

Introduction

The Kionix Android logger's purpose is to collect sensor data over Bluetooth low energy connection.

Installation A

Download the Android logger alpha version from google play:
<https://play.google.com/apps/testing/com.kionix.datalogger>. Make sure you are a part of alpha testers group. Send request for joining as an alpha tester to jniemi@kionix.com.

Installation B

The Android logger can be found in the <https://bitbucket.org/Kionix-Rohm/stellasw/> repository, under the common/kionix_logger folder. Select the branch "json_kionix_logger". Release packages and notes can be found in the apk-release folder. The ADB tool and the Google USB driver need to be installed before setting up the logger application.

Install USB driver and ADB tool

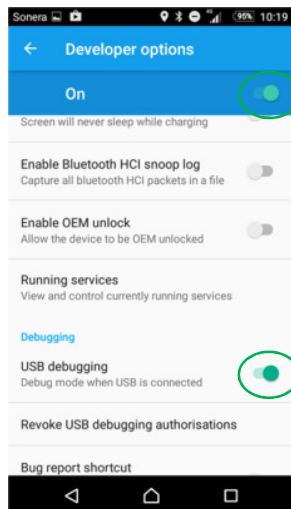
Install [ADB tool](#) to PC

Install [Google USB driver](#) to PC
Connect the Android phone to the PC via USB.

Enable developer mode from Android device

On Android 4.2 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 the **Developer options** at the bottom. On some devices, the Developer options screen may be located or named differently.

To use the adb socket connection, you must enable **USB debugging** in the device system settings, under **Developer options**.



Set adb.exe to system path

Control panel -> system -> advanced system settings:

- Open environmental variables
- Highlight the **Path** variable in the "System variables" section and click the **Edit** button. Add the path line with the paths you want the computer to access.

Test connection

After running the "adb devices" command the device should found

```
adb.exe devices
```

```
List of devices attached  
TA10408MVK      device
```

Set-up Android data logger

The Data logger can be installed using the following command `adb.exe install "apk name"`.

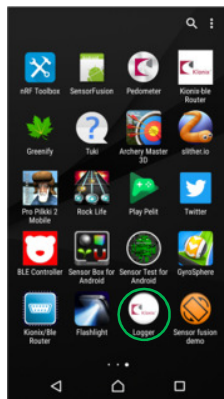
The CSV -log files are written to the phone's internal storage under the "download/KionixLogs" folder.

The Logger configuration files can be found in the phone's internal storage under the "kionix" folder. Files can be manually edited and copied to the kionix folder.

Using Android data logger

Android application

1. Launch the Kionix logger by pressing the logger icon. When the application is started for the first time the configuration files are downloaded from Bitbucket and extracted to the phone's internal storage (to kionix folder).

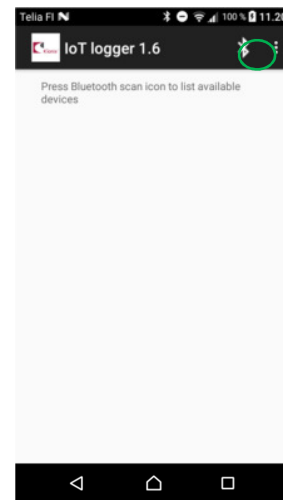


Link to Bit bucket:

<https://bitbucket.org/Kionix-Rohm/slash/get/master.zip>

License agreement, access rights and battery optimization permission are requested from the user and must be accepted to guarantee the correct functionality of the logger software.

2. Press the Bluetooth "scan" icon to detect the Kionix IoT devices. Make sure that the Kionix IoT device is powered on.



3. Kionix IoT device(s) should appear on the list. Select the device from the list to make the BLE connection.



4. When the device is successfully connected; the Bluetooth

connection icon is changes to blue and the list view shows the "connected device" text.

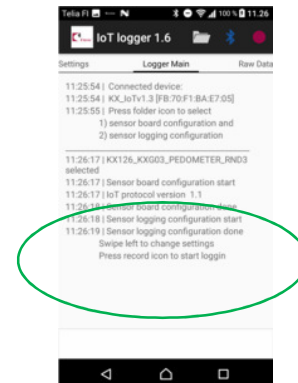
Press the folder icon to select the configuration files to initialize device for logging.



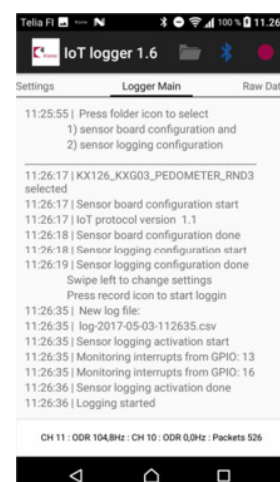
5. Select the proper board and logging configuration files according the IoT node hardware characteristics. Press the "set configure" button to start the IoT board initialize phase.



6. When the initializing is done successfully the "record" icon will be enabled and the logging can be started by pressing the icon.

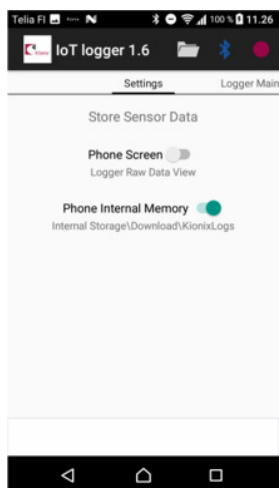


7. Pressing the "record" icon the logging can be either started or stopped. When the icon is pressed for starting, the stream activation messages are sent to the Kionix IoT device and the logging has started. The status of the activation messages can be seen in the list view. The status bar at bottom of the screen indicates; active channel(s) ODR and also the total packets received via BLE. If Stream is not activated the protocol error is seen in the list view.

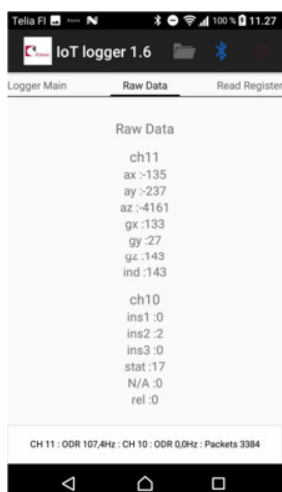


The log file is stored to the Phone's internal storage download/KionixLogs folder.

8. With a swipe the user can navigate through the different views. The main view is the "Logger Main". Swipe to the left from the main view to see the settings view. The raw data view can be enabled by toggling the Phone screen switch. The logging to file can be disabled by toggling the internal memory switch.



9. Swipe right from the "main view" to see the raw data view.



10. Swipe twice to the right from the main view to see the register dump view. The register dump can be set to the JSON stream configuration file(s) from the "register_dump" section.



Download log –files to PC

1. Connect Android phone to the PC via USB cable.
2. Select MTP or USB mass storage mode from the phone for a file transfer.
3. Go to PC file explorer and select your Android device. Browse \internal storage\Download\KionixLogs folder. Copy log(s) file to PC.

Data formats (JSON stream files):

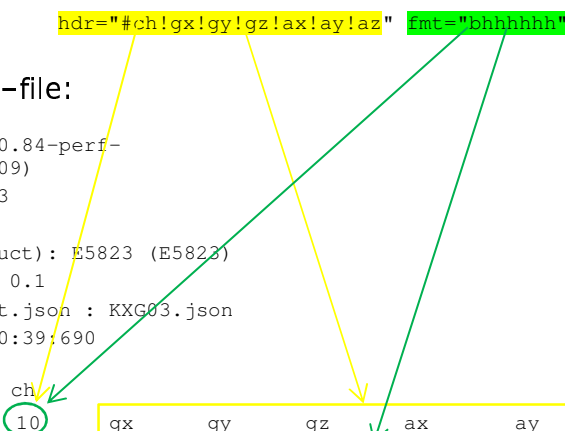
The `<start_req>` tag in the stream JSON file determines the format of the log file. The tag also specifies how to extract data indication messages. The "hdr" attribute determines the log file header part; which can be defined by the user. The channel "ch", is replaced with the indication id which is allocated from the IoT device during a stream activation phase. The rest of the values are split to their own columns for the CSV file.

The "fmt" attribute defines how the stream indication message is extracted. The following example extracts one signed byte (data indication message id) and 6*16bit signed values (sensor data) from the stream. The following format characters are supported by the "fmt" attribute.

Format	C Type	Primitive java type	Standard size
c	char	string of length 1	1
b	signed char	Byte	1
B	unsigned char	Byte	1
h	short	Short (litte endian)	2
H	unsigned short	Integer (litte endian)	4, note: two bytes used!

Start message example (KXG03_ODR_100.json):

Example of log –file:



```
#
# OS Version: 3.10.84-perf-
# g19d6d92 (3574277109)
# OS API Level: 23
# Device: E5823
# Model (and Product): E5823 (E5823)
# Logger version: 0.1
# KX122_board_init.json : KXG03.json
# 2016-11-23 15:10:39:690
#
# timestamp
#
```

	ch	gx	gy	gz	ax	ay	az
0	10	59	-129	-59	13231	779	-9509
0.002638	10	59	-129	-61	13213	804	-9597

0.003319 10 61 -128 -61 13211 819 -9575

In the "fmt" attribute the first "b" represents the stream id which is given by the IoT protocol in the stream allocation phase. The id must be set. The BLE maximum packed size is 20 bytes. 18 bytes are reserved for the payload (the actual sensor data). This is the restriction of the "fmt" attribute.

In the previous example the stream activation message `8,7,16,1,2,79,2,12` is interpreted in following way.

8	7	16	1	2	79	2	12
message length	protocol message id: Stream request	GPIO number to be enabled.	Sense: active low	pin pull: pull down	sensor address: 0x4F	Read start address	Read length

This means that the protocol reads the sensor which slave address is 0x4f. The read address determines the starting point of the read and the length indicates how many bytes will be read over the i2c bus. In this particular example the kxg03 data registers are read.

Set register dump to stream configuration file

1. Create a register dump section if it does not exist. Add it to the same level as the configure, activate and deactivate sections.
2. Set the sensor address, register and length to be read.
3. Set the column names, separated by a comma.
4. See how the data is extracted from the protocol read message.

See pedometer step count example:

```
"register_dump": [
  [
    "KX126",
    14,
    2,
    "PED_STP_L",
    "<H"
  ]
]
```

Yellow: Sensor address, register and the length of the read.

Grey: Column name to see in the register dump view.

Green: How to extract read register data. In pedometer case: 16bit unsigned short.