (1485, 2420, 16224)
6 (1495, 2413, 16225)
7 (1499, 2415, 16229)
8 (1496, 2411, 16229)
9 (1487, 2412, 16235)
INFO :  Probe with kmx62_driver
INFO :  KMX62 found
INFO :  Sensor found from slave address 0x0e
Reading 10 data samples with kmx62_driver.
0 (-1637, -532, -16091, -20481, 30591, -4097)
1 (-1715, -548, -16117, 746, -2635, 2687)
2 (-1703, -545, -16116, 742, -2632, 2684)
3 (-1699, -549, -16110, 731, -2636, 2685)
4 (-1694, -538, -16128, 725, -2644, 2671)
5 (-1687, -516, -16125, 729, -2639, 2673)
6 (-1684, -519, -16128, 727, -2642, 2676)
7 (-1689, -548, -16117, 723, -2648, 2679)
8 (-1680, -537, -16120, 721, -2651, 2670)
9 (-1692, -534, -16145, 710, -2661, 2666)
INFO :  Probe with kxg03_driver
INFO :  kxg03 found
INFO :  Sensor found from slave address 0x4f
Reading 10 data samples with kxg03_driver.
0 (-42, 211, 404, -2648, -2005, -17679)
1 (-41, 193, 346, -2628, -2000, -17688)
2 (-40, 188, 318, -2664, -2010, -17665)
3 (-45, 174, 286, -2644, -1988, -17669)
4 (-39, 179, 280, -2634, -2021, -17660)
5 (-43, 172, 272, -2656, -2003, -17685)
6 (-43, 172, 274, -2644, -2009, -17672)
7 (-38, 164, 273, -2641, -1999, -17669)
8 (-36, 160, 265, -2639, -2001, -17648)
9 (-32, 169, 268, -2648, -1999, -17664)
INFO :  Probe with kxg08_driver
INFO :  Probe with kxtj2_driver
INFO :  Probe with kxtj3_driver
INFO :  Probe with kxcnl_driver
INFO :  Probe with bh1726_driver
INFO :  Probe with bh1730_driver
INFO :  Probe with bh1745_driver
INFO :  Probe with bh1790_driver
INFO :  Probe with bm1383glv_driver
INFO :  Probe with bm1383aglv_driver
INFO :  Sensor found from slave address 0x5d
Reading 10 data samples with
bm1383aglv_driver.
0 (0, 0, 0, 0, 0)
1 (124, 15, 204, 3, 184)
2 (124, 20, 208, 3, 186)
3 (124, 20, 76, 3, 187)
4 (124, 20, 24, 3, 188)
5 (124, 20, 24, 3, 188)
6 (124, 18, 120, 3, 188)
7 (124, 19, 60, 3, 189)
8 (124, 17, 248, 3, 189)
9 (124, 16, 216, 3, 189)
INFO :  Probe with bm1422gmv_driver
INFO :  Probe with rpr0521_driver
>
```

6.4. File structure of the evaluation kit

The overall structure of the *Kionix Multi-OS Evaluation Software* is the following:

```
Kionix-Multi-OS-Evaluation-Software/
settings.cfg
hello_sensor.py
plot.py
validate.py
connection_setup.py
├──bh1726
├──bh1730
├──bh1745
├──bh1790
├──bm1383aglv
├──bm1383glv
├──bm1422gmv
├──kmx62
├──kx022_kx122
├──kx126
├──kxcnl
├──kxg03
├──kxg08
├──kxtj2
├──kxtj3
├──lib
├──license
└──rpr0521
```

Each sensor has its own directory containing:

- Register definitions (for example `kx022_registers.py`)
- Reference driver implementation (for example `kx022_driver.py`)
- Test application for reading sensor data (for example `kx022_data_logger.py`)
- Possible other test applications

Lib directory contains *Kionix Multi-OS Evaluation Software* middleware.

6.5. Running test applications

Test applications are sensor specific and located in corresponding folders. For example: data logging application for KX022 is located at `Kionix-Multi-OS-Evaluation-Software\kx022_kx122` application and it is executed with command:

```
python kx022_data_logger.py
```

The output of the application is the following (CTRL+C will stop the application).

```
INFO :  Bus serial_com_nrf51dk_rohm_evk_101 selected
INFO :  Automatic port choice: COM11
INFO :  Probe with kx022_driver
INFO :  KX112/KX122/KX123/KX124 found
INFO :  Sensor found from slave address 0x1e
#time;  10;     ax;     ay;     az
0.039989;       10;     -9;     -196;   16372
0.086040;       10;     5;      -221;   16354
0.137019;       10;     -4;     -214;   16356
0.185797;       10;     -11;    -207;   16353
0.236873;       10;     -4;     -200;   16353
0.286981;       10;     10;     -209;   16358
0.336043;       10;     -9;     -205;   16360
0.387093;       10;     -17;    -207;   16362
0.435952;       10;     -6;     -213;   16367
0.485743;       10;     12;     -212;   16454
```

Sensor data can be directed also to file:

```
python kx022_data_logger.py > kx022.csv
```

6.5.1. High ODR logging

Synchronous sensor reading in the *Kionix Multi-OS Evaluation Software* uses "polling" monitoring DRDY bit or corresponding interrupt line. When using the *Kionix IoT Evaluation Kit* with a bus adapter which uses *Kionix Evaluation Kit Firmware* then it is possible to achieve higher ODRs. This functionality requires that interrupt line(s) from the sensor are connected to the Bus adapter and interrupt lines are configured properly. Ref chapter 6.2.2.

All data logger test scripts are supporting this feature. The feature is enabled with `-s` switch. (Switch `-l` will stop logging after given number of data samples is read.)

```
python kx022_data_logger.py -s -l 4000 > kx022.csv
```

6.5.2. Other provided tools

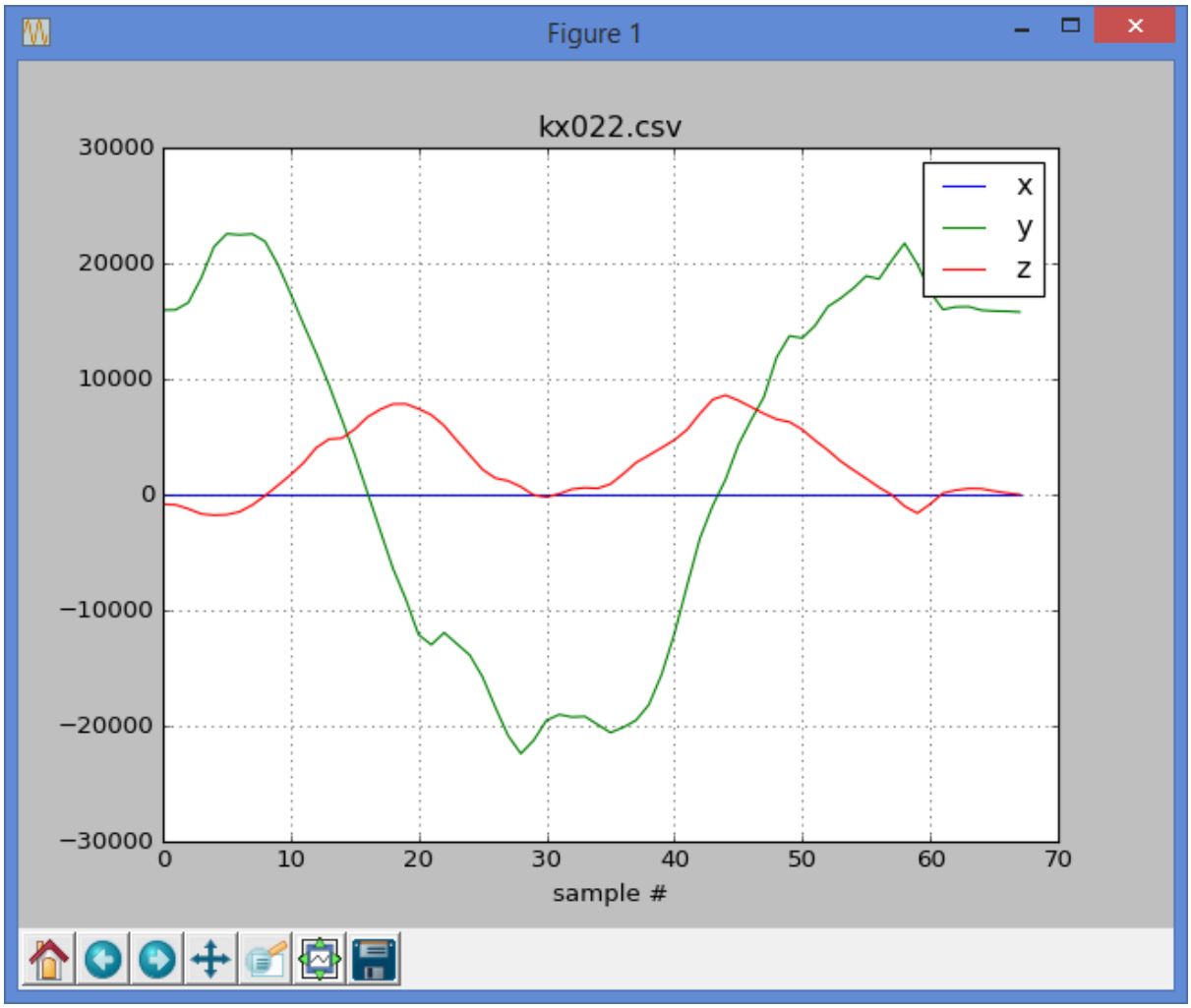
It is possible to verify the log validity:

```
kx022_kx122>..\validate.py kx022.csv
start time 0.00990 (s)
stop time  9.99000 (s)
duration   9.98010 (s)
samples    4000
max delta  5.16900 (ms) / 193 (Hz) at sample 2148
min delta  0.03900 (ms) / 25641 (Hz) at sample 692
avg delta  2.49502 (ms)
ODR avg    400.80 (Hz)
```

NOTE: Timestamping is done on a PC and it is not accurate with high ODRs. This causes an impact to delta time statistics.

It is possible to view the recorded log with the plot.py application:

```
kx022_kx122>python ..\plot.py kx022.csv -l x y z -c 2 3 4
```



6.6. Changing test application configuration

Test applications list the most common sensor configuration alternatives, including g-range and ODR settings. The required setting can be selected by commenting / un-commenting the corresponding lines of code:

kx022_data_logger.py

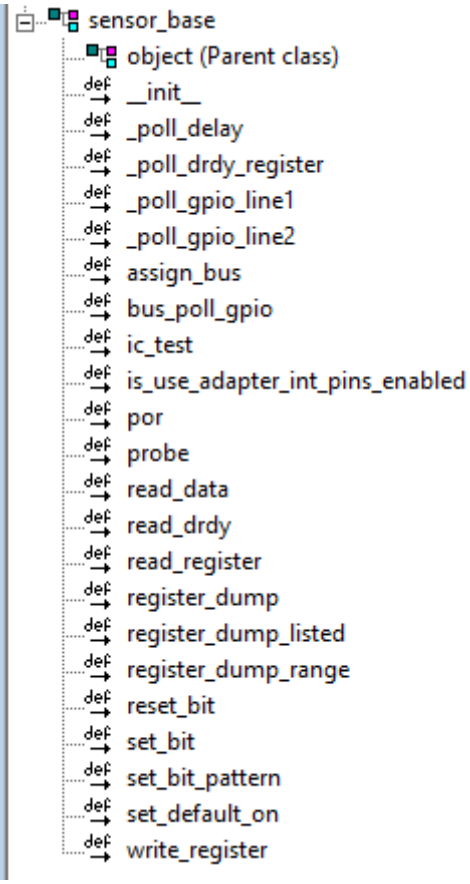
```
def test_data_logging(sensor):
    try:
        sensor.set_power_off()                          # this sensor request PC=0 to PC=1 before valid settings

        ## select ODR
        #sensor.set_odr(b.KX022_ODCNTL_OSA_0P781)       #
        #sensor.set_odr(b.KX022_ODCNTL_OSA_1P563)       #
        #sensor.set_odr(b.KX022_ODCNTL_OSA_3P125)       #
        #sensor.set_odr(b.KX022_ODCNTL_OSA_6P25)        #
        #sensor.set_odr(b.KX022_ODCNTL_OSA_12P5)        #
        sensor.set_odr(b.KX022_ODCNTL_OSA_25)           # odr setting for basic data logging
        #sensor.set_odr(b.KX022_ODCNTL_OSA_50)          #
        #sensor.set_odr(b.KX022_ODCNTL_OSA_100)         # works with Aardvark + I2C
        #sensor.set_odr(b.KX022_ODCNTL_OSA_200)         # works with Aardvark + SPI
        #sensor.set_odr(b.KX022_ODCNTL_OSA_800)         #
        #sensor.set_odr(b.KX022_ODCNTL_OSA_1600)        #
        ##higher than 1600Hz ODR are only for KX122, KX112, KX123, KX124
        #sensor.set_odr(b122.KX122_ODCNTL_OSA_3200)     #
        #sensor.set_odr(b122.KX122_ODCNTL_OSA_6400)     #
        #sensor.set_odr(b122.KX122_ODCNTL_OSA_12800)    #
        #sensor.set_odr(b122.KX122_ODCNTL_OSA_25600)    #

        ## select g-range
        sensor.set_range(b.KX022_CNTL1_GSEL_2G)
        #sensor.set_range(b.KX022_CNTL1_GSEL_4G)
        #sensor.set_range(b.KX022_CNTL1_GSEL_8G)

        ## resolution / power mode selection
        sensor.set_bit(r.KX022_CNTL1, b.KX022_CNTL1_RES)    # high resolution mode
        #sensor.reset_bit(r.KX022_CNTL1, b.KX022_CNTL1_RES)  # low resolution mode
        #sensor.average(0,b.KX022_LP_CNTL_AVC_NO_AVG)        # low resolution mode average
```

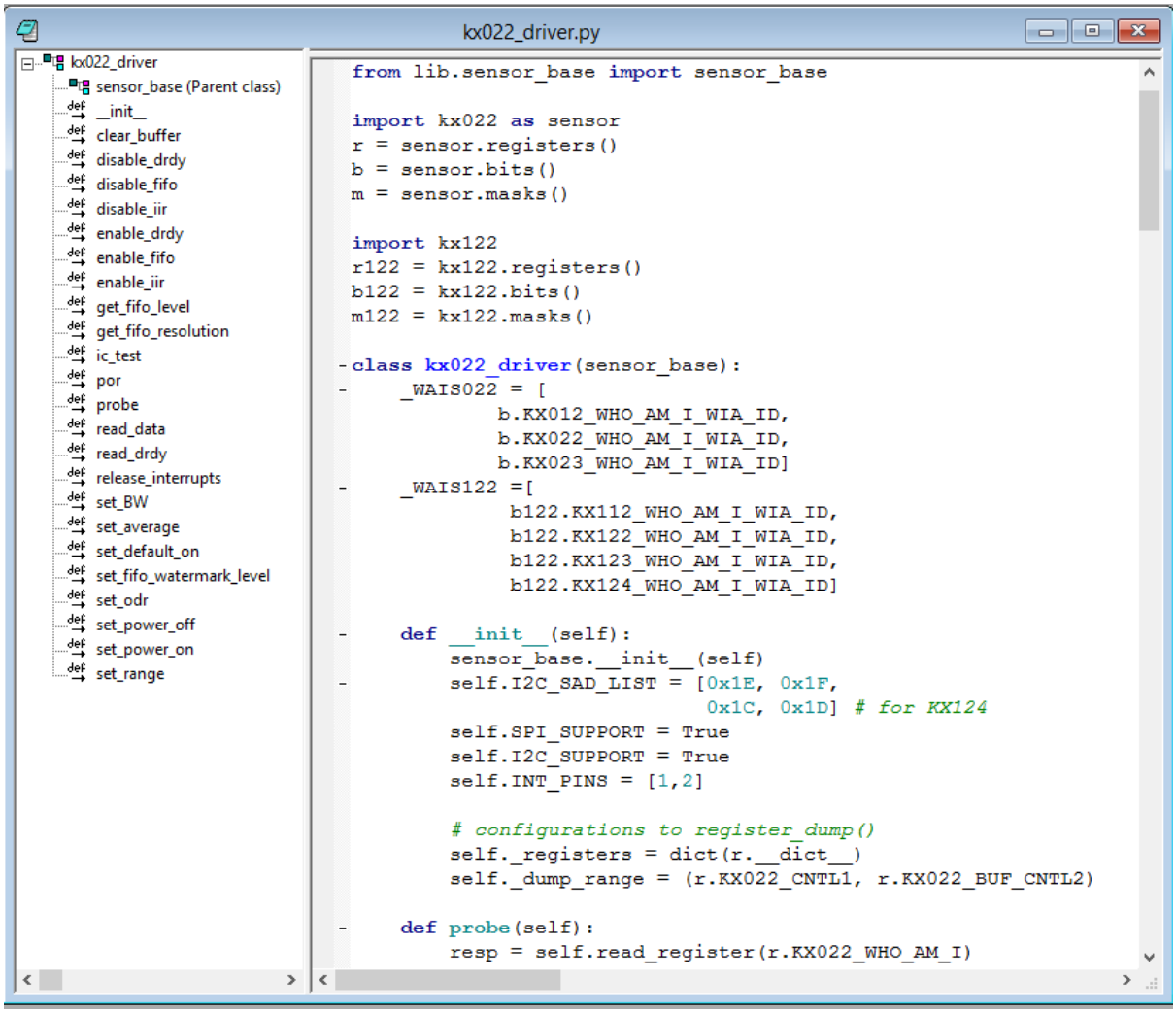
If the needed option is not listed, it can be easily added by referencing the corresponding register definition header file.

6.7. Reference driver implementation

The *Kionix Multi-OS Evaluation Software* offers platform independent reference sensor driver software. Since only the sensor related configurations are in the driver, and the platform specific dependencies are not visible, it gives a good starting point for porting the driver software to a desired target platform.

All drivers follow the same interface defined in `lib\sensor_base.py`





7. Troubleshooting and known issues

7.1. Communications

- USB communication may miss sensor data samples or the USB connection is lost randomly: Use good quality USB cables which are USB certified.

- The *Kionix Evaluation Kit SW* may not connect to the *Kionix IoT Sensor Node* when USB connection is used and the battery is connected to the *Kionix IoT Sensor Node*: the *Kionix IoT Sensor Node* must be powered on before connecting it to the PC with a USB cable. Otherwise the FTDI IC may get powered on slightly before nRF51822 SoC and thus may not detect nRF51822 SoC properly.

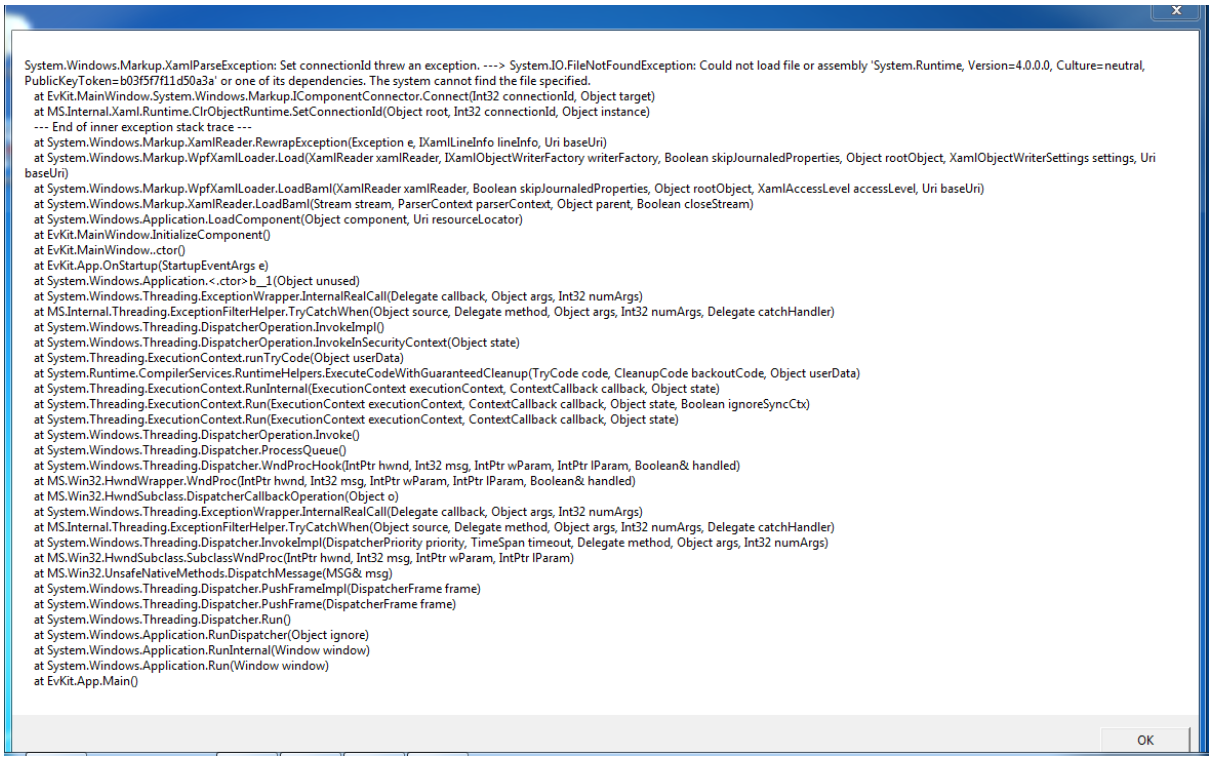
- Windows firewall may block TCP/IP port used by *Kionix BLE Router*.

7.2. *Kionix Windows Sensor Evaluation Software*

- When running the application for the first time Windows may inform that it is a security risk to run an unknown application. This can be solved by selecting *more info / run anyway*.
- When using Windows BLE, ODR can occasionally be slow. Turning off WLAN usually helps to improve the speed. Windows BLE connection speed is highly dependent on your PC load.

- When selecting ANDROID_BLE connection, a pop-up reading "please enable streaming…" is shown even when the *Kionix BLE Router* is not running in the Android device.

- If this kind of an error message appears, the Windows .NET installation is not up to date. Please run Windows update to in order to solve this.



7.3. *Kionix Multi-OS Evaluation Software*

- In case there are issues with *Kionix Multi-OS Evaluation Software* operations, it is possible to change logging level from default INFO for example to DEBUG.

- `logging_level = ERROR ; DEBUG / INFO / WARNING / ERROR / CRITICAL`

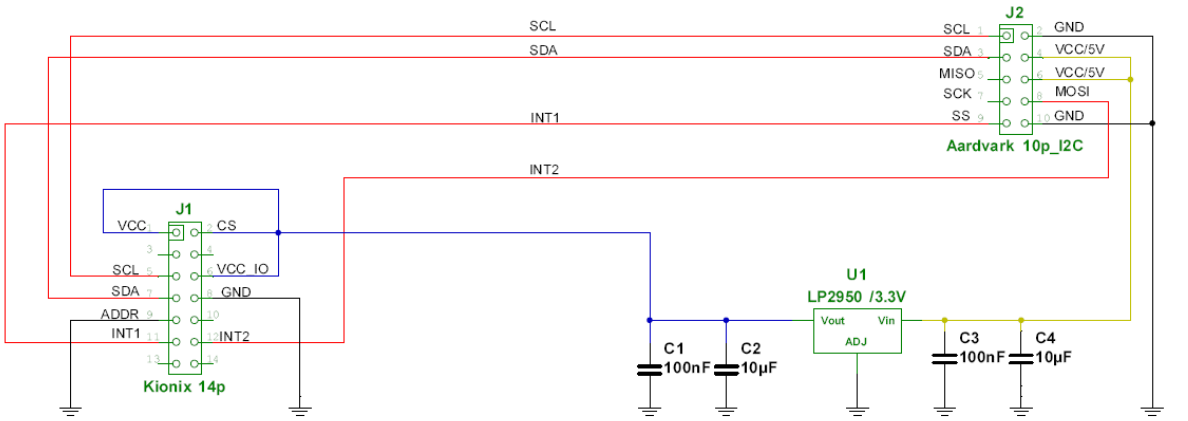
- Additionally logs can be saved to file by defining a file name for the key `log_file`, which is empty (e.g. no logging to file) by default.

- The *Kionix Multi-OS Evaluation Software* verifies that it can capture all interrupts (before starting to monitor interrupt line it is first verified that interrupt is not yet triggered). If `logging_level` is set to `WARNING` or higher then warning printout is given if interrupt speed (for example ODR) is too big.

- If the driver or lib code have been modified and execution fails to import error, delete all *.pyc files.

8. Appendix

8.1. Connecting *Kionix Evaluation Board* to Aardvark I2C

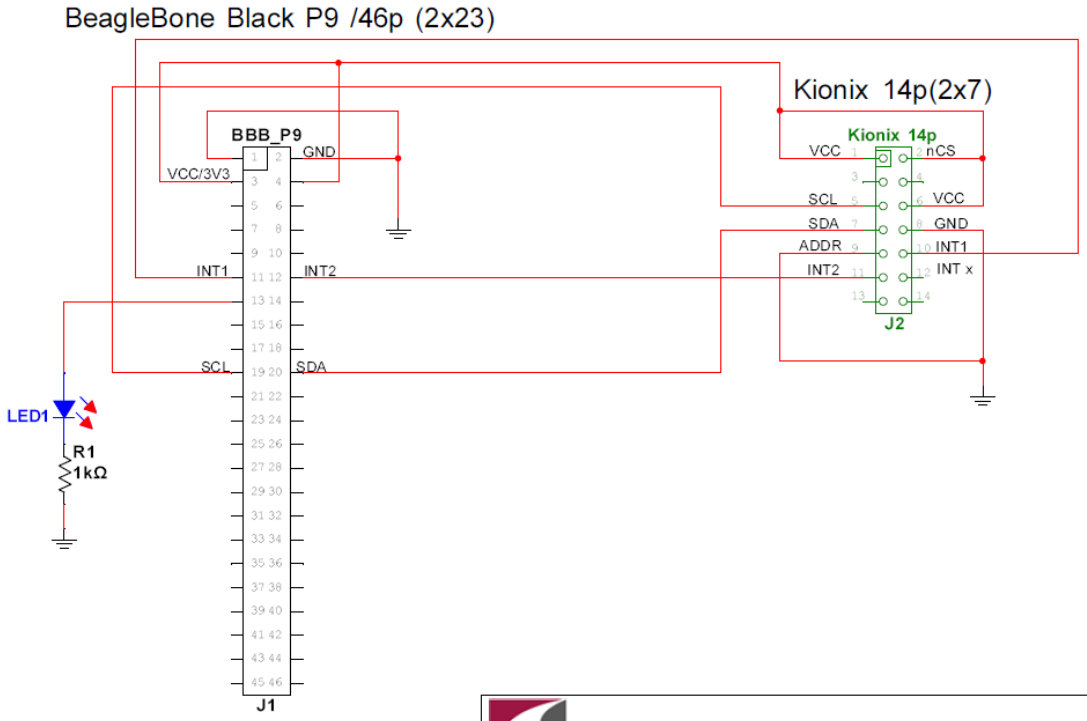


Kionix

Example: setup for KXG03

Title: Simple Aarvark w Kionix	Desc.: Aardvark_I2C_Kionix_S	
Designed by: VKo	Document No:	Revision: A01
Checked by:	Date: 28.4.2016	Size: A4
Approved by:	Sheet 1 of 1	**for I2C bus control**

8.2. Connecting *Kionix Evaluation Board* to Beagle Bone Black



Kionix

Title: BBB_Kionix	Desc.: Kionix14p_to_BBB	
Designed by: VKo	Document No:	Revision:
Checked by:	Date: 4.8.2015	Size: A4
Approved by:	Sheet 1 of 1	