Getting started with MotionAD airplane detection library in X-CUBE-MEMS1 expansion for STM32Cube

Introduction

The MotionAD is a middleware library part of X-CUBE-MEMS1 software and runs on STM32. It provides real-time information about airplane mode detection based on data from a hand-held device.

The library is able to detect airplane mode from hand-held devices (mobile phone, laptop, tablet) to automatically avoid potential hazards such as interference with wireless communication and battery explosion due to high current drawn by the airplane outlet.

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®-M33, 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 an X-NUCLEO-IKS01A3 or X-NUCLEO-IKS4A1 expansion board plugged on a NUCLEO-F401RE, NUCLEO-L152RE or NUCLEO-U575ZI-Q 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 MotionAD middleware library for X-CUBE-MEMS1 software expansion for STM32Cube

2.1 MotionAD overview

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

The library is able to detect airplane mode from hand-held devices (mobile phone, laptop, tablet) to automatically avoid potential device hazard such as interference with wireless communication and explosion of battery due to the high current drawn by the airplane outlet.

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 documented behavior.

Sample implementation is available for the X-NUCLEO-IKS01A3 or X-NUCLEO-IKS4A1 expansion board, mounted on a NUCLEO-F401RE, NUCLEO-L152RE or NUCLEO-U575ZI-Q development board.

2.2 MotionAD library

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

2.2.1 MotionAD library description

The MotionAD airplane detection library manages the data acquired from the accelerometer, pressure and temperature sensors; it features:

- possibility to distinguish the airplane mode (on land, take off, landing)
- intended for hand-held devices
- recognition based on accelerometer, pressure and temperature data
- required accelerometer data sampling frequency of 100 Hz
- required pressure data sampling frequency higher than 2 Hz
- available for ARM® Cortex®-M3, ARM® Cortex®-M33, ARM® Cortex®-M4 and ARM® Cortex®-M7 architectures
  - resources requirements:
    - Cortex-M3: 4.7 kB of code and 1.2 kB of data memory
    - Cortex-M33: 4.8 kB of code and 1.2 kB of data memory
    - Cortex-M4: 4.8 kB of code and 1.2 kB of data memory
    - Cortex-M7: 4.8 kB of code and 1.2 kB of data memory

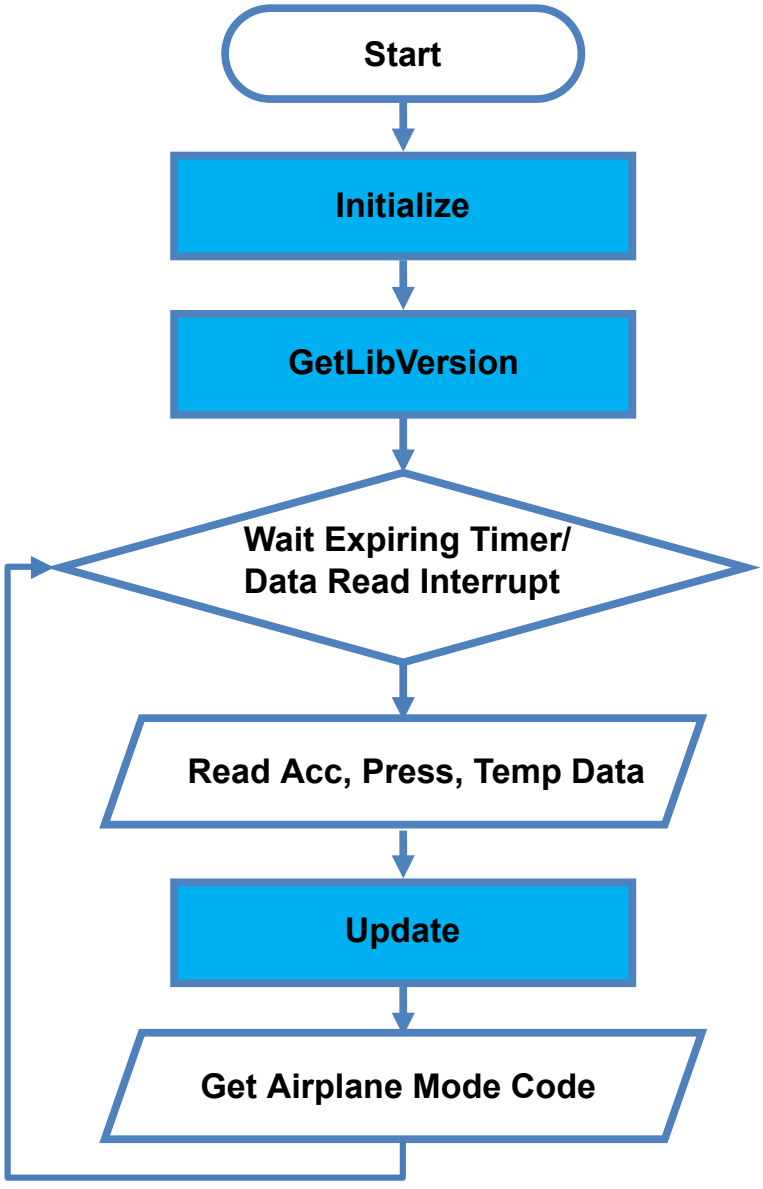
2.2.2 MotionAD APIs

The MotionAD library APIs are:

- `uint8_t MotionAD_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 MotionAD_Initialize(int xl_odr)`
  - performs MotionAD 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
  - `xl_odr` is an accelerometer ODR in Hz (nearest int)
- `void MotionAD_Update(MAD_input_t *data_in, MAD_output_t *data_out)`
  - executes airplane detection algorithm
  - `*data_in` parameter is a pointer to a structure with input data
  - the parameters for the structure type `MAD_input_t` are:
    - `Acc[3]` is the array of accelerometer sensor values in X, Y, Z axes in g (float)
    - `Press` is the pressure sensor value in Pa (unsigned int)
    - `Temp` is the temperature sensor value in °C (float)
    - `*data_out` parameter is a pointer to an enum with the following values:
      - `MAD_ONLAND = 0`
      - `MAD_TAKEOFF = 1`
      - `MAD_LANDING = 2`

2.2.3 API flow chart

Figure 1. MotionAD API logic sequence



2.2.4 Demo code

The following demonstration code reads data from the accelerometer, temperature and pressure sensors and detects the airplane mode code.

```
[…]

#define VERSION_STR_LENG 35

#define ALGO_FREQ 100
[…]

/*** Initialization ***/

char lib_version[VERSION_STR_LENG];

/* Airplane Detection API initialization function */

MotionAD_Initialize(ALGO_FREQ);

/* Optional: Get version */

MotionAD_GetLibVersion(lib_version);

[…]

/*** Using Airplane Detection algorithm ***/


Timer_OR_DataRate_Interrupt_Handler()


{

MAD_input_t data_in;

MAD_output_t data_out;

/* Get acceleration X/Y/Z in g */

MEMS_Read_AccValue(&data_in.Acc[0], &data_in.Acc[1], &data_in.Acc[2]);

/* Get presure in Pa */

MEMS_Read_PressValue(&data_in.Press);

/* Get temperature in °C */

MEMS_Read_TempValue(&data_in.Temp);

/* Airplane Detection algorithm update */

MotionAD_Update(&data_in, &data_out);

}
```

2.2.5 Algorithm performance

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

Cortex-M4 STM32F401RE at 84 MHz			Cortex-M3 STM32L152RE at 32 MHz		
Min	Avg	Max	Min	Avg	Max
4	24	95	36	110	371

Table 3. **Cortex-M33 and Cortex-M7: elapsed time (µs) algorithm**

Cortex-M33 STM32U575ZI-Q at 160 MHz			Cortex-M7 STM32F767ZI at 96 MHz		
Min	Avg	Max	Min	Avg	Max
2	14	56	46	49	141

2.3 Sample application

The MotionAD 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-L152RE or NUCLEO-U575ZI-Q development board connected to an X-NUCLEO-IKS01A3 or X-NUCLEO-IKS4A1 expansion board.

The board is powered by the PC via USB connection, which is required to monitor real-time data or feed the library with offline data.

This working mode allows the user to display detected airplane mode, accelerometer, pressure and temperature data, time stamp and eventually other sensor data, in real-time, using the MEMS-Studio GUI application.

.

2.4 MEMS-Studio application

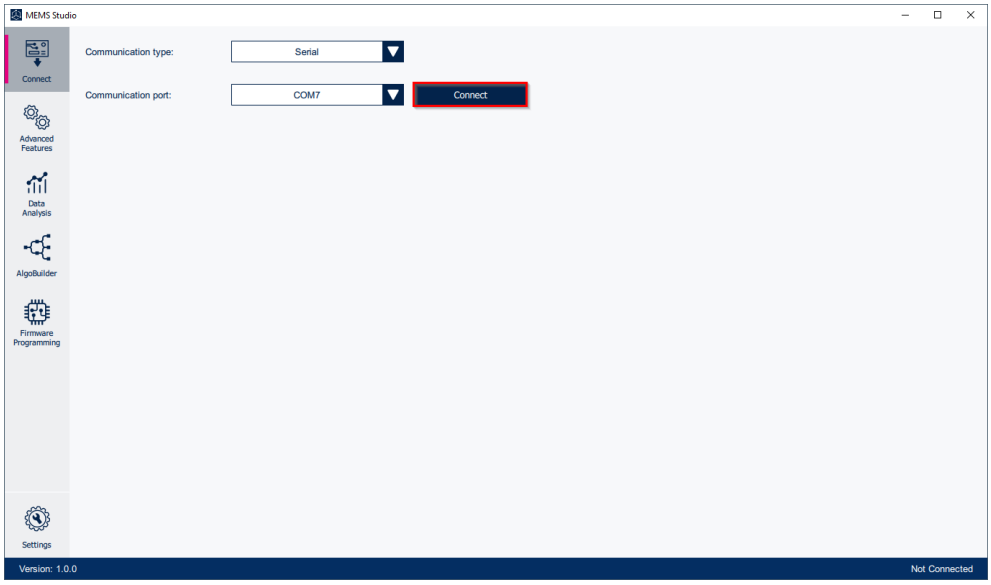
The sample application uses the MEMS-Studio GUI application, which can be downloaded from www.st.com.

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 MEMS-Studio application to open the main application window.

If an STM32 Nucleo board with supported firmware is connected to the PC, it is automatically detected the appropriate COM port. Press **Connect** button to open this port.

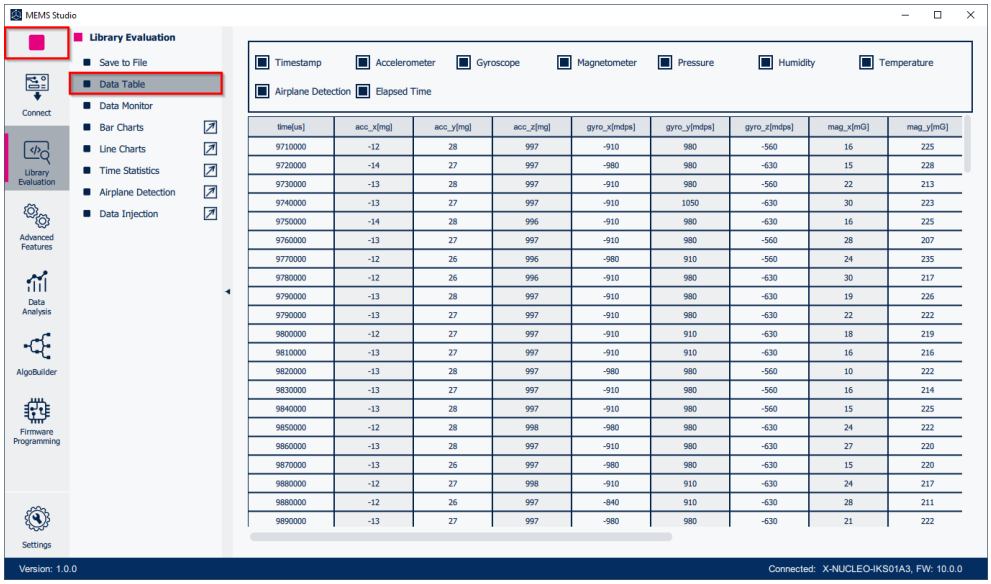
Figure 2. **MEMS-Studio - Connect**



Step 3. When connected to STM32 Nucleo board with supported firmware *Library Evaluation* tab is opened.

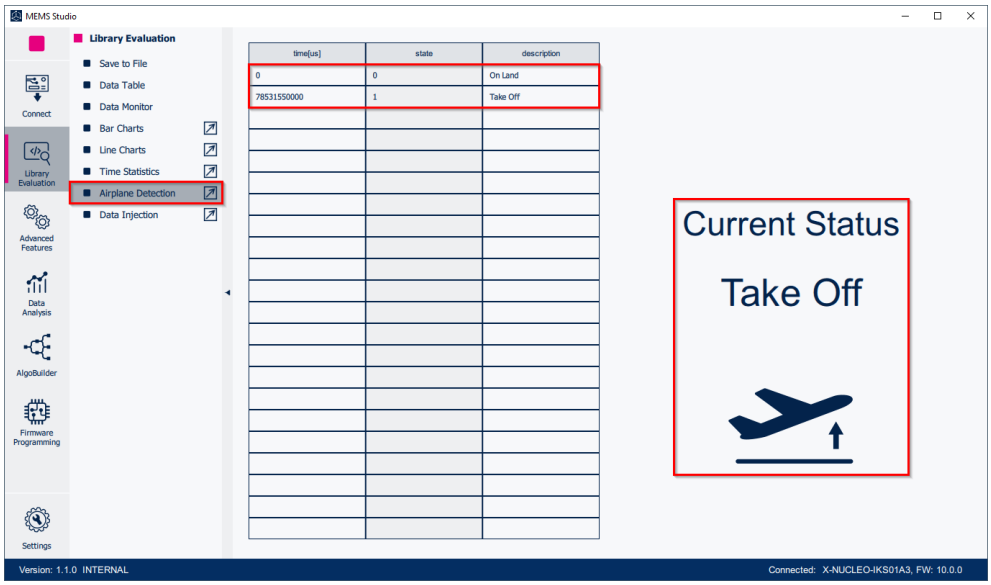
To start and stop data streaming toggle the appropriate start / stop button on the outer vertical tool bar.
The data coming from the connected sensor can be viewed selecting the *Data Table* tab on the inner vertical tool bar.

Figure 3. MEMS-Studio - Library Evaluation - Data Table



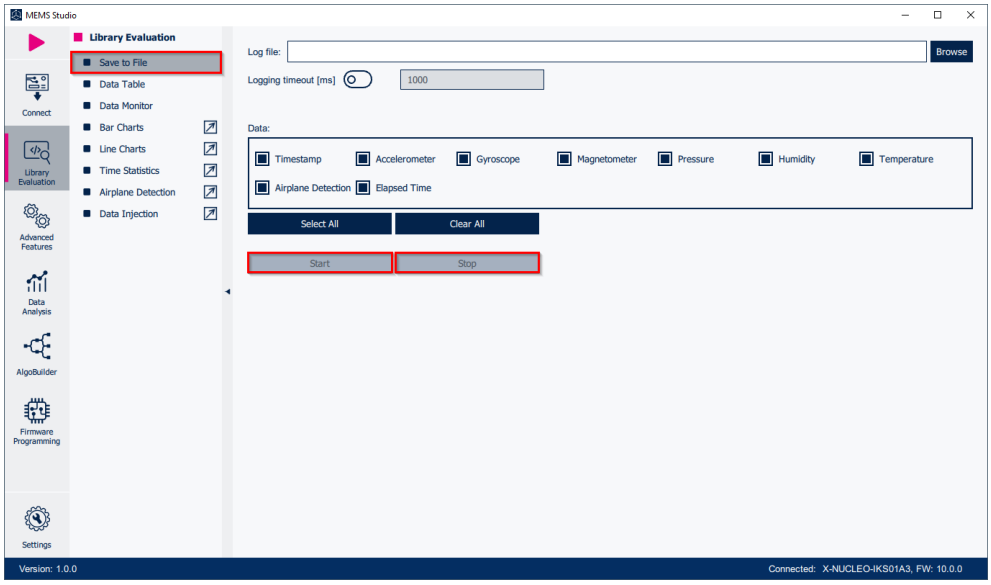
Step 4. Select the *Airplane Detection* tab on the inner vertical tool bar to open the dedicated application status view.

Figure 4. MEMS-Studio - Library Evaluation - Accelerometer Calibration



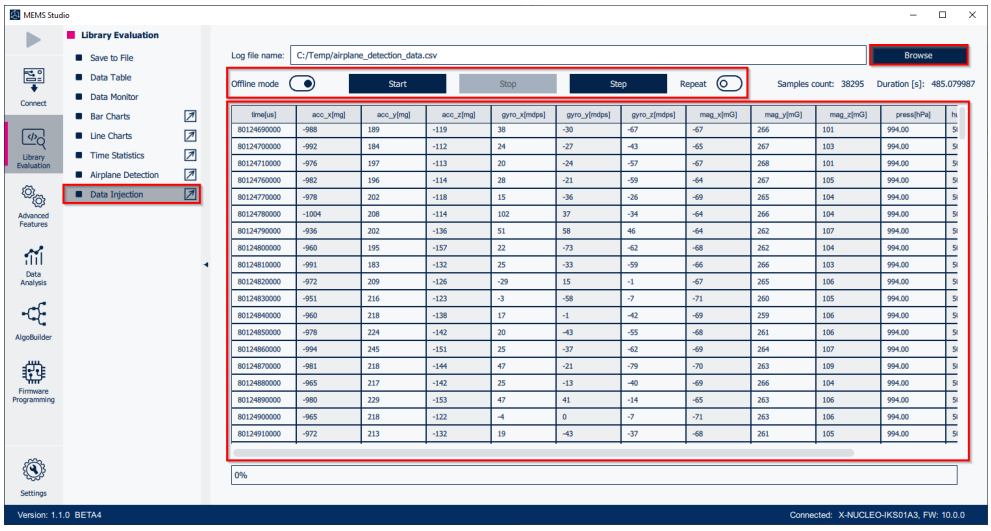
Step 5. Select the *Save to File* tab on the inner vertical tool bar to open the data logging configuration window. Select which sensor and activity data to save to log file. You can start or stop saving by clicking on the corresponding Start / Stop button.

Figure 5. MEMS-Studio - Library Evaluation - Save to File



Step 6. Select the *Data Injection* tab on the inner vertical tool bar to open the view dedicated to process the previously captured data. The data are processed by the firmware in MCU.

Figure 6. MEMS-Studio - Library Evaluation - Data Injection



Step 7. Click on the *Browse* button to select the file with the previously captured data in CSV format. The data will be loaded into the table in the current view.

Other buttons above table in the current view will become active. You can click on:

– *Offline mode* button to switch on/off the firmware *offline mode* (mode utilizing the previously captured data)

– *Start/Stop/Step/Repeat* buttons to control the data feed from the MEMS-Studio to the firmware

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 boards (MB1136)
3. UM3233: Getting started with MEMS-Studio

Revision history

Table 4. **Document revision history**

Date	Version	Changes
13-May-2020	1	Initial release.
28-Mar-2024	2	Updated Section Introduction, Section 2.1: MotionAD overview, Section 2.2.1: MotionAD library description, Section 2.2.2: MotionAD APIs, Section 2.2.4: Demo code, Section 2.2.5: Algorithm performance, Section 2.3: Sample application and Section 2.3: Sample application.

Contents

List of tables

List of figures