FXOS8700 STM32 library is designed to measure tilt-compensated magnetic heading.

Library consists of two files: FXOS8700.c and FXOS8700.h and goes along with an example of usage.

How to use in code:

1)add

#include <FXOS8700.h>

and

#include <math.h>

to your project

2)create a variable to store calibration result for exapmple:

CALIBRATIONRESULT calResBuf;

3)create a variable to store the measured magnetic heading for example:

int32\_t magneticHeading;

4)start calibration procedure:

calibrateFXOS8700(&calResBuf,1);

use 1 as a second parameter to perform precise approximation algorithm, and use 0 for the regular one which is about 10 times faster.

The regular algorithm approximates the pointcloud with an ellipsoid using the gradient approximation method.

The precise algorithm in addition to the regular one varies all parameters in the same iteration loop witch leads to the more accurate solution. It’s better to use precise algorithm when less than 20 calibration points used.

5)get tilt-compensated heading sending previously obtained calibration result as a parameter:

magneticHeading = FXOS8700\_GET\_MAGNETIC\_HEADING(calResBuf);

Calibration procedure.

When the calibration procedure is started, all parameters are being measured and result is being filtrated with a low-pass filter (#define InputLPFKoeff 300).

If the acceleration measure changes over time less than (#define AccCalThreshold 1) and the distance between the last calibration point and the current point exceeds (#define CalibrationDistanceThreshold 200) so the new calibration point added.

In another words, change device position and leave it with no movement so the algorithm calculates no movement and it is possible to add a new calibration point. If you configure InputLPFKoeff as 2 and AccCalThreshold as 20 the calibration becomes possible with the slow rotation of the device holding in hand, but way more calibration points needed, about 100 (see #define NumberOfCalibrationPoints 10) and it is les accurate.

You may use up to 255 calibration points.

InputLPFKoeff 300 provides about 2 measures per second with 1 degree tolerance.

If InputLPFKoeff is defined as 2 so it’s the maximum reading speed but tolerance in about 30 deg.

If you define as follows:

#define InputLPFKoeff 1000

#define AccCalThreshold 2

#define NumberOfCalibrationPoints 200

you will got a precise instrument with maximum measuring speed of about 1 time per 2-3 sec.