
STM32Cube 的 X-CUBE-MEMS1 扩展软件中 MotionFX 传感器融合库使用入门

前言

MotionFX 是 X-CUBE-MEMS1 软件的中间件库组件，可在 STM32 上运行。它可提供实时运动传感器数据融合。它还执行陀螺仪偏差和磁力计硬铁校准。

该算法以静态库格式提供，旨在用于基于 ARM® Cortex®-M4 架构的 STM32 微控制器。

它以 STM32Cube 软件技术为基础而构建，便于在不同 STM32 微控制器之间移植。

该软件附带了 NUCLEO-F401RE 或 NUCLEO-L476RG 开发板上的 X-NUCLEO-IKS01A1（可选 STEVAL-MKI160V1）或 X-NUCLEO-IKS01A2 扩展板上运行的示例实现。

目录

1	缩写和缩略语	5
2	MotionFX middleware library in X-CUBE-MEMS1 software expansion for STM32Cube	6
2.1	MotionFX 概述	6
2.2	MotionFX 库	6
2.2.1	MotionFX 库说明	6
2.2.2	MotionFX 6 轴和 9 轴传感器融合模式	6
2.2.3	MotionFX 库操作	7
2.2.4	MotionFX 库参数	7
2.2.5	融合算法库数据输出频率	9
2.2.6	MotionFX 库中的传感器校准	9
2.2.7	MotionFX API	11
2.2.8	API 流程图	14
2.2.9	演示代码	15
3	应用示例	16
3.1	Unicleo-GUI 应用程序	18
4	参考	21
5	版本历史	22

表格索引

表 1: 缩略语列表	5
表 2: 文档版本历史	22
表 3: 中文文档版本历史	22

图片索引

图 1: 传感器定向示例.....	8
图 2: 校准过程中的 STM32 Nucleo 板旋转.....	11
图 3: MotionFX API 逻辑时序	14
图 4: STM32 Nucleo: LED, 按钮, 跳线.....	16
图 5: Unicleo 主窗口	18
图 6: 用户消息选项卡.....	19
图 7: 融合窗口.....	19
图 8: 磁力计散点图 (正确校准的磁力计)	20
图 9: 数据记录窗口	20

1 缩写和缩略语

表 1：缩略语列表

缩略语	说明
API	应用编程接口
BSP	板级支持包
GUI	图形用户界面
HAL	硬件抽象层
IDE	集成开发环境

2 MotionFX middleware library in X-CUBE-MEMS1 software expansion for STM32Cube

2.1 MotionFX 概述

MotionFX 库扩展了 X-CUBE-MEMS1 软件的功能。

该库从加速度计、陀螺仪（6 轴融合）和磁力计（9 轴融合）获取数据，并提供实时运动传感器数据融合。

无论环境条件如何，MotionFX 滤波和预测软件都利用高级算法智能地对来自多个 MEMS 传感器的输出进行整合，可以实现最佳性能。

X-NUCLEO-IKS01A2 和 X-NUCLEO-IKS01A1（可选 STEVAL-MKI160V1）扩展板上提供了一个示例实现，安装在 NUCLEO-F401RE 或 NUCLEO-L476RG 开发板上。

2.2 MotionFX 库

在“Documentation”文件夹的 HTML 文件（使用 MotionFX_Package.chm 编译）中，提供了完整描述 MotionFX API 的功能和参数的技术信息。

2.2.1 MotionFX 库说明

MotionFX 传感器融合库可管理从加速度计、陀螺仪和磁力计传感器获取的数据；它能够：

- 实时 9 轴运动传感器数据融合（加速度计，陀螺仪，磁力计）
- 实时 6 轴运动传感器数据融合（加速度计，陀螺仪）
- 计算旋转、四元数、重力和线性加速度数据
- 陀螺仪偏差校准
- 磁力计硬铁校准
- 建议传感器数据采样频率为 100 Hz
- 45 kB 的代码和 8 kB 的数据存储器
实际空间大小对于不同的 IDE（工具链）可能不同
- 适用于 ARM Cortex-M4 架构

2.2.2 MotionFX 6 轴和 9 轴传感器融合模式

MotionFX 库实现了传感器融合算法，用来估算空间中的 3D 方向。它使用基于 Kalman 理论的数字滤波器来融合来自多个传感器的数据，并补偿单个传感器的限制。例如，陀螺仪数据可能会漂移，这会影响定向估计；这个问题可以通过使用磁力计提供绝对定向信息来解决。

类似地，磁力计带宽不高，并且受磁力干扰，但是这些弱点可利用陀螺仪来进行补偿。

9 轴传感器融合使用来自加速度计、陀螺仪和磁力计的数据，并提供 3D 空间中的绝对定向，包括航向（即磁性北方向）。

6 轴传感器融合仅使用加速度计和陀螺仪数据。它对计算要求较低，但不能提供有关器件绝对定向的信息。

6 轴传感器融合适用于快速移动（例如，游戏中）并且不需要绝对定位时的场景。

2.2.3 MotionFX 库操作

MotionFX 库将 6 轴和 9 轴传感器融合算法集成在一个库中；甚至可以同时运行它们。

该库实现了传感器融合计算相关的以下关键内部函数：

1. `MotionFX_propagate` 是用于估计 3D 空间中定向的预测函数；在此阶段给予了陀螺仪数据更高权重。
2. `MotionFX_update` 是校正函数，可在需要时调整预测值；这个阶段给予了加速度计和磁力计数据更高权重。

每当调用 `MotionFX_propagate` 时，都可调用 `MotionFX_update` 函数，在计算能力较小的系统中不经常调用该函数。

`MotionFX_update` 函数大约比 `MotionFX_propagate` 函数多花费三倍的 MCU 计算时间。

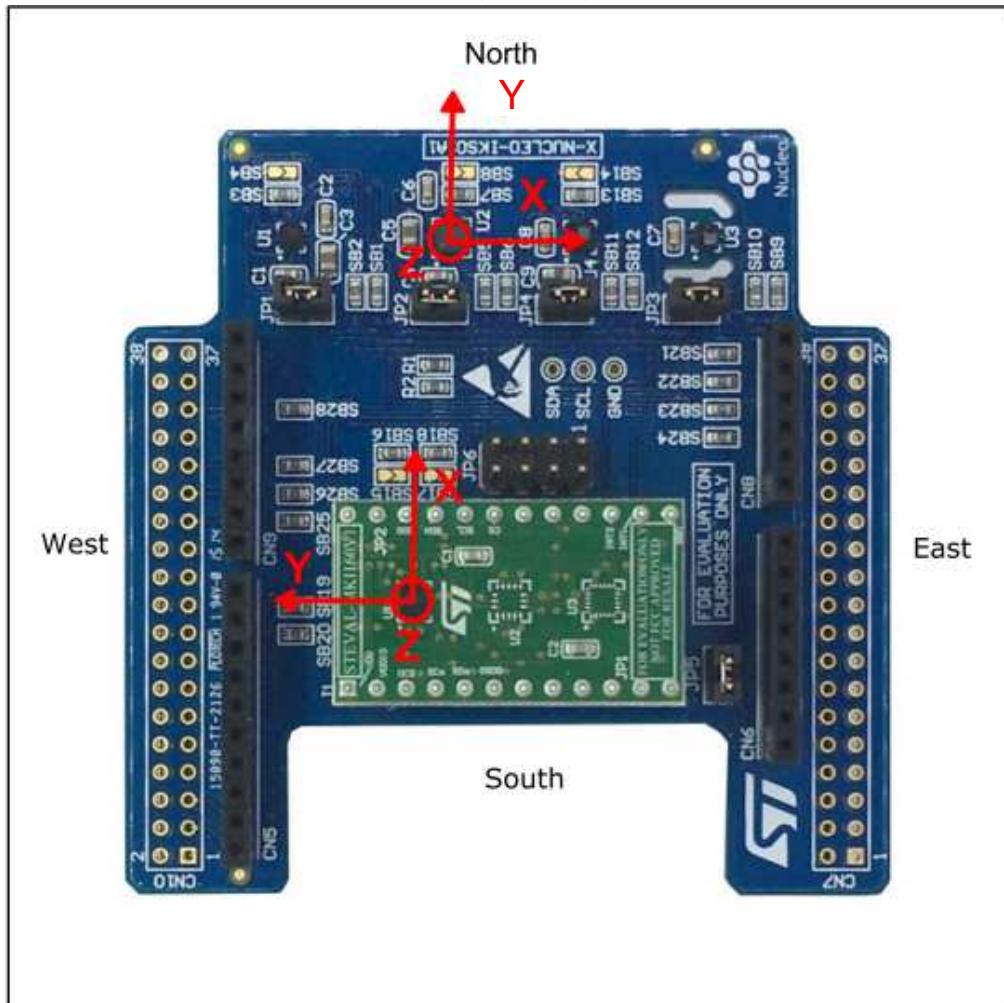
2.2.4 MotionFX 库参数

MotionFX 库使用 `MFX_knobs_t` 结构体进行“参数化”。

结构体参数如下：

- `ATime`、`MTime`、`FrTime` 表示用于预测的传感器（信任因子）的加权稳定性，从 0 到 1。建议使用默认值。
- `LMode` 代表陀螺仪偏差学习模式；库能够自动监测校准条件并进行零偏校准。此可能的参数值是：
 - `LMode = 0` – 学习关闭；如果陀螺仪已校准，则使用此模式
 - `LMode = 1` – 静态学习；仅当系统不动时进行学习
 - `LMode = 2` – 动态学习；系统移动时也进行学习
- `gbias_acc/gyro/mag_th_sc_6X`, `gbias_acc/gyro/mag_th_sc_9X` 表示阈值，低于这些阈值时 `gbias` 算法自动启动。这些值应通过测试来确定（不同型号器件对应不同的阈值）。示例项目中的值通常是正确的
 - `modx` 表示 `MotionFX_update` 调用频率的抽取
 - `output_type` 用于定义融合库输出数据的坐标系统类型：0 = NED, 1 = ENU
 - `start_automatic_gbias_calculation` 表示一个标志，当该标志设置为 1 时可重新启动陀螺仪偏差校准
 - `acc/gyro/mag_orientation` 是由三个字符组成的 acc/gyro/mag 数据定向字符串，指示速度计数据输出使用的参考框架的每个正定向的方向，顺序为 x、y、z。有效值为：n（北）或 s（南），w（西）或 e（东），u（上）或 d（下）。如下图所示，X-NUCLEO-IKS01A1 加速度计传感器采用 ENU 定向（x - 东，y - 北，z - 上），因此字符串为“enu”，而 STEVAL-MKI160V1 中的加速度计传感器为 NWU 定向（x-北，y-西，z-上），字符串为“nwu”。

图 1：传感器定向示例



`MotionFX_propagate` 和 `MotionFX_update` 函数从 `MFX_input_t` 结构体获取传感器数据：

- mag 表示校准后的磁力计数据, 以 $\mu\text{T}/50$ 计
 - acc 表示加速度计数据, 以 g 计
 - gyro 表示陀螺仪数据, 以 dps 计

MotionFX propagate 和 MotionFX update 将融合算法输出保存到 MFX output t 结构体中

2

- `rotation_6/9X` 表示系统定向为三角格式：偏航角，俯仰角，翻滚角
 - `quaternion_6/9X` 表示系统定向四元素格式；此格式可提供与 `rotation_6/9X` 同样的信息，但它在计算方面具有优势，因此通常用于其他算法（基于传感器融合）。
 - `gravity_6/9X` 表示从加速度数据中所得到的静态加速度（即，地球重力）向量
 - `linear acceleration 6/9X` 表示从加速度数据中得到的动态加速度（即，移动）向量。

2.2.5 融合算法库数据输出频率

正确设置传感器融合库输出数据速率非常重要；建议为 100 Hz。

输出数据速率为：

- 陀螺仪和加速度计应等于或大于 100 Hz；
- 磁力计可以更低 - 对于磁场传感器来说，通常 20/40 Hz 比较好。

可以根据 MCU/MPU 负载来调整库的系统要求。由于 MotionFX_update 函数需要的运算能力大约高于 MotionFX_propagate 函数的三倍，因此如果系统资源有限（例如，在嵌入式系统中），可以用比库输出数据速率更低的频率调用它。

使用 MFX_knobs_t 结构中的 modx 参数来降低 MotionFX_update 函数调用频率。例如，将 modx 设置为 2，实现每调用 MotionFX_propagate 函数两次，可调用 MotionFX_update 函数一次。

推荐设置为：

- modx = 1，适用于平板电脑或其他具有 MCU/MPU 的系统和 STM32F4；
- modx = 2，适用于 STM32F1。

2.2.6 MotionFX 库中的传感器校准

陀螺仪和加速度计校准

除了要求极高定向精度的应用之外，传感器融合不需要加速度计校准；加速度计校准可根据重力方向使系统调整为几种位置。

陀螺仪校准通过连续补偿零速率偏移效应由 MotionFX 库自动处理。

磁力计校准

MotionFX 库包含了用于磁力计硬铁校准的例程。

磁力计受下述硬铁和软铁现象的影响。

硬铁畸变

硬铁畸变通常由具有永久磁场的铁磁材料产生，该永久磁场是所用物体（例如，平板）的一部分。这些材料可以是永磁体或磁化铁或钢。它们是时不变的，并且使局部磁场发生畸变，在不同方向上产生不同偏移。

由于每块板可以被不同地磁化，因此硬铁效应是专门针对于单块板的。

如果您在理想环境下（无硬铁/软铁畸变）将板移动到近似（尽可能接近）3D 球体的空间，并画出所采集的磁传感器原始数据，其结果将是无偏移的完美球体。

硬铁畸变效应是使球体沿着 x、y 和 z 轴偏移；在 x-y 平面上，硬铁畸变利用理想圆原点到 (0, 0) 的偏移来识别，XY 和 XZ 轴的散点图能够用来确定是否存在偏移。

软铁畸变

软铁畸变由软磁材料或载流 PCB 走迹线产生。硬铁畸变是恒定的，与方向无关，而软铁畸变会随着对象在地球场中的取向而改变（软磁材料会改变其磁化方向）。

局部地磁场会发生变形，在不同方向上有不同增益。软铁效应是使理想的全圆球体畸变成倾斜的椭球体；在 x-y 平面上，软铁畸变利用原点在 XY 轴 (XZ) 的 (0, 0) 的倾斜椭球体来识别。

软铁效应在所有相同设计的板上是相同的，这就是为什么良好的 PCB 设计（考虑了磁力计放置，高电流迹线/其他部件间隙）通常可以避免软铁效应（对于 X-NUCLEO-IKS01A1 和 X-NUCLEO-IKS01A2 扩展板有效）。

校准过程

MotionFX 库磁力计校准库可补偿硬铁畸变。

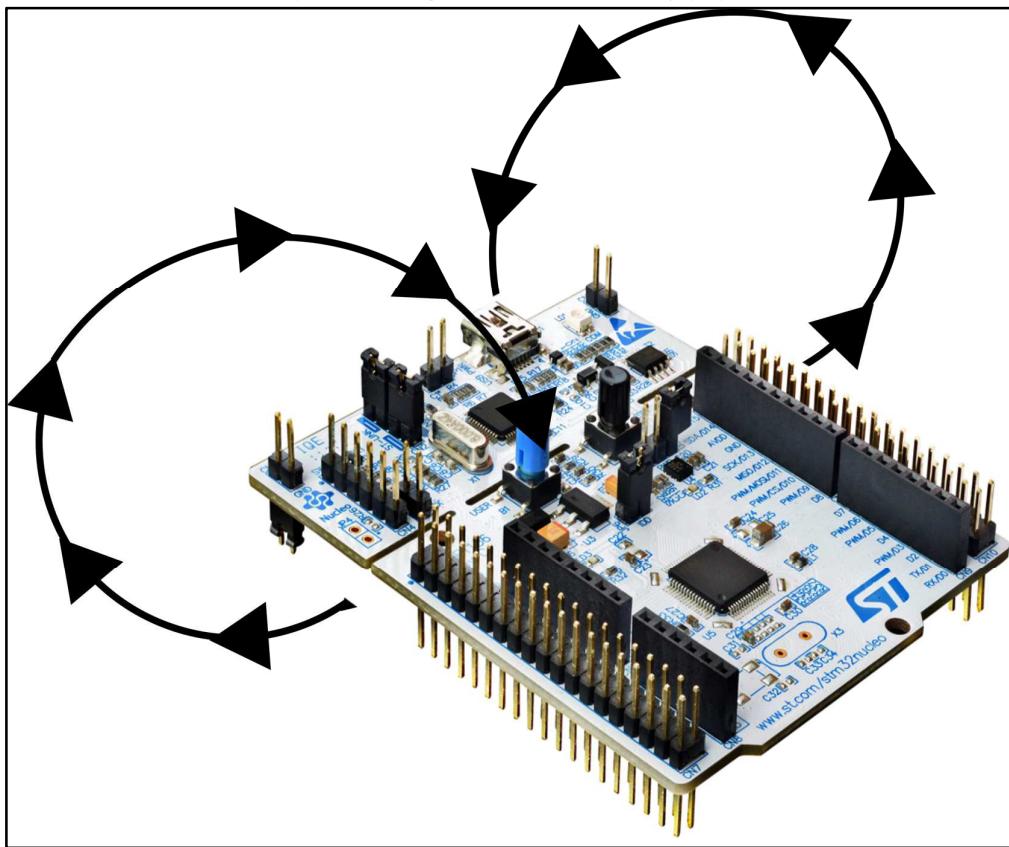
磁力计校准可以低于传感器融合输出数据速率（例如 25Hz）的频率执行。

要进行校准：

1. 初始化磁力计校准库 (MotionFX_MagCal_init)
2. 周期调用校准函数 (MotionFX_MagCal_run)，直至成功执行校准
3. 检查校准是否成功 (MotionFX_MagCal_getParams)；如果该函数返回 `mag_data_out.cal_quality = MFX_MAGCALGOOD`，则校准已成功执行
4. 应用校准结果：
 - `MAG_Calibrated.AXIS_X = MAG_Value.AXIS_X - MAG_Offset.AXIS_X`
 - `MAG_Calibrated.AXIS_Y = MAG_Value.AXIS_Y - MAG_Offset.AXIS_Y`
 - `MAG_Calibrated.AXIS_Z = MAG_Value.AXIS_Z - MAG_Offset.AXIS_Z`

校准程序初始化后，在空中进行划“8”字校准。进行此移动时，请将器件远离其他磁性物体，如手机、电脑和其他钢质物体。

图 2: 校准过程中的 STM32 Nucleo 板旋转



要检查校准是否正确执行, 请使用散点图检查磁力计数据 (应用校准结果后的)。

2.2.7 MotionFX API

MotionFX API 是:

- `uint8_t MotionFX_GetLibVersion(char *version)`
 - 检索库的版本
 - `*version` 是一个指针, 指向 35 个字符的数组
 - 返回版本字符串中的字符数
- `void MotionFX_Initialize(void)`
 - 执行 MotionFX 库的初始化和内部机制设置。



此函数必须在使用传感器融合库之前调用。

- `void MotionFX_SetKnobs(MFX_knobs_t *knobs)`
 - 设置内部 knobs
 - `*knobs` 参数是指向 knobs 的结构体的指针

- `void MotionFX_getKnobs(MFX_knobs_t *knobs)`
 - 得到当前内部 knobs
 - *knobs 参数是指向 knobs 的结构的指针
- `MFX_engine_state_t MotionFX_getStatus_6X(void)`
 - 得到 6 轴库状态
 - 如果使能则返回 1, 如果禁用则返回 0
- `MFX_engine_state_t MotionFX_getStatus_9X(void)`
 - 得到 9 轴库状态
 - 如果使能则返回 1, 如果禁用则返回 0
- `void MotionFX_enable_6X(MFX_engine_state_t enable)`
 - 使能或禁用 6 轴函数 (ACC + GYRO)
 - enable 参数为 1 则使能, 为 0 则禁用
- `void MotionFX_enable_9X(MFX_engine_state_t enable)`
 - 使能或禁用 9 轴函数 (ACC + GYRO+MAG)
 - enable 参数为 1 则使能, 为 0 则禁用
- `void MotionFX_setGbias(float *gbias)`
 - 设置初始 gbias
 - *gbias 参数是指向一个浮点数组的指针, 该数组中含有每个轴的陀螺仪偏差值
- `void MotionFX_getGbias(float *gbias)`
 - 得到初始 gbias
 - *gbias 参数是指向一个浮点数组的指针, 该数组中含有每个轴的陀螺仪偏差值
- `void MotionFX_update(MFX_output_t *data_out, MFX_input_t *data_in, float eml_deltatime, float *eml_q_update)`
 - 进行 Kalman 滤波更新
 - *data_out 参数是指向输出数据结构体的指针
 - *data_in 参数是指向输入数据结构体的指针
 - eml_deltatime 参数是两次传导调用之间的时间差[sec]
 - *eml_q_update 参数是一个设置为 NULL 的指针
- `void MotionFX_propagate(MFX_output_t *data_out, MFX_input_t *data_in, float eml_deltatime)`
 - 进行 Kalman 滤波传导
 - *data_out 参数是指向输出数据结构体的指针
 - *data_in 参数是指向输入数据结构体的指针
 - eml_deltatime 参数是两次传导调用之间的时间差[sec]
- `void MotionFX_MagCal_init(int sampletime, unsigned short int enable)`
 - 初始化磁力计校准库
 - sampletime 参数是更新函数调用的周期, 单位为毫秒[ms]
 - enable 参数用来使能 (1) 或禁用 (0) 库
- `void MotionFX_MagCal_run(MFX_MagCal_input_t *data_in)`
 - 运行磁力计校准算法
 - *data_in 参数是指向输入数据结构体的指针
 - 结构体类型 MFX_MagCal_input_t 的参数为:
 - mag 是未校准的磁力计数据[μ T]/50
 - time_stamp 为时间戳[ms]

- `void MotionFX_MagCal_getParams(MFX_MagCal_output_t *data_out)`
 - 得到磁力计校准参数
 - `*data_out` 参数是指向输出数据结构体的指针
 - 结构体类型 `MFX_MagCal_output_t` 的参数为:
 - `hi_bias` 为硬铁偏移数组[μ T]/50
 - `cal_quality` 为校准品质因数:
 - `MFX_MAGCALUNKNOWN` = 0; 校准参数精度未知
 - `MFX_MAGCALPOOR` = 1; 校准参数精度差, 不可信
 - `MFX_MAGCALOK` = 2; 校准参数精度可以
 - `MFX_MAGCALGOOD` = 3; 校准参数精度良好

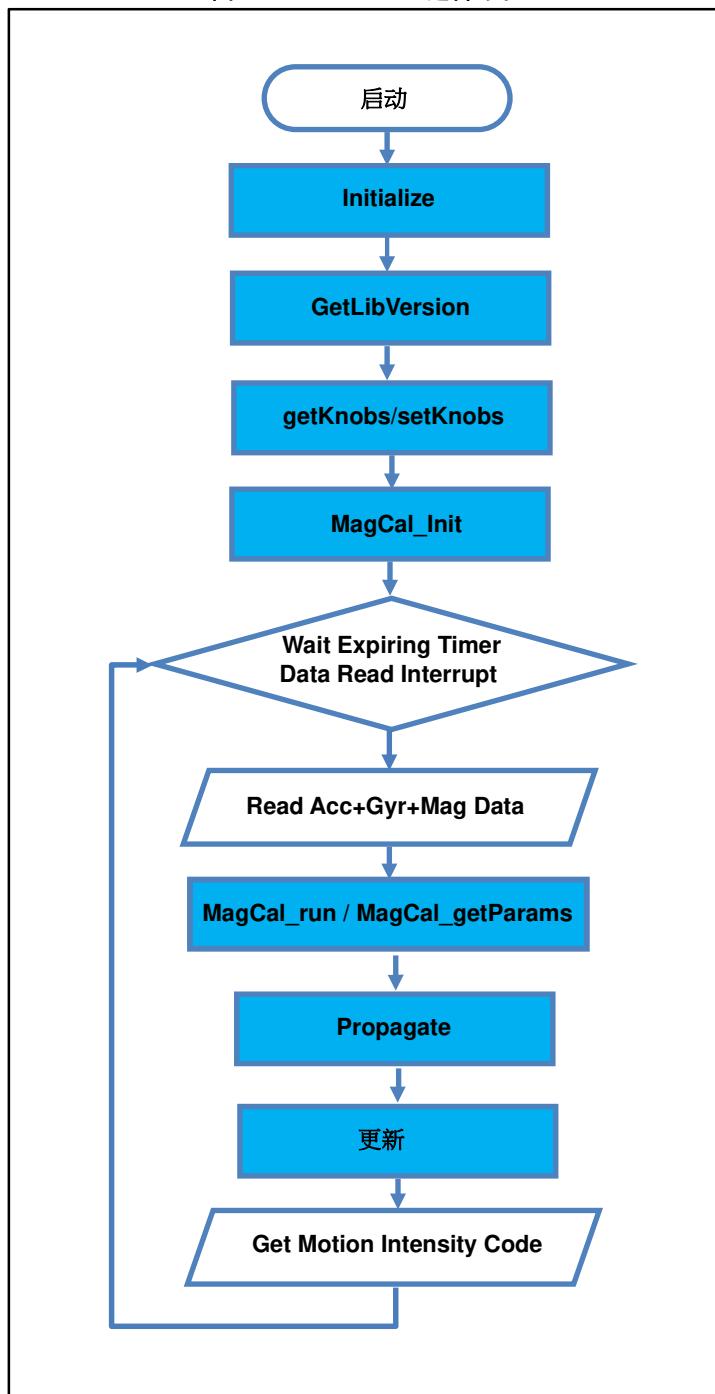
存储和加载磁力计校准参数

对于每个目标平台必须专门执行以下函数:

- `char MotionFX_LoadMagCalFromNVM(unsigned short int dataSize, unsigned int *data)`
 - 此函数用来从存储数据中取回校准参数, 当磁力计校准库使能时调用此函数
 - `dataSize` 是数据的字节数
 - `*data` 是数据位置指针
 - 正确加载则返回 0, 否则返回 1
- `char MotionFX_SaveMagCalInNVM(unsigned short int dataSize, unsigned int *data)`
 - 此函数用来存储校准参数, 当磁力计校准库被禁用时调用
 - `dataSize` 是数据的字节数
 - `*data` 是数据位置指针
 - 正确保存则返回 0, 否则返回 1

2.2.8 API 流程图

图 3: MotionFX API 逻辑时序



2.2.9 演示代码

以下演示代码从加速度计、陀螺仪和磁力计传感器读取数据，并获得旋转、四元数、重力和线性加速度。

```
[...]
#define VERSION_STR LENG 35
#define MFX_DELTATIME 10
[...]

/** Initialization */
char lib_version[VERSION_STR LENG];
char acc_orientation[3];
MFX_knobs_t iKnobs;

/* Sensor Fusion API initialization function */
MotionFX_initialize();

/* Optional: Get version */
MotionFX_GetLibVersion(lib_version);

MotionFX_getKnobs(&iKnobs);
/* Modify knobs settings */
MotionFX_setKnobs(&iKnobs);

/* Enable 9-axis sensor fusion */
MotionFX_enable_9X(MFX_ENGINE_ENABLE);

[...]

/** Using Sensor Fusion algorithm */
Timer_OR_DataRate_Interrupt_Handler()
{
MFX_input_t data_in;
MFX_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 angular rate X/Y/Z in dps */
MEMS_Read_GyroValue(data_in.gyro[0], data_in.gyro[1], data_in.gyro[2]);

/* Get magnetic field X/Y/Z in uT/50 */
MEMS_Read_MagValue(data_in.mag[0], data_in.mag[1], &data_in.mag[2]);

/* Run Sensor Fusion algorithm */
MotionFX_propagate(&data_in, &data_out, MFX_DELTATIME);
MotionFX_update(&data_in, &data_out, MFX_DELTATIME, NULL);
}
```

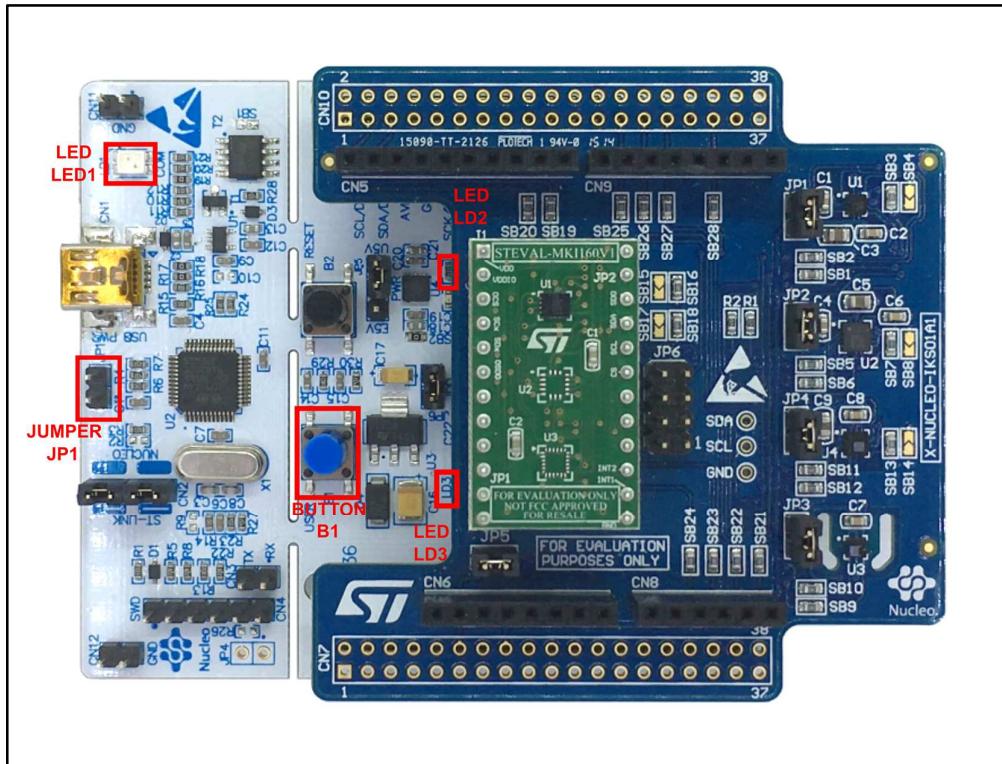
3 应用示例

可以轻松操作 MotionFX 中间件来构建用户应用程序。Application 文件夹中提供了一个示例应用程序。

根据设计，它可以在连接了 X-NUCLEO-IKS01A1 (基于 LSM6DS0) 或 X-NUCLEO-IKS01A2 (基于 LSM6DSL) 扩展板以及可选 STEVAL-MKI160V1 板 (基于 LSM6DS3) 的 NUCLEO-F401RE 或 NUCLEO-L476RG 开发板上运行。

该应用程序提供了实时运动传感器数据融合，并返回旋转、四元数、重力和线性加速度。

图 4: STM32 Nucleo: LED, 按钮, 跳线



上图显示了 NUCLEO-F401RE 板的用户按钮 B1 和三个 LED。板通电时，LED LD3 (PWR) 将打开。



为板供电后，LED LD2 闪烁一次，表示应用程序已准备就绪。

开始时，磁力计校准数据从 MCU 内部闪存加载，检查数据有效性并确保具有良好的校准质量。

如果数据有效且校准质量良好，则 LED2 接通，否则磁力计需要校准，LED2 关闭；这种情况下，初始化校准程序。

要校准磁力计，请执行图 8 的校准活动。



只有当传感器融合被激活时，才能校准磁力计。

按下用户按钮 B1 时，系统会清除存储在闪存中的原有磁力计校准数据，并再次启动校准程序。

一旦磁力计校准完成，在获得足够的数据后，LED2 会亮起，表示校准质量良好，并且校准数据会存储在闪存中。

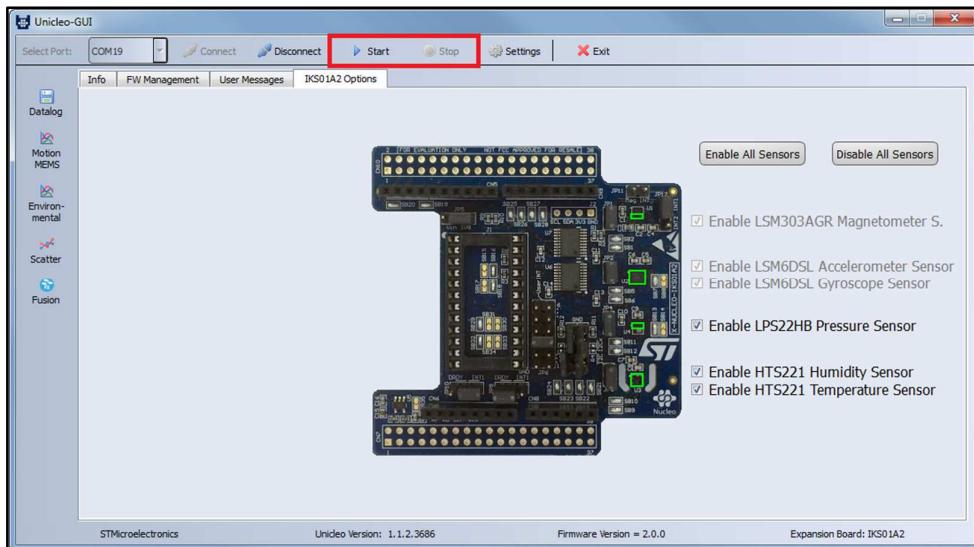
3.1 Unicleo-GUI 应用程序

示例应用程序使用 Windows Unicleo-GUI 实用工具，可从 www.st.com 下载。

- 1 确保已安装必要的驱动，并且 STM32 Nucleo 板以及适当的扩展板已连接 PC。
- 2 启动 Unicleo-GUI 应用程序，打开主应用程序窗口。

如果 STM32 Nucleo 板及所支持的固件连接到 PC，则会自动对其进行检测并打开相应的 COM 端口。

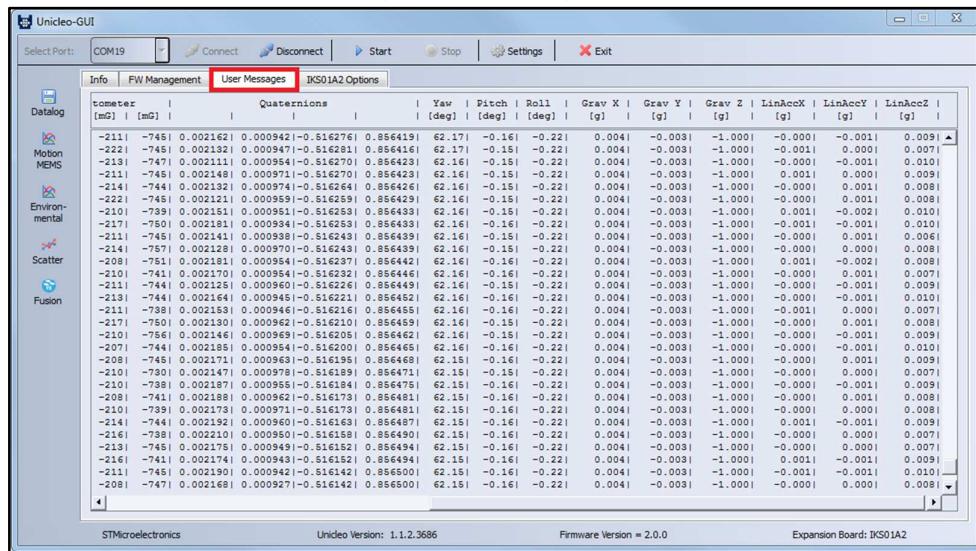
图 5: Unicleo 主窗口



- 3 使用垂直工具栏上的相应按钮开始和停止数据流。

来自所连接传感器的数据可以在“用户消息”选项卡中查看。

图 6: 用户消息选项卡

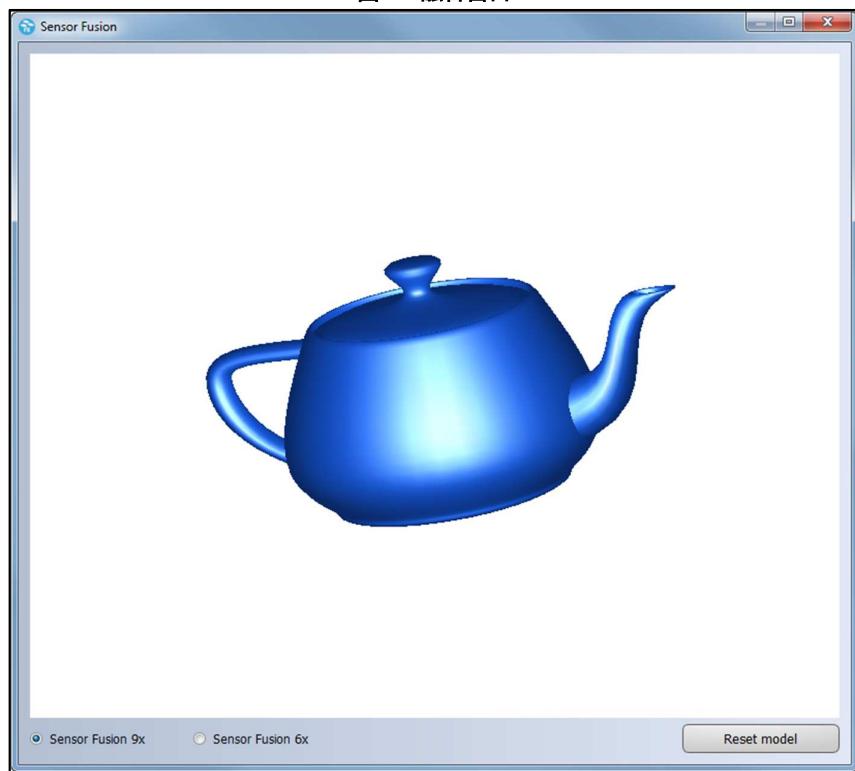


4 点击垂直工具栏中的 **Fusion** 图标打开特定应用程序窗口。

要在 9 轴和 6 轴传感器融合之间进行切换, 请点击相应的按钮。

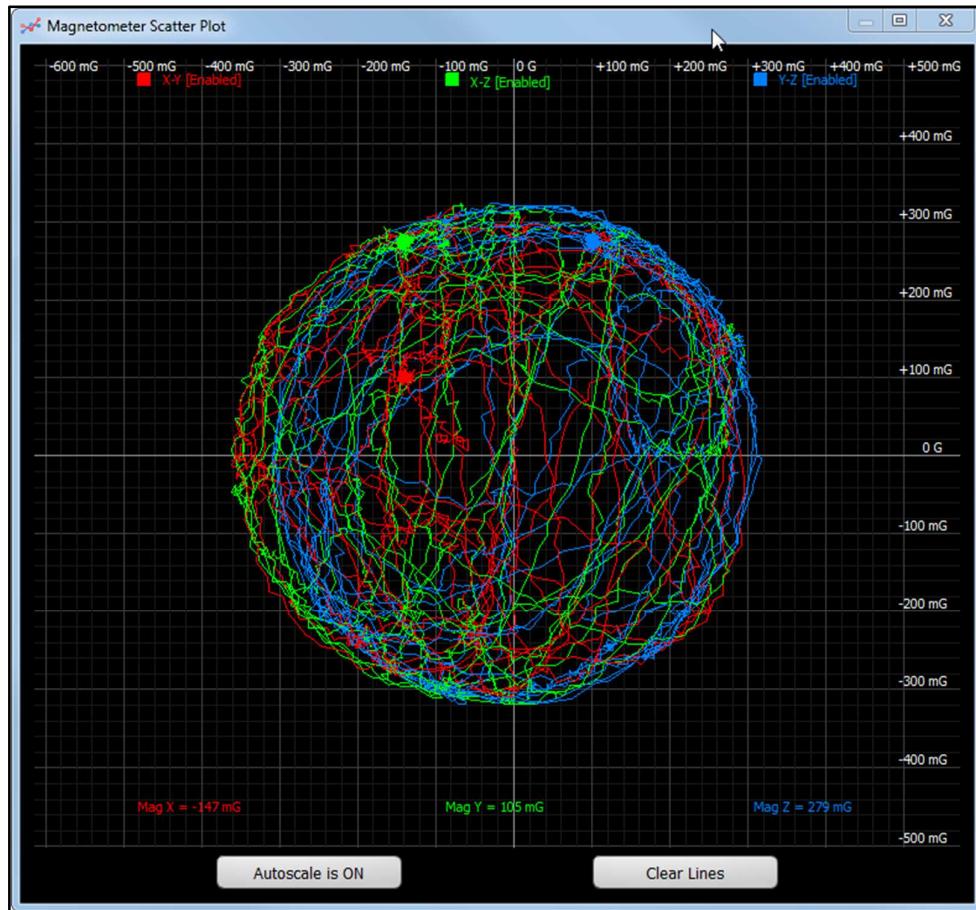
要对齐茶壶位置, 应使 Nucleo 板朝向屏幕, 并按下 **Reset** 模型按钮。

图 7: 融合窗口



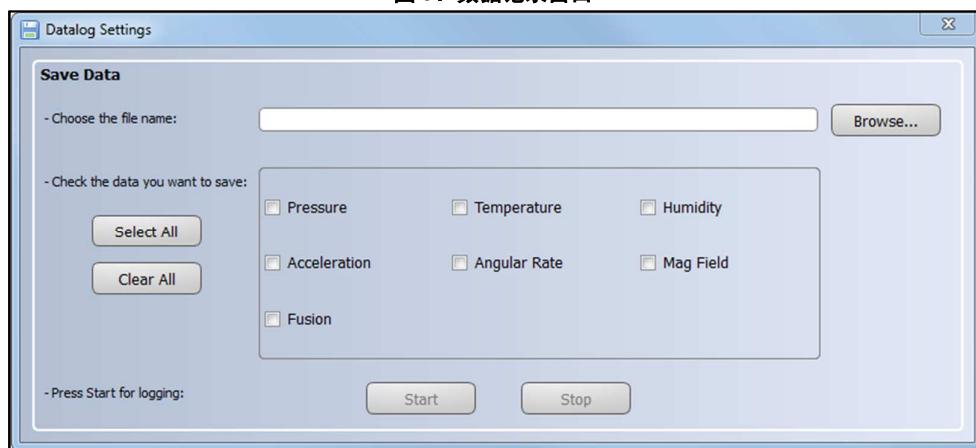
- 5 点击 Scatter Plot 图标，检查磁力计校准质量。

图 8：磁力计散点图（正确校准的磁力计）



- 6 点击垂直工具栏中的 Datalog 图标，打开数据记录配置窗口：您可以选择要保存在文件中的传感器和融合数据。可以通过点击相应的按钮来启动或停止保存。

图 9：数据记录窗口



4 参考

以下所有资源均可在 www.st.com 上免费获得。

1. UM1859: STM32Cube 的 X-CUBE-MEMS1 运动 MEMS 和环境传感器扩展软件使用入门
2. UM1724: STM32 Nucleo-64 板
3. UM2128: STM32Cube 的 Unicleo-GUI 运动 MEMS 和环境传感器扩展软件使用入门

5 版本历史

表 2：文档版本历史

日期	版本	变更
2017 年 5 月 12 日	1	初始版本。

表 3：中文文档版本历史

日期	版本	变更
2017 年 11 月 13 日	1	中文初始版本。

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。本文档的中文版本为英文版本的翻译件，仅供参考之用；若中文版本与英文版本有任何冲突或不一致，则以英文版本为准。

© 2017 STMicroelectronics - 保留所有权利