User manual

Getting started with MotionTL2 2-axis tilt measurement library in X-CUBE-MEMS1 expansion for STM32Cube

Introduction

The MotionTL2 middleware library is part of the X-CUBE-MEMS1 software and runs on STM32. It provides real-time information about the tilt angles of the IIS2ICLX 2-axis accelerometer.

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®-M0+, ARM® Cortex®-M3, ARM® Cortex®-M4 or ARM® Cortex®-M7 architecture.

It is built on top of STM32Cube software technology to ease portability across different STM32 microcontrollers.

The software comes with sample implementation running on a STEVAL-MKI209V1K MEMS inclinometer kit based on IIS2ICLX 2-axis accelerometer 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 MotionTL2 middleware library in X-CUBE-MEMS1 software expansion for STM32Cube

2.1 MotionTL2 overview

The MotionTL2 library expands the functionality of the X-CUBE-MEMS1 software.

The library acquires data from the 2-axis accelerometer and provides information about the tilt angles of the device.

This library is suitable for static inclinometer where system acceleration is negligible. The main applications are industrial, leveling, satellite antennas, solar panels and automotive.

The library is designed for ST MEMS only. Functionality and performance when using other MEMS sensors are not analyzed and can differ significantly from what is described herein.

A sample implementation is available for the STEVAL-MKI209V1K mounted on a NUCLEO-F401RE, NUCLEO-L476RG, NUCLEO-L152RE or NUCLEO-L073RZ development board.

2.2 MotionTL2 library

Technical information fully describing the functions and parameters of the MotionTL2 APIs can be found in the MotionTL2_Library.chm compiled HTML file located in the Documentation folder.

2.2.1 MotionTL2 library description

The MotionTL2 library implements a tilt computation algorithm for the estimation of single or dual axis orientation in space. The library includes the functionality to reduce the impact of vibration using knob settings and uses gravity to determine the tilt angle using calibrated accelerometer data as input.

The MotionTL2 library manages the data acquired from the 2-axis accelerometer; it features:

- real-time tilt computation
- single or dual plane mode configuration
- knob configuration to mitigate vibration noise
- tilt angle, validity flag and expected error outputs
- measurement based on the accelerometer data only
- recommended sensor data sampling frequency of 100 Hz and support for all full scale ranges
- resources requirements:
  - Cortex-M0+: 1.9 kB of code and 0.1 kB of data memory
  - Cortex-M3: 1.9 kB of code and 0.1 kB of data memory
  - Cortex-M4: 1.8 kB of code and 0.1 kB of data memory
  - Cortex-M7: 1.7 kB of code and 0.1 kB of data memory

2.2.2 MotionTL2 APIs

The MotionTL2 library APIs are:

- `uint8_t MotionTL2_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 MotionTL2_Init(void)`
  - performs MotionTL2 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 tilt library.

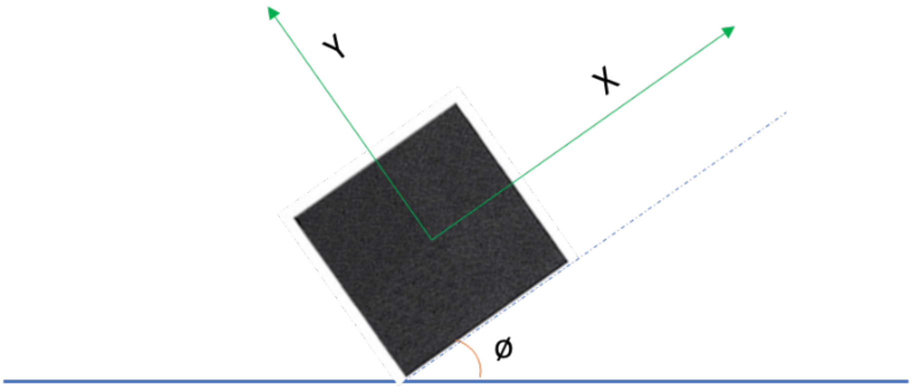
- `void MotionTL2_Update(MTL2_input_t *data_in, uint64_t timestamp_ms, MTL2_output_t *data_out)`
  - executes tilt algorithm
  - `*data_in` parameter is a pointer to a structure with input data
  - the parameters for the structure type `MTL2_input_t` are:
    - `acc_x` is the accelerometer sensor value in X axis in g
    - `acc_y` is the accelerometer sensor value in Y axis in g
  - `timestamp_ms` parameter is the timestamp in ms
  - `*data_out` parameter is a pointer to a structure with output data
  - the parameters for the structure type `MTL2_output_t` are:
    - `tilt_1x` indicates the angle between X axis and horizontal plane in single plane mode; the range of the angle is [-180, 180] degrees
    - `theta_2x` indicates the angle between X axis and horizontal plane in dual plane mode; the range of the angle is [-90, 90] degrees
    - `psi_2x` indicates the angle between Y axis and horizontal plane in dual plane mode; the range of the angle is [-90, 90] degrees
    - `phi_2x` indicates the angle between XY plane and horizontal plane in dual plane mode; the range of the angle is [0, 90] degrees
    - `err_deg` indicates the predicted angle error in both modes; the output can be used to accept/reject the tilt angle and the range of the angle is [0, 90] degrees
    - `valid` is used to show if the output is valid or not in both modes, if the accelerometer reading is showing high vibration or saturation at full scale. The library output is ‘0’ for valid field
- `void MotionTL2_GetKnobs(MTL2_knobs_t *knobs)`
  - gets current knob settings
  - `*knobs` parameter is a pointer to a structure with settings
  - the parameters for the structure type `MTL2_knobs_t` are:
    - `fullscale` parameter is the accelerometer full range in the unit of g. It is recommended to set full scale >1g for the sensor. A lower full scale can be selected if the tilt variation is limited and higher resolution is required for the application.
    - `k` parameter is the filtering coefficient. The range of k is [0.1 to ODR]. A lower value of k increases the filtering and removes the noise. For systems with high vibration, it is recommended to reduce the value of k.
    - `mode` parameter is the operational mode
      - `MTL2_SINGLE_PLANE = 0` enables the angle computation in single plane mode
      - `MTL2_DUAL_PLANE = 1` enables the angle computation in dual plane mode
    - `orn[2]` parameter is the acc data orientation string of two characters indicating the direction of each positive orientation of the reference frame used for the accelerometer data output, in the sequence x, y. Valid values are: n (north) or s (south), w (west) or e (east).
- `void MotionTL2_SetKnobs(MTL2_knobs_t *knobs)`
  - sets current knob settings
  - `*knobs` parameter is a pointer to a structure with settings

2.2.3 MotionTL2 modes

2.2.3.1 Single plane mode

In single plane mode, the inclination and sensor plane should always be in single plane.

Figure 1. **Single plane mode**



Single plane mode measures the angle of X axis with respect to the horizontal plane in anticlockwise direction as shown in the figure above.

In case of <1g full scale, the library outputs the angle but it might have a higher influence on external vibration or bias error.

The single plane output is stored in the `tilt_1x` variable.

The device orientation should be set to ensure the X axis is in the horizontal plane and Y axis points up.

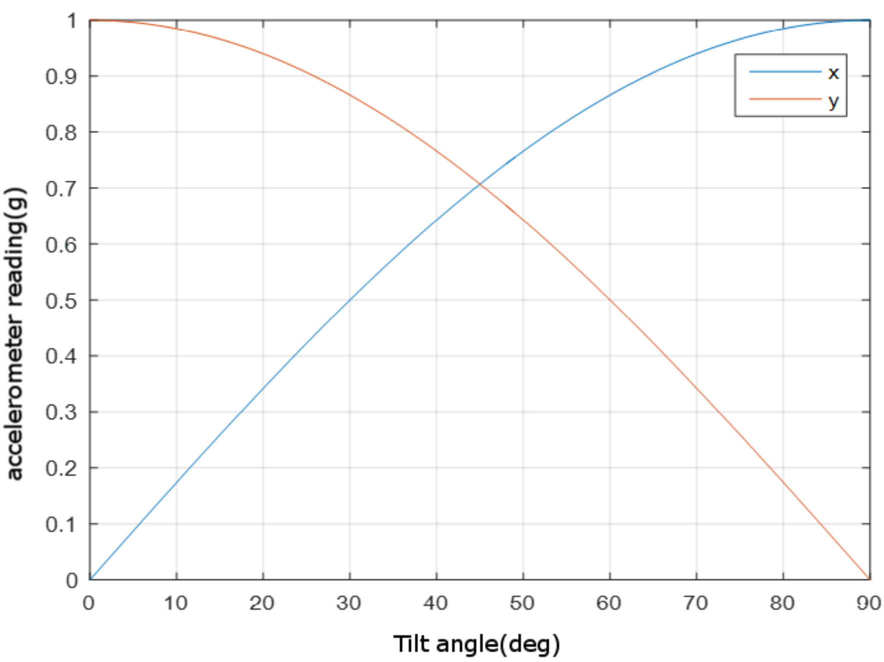
The reading of the calibrated accelerometer in presence of gravity (g) is represented by:

$$Ax = g \cdot \sin(\Phi) \quad (1)$$

$$Ay = g \cdot \cos(\Phi) \quad (2)$$

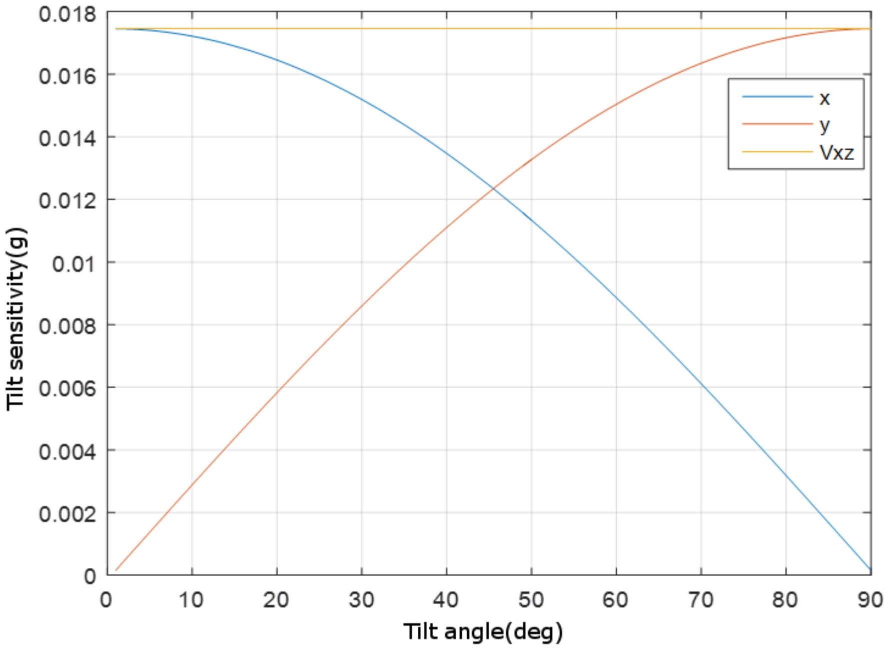
The figure below shows the typical variation in both axes due to the gravity applied to different angles.

Figure 2. **Accelerometer reading vs. tilt angle**



The figure below shows the individual axis slope varying across different tilt angles and sensitivity. It also shows the benefits of using both axes readings to estimate the tilt of the constant sensitivity VxZ across all the tilt angle readings.

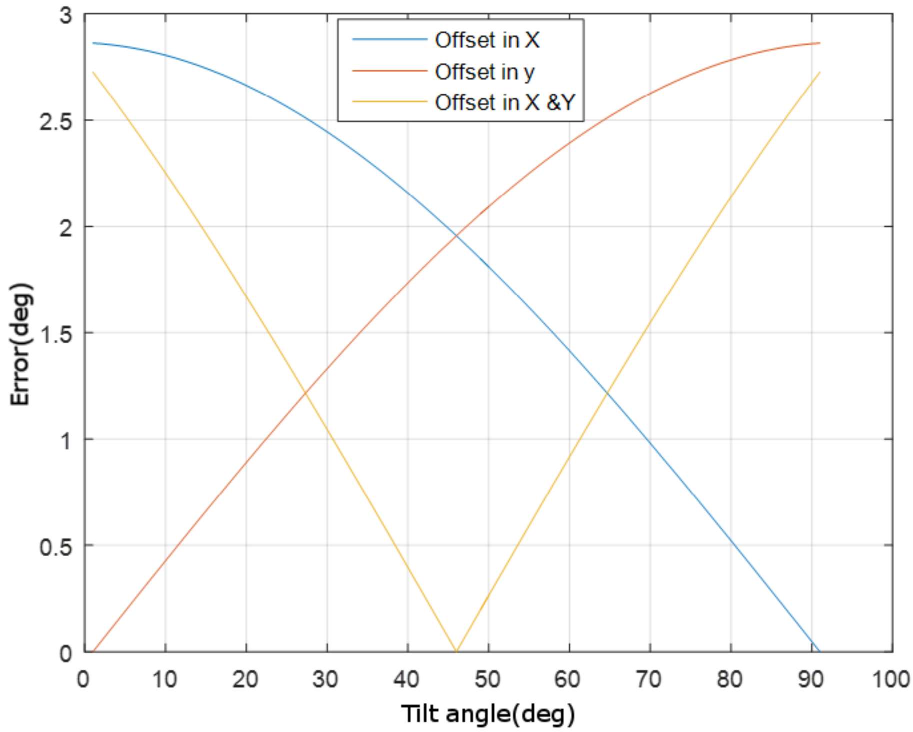
Figure 3. Tilt sensitivity vs. tilt angle



The accelerometer readings can be affected by various factors, such as bias, sensitivity error, noise, thermal drift and external acceleration.

The figure below demonstrates the impact of constant bias on the tilt final calculation: the error in the tilt angle appears if 50 mg offset is present in X, Y or both.

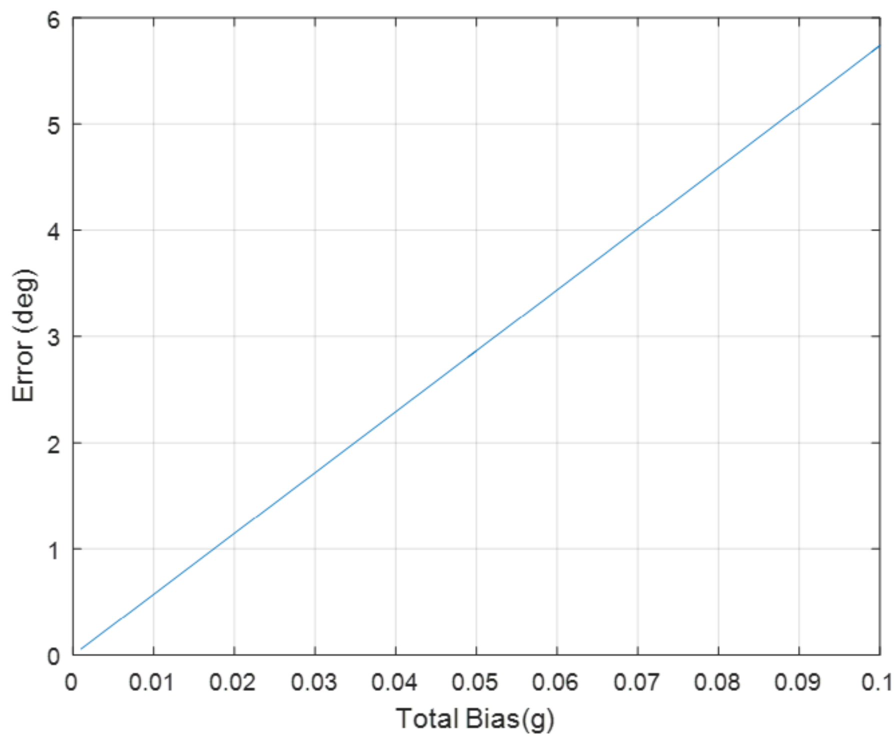
Figure 4. **Error due to bias in tilt estimation at 50 mg offset**



The figure below shows the maximum error when total bias (||Xoff+YOff||) error varies from 1 mg to 100 mg at the selected angle (30 degrees). The error rises linearly as the total bias increases.

It is recommended to perform accelerometer calibration to reduce the error.

Figure 5. **Maximum error at 30° deg in tilt estimation**



2.2.3.2 *Dual plane mode*

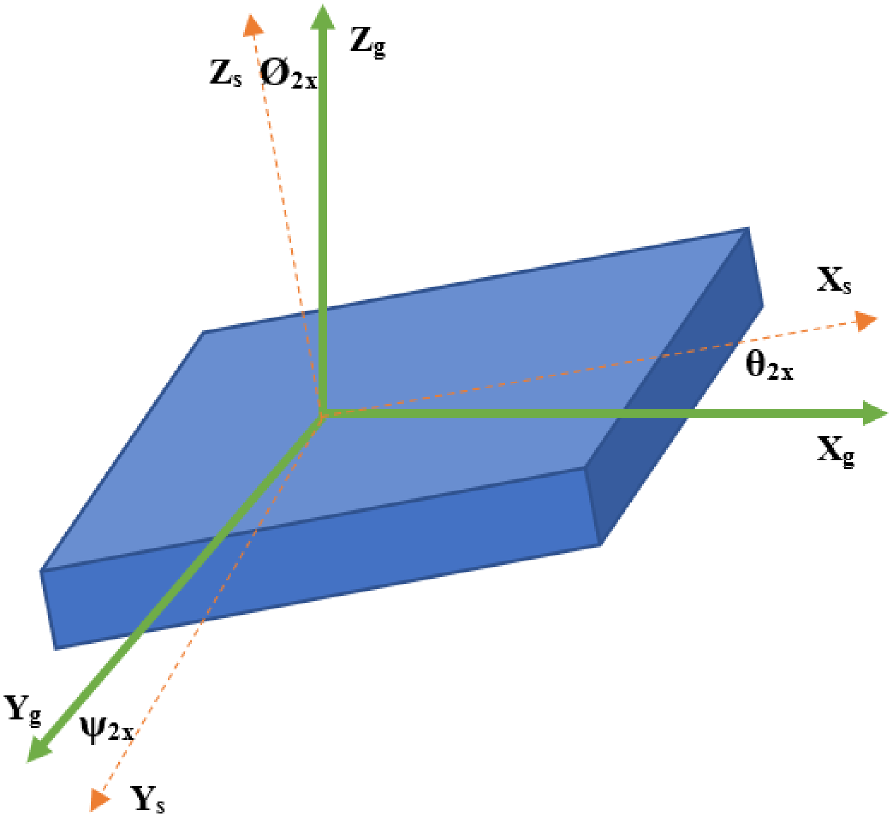
Dual plane mode is suitable for applications where the inclination is needed on both axes.

Dual plane mode computes the angle between X-axis, Y-axis and the horizontal plane. It also computes the gravity inclination (vertical axis and gravity vector) or angle between horizontal plane and sensor XY plane.

The dual plane mode output is stored in Theta_2x, Psi_2x, Phi_2x of data structure.

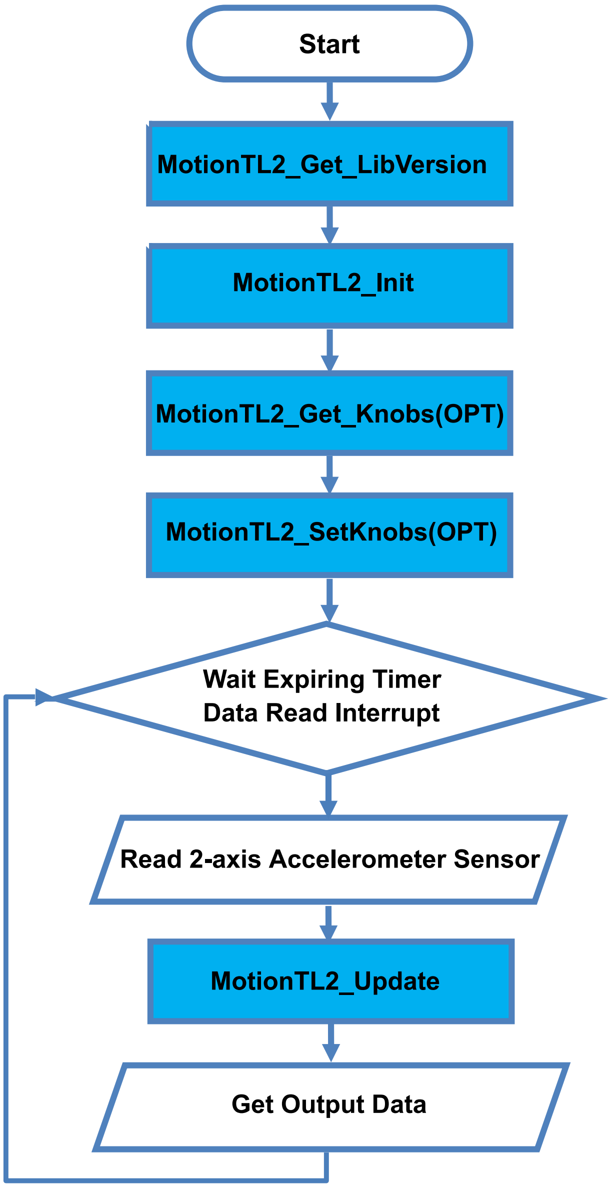
In normal operation condition, the X and Y axes are in the horizontal plane.

Figure 6. Dual plane mode



2.2.4 API flow chart

Figure 7. MotionTL2 API logic sequence



2.2.5 Demo code

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

```
[…]

#define VERSION_STR_LENG    35

[…]

/*** Initialization ***/

char lib_version[VERSION_STR_LENG];
MTL2_knobs_t Knobs;

/* Tilt API initialization function */

MotionTL2_manager_init();

/* OPTIONAL */

/* Get library version */

MotionTL2_manager_get_version(lib_version, &lib_version_len);

/* OPTIONAL */
MotionTL2_GetKnobs(&knobs);
/* Update scale factor, k, orn */
MotionTL2_SetKnobs(&knobs);

[…]

/*** Using tilt algorithm ***/

Timer_OR_DataRate_Interrupt_Handler()

{

MTL2_input_t data_in;

MTL2_output_t data_out;

AngleMode_t angle_mode = MODE_THETA_PSI_PHI;

/* Get acceleration X/Y in g */

MEMS_Read_AccValue(&data_in.acc_x, &data_in.acc_y);

/* Run tilt sensing algorithm */

MotionTL2_manager_Update(&data_in, timestamp_ms, &data_out);

MotionTL2_manager_GetAngles(&data_out, angleMode);

}
```

2.2.6 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								
STM32CubeIDE 1.3.0			IAR EWARM 8.32.3			Keil µVision 5.27			STM32CubeIDE 1.3.0			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
11	12	15	8	9	10	8	9	10	329	377	460	165	204	226	219	261	290

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

Cortex-M0+ STM32L073RZ at 32 MHz									Cortex-M7 STM32F767ZI at 96 MHz								
STM32CubeIDE 1.3.0			IAR EWARM 8.32.3			Keil µVision 5.27			STM32CubeIDE 1.3.0			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
<1000	<1000	1000	<1000	<1000	1000	<1000	<1000	<1000	10	17	22	7	10	15	8	11	16

3 Sample application

The MotionTL2 middleware can be easily manipulated to build user applications; a sample application is provided in the Application folder.

It is designed to run on NUCLEO-F401RE, NUCLEO-L476RG, NUCLEO-L152RE or NUCLEO-L073RZ development board when connected to a STEVAL-MKI209V1K.

The 2-axis tilt sensing algorithm only uses data from the accelerometer. It detects and provides real-time information about the tilt angles of the IIS2ICLX 2-axis accelerometer.

The application is also able to perform 2-axis accelerometer calibration thanks to integration of MotionAC2 library to increase the accuracy of tilt angles.

A USB cable connection is required to monitor real-time data. This allows the user to display calculated tilt angles, and raw and calibrated accelerometer data on the fly through the Unicleo-GUI.

3.1 Unicleo-GUI application

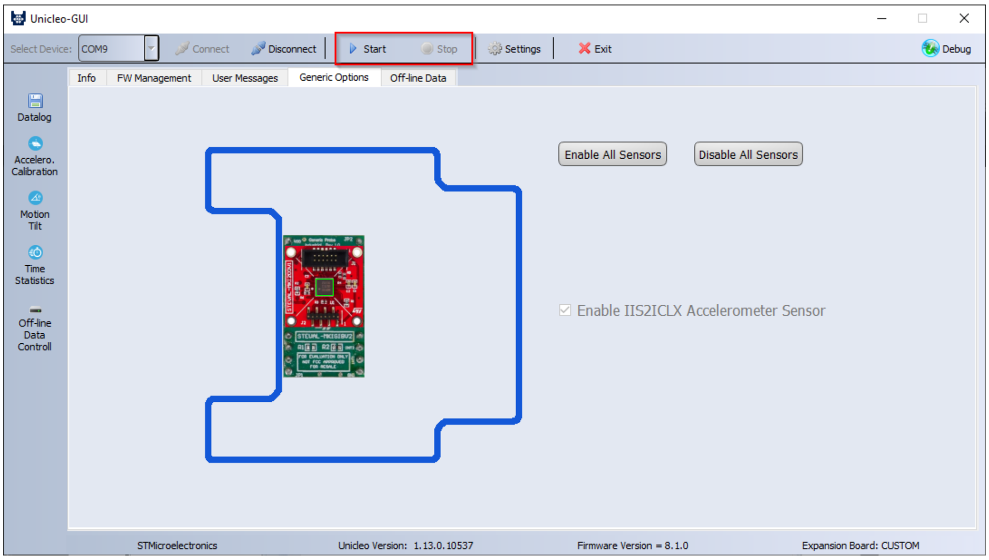
The sample application uses the Windows Unicleo-GUI utility, which can be downloaded from www.st.com.

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.

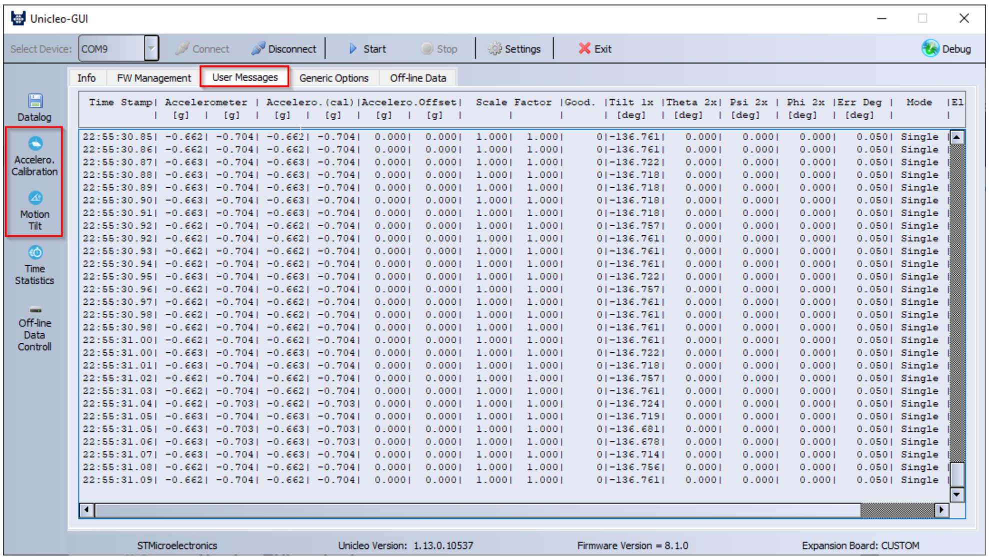
If an STM32 Nucleo board with supported firmware is connected to the PC, it is automatically detected and the appropriate COM port is opened.

Figure 8. Unicleo main window



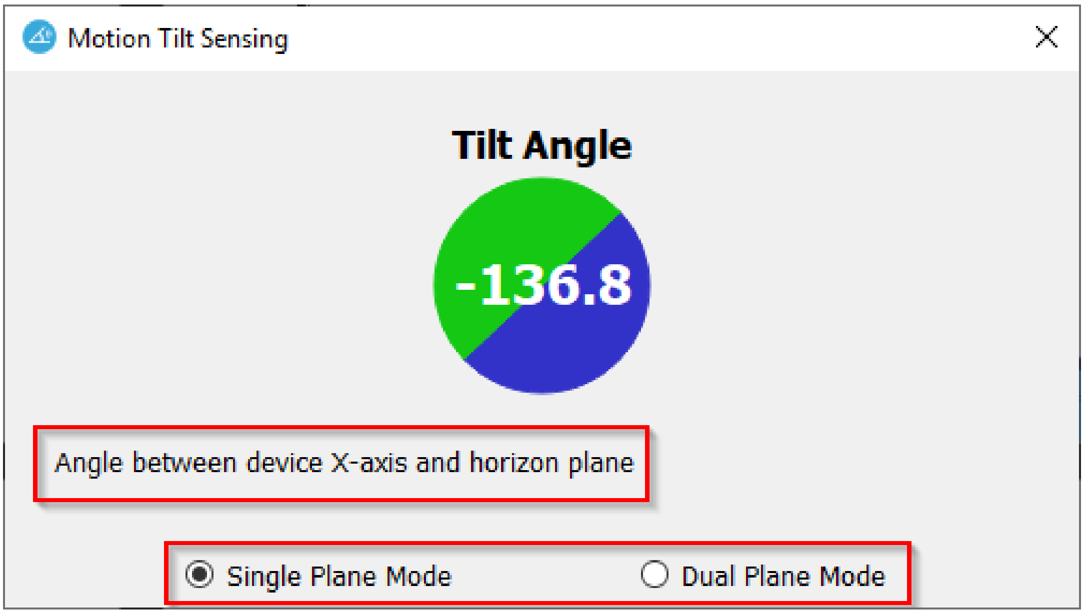
Step 3. Start and stop data streaming by using the appropriate buttons on the vertical tool bar. The data coming from the connected sensor can be viewed in the User Messages tab.

Figure 9. User Messages tab



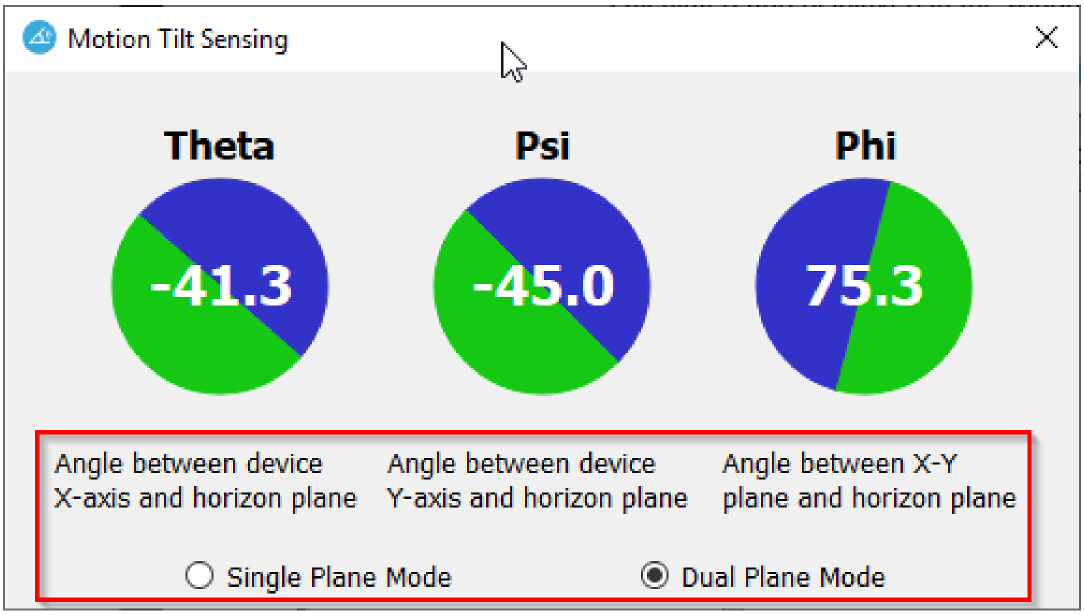
Step 4. Click on the [**Motion Tilt**] icon in the vertical tool bar to open the dedicated application window.

Figure 10. **Motion Tilt Sensing window 1**



You can switch between two angle modes. In the first mode the single plane tilt angle is displayed, in the second mode the dual plane Theta, Psi and Phi angles are displayed. The meaning of the angles in the second mode is displayed right below each indicator:

Figure 11. **Motion Tilt Sensing window 2**



Step 5. Click on the [**Accelero. Calibration**] icon in the vertical tool bar to open the dedicated application window.

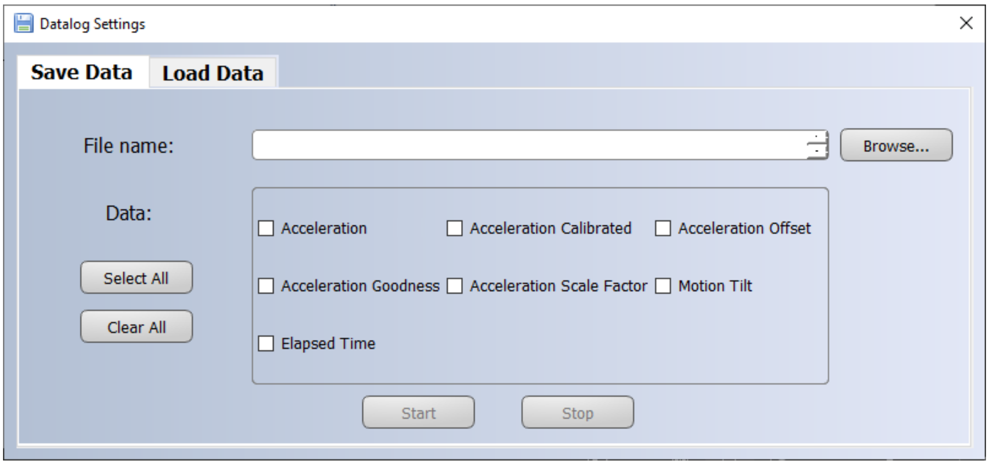
Figure 12. Accelerometer calibration window

After pressing the [**Start Calibration**] button in the Unicleo-GUI, you have to hold the device still in each of the four positions while the calibration data is collected. Then the calibration parameters (offset and gain for 2 axes) are calculated, sent to the Unicleo-GUI and displayed (for further details, see UM2774 freely available on www.st.com).

Step 6. 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.

You can also load the previously stored data.

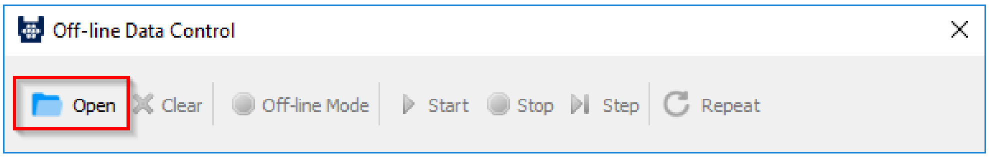
Figure 13. Datalog window



Step 7. To process the previously captured data, click on the [**Offline Data Control**] icon in the vertical tool bar and open the dedicated window.

The data are processed by the MCU firmware.

Figure 14. Offline data control menu bar



Step 8. Click on the [**Open**] button to select the file with offline data in CSV format. The data are loaded into the [**Offline Data**] tab.

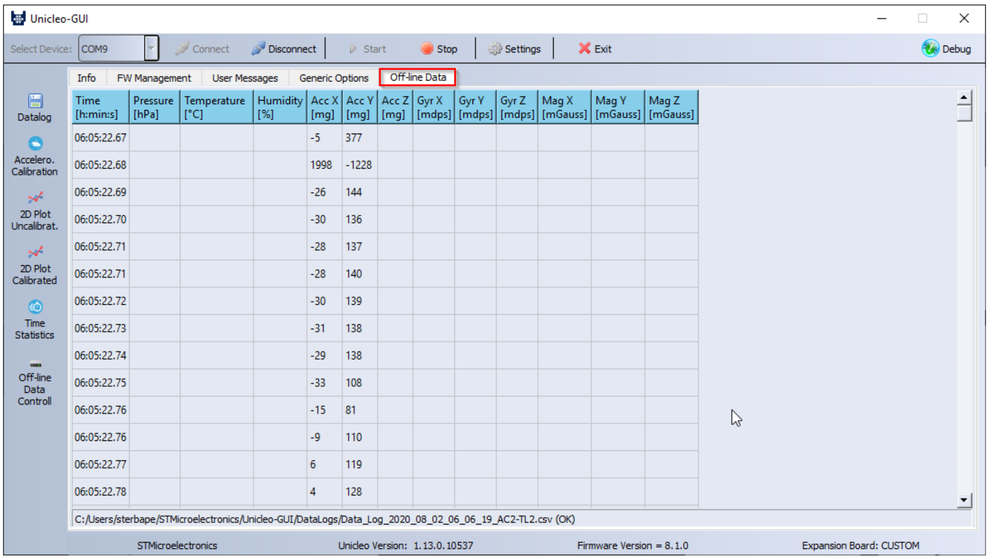


Figure 15. **Offline data control window**

Other buttons in the Offline Data Control menu bar are activated. You can click on:

- [**Offline Mode**] button to switch the firmware offline mode on/off.
- [**Start**]/[**Stop**]/[**Step**]/[**Repeat**] buttons to control the data sent by Unicleo-GUI to the firmware.
- [**Clear**] button to remove data from Unicleo-GUI.

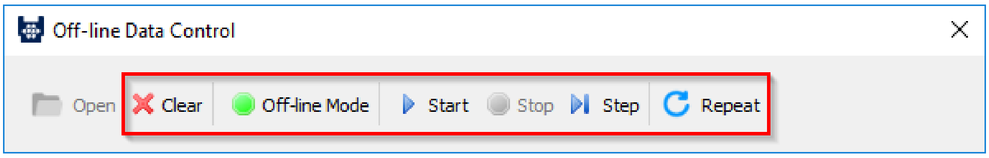


Figure 16. **Offline Data Control window - other buttons**

4 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. 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
08-Sep-2020	1	Initial release.

Contents

List of tables

List of figures