

## STM32Cube 的 X-CUBE-MEMS1 扩展软件中 MotionCP 实时携带位置库入门

### 引言

MotionCP 中间件库为 X-CUBE-MEMS1 软件的一部分，运行于 STM32 上。它提供了用户如何携带设备（如手机）的实时信息。它能够区别如下位置：在桌子上、在手上、在头部附近、在衬衫口袋、在裤子口袋、在手里摇晃、在夹克口袋。

该库仅可用于 ST MEMS。

该算法以静态库格式提供，旨在用于基于 ARM® Cortex®-M3 或 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。

该软件附带有在 NUCLEO-F401RE、NUCLEO-L476RG 或 NUCLEO-L152RE 开发板上的 X-NUCLEO-IKS01A2 或 X-NUCLEO-IKS01A3 扩展板上运行的实现示例。

## 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 STM32Cube X-CUBE-MEMS1 软件扩展中的 MotionCP 中间件库

### 2.1 MotionCP 概述

MotionCP 库扩展了 X-CUBE-MEMS1 软件的功能。

该库从加速度计获取数据，提供用户如何携带设备的信息。

该库专为 ST MEMS 而设计。在使用其他 MEMS 传感器时，不分析其功能和性能，并且可能与文档中的描述明显不同。

X-NUCLEO-IKS01A2 和 X-NUCLEO-IKS01A3 扩展板上提供了一个示例实现，安装在 NUCLEO-F401RE、NUCLEO-L476RG 或 NUCLEO-L152RE 开发板上。

### 2.2 MotionCP 库

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

#### 2.2.1 MotionCP 库说明

MotionCP 携带位置识别库管理从加速度计获取的数据；它具有如下功能：

- 能够区别如下位置：在桌子上、在手上、在头部附近、在衬衫口袋、在裤子口袋、在手里摇晃、在夹克口袋
- 仅基于加速度计数据识别
- 需要加速度计数据采样频率为 50 Hz
- 资源需求：
  - Cortex-M3: 6.7 kB 的代码和 11.0 kB 的数据存储器
  - Cortex-M4: 6.7 kB 的代码和 11.0 kB 的数据存储器
- 可用于 ARM® Cortex® -M3 和 ARM Cortex-M4 架构

#### 2.2.2 MotionCP API

MotionPE 库 API 为：

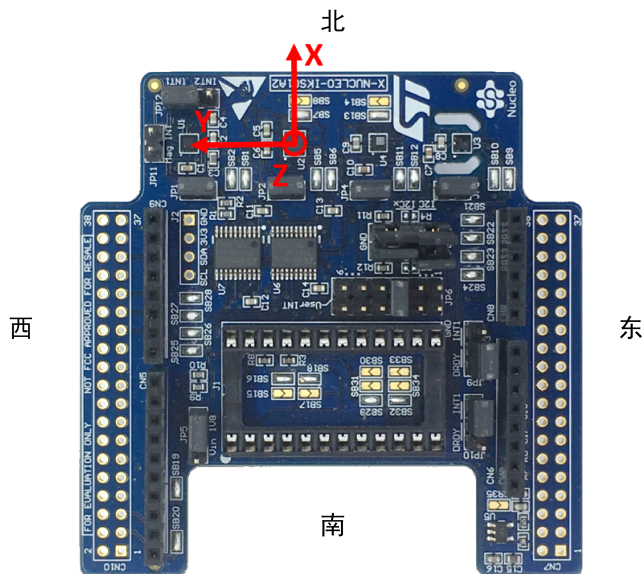
- `uint8_t MotionCP_GetLibVersion(char *version)`
  - 检索库版本
  - \*version 是一个指针，指向 35 个字符的数组
  - 返回版本字符串中的字符数
- `void MotionCP_Initialize(void)`
  - 执行 MotionCP 库的初始化和内部机制设置
  - 使用该库之前，应启用 STM32 微控制器中的 CRC 模块（在 RCC 外设时钟使能寄存器中）。

**提示** 此函数必须在使用携带位置库之前调用

- `void MotionCP_Update(MCP_input_t *data_in, MCP_output_t *data_out)`
  - 执行携带位置算法
  - `*data_in` 参数是指向输入数据结构的指针
  - 结构体类型 `MCP_input_t` 的参数为：
    - `AccX` 为 X 轴的加速度计传感器值，单位为 `g`
    - `AccY` 为 Y 轴的加速度计传感器值，单位为 `g`
    - `AccZ` 为 Z 轴的加速度计传感器值，单位为 `g`
  - `*Data_out` 参数为指向枚举型的指针，含有如下项：
    - `MPE_UNKNOWN = 0`
    - `MCP_ONDESK = 1`
    - `MCP_INHAND = 2`
    - `MCP_NEARHEAD = 3`
    - `MCP_SHIRTPOCKET = 4`
    - `MCP_TROUSERPOCKET = 5`
    - `MCP_ARMSWING = 6`
    - `MCP_JACKETPOCKET = 7`
- `void MotionCP_SetOrientation_Acc(const char *acc_orientation)`
  - 此函数用于设置加速度计数据方向
  - 配置通常在 `MotionCP_Initialize` 函数调用后立即执行
  - `*acc_orientation` 参数是由三个字符组成的字符串的指针，指示加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）。

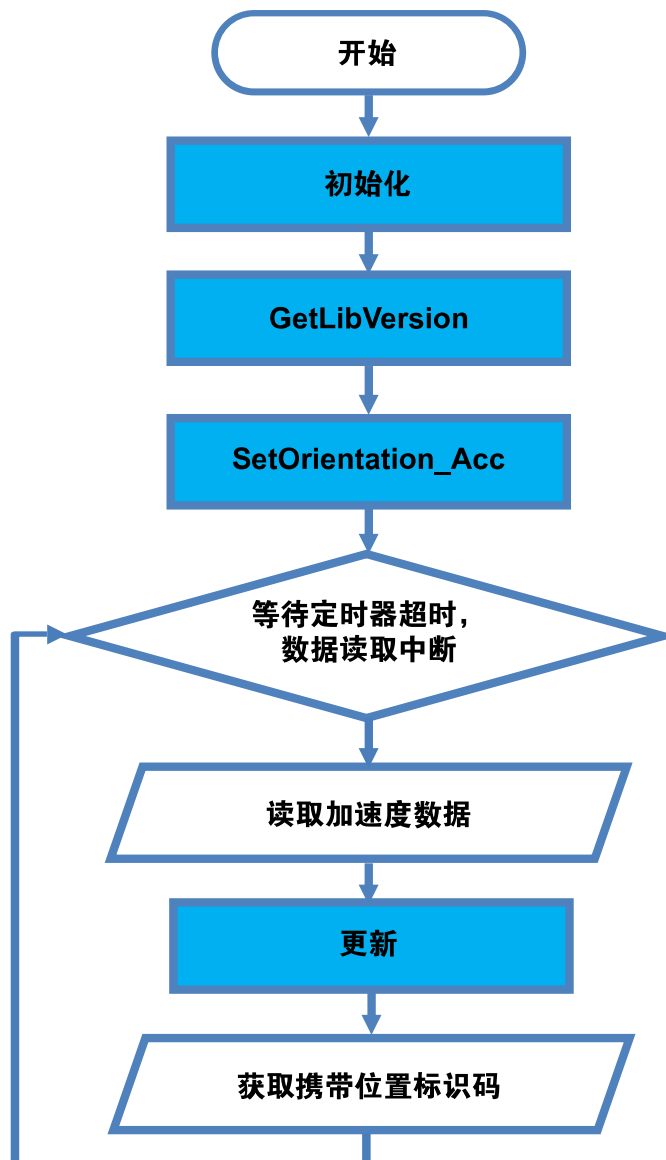
如下图所示，**X-NUCLEO-IKS01A2** 加速度计传感器具有 **NWU** 方向（`x`-北，`y`-西，`z`-上），所以字符串是：“`nwu`”。

图 1. 传感器定向示例



### 2.2.3 API 流程图

图 2. MotionCP API 逻辑时序



## 2.2.4 演示代码

下面的演示代码从加速度计传感器读取数据，得到携带位置代码。

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

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

/* Carry position API initialization function */
MotionCP_Initialize();

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

/* Set accelerometer orientation */
acc_orientation[0] = 'n';
acc_orientation[1] = 'w';
acc_orientation[2] = 'u';
MotionCP_SetOrientation_Acc(acc_orientation);

[...]

/** Using Carry Position algorithm **/
Timer_OR_DataRate_Interrupt_Handler()
{
  MCP_input_t data_in;
  MCP_output_t data_out;

  /* Get acceleration X/Y/Z in g */
  MEMS_Read_AccValue(&data_in.AccX, &data_in.AccY, &data_in.AccZ);

  /* Carry Position algorithm update */
  MotionCP_Update(&data_in, &data_out);
}
```

## 2.2.5 算法性能

携带位置识别算法只使用来自加速度计的数据，并以低频率（50 Hz）运行从而降低功耗。

因为算法对方向敏感，所以检测到的位置为手机典型的携带位置，尤其是在手中和接近头部位置。一些其它的携带位置（如在手中摇晃和裤子口袋）仅当用户在走路时才能检测到。

提示

当使用 **STM 32 Nucleo** 板模拟手机行为时，请确保 **USB** 连接器方向向下，和在手机中一样（见下图）。

图 3. STM32 Nucleo 和手机方向

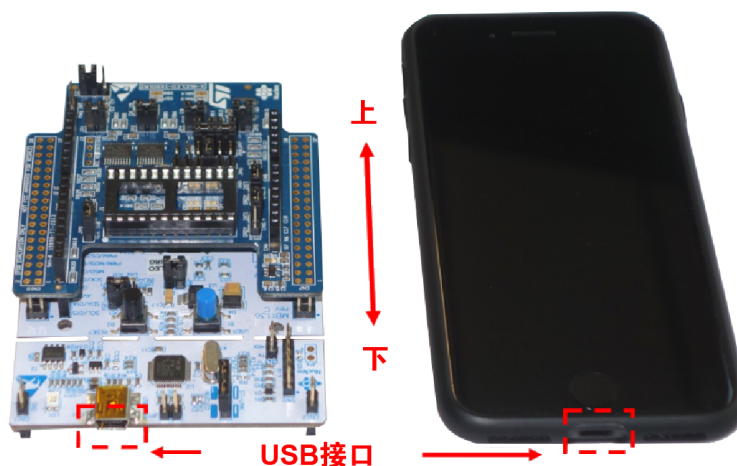


表 2. 算法性能数据

携带位置	检测概率（典型值） <sup>(1)</sup>	最佳性能	易受影响
在桌子上	95.31%	当手机在桌子上时的正常用例	易受持续振动影响，如敲打桌子或连续触碰手机
在手上	99.31%	正确的方向；即，当查看、阅读或打字时手机在手中的自然携带位置。可灵活适应静止、走路和快速走路的场景。	不考虑垂直方向/全部方向
在头附近	96.31%	正确方向，即，当打电话时携带手机的位置。可灵活适应静止、走路和快速走路的场景。	错误的方向
在衬衣口袋里	98.29%	可稳定适应走路和快速走路的场景。	对于静止场景，身体姿势决定了算法的性能
在裤子口袋里	98.68%	稳定适应走路场景，适用于前、后裤子口袋和手机在裤子口袋中携带时的多个方向。	静止
在手中摇晃	97.68%	行走	静止
夹克口袋	94.73%	行走	静止

1. 这些技术指标是典型值，不作保证。

典型的检测时延为 5 秒。

表 3. 运行时间（μs）算法

Cortex-M4 STM32F401RE @ 84 MHz									Cortex-M3 STM32L152RE @ 32 MHz								
SW4STM32 2.6.0 (GCC 7.2.1)			IAR EWARM 7.80.4			Keil μVision 5.24			SW4STM32 2.6.0 (GCC 7.2.1)			IAR EWARM 7.80.4			Keil μVision 5.24		
最小值	平均值	最大值	最小值	平均值	最大值	最小值	平均值	最大值	最小值	平均值	最大值	最小值	平均值	最大值	最小值	平均值	最大值
170	23	3606	150	201	3048	152	462	6562	518	754	11261	498	656	9566	502	1421	19656

## 2.3 应用示例

可以轻松操作 MotionCP 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。其设计目的是在 [NUCLEO-F401RE](#)、[NUCLEO-L476RG](#) 或 [NUCLEO-L152RE](#) 开发板（连接 [X-NUCLEO-IKS01A2](#) 或 [X-NUCLEO-IKS01A3](#) 扩展板）上运行。

应用实时识别携带位置。数据可通过 GUI 实时显示或储存在板上供离线分析。

算法能够识别如下位置：在桌子上、在手上、在头部附近、在衬衫口袋、在裤子口袋、在手里摇晃、在夹克口袋。

单机模式

在单机模式，应用示例让用户能够检测执行的姿势，并将其保存在 MCU flash 存储器中。

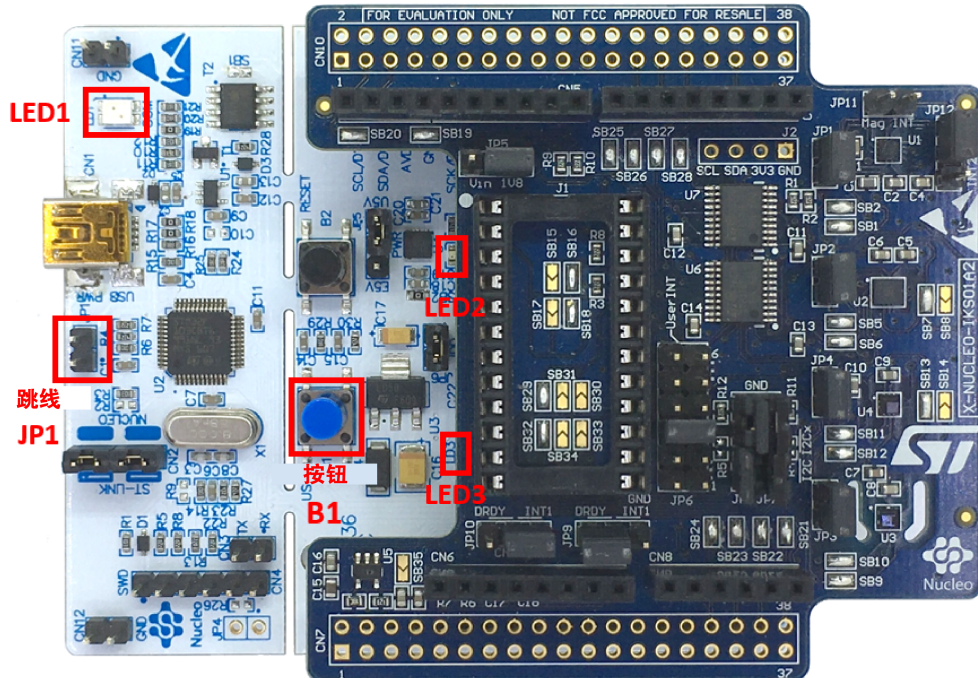
可通过便携式电池组为 STM32 Nucleo 板供电，以实现更舒适、轻便的用户体验且无任何 PC 连接。

表 4. 供电方案

电源	JP1 设置	工作模式
USB PC 连接线	JP1 断开	PC GUI 驱动模式
电池组	JP1 闭合	单机模式



图 4. STM32 Nucleo: LED, 按钮, 跳线



上图显示了 NUCLEO-F401RE 板上的用户按钮 B1 和三个 LED。电路板一旦上电，LED LD3 (PWR) 将点亮，由于缺失 USB 枚举，三色 LED LD1 (COM) 开始缓慢闪烁 (请参考 [www.st.com](http://www.st.com) 上的 UM1724 了解更多信息)。

提示

为板供电后，LED LD2 闪烁一次，表示应用程序已准备就绪。

在按下用户按钮 B1 时，系统开始从加速度计传感器采集数据并检测携带位置。在该采集模式下，LED LD2 快速闪烁，表明算法正在运行；检测到的用户姿势储存在 MCU 内部 flash 存储器中。为避免意外的电源故障导致过多数据损失，每隔五分钟自动保存一次数据。

第二次按下 B1 按钮会停止算法和数据存储，LED LD2 关闭。

再次按下按钮将再次启动算法和数据存储。

专用于数据存储的闪存扇区为 128 KB，可以存储超过 16,000 个数据集。

若需获取这些数据，必须将板连至运行 Unicleo-GUI 的 PC。在通过 GUI 检索保存的数据时，将清空专用于此目的的 MCU 闪存扇区。

如果板通电后 LED LD2 点亮，此为警告消息，表明 flash 存储器已满。

提示

可选择擦除 MCU 存储器，方法是按住用户按钮至少 5 秒。LED LD2 关闭然后闪烁 3 次，指示 MCU 中储存的数据已被擦除。这一选项仅在开机或复位板后有效，这时 LED LD2 点亮，表明 flash 存储器已满。

当应用运行于单机模式，若 flash 存储器已满，则应用切换至 PC GUI 驱动模式，LED LD2 关闭。

必须擦除 flash 存储器，方法是通过 Unicleo-GUI 或用户按钮下载数据 (见上面的注释)。

PC GUI 驱动模式

此模式需要使用 USB 连接线监控实时数据。板由 PC 通过 USB 连接供电。此工作模式允许用户使用 Unicleo-GUI 显示实时计数器值、加速度计和压力数据、时间戳和其它传感器数据。

在此工作模式，数据不储存在 MCU flash 存储器中。

## 2.4 Unicleo-GUI 应用程序

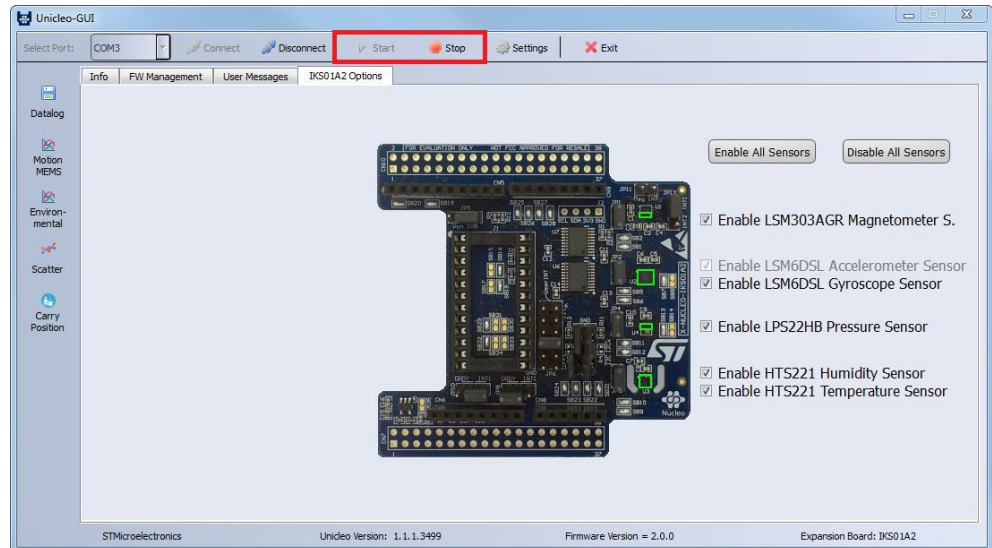
示例应用程序使用 Windows Unicleo-GUI 实用工具，可从 [www.st.com](http://www.st.com) 下载。

**Step 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。



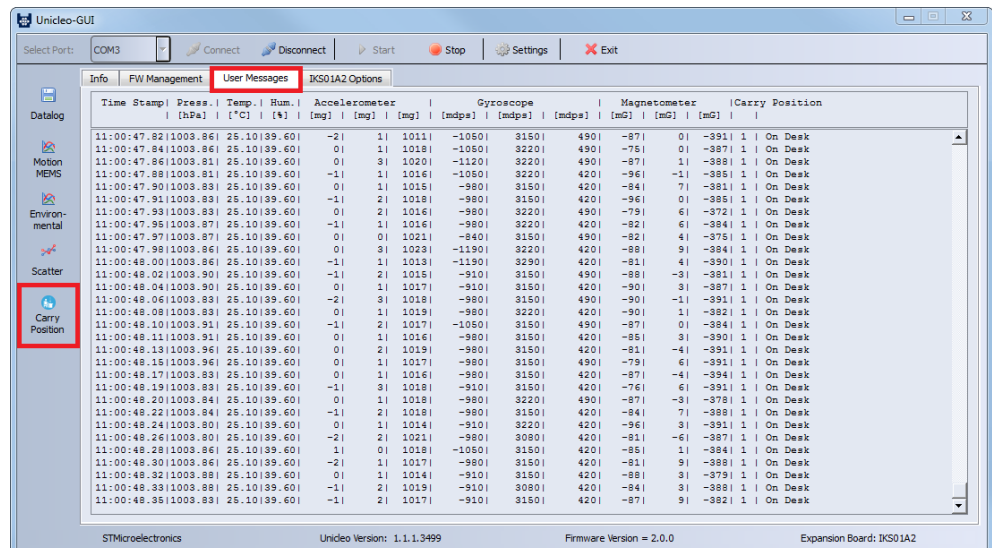
- Step 2.** 启动 Unicleo-GUI 应用程序，打开主应用程序窗口。  
如果 STM32 Nucleo 板及所支持的固件连接到 PC，则会自动对其进行检测并打开相应的 COM 端口。

图 5. Unicleo 主窗口



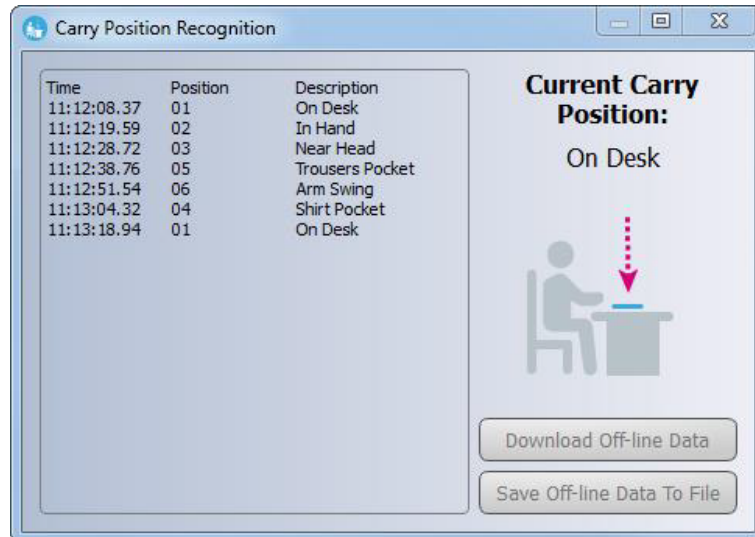
- Step 3.** 使用垂直工具栏上的相应按钮开始和停止数据流。  
来自所连接传感器的数据可以在“用户消息”选项卡中查看。

图 6. 用户消息选项卡



**Step 4.** 点击垂直工具栏中的携带位置图标打开特定应用程序窗口。

图 7. 携带位置窗口



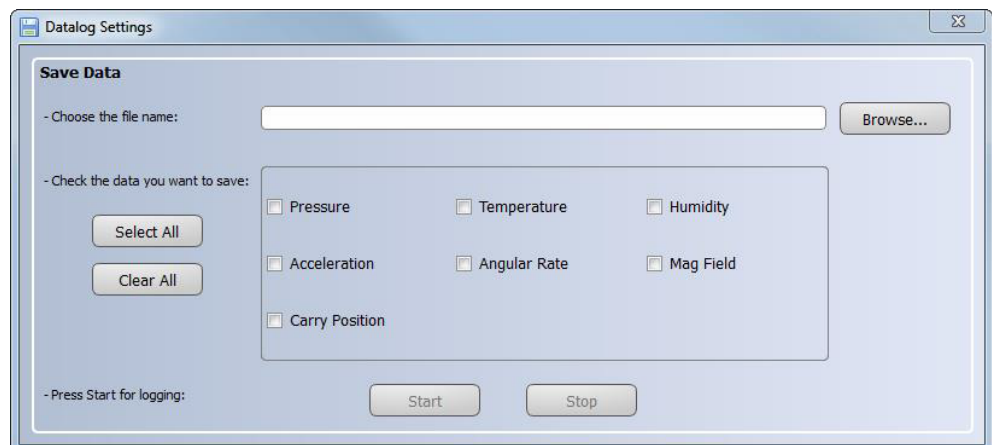
若板已经工作于单机模式时，用户想获取储存的数据，则可按下下载离线数据按钮，将储存的活动数据上传至应用。此操作会自动删除从微控制器获取的数据。

按下将离线数据储存至文件按钮可将上传的数据储存至 .tsv 文件中。

**Step 5.** 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活动数据。可以通过点击相应的按钮来启动或停止保存。

图 8. 数据记录窗口



### 3 参考

---

以下所有资源均可在 [www.st.com](http://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



版本历史

表 5. 文档版本历史

日期	版本	变更
2017 年 5 月 15 日	1	初始版本。
2018 年 1 月 25 日	2	增加了对 NUCLEO-L152RE 开发板和表 3. 运行时间 (µs) 算法的引用。
2018 年 3 月 20 日	3	更新了简介、第 2.1 节 MotionCP 概述和第 2.2.5 节算法性能。
2019 年 2 月 18 日	4	更新了图 1. 传感器定向示例、图 3. STM32 Nucleo 和手机方向、表 3. 运行时间 (µs) 算法和图 4. STM32 Nucleo: LED, 按钮, 跳线。 增加了 X-NUCLEO-IKS01A3 扩展板兼容性信息。

## 目录

<b>1</b>	缩写和缩略语 .....	<b>2</b>
<b>2</b>	STM32Cube X-CUBE-MEMS1 软件扩展中的 MotionCP 中间件库.....	<b>3</b>
<b>2.1</b>	MotionCP 概述.....	<b>3</b>
<b>2.2</b>	MotionCP 库.....	<b>3</b>
<b>2.2.1</b>	MotionCP 库说明.....	<b>3</b>
<b>2.2.2</b>	MotionCP API .....	<b>3</b>
<b>2.2.3</b>	API 流程图 .....	<b>4</b>
<b>2.2.4</b>	演示代码.....	<b>6</b>
<b>2.2.5</b>	算法性能.....	<b>6</b>
<b>2.3</b>	应用示例 .....	<b>7</b>
<b>2.4</b>	Unicleo-GUI 应用程序.....	<b>8</b>
<b>3</b>	参考.....	<b>11</b>
	版本历史 .....	<b>12</b>



表一览

表 1.	缩略语列表 .....	2
表 2.	算法性能数据 .....	7
表 3.	运行时间 (µs) 算法 .....	7
表 4.	供电方案 .....	7
表 5.	文档版本历史 .....	12



图一览

图 1.	传感器定向示例 .....	4
图 2.	MotionCP API 逻辑时序 .....	5
图 3.	STM32 Nucleo 和手机方向 .....	6
图 4.	STM32 Nucleo: LED, 按钮, 跳线 .....	8
图 5.	Unicleo 主窗口 .....	9
图 6.	用户消息选项卡 .....	9
图 7.	携带位置窗口 .....	10
图 8.	数据记录窗口 .....	10





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。

© 2019 STMicroelectronics - 保留所有权利