invn.algo.sw-integration - InvnAlgoAMLLib documentation

1.6.0

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

Module Documentation

3.1 AML

AIR MOTION Library provides pointing from 6-axis sensor and additionally swipe motion recognition.

Data Structures

- struct InvnAlgoAMLInput

    AML input structure (6-axis raw data sensor and click button)
- struct InvnAlgoAMLOutput

    AML output structure (calibrated sensor offsets, pointing output, swipe gestures detected and quaternion)
- struct InvnAlgoAMLConfig

    AML configuration structure (gain for X and Y axis, sensor offsets, sensor full scale range)

Functions

- const char ∗ invn_algo_aml_version (void)

    Return library version x.y.z-suffix as a char array.
- uint8_t invn_algo_aml_init (struct inv_imu_device ∗icm_device, const InvnAlgoAMLConfig ∗config)

    Initializes algorithms with default parameters and reset states.
- void invn_algo_aml_process (const InvnAlgoAMLInput ∗inputs, InvnAlgoAMLOutput ∗outputs)

    Performs algorithm computation.
- void invn_algo_aml_reset_swipe_recognition (void)

    Performs reset of swipes gesture recognition.

3.1.1 Detailed Description

AIR MOTION Library provides pointing from 6-axis sensor and additionally swipe motion recognition.

Warning

> Supported sampling frequency is [100 Hz]

3.1.2 Introduction

The Air Motion Library is a highly programmable API for managing mouse cursor motion from 3 axes gyroscope and 3 axes accelerometer. Embedded TDK proprietary algorithms are responsible for converting motion sensors data into delta X and delta Y pointer movements. The library is intended to be used in free space pointing devices to operate in-air point and click navigation, just like a classic 2D mouse will do on a desk. In addition to its pointing feature, Air Motion Library is also capable of swipe motion recognition: 'Up', 'Down', 'Left', 'Right', 'Clockwise' and 'CounterClockwise' swipes are currently supported.

3.1.3 AIR MOTION Library sensors referential

- When the device lies flat on its back, accelerometer must see +1G on Z axis

- When the device lies on its right side, accelerometer must see +1G on Y axis

- When the device is held vertically pointing down, accelerometer must see +1G on X axis

- A clockwise rotation around the gyroscope X axis generates positive values

- A clockwise rotation around the gyroscope Y axis generates positive values i.e. negative delta Y on screen (cursor moves to the top)

- A clockwise rotation around the gyroscope Z axis generates positive values i.e. positive delta X on screen (cursor moves to the right)

3.1.4 API

The AIR MOTION Library uses the following functions :

- invn_algo_aml_init() : This function must be called for library initialization, each time processing (re)starts.

- invn_algo_aml_process() : This function must be called each time new sensors values are available. The library is designed to run at 100Hz.

- invn_algo_aml_reset_swipe_recognition() : This function must be called to trigger each new swipe gesture detection.

3.1.5 Calibration

Gyroscope offsets values can fluctuate over time depending on various parameters. The library supports a continuous calibration which computes gyroscope offsets values in real time.

Each time the invn_algo_aml_process() is called, gyroscope instant "staticness" is checked. If the device is considered static, new gyroscope offset values are computed. The remote can then be considered as calibrated. You can check 'InvnAlgoAMLOutput.status' to track calibration status over time.

'InvnAlgoAMLConfig.gyr_fsr' parameter is used to calibrate motion algorithms. 'InvnAlgoAMLConfig.acc_fsr' parameter is used for roll-compensation feature.

3.1.6 RollCompensation

The Air Motion Library embeds a roll-compensation algorithm along with its pointing feature. Roll-compensation allows user's movement to be reliably reproduced on the screen, independently from the device roll orientation.

3.1.7 EasyClick

This feature allows user to perform more accurate and stable mouse clicks for a better experience. On click press, the library will freeze the pointer (dX and dY null) to avoid undesired movement. During this period, if the device movement quantity is too important, the pointer will be released sooner.

3.1.8 Function Documentation

3.1.8.1 invn_algo_aml_init()

```
uint8_t invn_algo_aml_init (
            struct inv_imu_device * icm_device,
            const InvnAlgoAMLConfig * config )
```

Initializes algorithms with default parameters and reset states.

Parameters

in	*icm_device*	Invensense IMU device pointer.
in	*config*	algo configuration structure in InvnAlgoAMLConfig.

Returns

    initialization success indicator.

Return values

0	Success
1	Fail

3.1.8.2 invn_algo_aml_process()

```
void invn_algo_aml_process (
            const InvnAlgoAMLInput * inputs,
            InvnAlgoAMLOutput * outputs )
```

Performs algorithm computation.

Parameters

in	*inputs*	algorithm input in InvnAlgoAMLInput.
out	*outputs*	algorithm output. Output status reports updated outputs in InvnAlgoAMLOutput.

3.1.8.3 invn_algo_aml_reset_swipe_recognition()

```
void invn_algo_aml_reset_swipe_recognition (
            void  )
```

Performs reset of swipes gesture recognition.

Allows to restart gesture processing without resetting pointing.

3.1.8.4 invn_algo_aml_version()

```
const char* invn_algo_aml_version (
            void  )
```

Return library version x.y.z-suffix as a char array.

Return values

library	version a char array "x.y.z-suffix"

Chapter 4

Data Structure Documentation

4.1 InvnAlgoAMLConfig Struct Reference

AML configuration structure (gain for X and Y axis, sensor offsets, sensor full scale range)

```
#include <invn_algo_aml.h>
```

Data Fields

- int8_t delta_gain [2]

    Delta gain values for X and Y axes.
- int16_t gyr_offset [3]

    Initial 3-axis raw gyroscope offsets [q4].
- uint16_t acc_fsr

    Accelerometer full scale range [g].
- uint16_t gyr_fsr

    Gyroscope full scale range [dps].
- uint8_t gestures_auto_reset

    En/disable gestures auto reset.

4.1.1 Detailed Description

AML configuration structure (gain for X and Y axis, sensor offsets, sensor full scale range)

The documentation for this struct was generated from the following file:

- invn_algo_aml.h

4.2 InvnAlgoAMLInput Struct Reference

AML input structure (6-axis raw data sensor and click button)

```
#include <invn_algo_aml.h>
```

Data Fields

- int16_t racc_data [3]

    3-axis raw accelerometer data
- int16_t rgyr_data [3]

    3-axis raw gyroscope data
- uint8_t click_button

    Click button state.

4.2.1 Detailed Description

AML input structure (6-axis raw data sensor and click button)

The documentation for this struct was generated from the following file:

- invn_algo_aml.h

4.3 InvnAlgoAMLOutput Struct Reference

AML output structure (calibrated sensor offsets, pointing output, swipe gestures detected and quaternion)

```
#include <invn_algo_aml.h>
```

Data Fields

- uint8_t status

    Mask to specify status outputs.
- int16_t gyr_offset [3]

    Computed 3-axis raw gyroscope offsets [q4].
- int8_t delta [2]

    Computed delta values for pointing.
- uint8_t swipes_detected

    Bitfield of detected swipes.
- int32_t quaternion [4]

    Computed W, X, Y and Z quaternion coefficients [q10].
- RemotePosition remote_position

    Orientation of remote.

4.3.1 Detailed Description

AML output structure (calibrated sensor offsets, pointing output, swipe gestures detected and quaternion)

The documentation for this struct was generated from the following file:

- invn_algo_aml.h

Index