User manual

Getting started with MotionID motion intensity detection library in X-CUBE-MEMS1 expansion for STM32Cube

Introduction

The MotionID is a middleware library part of X-CUBE-MEMS1 software and runs on STM32. It provides real-time information about the user motion intensity based on data from a device.

It is able to distinguish motion intensity in a range from 0 to 10 (according to the library indexes) corresponding to the following activities: on desk, hand on bed/couch/cushion, light movements, biking, typing/writing, high intensity typing/slow walking, washing hands/walking, fast walking/jogging, running/brushing teeth, sprinting. The library is intended for wrist-based devices.

This library is intended to work with ST MEMS only.

The algorithm is provided in static library format and is designed to be used on STM32 microcontrollers based on the ARM Cortex-M3, ARM Cortex-M4 or ARM Cortex-M0+ architecture.

The algorithm is provided in static library format and is designed to be used on STM32 microcontrollers based on the ARM® Cortex®-M0+, ARM® Cortex®-M3, ARM® Cortex®-M4 or ARM® Cortex®-M7 architecture.

It is built on top of STM32Cube software technology that eases portability across different STM32 microcontrollers.

The software comes with sample implementation running on X-NUCLEO-IKS01A2 or X-NUCLEO-IKS01A3 expansion board on a NUCLEO-F401RE, NUCLEO-L476RG, NUCLEO-L152RE or NUCLEO-L073RZ development board.

1 Acronyms and abbreviations

Table 1. List of acronyms

Acronym	Description
API	Application programming interface
BSP	Board support package
GUI	Graphical user interface
HAL	Hardware abstraction layer
IDE	Integrated development environment

2 MotionID middleware library in X-CUBE-MEMS1 software expansion for STM32Cube

2.1 MotionID overview

The MotionID library expands the functionality of the X-CUBE-MEMS1 software.

The library acquires data from the accelerometer and provides information about the user motion intensity based on data from a device.

The library is designed for ST MEMS only. Functionality and performance when using other MEMS sensors are not analyzed and can be significantly different from what described in the document.

Sample implementation is available for X-NUCLEO-IKS01A2 and X-NUCLEO-IKS01A3 expansion board, mounted on a NUCLEO-F401RE, NUCLEO-L476RG, NUCLEO-L152RE or NUCLEO-L073RZ development board.

2.2 MotionID library

Technical information fully describing the functions and parameters of the MotionID APIs can be found in the MotionID_Package.chm compiled HTML file located in the Documentation folder.

2.2.1 MotionID library description

The MotionID intensity detection library manages the data acquired from the accelerometer; it features:

- possibility to distinguish motion intensity in a range from 0 to 10 (according to the library indexes) corresponding to the following activities: on desk, hand on bed/couch/cushion, light movements, biking, typing/writing, high intensity typing/slow walking, washing hands/walking, fast walking/jogging, running/brushing teeth, sprinting.
- intended for wrist-based devices
- recognition based only on accelerometer data
- required accelerometer data sampling frequency of 16 Hz
- resources requirements:
  – Cortex-M0+: 1.0 kB of code and 0.6 kB of data memory
  – Cortex-M3: 1.1 kB of code and 0.6 kB of data memory
  – Cortex-M4: 1.0 kB of code and 0.6 kB of data memory
  – Cortex-M7: 0.9 kB of code and 0.6 kB of data memory
- available for ARM Cortex-M3, ARM Cortex-M4, ARM Cortex-M0+ or ARM Cortex-M7 architectures

2.2.2 MotionID APIs

The MotionID library APIs are:

- `uint8_t MotionID_GetLibVersion(char *version)`
  – retrieves the library version
  – `*version` is a pointer to an array of 35 characters
  – returns the number of characters in the version string
- `void MotionID_Initialize(void)`
  – performs MotionID library initialization and setup of the internal mechanism
  – the CRC module in STM32 microcontroller (in RCC peripheral clock enable register) has to be enabled before using the library

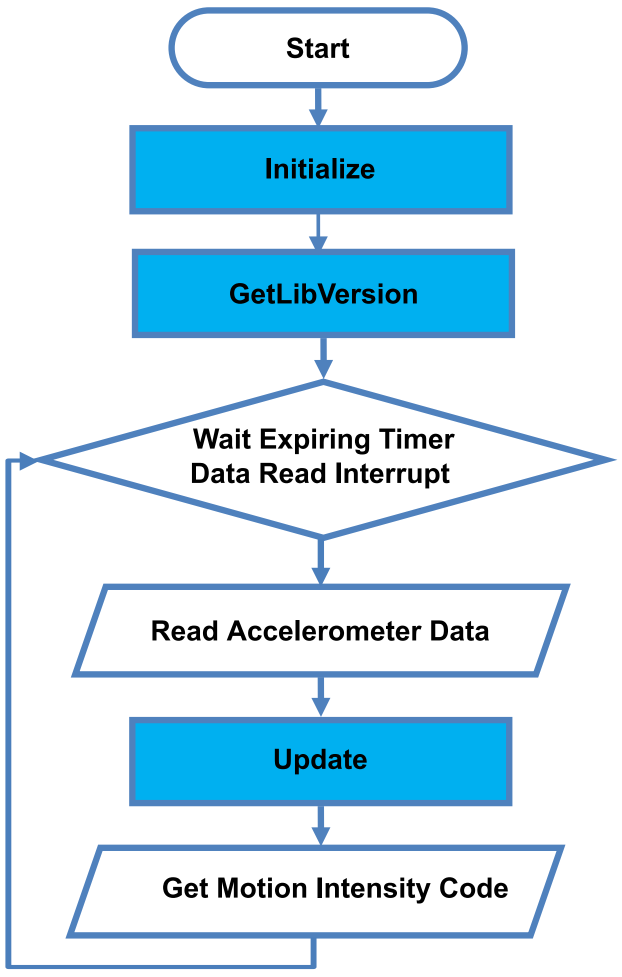
Note: *This function must be called before using the intensity detection library.*

- `void MotionID_ResetLib(void)`
  – resets the library

- `void MotionID_Update(MID_input_t *data_in, MID_output_t *data_out)`
  - executes intensity detection algorithm
  - `*data_in` parameter is a pointer to a structure with input data
  - the parameters for the structure type `MID_input_t` are:
    - `AccX` is the accelerometer sensor value in X axis in g
    - `AccY` is the accelerometer sensor value in Y axis in g
    - `AccZ` is the accelerometer sensor value in Z axis in g
  - `*data_out` parameter is a pointer to an enum with the following with:
    - `MID_ON_DESK = 0`
    - `MID_BED_COUCH_PILLOW = 1`
    - `MID_LIGHT_MOVEMENTS = 2`
    - `MID_BIKING = 3`
    - `MID_TYPING_WRITING = 4`
    - `MID_HI_TYPING__SLOW_WALKING = 5`
    - `MID_WASHING_HANDS_WALKING = 6`
    - `MID_FWALKING = 7`
    - `MID_FWALKING_JOGGING = 8`
    - `MID_FJOGGING_BRUSHING = 9`
    - `MID_SPRINTING = 10`

2.2.3 API flow chart

Figure 1. MotionID API logic sequence



2.2.4 Demo code

The following demonstration code reads data from the accelerometer sensor and gets the motion intensity code.

```
[…]
#define VERSION_STR_LENG 35
[…]

/*** Initialization ***/
char lib_version[VERSION_STR_LENG];

/* Intensity Detection API initialization function */
MotionID_Initialize();

/* Optional: Get version */
MotionID_GetLibVersion(lib_version);

[…]

/*** Using Intensity Detection algorithm ***/
Timer_OR_DataRate_Interrupt_Handler()
{
MID_input_t data_in;
MID_output_t data_out;

/* Get acceleration X/Y/Z in g */
MEMS_Read_AccValue(&data_in.AccX, &data_in.AccY, &data_in.AccZ);

/* Intensity Detection algorithm update */
MotionID_Update(&data_in, &data_out);
}
```

2.2.5 Algorithm performance

The intensity detection algorithm only uses data from the accelerometer and runs at a low frequency (16 Hz) to reduce power consumption.

Algorithm latency is 1 second.

Table 2. **Cortex-M4 and Cortex-M3: elapsed time (µs) algorithm**

Cortex-M4 STM32F401RE at 84 MHz									Cortex-M3 STM32L152RE at 32 MHz								
STM32CubeIDE 1.2.0			IAR EWARM 8.32.3			Keil µVision 5.27			STM32CubeIDE 1.2.0			IAR EWARM 8.32.3			Keil µVision 5.27		
Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
1	2	32	1	2	28	1	2	47	4	31	885	4	25	692	4	32	923

Table 3. **Cortex-M0+ and Cortex-M7: elapsed time (µs) algorithm**

Cortex-M0+ STM32L073RZ at 32 MHz									Cortex-M7 STM32F767ZI at 96 MHz								
STM32CubeIDE 1.2.0			IAR EWARM 8.32.3			Keil µVision 5.27			STM32CubeIDE 1.2.0			IAR EWARM 8.32.3			Keil µVision 5.27		
Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
9	70	1845	9	44	1371	9	75	2755	2	2	37	1	1	22	1	1	141

2.3 Sample application

The MotionID middleware can be easily manipulated to build user applications; a sample application is provided in the Application folder.

It is designed to run on a NUCLEO-F401RE, NUCLEO-L476RG, NUCLEO-L152RE or NUCLEO-L073RZ development board connected to an X-NUCLEO-IKS01A2 or X-NUCLEO-IKS01A3 expansion board.

The application recognizes the user motion intensity in real-time. Data can be displayed through a GUI or stored in the board for offline analysis.

Stand-alone mode

In stand-alone mode, the sample application allows the user to detect motion intensity and store it in the MCU flash memory.

The STM32 Nucleo board may be supplied by a portable battery pack (to make the user experience more comfortable, portable and free of any PC connections).

Table 4. **Power supply scheme**

Power source	JP1 settings	Working mode
USB PC cable	JP1 open	PC GUI driven mode
Battery pack	JP1 closed	Stand-alone mode

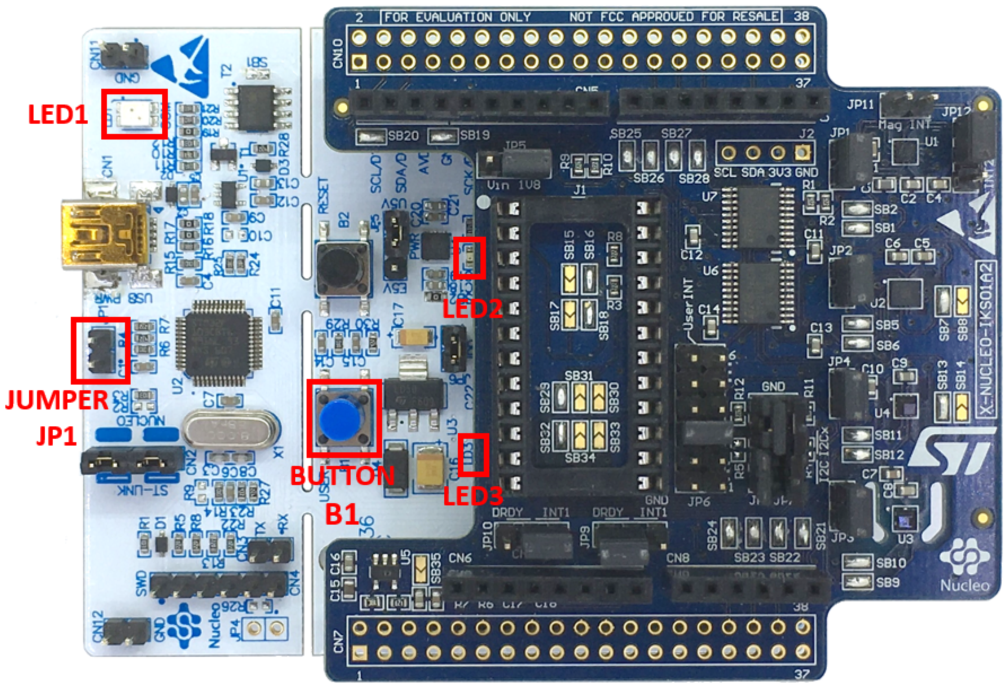


Figure 2. **STM32 Nucleo: LEDs, button, jumper**

The above figure shows the user button B1 and the three LEDs of the NUCLEO-F401RE board. Once the board is powered, LED LD3 (PWR) turns ON and the tricolor LED LD1 (COM) begins blinking slowly due to the missing USB enumeration (refer to UM1724 on www.st.com for further details).

Note: *After powering the board, LED LD2 blinks once indicating the application is ready.*

When the user button B1 is pressed, the system starts acquiring data from the accelerometer sensor and detects the user motion intensity; during this acquisition mode, a fast LED LD2 blinking indicates that the algorithm is running. During this phase, the detected user motion intensity is stored in the MCU internal flash memory. Data are automatically saved every 5 minutes to avoid excessive data loss in case of an unforeseen power fault.

Pressing button B1 a second time stops the algorithm and data storage and LED LD2 switches off.

Pressing the button again starts the algorithm and data storage once again.

The flash sector dedicated to data storage is 128 KB, allowing memorization of more than 16,000 data sets.

To retrieve those data, the board has to be connected to a PC, running Unicleo-GUI. When stored data is retrieved via the GUI, the MCU flash sector dedicated to this purpose is cleared.

If LED LD2 is ON after powering the board, it represents a warning message indicating the flash memory is full.

Note: *Optionally, the MCU memory can be erased by holding the user push button down for at least 5 seconds. LED LD2 switches OFF and then blinks 3 times to indicate that the data stored in the MCU has been erased. This option is available only after power ON or reset of the board while LED LD2 is ON indicating the flash memory is full.*

When the application runs in stand-alone mode and the flash memory is full, the application switches to PC GUI drive mode and LED LD2 switches OFF.

The flash memory must be erased by downloading data via the Unicleo-GUI or the user push button (see the above note).

PC GUI drive mode

In this mode, a USB cable connection is required to monitor real-time data. The board is powered by the PC via USB connection. This working mode allows the user to display detected user motion intensity, accelerometer data, time stamp and eventually other sensor data, in real-time, using the Unicleo-GUI.

In this working mode, data are not stored in the MCU flash memory.

2.4 Unicleo-GUI application

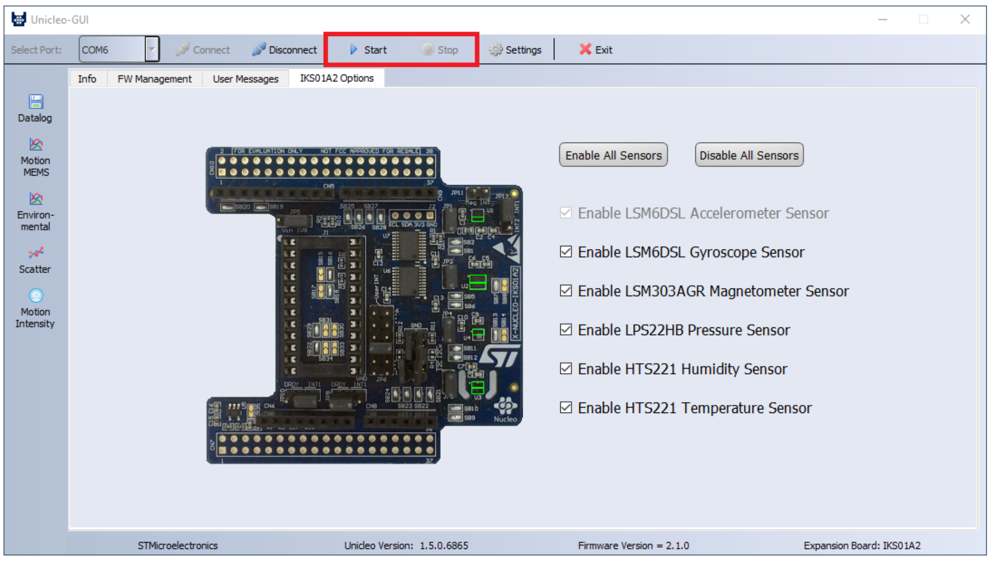
The sample application uses the Windows Unicleo-GUI utility, which can be downloaded from www.st.com.

Step 1. Ensure that the necessary drivers are installed and the STM32 Nucleo board with appropriate expansion board is connected to the PC.

Step 2. Launch the Unicleo-GUI application to open the main application window.

If an STM32 Nucleo board with supported firmware is connected to the PC, it is automatically detected and the appropriate COM port is opened.

Figure 3. Unicleo main window

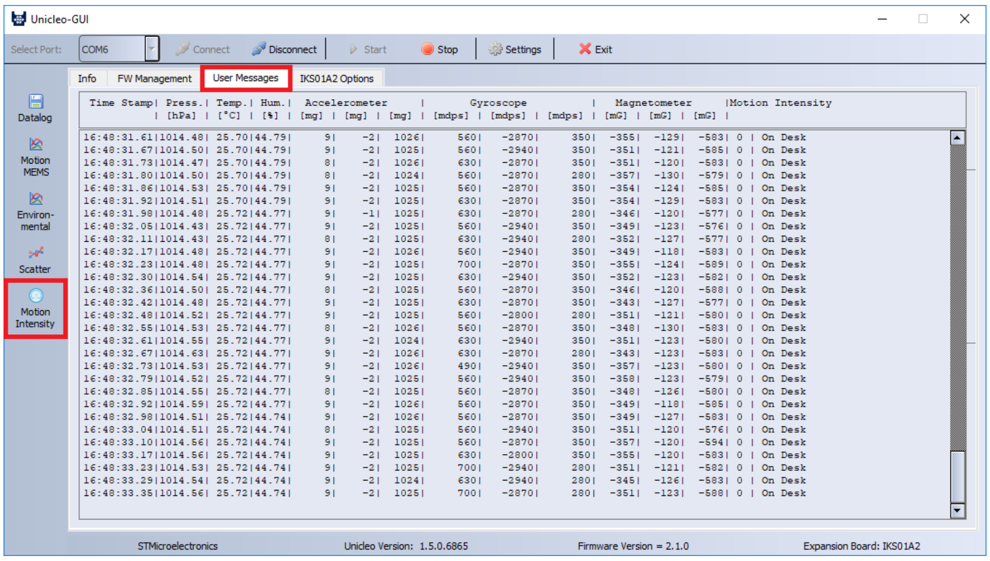


☑ Enable LSM6DSL Accelerometer Sensor
☑ Enable LSM6DSL Gyroscope Sensor
☑ Enable LSM303AGR Magnetometer Sensor
☑ Enable LPS22HB Pressure Sensor
☑ Enable HTS221 Humidity Sensor
☑ Enable HTS221 Temperature Sensor

STMicroelectronics   Unicleo Version: 1.5.0.6865   Firmware Version = 2.1.0   Expansion Board: IKS01A2

Step 3. Start and stop data streaming by using the appropriate buttons on the vertical tool bar.
The data coming from the connected sensor can be viewed in the User Messages tab.

Figure 4. User Messages tab

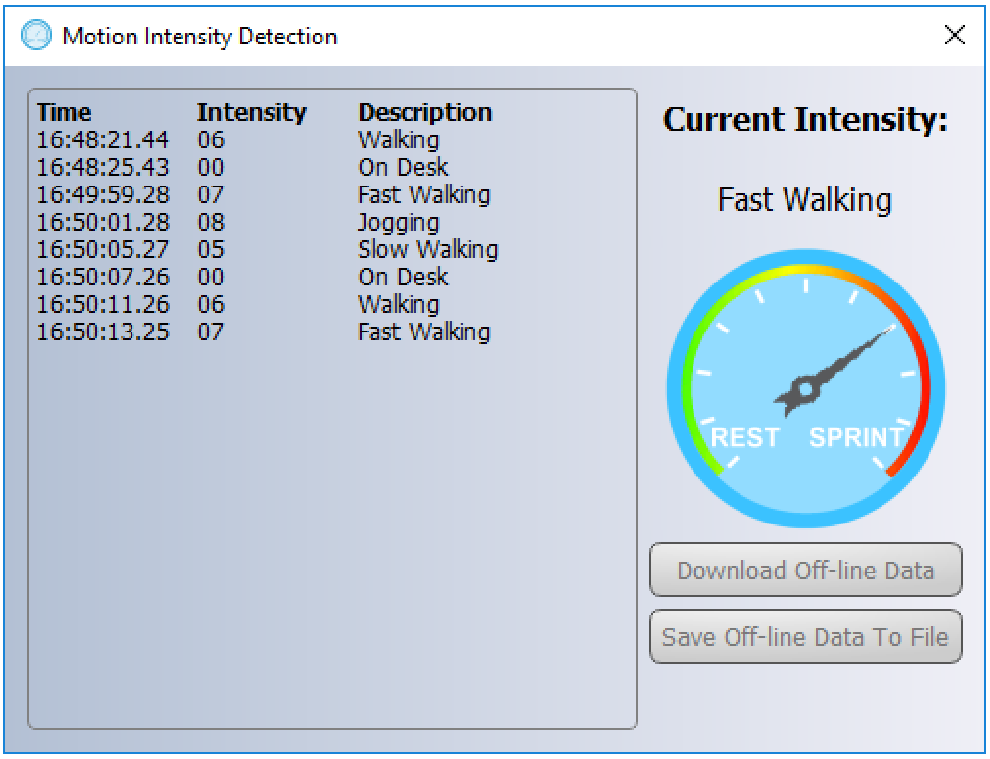


```
 Time Stamp| Press.| Temp.| Hum.|  Accelerometer     |      Gyroscope         |   Magnetometer     |Motion Intensity
           | [hPa] | [°C] | [%] | [mg] | [mg] | [mg] | [mdps] | [mdps] | [mdps] | [mG] | [mG] | [mG] |
16:48:31.61|1014.48| 25.70|44.79|     9|    -2|  1026|     560|   -2870|     350|  -355|  -129|  -583| 0 | On Desk
16:48:31.67|1014.50| 25.70|44.79|     9|    -2|  1025|     560|   -2940|     350|  -351|  -121|  -585| 0 | On Desk
16:48:31.73|1014.47| 25.70|44.79|     8|    -2|  1026|     630|   -2870|     350|  -351|  -120|  -583| 0 | On Desk
16:48:31.80|1014.50| 25.70|44.79|     8|    -2|  1024|     560|   -2870|     280|  -357|  -130|  -579| 0 | On Desk
16:48:31.86|1014.53| 25.70|44.79|     9|    -2|  1025|     560|   -2870|     350|  -354|  -124|  -585| 0 | On Desk
16:48:31.92|1014.51| 25.70|44.79|     9|    -2|  1025|     630|   -2870|     350|  -354|  -129|  -583| 0 | On Desk
16:48:31.98|1014.48| 25.72|44.77|     9|    -1|  1025|     630|   -2870|     280|  -346|  -120|  -577| 0 | On Desk
16:48:32.05|1014.43| 25.72|44.77|     9|    -2|  1025|     560|   -2940|     350|  -349|  -123|  -576| 0 | On Desk
16:48:32.11|1014.43| 25.72|44.77|     8|    -2|  1025|     630|   -2940|     280|  -352|  -127|  -577| 0 | On Desk
16:48:32.17|1014.48| 25.72|44.77|     9|    -2|  1026|     560|   -2940|     350|  -349|  -118|  -583| 0 | On Desk
16:48:32.23|1014.48| 25.72|44.77|     9|    -2|  1025|     700|   -2870|     350|  -355|  -124|  -589| 0 | On Desk
16:48:32.30|1014.54| 25.72|44.77|     9|    -2|  1025|     630|   -2940|     350|  -352|  -123|  -582| 0 | On Desk
16:48:32.36|1014.50| 25.72|44.77|     8|    -2|  1025|     560|   -2870|     350|  -346|  -120|  -588| 0 | On Desk
16:48:32.42|1014.48| 25.72|44.77|     9|    -2|  1025|     630|   -2870|     350|  -343|  -127|  -577| 0 | On Desk
16:48:32.48|1014.52| 25.72|44.77|     9|    -2|  1025|     560|   -2800|     280|  -351|  -121|  -580| 0 | On Desk
16:48:32.55|1014.53| 25.72|44.77|     8|    -2|  1026|     560|   -2870|     350|  -348|  -130|  -583| 0 | On Desk
16:48:32.61|1014.55| 25.72|44.77|     9|    -2|  1024|     630|   -2940|     350|  -351|  -123|  -580| 0 | On Desk
16:48:32.67|1014.63| 25.72|44.77|     9|    -2|  1026|     630|   -2870|     280|  -343|  -123|  -583| 0 | On Desk
16:48:32.73|1014.53| 25.72|44.77|     9|    -2|  1026|     490|   -2940|     350|  -357|  -123|  -580| 0 | On Desk
16:48:32.79|1014.52| 25.72|44.77|     9|    -2|  1025|     560|   -2940|     350|  -358|  -123|  -579| 0 | On Desk
16:48:32.85|1014.55| 25.72|44.77|     8|    -2|  1025|     560|   -2870|     350|  -348|  -126|  -580| 0 | On Desk
16:48:32.92|1014.59| 25.72|44.77|     9|    -2|  1026|     560|   -2870|     350|  -349|  -118|  -585| 0 | On Desk
16:48:32.98|1014.51| 25.72|44.74|     9|    -2|  1026|     560|   -2870|     350|  -349|  -127|  -583| 0 | On Desk
16:48:33.04|1014.51| 25.72|44.74|     8|    -2|  1025|     560|   -2940|     350|  -351|  -120|  -576| 0 | On Desk
16:48:33.10|1014.56| 25.72|44.74|     9|    -2|  1025|     560|   -2870|     350|  -357|  -120|  -594| 0 | On Desk
16:48:33.17|1014.56| 25.72|44.74|     9|    -2|  1025|     630|   -2800|     350|  -355|  -120|  -583| 0 | On Desk
16:48:33.23|1014.53| 25.72|44.74|     9|    -2|  1025|     700|   -2940|     280|  -351|  -121|  -582| 0 | On Desk
16:48:33.29|1014.54| 25.72|44.74|     9|    -2|  1024|     630|   -2940|     280|  -345|  -126|  -583| 0 | On Desk
16:48:33.35|1014.56| 25.72|44.74|     9|    -2|  1025|     700|   -2870|     280|  -351|  -123|  -588| 0 | On Desk
```

STMicroelectronics   Unicleo Version: 1.5.0.6865   Firmware Version = 2.1.0   Expansion Board: IKS01A2

Step 4. Click on the Motion Intensity icon in the vertical tool bar to open the dedicated application window.

Figure 5. Motion intensity detection window



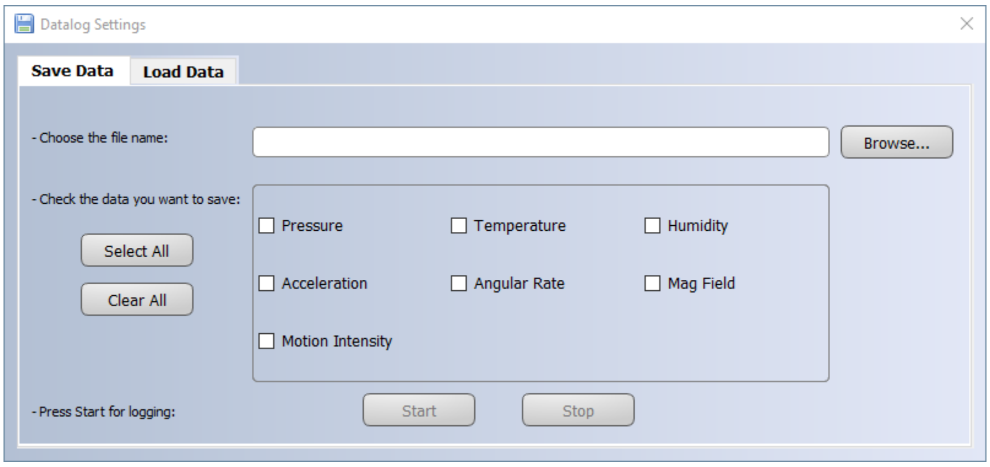
Time	Intensity	Description
16:48:21.44	06	Walking
16:48:25.43	00	On Desk
16:49:59.28	07	Fast Walking
16:50:01.28	08	Jogging
16:50:05.27	05	Slow Walking
16:50:07.26	00	On Desk
16:50:11.26	06	Walking
16:50:13.25	07	Fast Walking

If the board has been working in stand-alone mode and the user wants to retrieve stored data, press Download Off-line Data button to upload the stored activities data to the application. This operation automatically deletes acquired data from microcontroller.

Press the Save Off-line Data to File button to save the uploaded data in a .tsv file.

Step 5. Click on the Datalog icon in the vertical tool bar to open the datalog configuration window: you can select which sensor and activity data to save in files. You can start or stop saving by clicking on the corresponding button.

Figure 6. Datalog window



3 References

All of the following resources are freely available on www.st.com.

1. UM1859: Getting started with the X-CUBE-MEMS1 motion MEMS and environmental sensor software expansion for STM32Cube
2. UM1724: STM32 Nucleo-64 board
3. UM2128: Getting started with Unicleo-GUI for motion MEMS and environmental sensor software expansion for STM32Cube

Revision history

Table 5. **Document revision history**

Date	Version	Changes
10-May-2017	1	Initial release.
26-Jan-2018	2	Added references to NUCLEO-L152RE development board and Table 2. Elapsed time (µs) algorithm.
20-Mar-2018	3	Updated Introduction and Section 2.1 MotionID overview.
21-Nov-2018	4	Updated Figure 2. STM32 Nucleo: LEDs, button, jumper. Added Table 3. Cortex-M0+: elapsed time (µs) algorithm and references to ARM® Cortex®-M0+ and NUCLEO-L073RZ development board.
20-Feb-2019	5	Updated Table 2. Cortex-M4 and Cortex-M3: elapsed time (µs) algorithm, Table 3. Cortex-M0+: elapsed time (µs) algorithm, Figure 3. Unicleo main window, Figure 4. User Messages tab, Figure 5. Motion intensity detection window and Figure 6. Datalog window.
25-Mar-2020	6	Updated Introduction, Section 2.2.1 MotionID library description and Section 2.2.5 Algorithm performance. Added ARM Cortex-M7 architecture compatibility information.

Contents

List of tables

List of figures