UM2275

User manual

Getting started with MotionFD real-time fall detection library in X-CUBE-MEMS1 expansion for STM32Cube

Introduction

The MotionFD is a middleware library part of X-CUBE-MEMS1 software and runs on STM32. It provides real-time information about the user fall event based on data from a device. It is able to distinguish whether the user fall occurred or not.

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 ARM® 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 X-NUCLEO-IKS01A2 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 MotionFD middleware library in X-CUBE-MEMS1 software expansion for STM32Cube

2.1 MotionFD overview

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

The library acquires data from the accelerometer and pressure sensor and provides information about the user fall event 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 the X-NUCLEO-IKS01A2 expansion board, mounted on a NUCLEO-F401RE, NUCLEO-L476RG or NUCLEO-L152RE development board.

2.2 MotionFD library

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

2.2.1 MotionFD library description

The MotionFD fall detection library manages the data acquired from the accelerometer and pressure sensor; it features:

- possibility to distinguish whether the user fall occurred or not
- recognition based only on accelerometer and pressure sensor data
- required accelerometer and pressure sensor data sampling frequency is 25 Hz
- occupies 2.4 kByte of code memory and 3.4 kByte of data memory

  Note: Real size might differ for different IDEs (toolchain)
- available for ARM Cortex-M3 and Cortex-M4 architectures

2.2.2 MotionFD APIs

The MotionFD library APIs are:

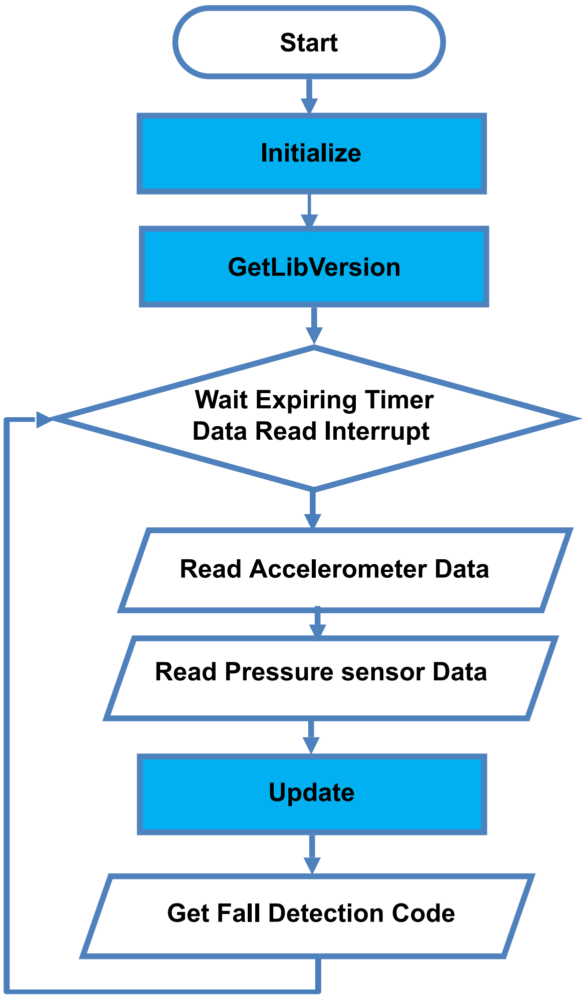
- `uint8_t MotionFD_GetLibVersion(char *version)`
  – retrieves the library version
  – `*version` is a pointer to an array of 35 characters
  – returns the number of characters in the version string
- `void MotionFD_Initialize(void)`
  – performs MotionFD library initialization and setup of the internal 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 fall detection library.
- `void MotionFD_Update (MFD_input_t *data_in, MFD_output_t *data_out)`
  – executes fall detection algorithm
  – `*data_in` parameter is a pointer to a structure with input data
  – the parameters for the structure type `MFD_input_t` are:
    ◦ `AccX` is the accelerometer sensor value in X axis in mg
    ◦ `AccY` is the accelerometer sensor value in Y axis in mg
    ◦ `AccZ` is the accelerometer sensor value in Z axis in mg
    ◦ `Press` is the pressure sensure value in hPa

– *data_out parameter is a pointer to an enum with the following items:
  ◦ MFD_NOFALL = 0
  ◦ MFD_FALL = 1

2.2.3 API flow chart

Figure 1. MotionFD API logic sequence



2.2.4 Demo code

The following demonstration code reads data from the accelerometer and pressure sensor and gets the fall event code.

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

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

/* Fall Detection API initialization function */
MotionFD_Initialize();
```

```
/* Optional: Get version */
MotionFD_GetLibVersion(lib_version);

[…]

/*** Using Fall Detection algorithm ***/
Timer_OR_DataRate_Interrupt_Handler()
{
  MFD_input_t data_in;
  MFD_output_t data_out;

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

  /* Get pressure in hPa */
  MEMS_Read_PressValue(&data_in.Press);

  /* Fall Detection algorithm update */
  MotionFD_Update(&data_in, &data_out);
}
```

2.2.5 Algorithm performance

The fall detection algorithm only uses data from the accelerometer and pressure sensor and runs at a low frequency (25 Hz) to reduce power consumption.

Table 2. **Elapsed time (µs) algorithm**

Cortex-M4 STM32F401RE at 84 MHz									Cortex-M3 STM32L152RE at 32 MHz								
SW4STM32 1.13.1 (GCC 5.4.1)			IAR EWARM 7.80.4			Keil µVision 5.22			SW4STM32 1.13.1 (GCC 5.4.1)			IAR EWARM 7.80.4			Keil µVision 5.22		
Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
13	130	133	80	81	131	77	252	256	9	466	548	282	285	470	272	828	894

2.3 Sample application

The MotionFD 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 (based on LSM6DSL) expansion board.

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

Stand-alone mode

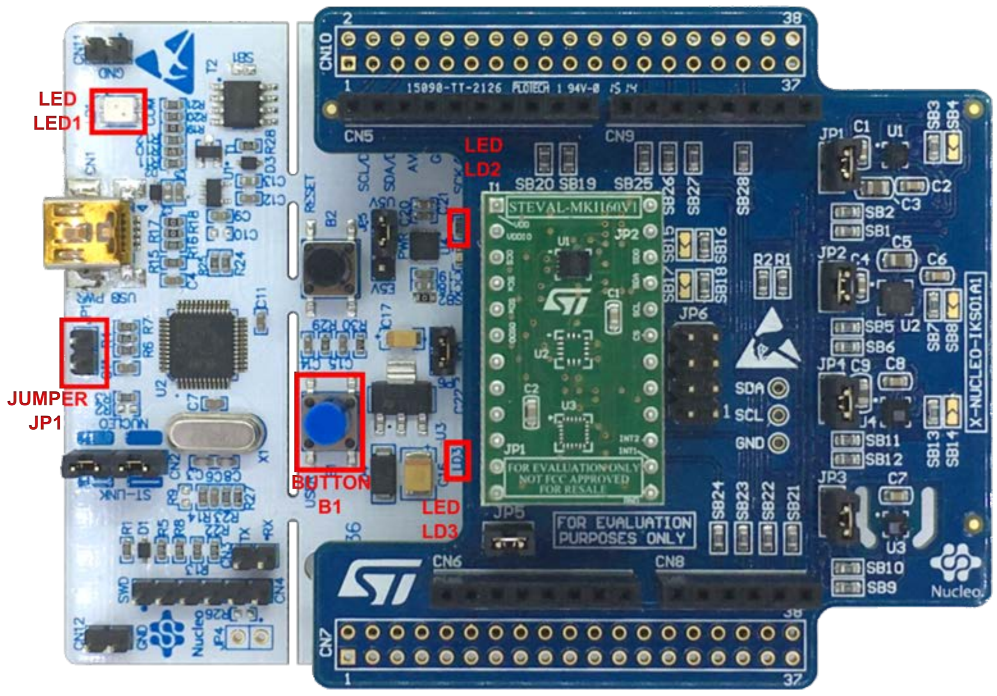
In stand-alone mode, the sample application allows detecting the user fall event and store it in the MCU flash memory.

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 3. **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 and pressure sensor and detects the user fall event; during this acquisition mode, a fast LED LD2 blinking indicates that the algorithm is running. During this phase, the detected user fall event 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 detected user fall event, accelerometer and pressure sensor 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

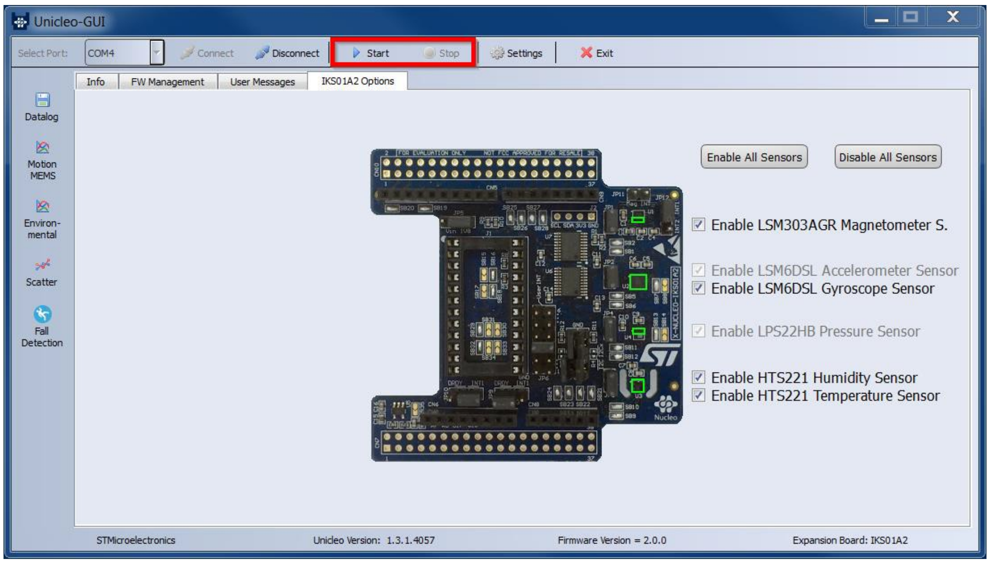
The sample application uses the Windows Unicleo-GUI utility, which can be downloaded from www.st.com.

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



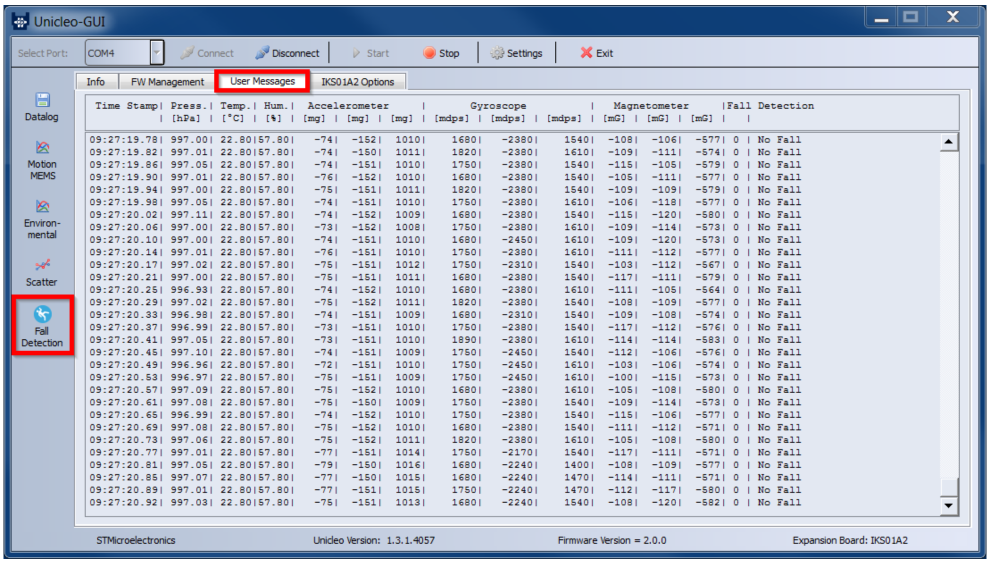
☑ Enable LSM303AGR Magnetometer S.
☑ Enable LSM6DSL Accelerometer Sensor
☑ Enable LSM6DSL Gyroscope Sensor
☑ Enable LPS22HB Pressure Sensor
☑ Enable HTS221 Humidity Sensor
☑ Enable HTS221 Temperature Sensor

STMicroelectronics Unicleo Version: 1.3.1.4057 Firmware Version = 2.0.0 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

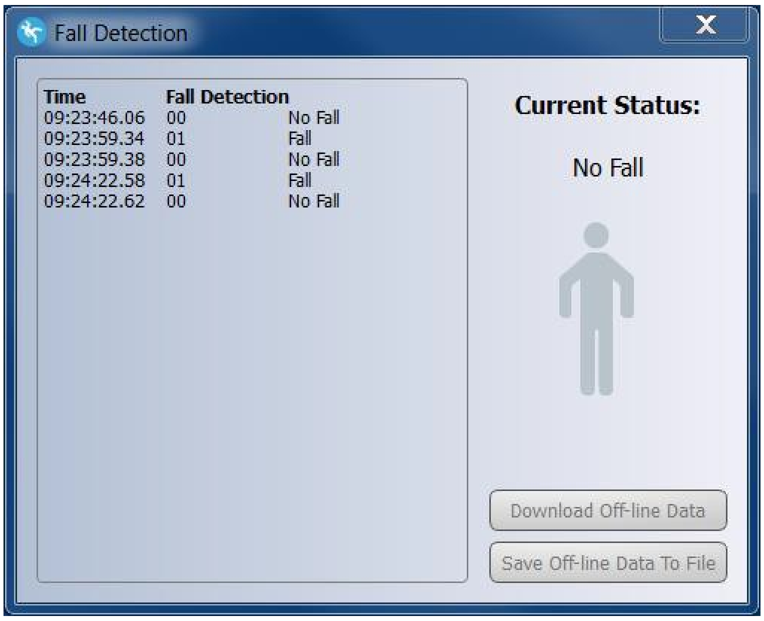


```
 Time Stamp| Press.| Temp.| Hum.| Accelerometer      |       Gyroscope        |    Magnetometer     |Fall Detection
           | [hPa] | [°C] | [%] | [mg] | [mg] | [mg] | [mdps] | [mdps] | [mdps] | [mG] | [mG] | [mG] |     |
09:27:19.78| 997.00| 22.80|57.80|   -74|  -152|  1010|    1680|   -2380|    1540|  -108|  -106|  -577| 0 | No Fall
09:27:19.82| 997.01| 22.80|57.80|   -74|  -150|  1011|    1820|   -2380|    1610|  -109|  -111|  -574| 0 | No Fall
09:27:19.86| 997.05| 22.80|57.80|   -74|  -151|  1010|    1750|   -2380|    1540|  -115|  -105|  -579| 0 | No Fall
09:27:19.90| 997.01| 22.80|57.80|   -76|  -152|  1010|    1680|   -2380|    1540|  -105|  -111|  -577| 0 | No Fall
09:27:19.94| 997.00| 22.80|57.80|   -75|  -151|  1011|    1820|   -2380|    1540|  -109|  -109|  -579| 0 | No Fall
09:27:19.98| 997.05| 22.80|57.80|   -74|  -151|  1010|    1750|   -2380|    1610|  -106|  -118|  -577| 0 | No Fall
09:27:20.02| 997.11| 22.80|57.80|   -74|  -152|  1009|    1680|   -2380|    1540|  -115|  -120|  -580| 0 | No Fall
09:27:20.06| 997.00| 22.80|57.80|   -73|  -152|  1008|    1750|   -2380|    1610|  -109|  -114|  -573| 0 | No Fall
09:27:20.10| 997.00| 22.80|57.80|   -74|  -151|  1010|    1680|   -2450|    1610|  -109|  -120|  -573| 0 | No Fall
09:27:20.14| 997.01| 22.80|57.80|   -76|  -151|  1010|    1750|   -2380|    1610|  -111|  -112|  -577| 0 | No Fall
09:27:20.17| 997.02| 22.80|57.80|   -75|  -151|  1012|    1750|   -2310|    1540|  -103|  -112|  -567| 0 | No Fall
09:27:20.21| 997.00| 22.80|57.80|   -75|  -151|  1011|    1680|   -2380|    1540|  -117|  -111|  -579| 0 | No Fall
09:27:20.25| 996.93| 22.80|57.80|   -74|  -152|  1010|    1680|   -2380|    1610|  -111|  -105|  -564| 0 | No Fall
09:27:20.29| 997.02| 22.80|57.80|   -75|  -152|  1011|    1820|   -2380|    1540|  -108|  -109|  -577| 0 | No Fall
09:27:20.33| 996.98| 22.80|57.80|   -74|  -151|  1009|    1680|   -2310|    1540|  -109|  -108|  -574| 0 | No Fall
09:27:20.37| 996.99| 22.80|57.80|   -73|  -151|  1010|    1750|   -2380|    1540|  -117|  -112|  -576| 0 | No Fall
09:27:20.41| 997.05| 22.80|57.80|   -73|  -151|  1010|    1890|   -2380|    1610|  -114|  -114|  -583| 0 | No Fall
09:27:20.45| 997.10| 22.80|57.80|   -74|  -151|  1009|    1750|   -2450|    1540|  -112|  -106|  -576| 0 | No Fall
09:27:20.49| 996.96| 22.80|57.80|   -72|  -151|  1010|    1750|   -2450|    1610|  -103|  -106|  -574| 0 | No Fall
09:27:20.53| 996.97| 22.80|57.80|   -75|  -151|  1009|    1750|   -2450|    1610|  -100|  -115|  -573| 0 | No Fall
09:27:20.57| 997.09| 22.80|57.80|   -75|  -152|  1010|    1680|   -2380|    1610|  -105|  -108|  -580| 0 | No Fall
09:27:20.61| 997.08| 22.80|57.80|   -75|  -150|  1009|    1750|   -2380|    1540|  -109|  -114|  -573| 0 | No Fall
09:27:20.65| 996.99| 22.80|57.80|   -74|  -152|  1010|    1750|   -2380|    1540|  -115|  -106|  -577| 0 | No Fall
09:27:20.69| 997.08| 22.80|57.80|   -75|  -152|  1010|    1680|   -2380|    1540|  -111|  -112|  -571| 0 | No Fall
09:27:20.73| 997.06| 22.80|57.80|   -75|  -152|  1011|    1820|   -2380|    1610|  -105|  -108|  -580| 0 | No Fall
09:27:20.77| 997.01| 22.80|57.80|   -77|  -151|  1014|    1750|   -2170|    1540|  -117|  -111|  -571| 0 | No Fall
09:27:20.81| 997.05| 22.80|57.80|   -79|  -150|  1016|    1680|   -2240|    1400|  -108|  -109|  -577| 0 | No Fall
09:27:20.85| 997.07| 22.80|57.80|   -77|  -150|  1015|    1680|   -2240|    1470|  -114|  -111|  -571| 0 | No Fall
09:27:20.89| 997.01| 22.80|57.80|   -77|  -151|  1015|    1750|   -2240|    1470|  -112|  -117|  -580| 0 | No Fall
09:27:20.92| 997.03| 22.80|57.80|   -75|  -151|  1013|    1680|   -2240|    1540|  -108|  -120|  -582| 0 | No Fall
```

STMicroelectronics Unicleo Version: 1.3.1.4057 Firmware Version = 2.0.0 Expansion Board: IKS01A2

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

Figure 5. **Fall Detection window**



```
Time         Fall Detection
09:23:46.06  00              No Fall
09:23:59.34  01              Fall
09:23:59.38  00              No Fall
09:24:22.58  01              Fall
09:24:22.62  00              No Fall
```

Current Status:

No Fall

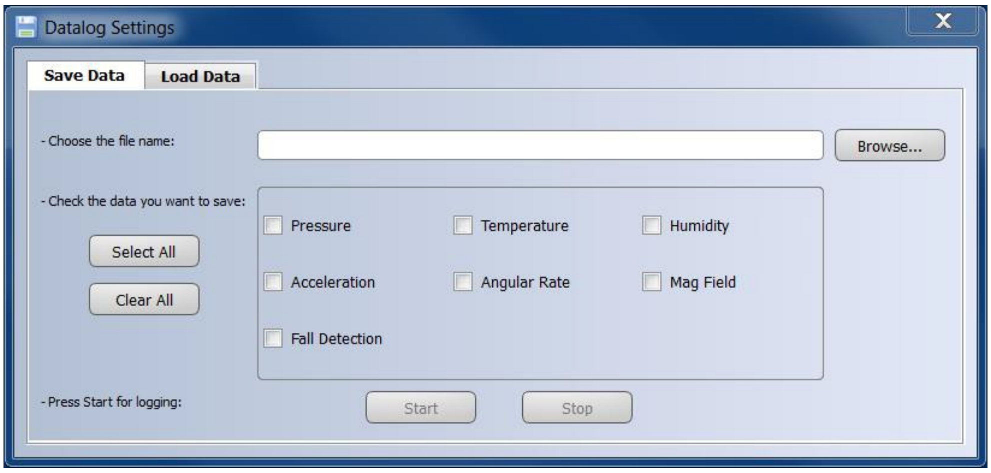
If the board has been working in stand-alone mode and the user wants to retrieve stored data, press **Download Off-line Data** button to upload the stored user fall event data to the application. This operation automatically deletes acquired data from microcontroller.

Note: ***Download Off-line Data*** *button is not available while data streaming is active.*

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 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 4. **Document revision history**

Date	Version	Changes
22-Sep-2017	1	Initial release.
06-Feb-2018	2	Added references to NUCLEO-L152RE development board and Table 2. Elapsed time (µs) algorithm.
21-Mar-2018	3	Updated Section ● Introduction and Section 2.1 MotionFD overview.

Contents

List of tables

List of figures