UM2194

User manual

Getting started with MotionAW activity recognition for wrist library in X-CUBE-MEMS1 expansion for STM32Cube

Introduction

The MotionAW is a middleware library part of X-CUBE-MEMS1 software and runs on STM32. It provides real-time information on the type of activity performed by the user.

It is able to distinguish the following activities: stationary, standing, sitting, lying, walking, fast walking, jogging, biking. The library is intended for wrist-worn 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 or Cortex-M4 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 an X-NUCLEO-IKS01A2 or X-NUCLEO-IKS01A3 expansion board on a NUCLEO-F401RE, NUCLEO-L476RG or NUCLEO-L152RE 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 MotionAW middleware library in X-CUBE-MEMS1 software expansion

2.1 MotionAW overview

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

The library acquires data from the accelerometer and provides information on the type of activity performed by the user.

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.

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

2.2 MotionAW library

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

2.2.1 MotionAW library description

The MotionAW real-time activity recognition library manages the data acquired from the accelerometer; it:

- can distinguish the following activities: stationary, standing, sitting, lying, walking, fast walking, jogging, biking
- is intended for wrist-worn devices
- is based only on accelerometer data
- requires accelerometer data sampling frequency of 16 Hz
- resources requirements:
  - Cortex-M3: 13.0 kB of code and 3.1 kB of data memory
  - Cortex-M4: 13.4 kB of code and 3.1 kB of data memory
- available for ARM Cortex-M3 and Cortex-M4 architectures

2.2.2 MotionAW APIs

The MotionAW library APIs are:

- `uint8_t MotionAW_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 MotionAW_Initialize(void)`
  - performs MotionAW 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 accelerometer calibration library
- `void MotionAW_Reset(void)`
  - resets activity recognition algorithms
- `void MotionAW_Update (MAW_input_t *data_in, MAW_output_t *data_out)`
  - executes activity recognition for wrist algorithm
  - `*data_in` parameter is a pointer to a structure with input data
  - the parameters for the structure type `MAW_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 a structure with output data
  - the parameters for the structure type `MAW_output_t` are:
    - `CurrectActivity` contains the following items
      - `MAW_NOACTIVITY = 0`
      - `MAW_STATIONARY = 1`
      - `MAW_STANDING = 2`
      - `MAW_SITTING = 3`
      - `MAW_LYING = 4`
      - `MAW_WALKING = 5`
      - `MAW_FASTWALKING = 6`
      - `MAW_JOGGING = 7`
      - `MAW_BIKING = 8`
    - `Confidence` is the confidence of the current activity from 0 to 5
    - `CurrentActivityDuration` is the duration in seconds for the current ongoing activity since the last activity change
    - `ActivityTotalDuration` is an array of the total duration in seconds for each activity since the last reset

- `void MotionAW_Reset_Activity_Duration(void)`
  - resets total activity duration counters

- `void MotionAW_SetOrientation_Acc (const char *acc_orientation)`
  - this function is used to set the accelerometer data orientation
  - configuration is usually performed immediately after the `MotionAW_Initialize` function call
  - `*acc_orientation` parameter is a pointer to a string of three characters indicating the direction of each of the positive orientations of the reference frame used for accelerometer data output, in the sequence x, y, z. Valid values are: n (north) or s (south), w (west) or e (east), u (up) or d (down).
  - As shown in the figure below, the X-NUCLEO-IKS01A2 accelerometer sensor has an NWU orientation (x-North, y-West, z-Up), so the string is: “nwu”.

Figure 1. Example of sensor orientations

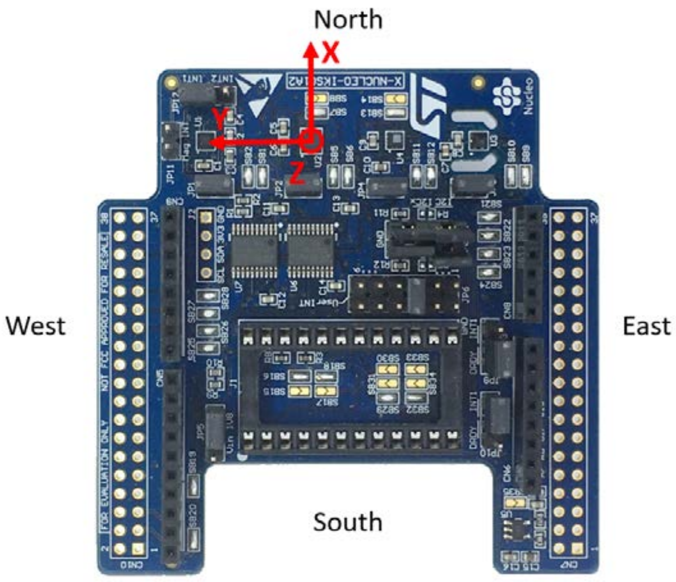
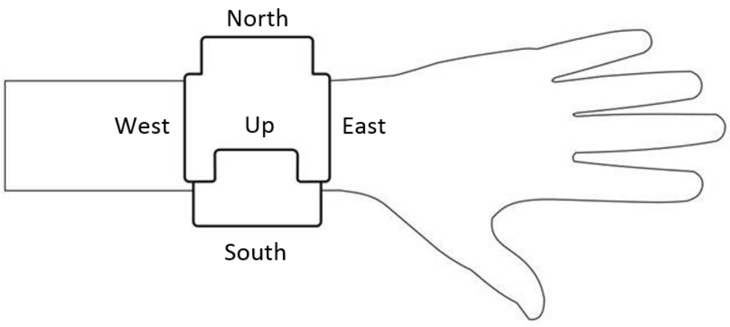
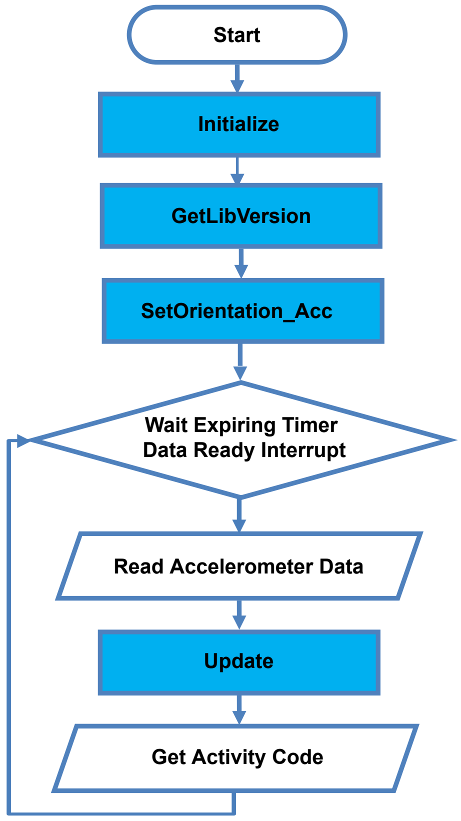


Figure 2. Orientation system for wrist-worn devices



2.2.3 API flow chart

Figure 3. MotionAW API logic sequence



2.2.4 Demo code

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

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

/*** Initialization ***/
char lib_version[VERSION_STR_LENG];
char acc_orientation[3];

/* Activity recognition API initialization function */
MotionAW_Initialize();

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

/* Set accelerometer orientation */
acc_orientation[0] ='n';
acc_orientation[1] ='w';
acc_orientation[2] ='u';
MotionAW_SetOrientation_Acc(acc_orientation);

[…]
```

```
/*** Using activity recognition algorithm ***/
Timer_OR_DataRate_Interrupt_Handler()
{
MAW_input_t data_in;
MAW_output_t activity;

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

/* Activity recognition algorithm update */
MotionAW_Update(&data_in, &activity);
}
```

2.2.5 Algorithm performance

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

Table 2. Algorithm performance

Activity	Detection probability (typical)[1]	Minimum Latency	Typical Latency
Stationary	98.48%	3 s	5 s
• Sitting	83.30%	15 s	30 s
• Standing	75.05%	15 s	30 s
• Lying	89.73%	3 min	-
Walking	93.22%	5 s	7 s
Fast walking	85.92%	5 s	7 s
Jogging	98.30%	3 s	6 s
Biking	89.35%	10 s	20 s

1. *Typical specifications are not guaranteed*

Table 3. Elapsed time (µs) algorithm

Cortex-M4 STM32F401RE at 84 MHz									Cortex-M3 STM32L152RE at 32 MHz								
SW4STM32 2.6.0 (GCC 7.2.1)			IAR EWARM 7.80.4			Keil µVision 5.24			SW4STM32 2.6.0 (GCC 7.2.1)			IAR EWARM 7.80.4			Keil µVision 5.24		
Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
6	18	286	10	17	241	10	21	384	98	345	7817	117	287	5736	117	282	5384

2.2.6 Sample application

The MotionAW 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 or NUCLEO-L152RE development board connected to an X-NUCLEO-IKS01A2 or X-NUCLEO-IKS01A3.

The application recognizes performed activities in real-time. Data can be displayed through a GUI or stored in the board for offline analysis. The algorithm recognizes stationary, walking, fast walking, jogging and bike riding.

Stand-alone mode

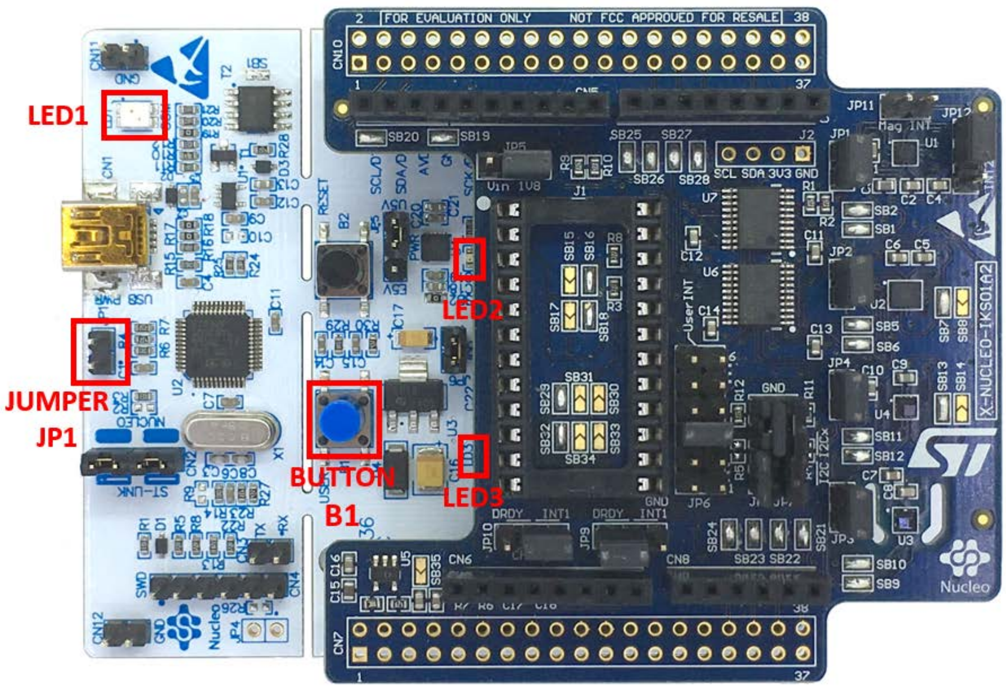
In stand-alone mode, the sample application allows the user to detect performed gesture and store it in the MCU flash memory.

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). The user can choose to display latest recognized activity through an on board LED. During acquisition and algorithm operation, recognized activities are stored in the microcontroller memory.

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 4. **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 (power) 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).

When the user button B1 is pressed, the system starts acquiring data from the accelerometer sensor and detects the performed activity; during this acquisition mode, a fast LED LD2 blinking indicates that the algorithm is running. During this phase the detected device gesture is stored in the MCU internal flash memory.

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 the LED LD2 is ON after powering the board, it represents a warning message indicating that 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 have been erased. This option is available only after power ON or board reset 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 (as mentioned in 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 the activity detected, 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.

2.2.6.1 *Data storage*

The sample application allows the user to detect the activity performed and store it in the MCU flash memory. Data are automatically saved every 5 minutes to avoid excessive loss in case of an unforeseen power fault. Data are also stored when the user stops acquisition by pressing button B1 to display data by LED.

When stored data are retrieved via the GUI, the dedicated MCU flash sector is cleared.

LED LD2 should be OFF at power-on, unless the flash memory is full or almost full. If the flash memory is full, the user can continue using the board for normal activity recognition and data storage (if there is enough available space) by pushing the user button (LED switches off in 5 seconds). Data are stored until the reserved sector is full but the algorithm keeps running.

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

- a minimum of 91 hours (activity changes and a new buffer is stored every 20 seconds)
- a maximum of 1365 hours (activity never changes and data are automatically stored every 5 minutes).

If LED LD2 switches ON at reset, no more than 1400 bytes are free (from one hour recording time if data are stored every 20 seconds to 15 hours if data are automatically saved every five minutes).

If a large amount of data needs to be stored and no PC is available for data download and flash memory clean up, the MCU memory can be erased when LED LD2 is ON, by holding the user push button down for at least 5 seconds. LED LD2 switches OFF and then starts blinking to indicate that acquisition mode has been activated and the activity data stored in the MCU have been erased.

2.2.7 Unicleo-GUI application

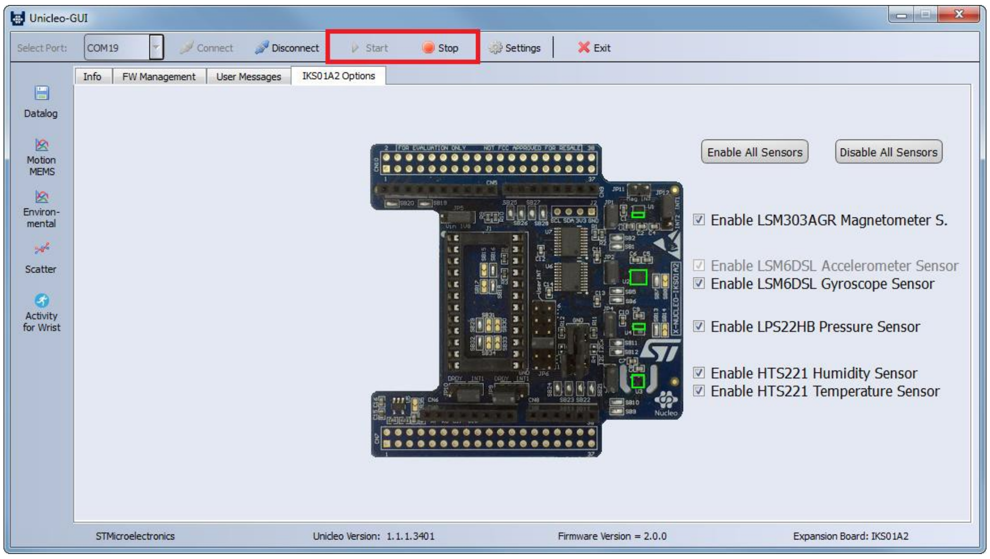
The sample application uses the Windows Unicleo-GUI utility, which can be downloaded from www.st.com.

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.

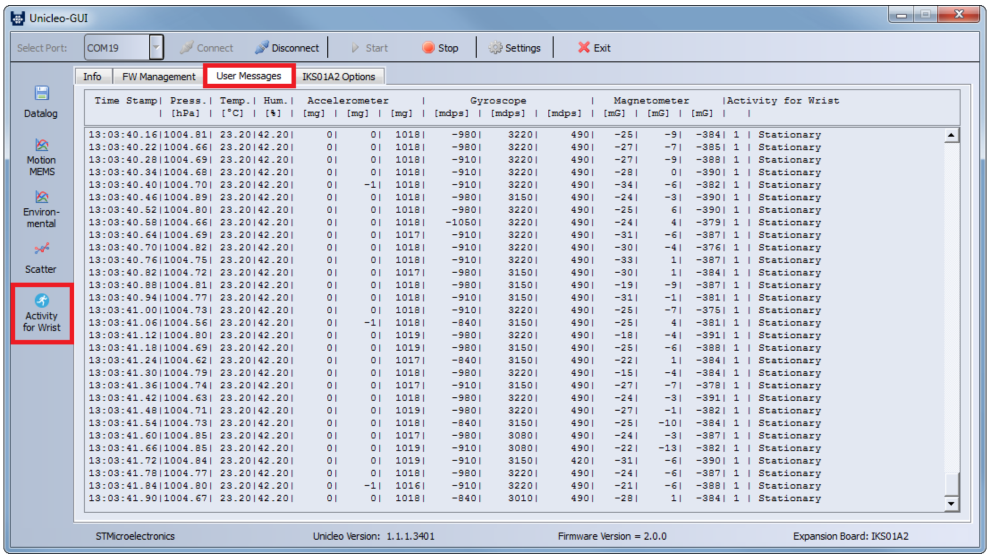
Figure 5. **Unicleo main window**



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 6. **User Messages tab**



```
 Time Stamp| Press.| Temp.| Hum.| Accelerometer       |       Gyroscope          |  Magnetometer    |Activity for Wrist
           | [hPa] | [°C] | [%] | [mg] | [mg] | [mg] | [mdps] | [mdps] | [mdps] | [mG] | [mG] | [mG] |    |
13:03:40.16|1004.81| 23.20|42.20|     0|     0|  1018|    -980|    3220|     490|   -25|    -9|  -384| 1 | Stationary
13:03:40.22|1004.66| 23.20|42.20|     0|     0|  1018|    -980|    3220|     490|   -27|    -7|  -385| 1 | Stationary
13:03:40.28|1004.69| 23.20|42.20|     0|     0|  1018|    -910|    3220|     490|   -27|    -9|  -388| 1 | Stationary
13:03:40.34|1004.68| 23.20|42.20|     0|     0|  1018|    -910|    3220|     490|   -28|     0|  -390| 1 | Stationary
13:03:40.40|1004.70| 23.20|42.20|     0|    -1|  1018|    -910|    3220|     490|   -34|    -6|  -382| 1 | Stationary
13:03:40.46|1004.89| 23.20|42.20|     0|     0|  1018|    -980|    3150|     490|   -24|    -3|  -390| 1 | Stationary
13:03:40.52|1004.80| 23.20|42.20|     0|     0|  1018|    -980|    3220|     490|   -25|     6|  -390| 1 | Stationary
13:03:40.58|1004.66| 23.20|42.20|     0|     0|  1018|   -1050|    3220|     490|   -24|     4|  -379| 1 | Stationary
13:03:40.64|1004.69| 23.20|42.20|     0|     0|  1017|    -910|    3220|     490|   -31|    -6|  -387| 1 | Stationary
13:03:40.70|1004.82| 23.20|42.20|     0|     0|  1018|    -910|    3220|     490|   -30|    -4|  -376| 1 | Stationary
13:03:40.76|1004.75| 23.20|42.20|     0|     0|  1018|    -910|    3220|     490|   -33|     1|  -387| 1 | Stationary
13:03:40.82|1004.72| 23.20|42.20|     0|     0|  1017|    -980|    3150|     490|   -30|     1|  -384| 1 | Stationary
13:03:40.88|1004.81| 23.20|42.20|     0|     0|  1018|    -980|    3150|     490|   -19|    -9|  -387| 1 | Stationary
13:03:40.94|1004.77| 23.20|42.20|     0|     0|  1018|    -910|    3150|     490|   -31|    -1|  -381| 1 | Stationary
13:03:41.00|1004.73| 23.20|42.20|     0|     0|  1018|    -910|    3220|     490|   -25|    -7|  -375| 1 | Stationary
13:03:41.06|1004.56| 23.20|42.20|     0|    -1|  1018|    -840|    3150|     490|   -25|     4|  -381| 1 | Stationary
13:03:41.12|1004.80| 23.20|42.20|     0|     0|  1019|    -980|    3220|     490|   -18|    -4|  -391| 1 | Stationary
13:03:41.18|1004.69| 23.20|42.20|     0|     0|  1019|    -980|    3150|     490|   -25|    -6|  -388| 1 | Stationary
13:03:41.24|1004.62| 23.20|42.20|     0|     0|  1017|    -840|    3150|     490|   -22|     1|  -384| 1 | Stationary
13:03:41.30|1004.79| 23.20|42.20|     0|     0|  1018|    -980|    3220|     490|   -15|    -4|  -384| 1 | Stationary
13:03:41.36|1004.74| 23.20|42.20|     0|     0|  1017|    -910|    3150|     490|   -27|    -7|  -378| 1 | Stationary
13:03:41.42|1004.63| 23.20|42.20|     0|     0|  1018|    -980|    3220|     490|   -24|    -3|  -391| 1 | Stationary
13:03:41.48|1004.71| 23.20|42.20|     0|     0|  1019|    -980|    3220|     490|   -27|    -1|  -382| 1 | Stationary
13:03:41.54|1004.73| 23.20|42.20|     0|     0|  1018|    -840|    3150|     490|   -25|   -10|  -384| 1 | Stationary
13:03:41.60|1004.85| 23.20|42.20|     0|     0|  1017|    -980|    3080|     490|   -24|    -3|  -387| 1 | Stationary
13:03:41.66|1004.85| 23.20|42.20|     0|     0|  1019|    -910|    3080|     490|   -22|   -13|  -382| 1 | Stationary
13:03:41.72|1004.84| 23.20|42.20|     0|     0|  1019|    -910|    3150|     420|   -31|    -6|  -390| 1 | Stationary
13:03:41.78|1004.77| 23.20|42.20|     0|     0|  1018|    -980|    3220|     490|   -24|    -6|  -387| 1 | Stationary
13:03:41.84|1004.80| 23.20|42.20|     0|    -1|  1016|    -910|    3220|     490|   -21|    -6|  -388| 1 | Stationary
13:03:41.90|1004.67| 23.20|42.20|     0|     0|  1018|    -840|    3010|     490|   -28|     1|  -384| 1 | Stationary
```

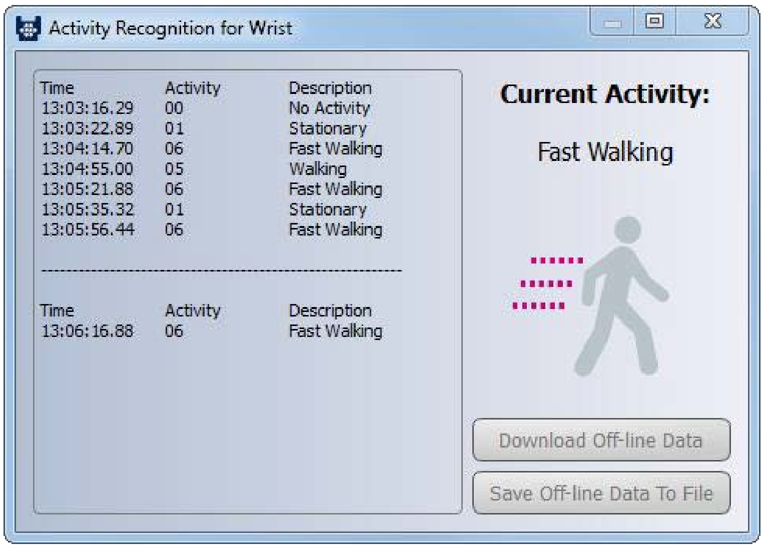
Step 4. Click on the Activity icon in the vertical tool bar to open the dedicated application window.

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.

Note: *Activities with a duration of less than 20 seconds are not memorized.*

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

Figure 7. **Activity recognition window**



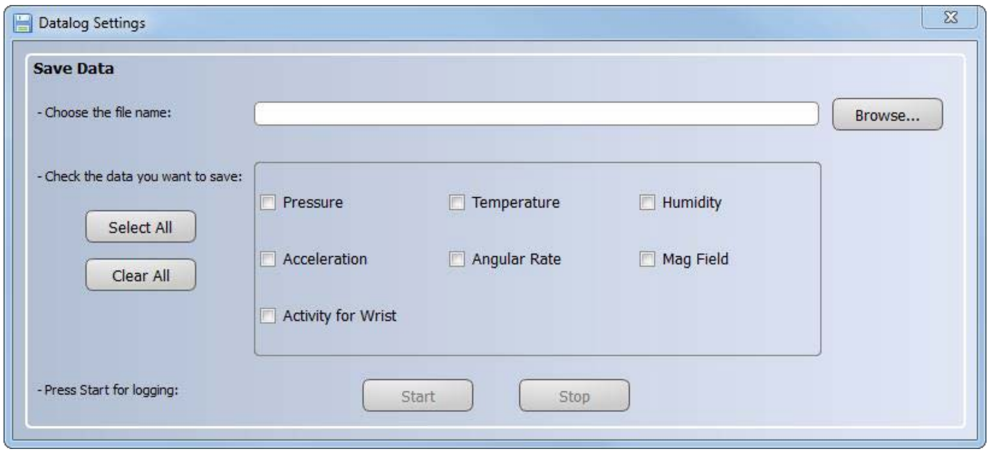
```
Time            Activity        Description
13:03:16.29     00              No Activity
13:03:22.89     01              Stationary
13:04:14.70     06              Fast Walking
13:04:55.00     05              Walking
13:05:21.88     06              Fast Walking
13:05:35.32     01              Stationary
13:05:56.44     06              Fast Walking

----------------------------------------------------------------

Time            Activity        Description
13:06:16.88     06              Fast Walking
```

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 8. **Datalog window**



Save Data

- Choose the file name:

- Check the data you want to save:

☐ Pressure	☐ Temperature	☐ Humidity
☐ Acceleration	☐ Angular Rate	☐ Mag Field
☐ Activity for Wrist		

- Press Start for logging:

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-Apr-2017	1	Initial release.
26-Jan-2018	2	Updated Section 2.3 Sample application. Added references to NUCLEO-L152RE development board, Figure 2. Orientation system for wrist-worn devices and Table 3. Elapsed time (µs) algorithm.
20-Mar-2018	3	Updated Introduction, Section 2.1 MotionAW overview and Section 2.2.5 Algorithm performance.
14-Feb-2019	4	Updated Figure 1. Example of sensor orientations, Table 3. Elapsed time (µs) algorithm and Figure 4. STM32 Nucleo: LEDs, button, jumper. Added X-NUCLEO-IKS01A3 expansion board compatibility information.

Contents

List of tables

List of figures

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2019 STMicroelectronics – All rights reserved