

## 基于 STM32Cube 的 STSW-BCNKT01 软件包使用入门(面向 STEVAL-BCNKT01V1 开发套件)

### 引言

BlueCoin 入门套件的 **STSW-BCNKT01** 固件包提供了用于开发自定义应用程序的示例项目。

它构建于 **STM32Cube** 软件技术之上，包括了所有底层驱动器，能够管理板上器件和系统级接口。

该软件包附带了 **Audio\_SD**、**DataLog**、**AudioLoop**、**BLE\_SampleApp** 和手势识别应用程序。

**Audio\_SD** 应用程序可将板载麦克风捕获的音频在 SD 卡上保存为常见的 **.wav** 文件。

**DataLog** 应用程序可以通过 USB（虚拟 COM 端口类）传输传感器的原始数据，并把的传感器数据存储在 SD 卡上。

**BLE\_SampleApp** 提供了一个蓝牙低功耗配置示例，使 **BlueCoin** 能够传输环境传感器数据；它可以与适用于 **Android** 和 **iOS** 操作系统的 **STBLESensor** 应用程序兼容。

手势识别应用程序利用 **Time-of-Flight** 测距传感器来检测目标物体的距离以及一些简单的手势，如定向滑动和点击。

## 1 什么是 STM32Cube?

STM32Cube™ 由意法半导体发起，通过减少开发工作量、时间和成本，让开发人员更轻松。STM32Cube 涵盖 STM32 产品系列。

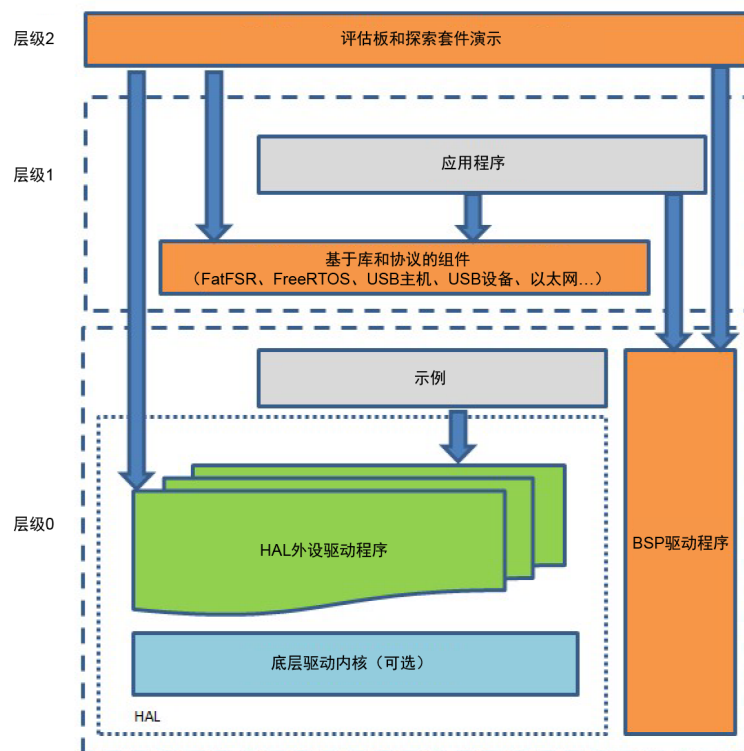
STM32Cube 1.x 版包括：

- 图形软件配置工具 STM32CubeMX，可通过图形向导生成初始化 C 代码。
- 针对于每个系列的综合嵌入式软件平台（例如 STM32F4 系列的 STM32CubeF4）包括：
  - 嵌入了 STM32Cube HAL 的抽象层软件，确保在 STM32 各个产品之间实现最大限度的可移植性
  - 一套一致的中间件，比如 RTOS、USB、TCP/IP 和图形
  - 所有嵌入式软件实用工具均配备一套完整的示例

### 1.1 STM32Cube 架构

STM32Cube 固件解决方案围绕三个独立层级构建，它们可以轻松地互相交互，如下图所示。

图 1. 固件架构



层级 0：此层级分为三个子层：

- 板级支持包（BSP）：该层提供了对应于硬件板中硬件组件的一系列 API（音频编解码器、IO 扩展器、触摸屏、SRAM 驱动器、LCD 驱动器、等）；它基于模块化架构，只需执行低层级例程，便可轻松移植到任何硬件上。它包含两部分：
  - 组件：是相对于板上外部设备的驱动程序，与 STM32 无关，组件驱动程序为 BSP 驱动程序的外部组件提供专用 API，并且可以移植到任何其他板子上。
  - BSP 驱动程序：将组件驱动程序链接到专用板上，并提供一组易于使用的 API。API 命名约定为 BSP\_FUNCT\_Action()：例如，BSP\_LED\_Init()、BSP\_LED\_On()。

- 硬件抽象层（HAL）：该层提供底层驱动程序和硬件接口的方法来与上层（应用程序、库和堆栈）进行交互。它提供了通用多实例且面向功能的 API，通过提供即用的进程来帮助用户减少应用程序的开发时间。例如，对于通信外设（I<sup>2</sup>C、UART 等），它提供了 API，用于外设初始化和配置，以及基于轮询、中断或 DMA 方式的数据传输管理和通信错误管理。HAL 驱动程序 API 分为两类：为所有 STM32 系列提供通用功能的通用 API，以及为特定系列或特定器件提供特殊定制功能的扩展 API。
- 基本外设使用示例：该层包含有围绕 STM32 构建外设（仅使用 HAL 和 BSP 资源）的示例。

层级 1：此层级分为两个子层：

- 中间件组件：一系列的库文件，包括：USB 主机和设备库、STemWin、FreeRTOS、FatFS、LwIP 和 PolarSSL。该层组件之间的水平交互是通过调用特征 API 来直接执行的，而与底层驱动程序的垂直交互是通过库系统调用接口中实现的特定回调函数和静态宏来管理的。例如，FatFs 实现磁盘 I/O 驱动程序，用来访问 microSD 驱动器或 USB Mass Storage Class。
- 基于中间件组件的示例：每个中间件组件都附带一个或多个示例（或应用程序）来显示如何使用它。还提供了使用多个中间件组件的集成示例。

层级 2：该层级是一个全局实时图形演示单层，基于中间件服务层、低层级抽象层和用于开发板基本功能的基础外设使用实例。

## 2 STM32Cube 的 STSW-BCNKT01 软件扩展

### 2.1 概述

该软件包扩展了 STM32Cube 平台的功能。

该软件包的主要功能是：

- 完整的固件套件在 [BlueCoin](#) 开发套件上构建应用程序，使用：
  - STM32F446 高性能微控制器
  - 运动传感器
  - 麦克风阵列
  - 压力传感器
  - 温度传感器
  - 接近传感器
- 基于 [STM32Cube](#)，一致且完整的 STM32 嵌入式软件，可最大限度地提高所有 STM32 系列产品之间的可移植性，并将用户从平台依赖性问题中解放出来。
- **Audio\_SD** 应用程序，可将板载麦克风捕获的音频在 SD 卡上保存为常见的 .wav 文件
- **DataLog** 应用程序允许通过串口将所有传感器数据实时传输到 PC，或将传感器数据保存/记录到 SD 卡的文件上
- **AudioLoop** 应用程序，通过 I<sup>2</sup>S 接口将麦克风采集的音频信号发送到 USB（Audio IN Class）和板载 DAC
- 手势识别应用程序，可用于配置 Time-of-Flight 测距传感器和手势检测中间件。
- **BLE\_SampleApp**，提供了一个蓝牙低功耗配置的示例
- 小型嵌入式系统的第三方 FAT 文件系统中间件
- 可以免费使用 [www.st.com](http://www.st.com) 网站提供的源代码

该软件可通过 I<sup>2</sup>C 或 I<sup>2</sup>S 实现不同传感器（如运动传感器、环境传感器和音频传感器）的数据采集。

利用所含 VCP USB 驱动程序的功能，该设备可被 Microsoft Windows 或任意 Unix 系统识别为虚拟 COM 端口。

如果将 SD 卡插入到相关连接器，也可以将数据保存在 SD 卡上。

### 2.2 架构

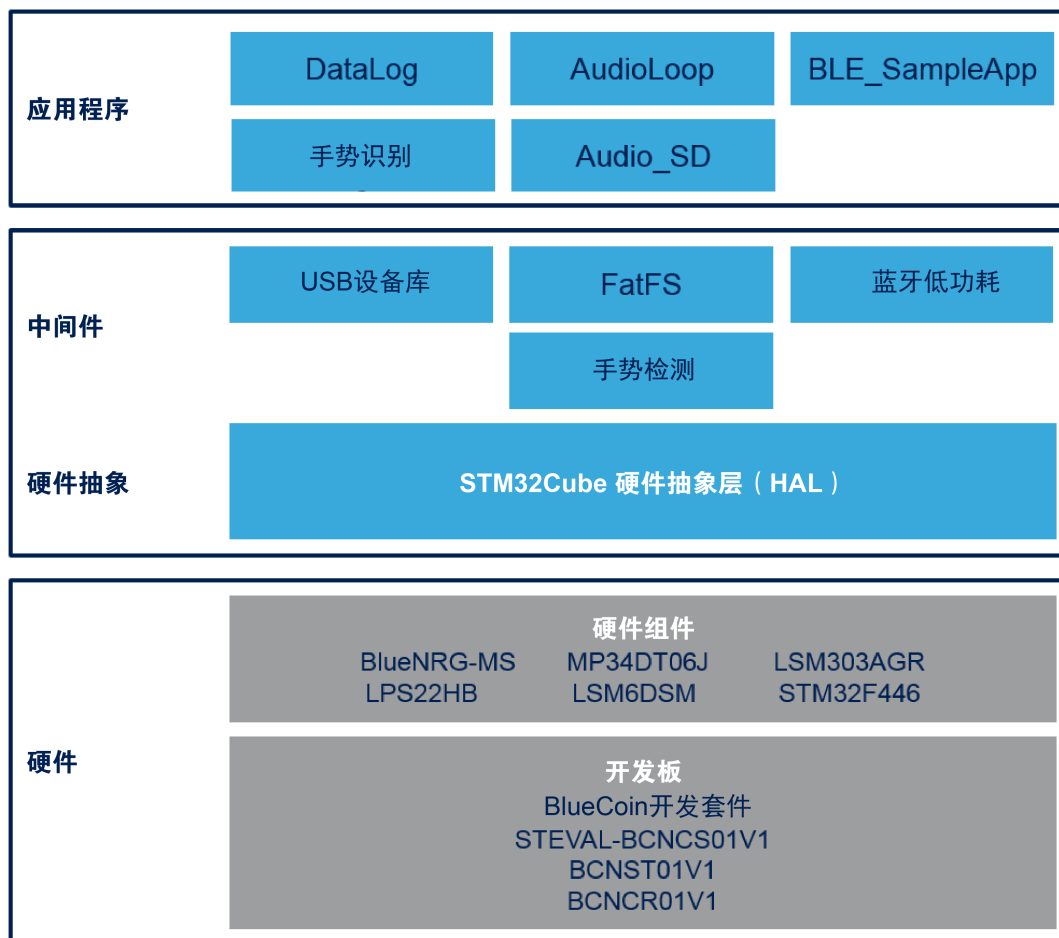
该软件是与 [STM32Cube](#) 完全兼容的扩展，可支持使用数字 MEMS 麦克风的应用开发。

该软件基于 STM32 微控制器的 STM32CubeHAL 硬件抽象层，并扩展了 STM32Cube，为麦克风扩展板提供了板级支持包（BSP），并可提供一些中间件组件用于音频处理和与 PC 的 USB 通信。

应用软件用来访问和使用麦克风扩展板的软件层是：

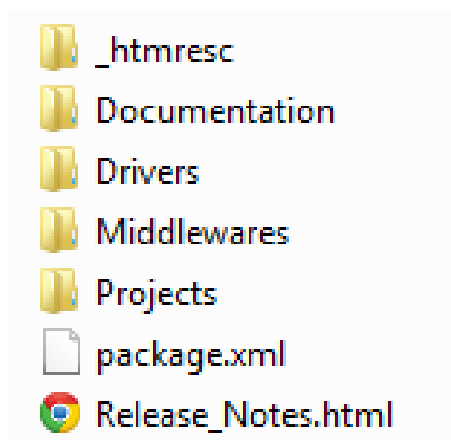
- **STM32Cube HAL 层**：提供通用、多实例的简单 API 集合（应用程序编程接口），以便与上层（应用、库和堆栈）交互。它由通用和扩展 API 构成。它直接围绕通用架构构建，允许在其基础上构建层（像中间件层），实现了它的功能又无需依赖给定微控制器单元（MCU）的特定硬件配置。此结构可提高库代码的可复用性，并确保可与其他设备轻松移植。
- **板级支持包（BSP）层**：包含支持除 MCU 外的 STM32 Nucleo 板外设的软件。

图 2. STSW-BCNKT01 软件架构



## 2.3 文件夹结构

图 3. STSW-BCNKT01 软件包文件夹结构



软件包包含下列文件夹：

- **Documentation:** 此文件夹包含从源码生成的已编译 HTML 文件，记录了软件组件和 API 详细信息。
- **Drivers:** 包括板级专用 HAL 驱动，可用于每个支持的开发板或硬件平台，包括板上组件，以及 ARM Cortex-M 系列处理器与 CMSIS 供应商无关的硬件抽象层。

- **Middlewares:** 包含用于虚拟 COM 端口 USB 驱动程序的 ST 库和第三方库 FatFS。
- **Projects:** 包含 DataLog 和 AudioLoop 示例应用程序，可以通过 ARM 的 IAR Embedded Workbench ([IAR-EWARM](#))、RealView 微控制器开发工具包 ([MDK-ARM-STM32](#)) 和面向 STM32 开发环境的 System Workbench ([SW4STM32](#)) 进行评估。

## 2.4 APIs

完整描述用户可用的 API 的详细技术信息，可以在位于软件包 **Documentation** 文件夹内已编译的 HTML 文件中找到。



## 2.5 STSW-BCNKT01 软件扩展应用程序

### 2.5.1 Audio\_SD

Audio\_SD 应用程序可将板载麦克风捕获的音频在 SD 卡上保存为常见的.wav 文件。

复位后，固件：

- 配置 HAL 和时钟
- 配置 SD 卡访问，基于 FatFS 中间件
- 配置传感器和麦克风
- 开始音频采集

如要打开文件和开始保存音频文件，需要在 BlueCoin 上按 SW2 按钮：LED 灯闪烁表示.wav 文件已经创建，并且记录已经初始化。

如要停止采集，必须再次按下 BlueCoin 按钮。

如果 LED 灯停止闪烁，则表示 *SensorTile\_Log\_N000.wav* 音频文件已正确保存在 SD 卡上。

现在可以重复以上操作并在另一个文件（*BlueCoin\_Log\_N001.wav*）重新启动数据记录。

### 2.5.2 DataLog

DataLog 应用程序有两种操作模式，可以在编译时通过更改 main.c 中的 LoggingInterface 变量来进行选择

- LoggingInterface = USB\_Datalog: 通过 USB 传输的传感器原始数据（虚拟 COM 端口类）
- LoggingInterface = SDCARD\_Datalog: 传感器原始数据存储在 SD 卡上。

复位后，固件：

1. 配置 HAL 和时钟
2. 初始化 USB 外设或 SDIO 以访问 SD 卡
3. 创建线程并激活 FreeRTOS 调度程序

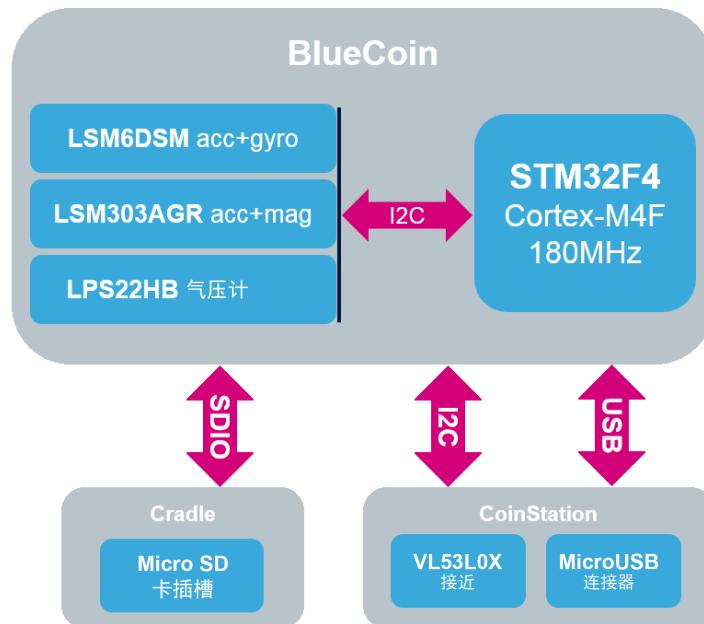
GetData\_Thread 和 WriteData\_Thread 线程由 FreeRTOS 以不同的优先级进行调度，并通过一个消息队列相互进行通信：

- GetData\_Thread: 高优先级任务，用来配置传感器、以给定频率读取数据，并将新数据推送入队列中。OS 定时器以给定频率触发线程执行。
- WriteData\_Thread: 低优先级任务，用来配置 SD 卡，并在传感器数据在队列中可用时立即将传感器数据写入。

这些优先级差异可以确保即使 SD 卡上的写入操作比采样周期长，应用程序也不会丢失传感器数据。如果发生这种情况，调度程序会挂起 WriteData\_Thread，以便以正确的时序来执行 GetData\_Thread。

如果选择了 SD 卡模式，则需要双击板子，开始将数据记录到 *BlueCoin\_Log\_N000.tsv* 上，然后再次双击来停止。后续操作会重新开始将数据记录到新文件（例如，*BlueCoin\_Log\_N001.tsv*）。

图 4. DataLog 应用程序框图



### 2.5.3

#### AudioLoop

**AudioLoop** 应用程序通过 I<sup>2</sup>S 和 USB 接口发送麦克风获取的音频信号，可允许用户在扬声器或耳机上播放这些声音，或将其录制到主机 PC 上。

复位后，固件：

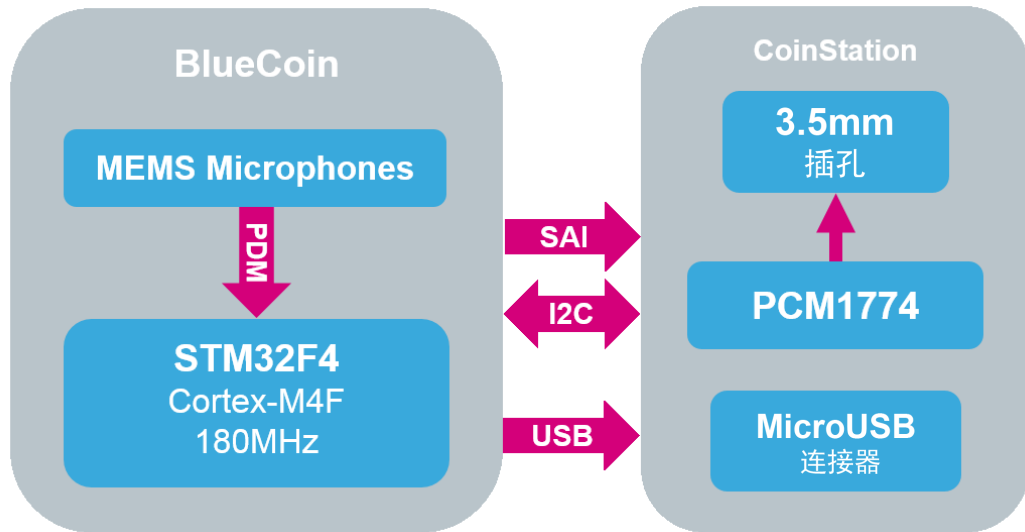
1. 配置 HAL 和时钟
2. 配置 LED1
3. 初始化 USB 外设
4. 将 STM32 串行音频接口外设（SAI）配置为 I<sup>2</sup>S 模式并配置外部 DAC
5. 配置 PDM 到 PCM 算法；BSP\_AUDIO\_IN\_ClockConfig 函数（在 BSP 中定义为弱优先级）在 main.c 中被重新定义为空函数，因为 BSP\_AUDIO\_OUT\_Init 函数中的时钟和 PLL 配置已经完成。
6. 开始音频采集

在这个应用程序中，主循环是空的，因为将从麦克风获取的音频流复制到串行音频接口所需的所有操作都是在 DMA 中断中执行的。

因此，AudioProcess() 函数由 BSP\_AUDIO\_IN\_TransferComplete\_Callback() 和 BSP\_AUDIO\_IN\_HalfTransfer\_Callback() 调用。



图 5. AudioLoop 应用程序框图



#### 2.5.3.1 麦克风采集过程

可以通过 SPI、I<sup>2</sup>S、GPIO 等不同的外设来对数字 MEMS 麦克风进行采集。它需要一个输入时钟，并以与输入时钟相同的频率输出一个 PDM 流。该 PDM 流会被进一步滤波和抽取，转换为 PCM 标准用于音频传输。

两个不同的数字 MEMS 麦克风可以连接在同一条数据线上，通过相应地设置每个麦克风的 L/R 引脚，将第一个数字 MEMS 麦克风配置为在时钟上升沿生成有效数据，将另一个数字 MEMS 麦克风配置为在下降沿生成有效数据。

在 BlueCoin（STEVAL-BCNCS01V1）上，麦克风输出信号通过 I<sup>2</sup>S 外设进行采集，产生所需的精确时钟，并读取时钟线上升沿和下降沿的 PDM 信号。

所采集的信号随后会从 PDM 发送到 PCM 算法以生成标准 PCM 流。

附加的软件高通滤波级去除了输出流中的所有直流偏移。

采用 DMA 来降低 MCU 负载。

#### 2.5.4 BLE\_SampleApp

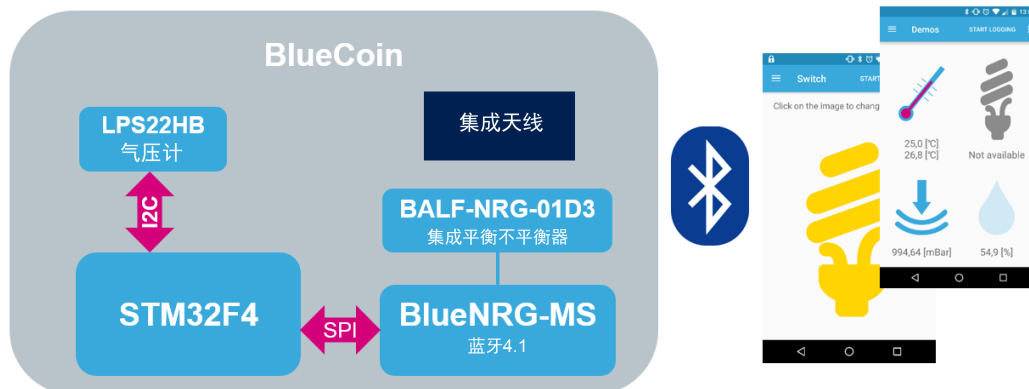
BLE\_SampleApp 提供了一个蓝牙低功耗配置示例，使 BlueCoin 能够传输环境传感器数据。

它可与适用于 Android 和 iOS 的 STBLESensor 应用程序兼容。

复位后，固件：

- 配置 HAL 和时钟
- 配置和禁用传感器芯片选择引脚
- 初始化目标平台：
  - USB 外设（用于调试）
  - LED1
  - 环境传感器
- 初始化蓝牙低功耗堆栈
- 初始化蓝牙低功耗服务
- 初始化定时器
- 开始主循环：
  - LED 管理
  - BLE 事件管理
  - 环境传感器数据管理

图 6. BLE\_SampleApp 应用程序框图



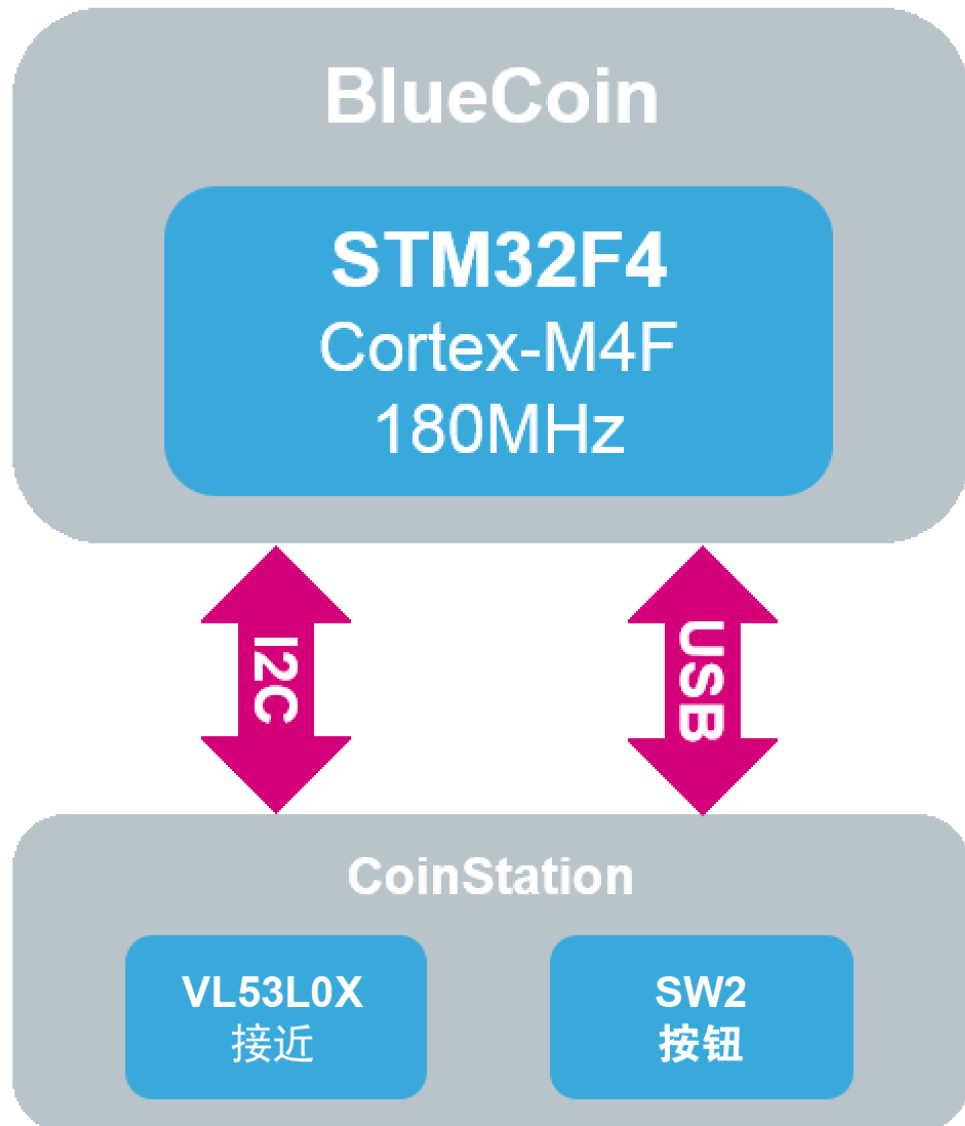
### 2.5.5 手势识别

手势识别应用程序利用 Time-of-Flight 传感器提供了一个示例，可实现非常准确的距离感应、定向滑动和点击检测。

复位后，固件：

- 配置 HAL 和时钟
- 初始化 LED 和按钮
- 初始化 Time-of-Flight 传感器
- 初始化手势检测中间件：
  - 点击检测
  - 定向滑动检测
- 开始主循环：
  - 按下 SW2 按钮在两个演示之间切换：
    - 带 LED 接口的距离感应
    - 手势检测：向左/向右滑动和点击

图 7. 手势识别应用程序框图



### 3 系统设置指南

#### 3.1 硬件配置和主板设置

下图显示了硬件配置：

图 8. BlueCoin 和 CoinStation

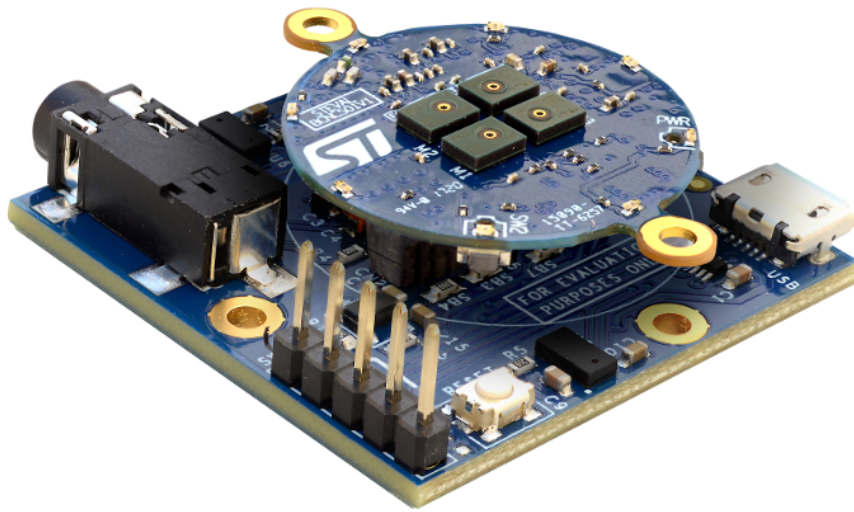
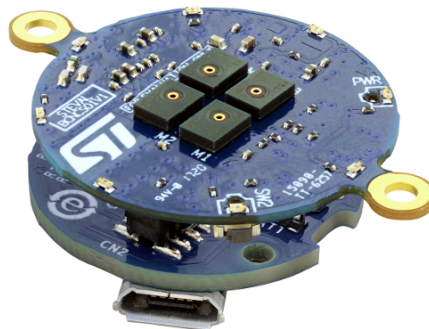


图 9. BlueCoin 和 Cradle 扩展

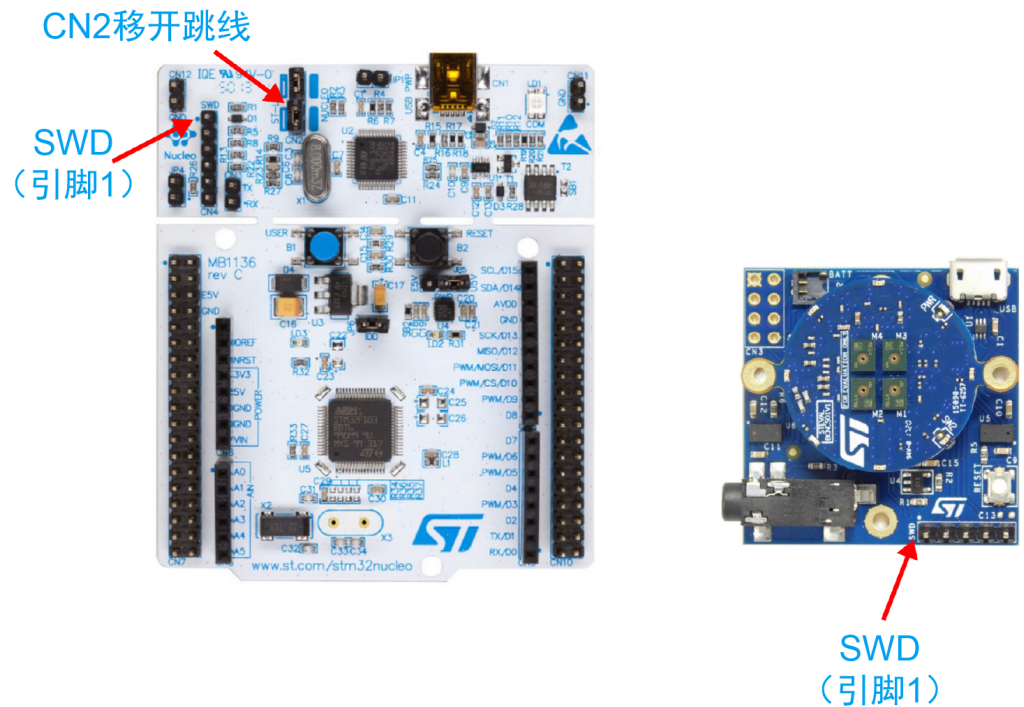


在 BlueCoin 硬件（STEVAL-BCNKT01V1）上运行软件示例：

- Step 1.** 将外部 ST-LINK 连接到 Cradle SWD 连接器（BlueCoin 套件包中提供了一个五引脚扁平电缆）最简单的方法是使用一个 STM32 Nucleo 板，其中包含了 ST-LINK V2.1 编程器。

- Step 2.** 确保 CN2 跳线为 OFF，然后，通过提供的线缆，将您的 STM32 Nucleo 板连接到 CoinStation（使极性正确）。  
如下所示，在 PCB 丝印上用点标记出引脚 1。

图 10. STM32 Nucleo 板，CoinStation SWD 连接器



- Step 3.** 选择其中一个软件包应用程序  
**Step 4.** 通过所支持的某一个 IDE 打开软件包  
**Step 5.** 将代码下载到主板上

## 3.2 软件和硬件兼容性

### 3.2.1 Datalog 3.2.1.1 USB 模式

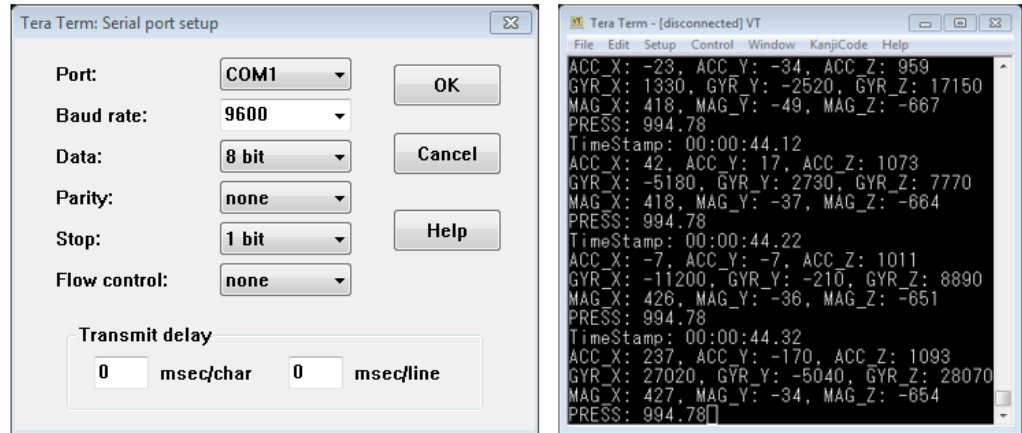
#### BlueCoin

USB 模式下 DataLog 应用程序在两种硬件配置下均可运行（请参见图 8. BlueCoin 和 CoinStation 和图 9. BlueCoin 和 Cradle 扩展），因为两个 Cradle 上唯一所需的接口都是 USB 连接器。

- Step 1.** 使用微型 USB 数据线将板连接到 PC：  
它被识别为一个虚拟 COM 端口。  
您可以从 ST 网站的 [VCP 驱动程序](#) 下载 Windows 驱动程序。
- Step 2.** 板子可对传感器进行配置并开始将数据传输到 PC。  
您可以通过任何 COM 端口软件（如 Putty 或 Tera Term）查看传入的数据。
- Step 3.** 检查 PC 上的 COM 端口号是否正确。

**Step 4.** 使用以下配置:

图 11. 串口配置和接收数据示例



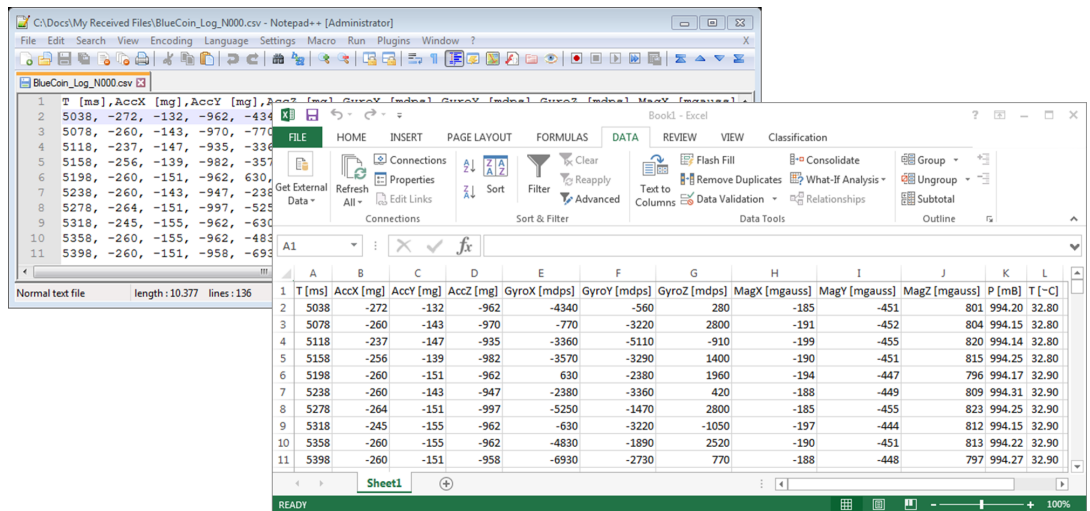
### 3.2.1.2 SD 卡模式

SD 卡模式下 DataLog 应用程序只能在 BlueCoin Cradle 硬件上运行。

软件包中不提供 SD 卡; 您必须使用 SDHC (安全数字高容量) Class 10 卡, 并且必须使用 FAT32 文件系统将其格式化 (例如, 通过使用 SD Formatter 工具)。

- Step 1.** 插入卡并通过开关给板子上电。  
当 SW2 按钮被按下时, 采集开始; 再次按下则采集停止。
- Step 2.** 按下 SW2 按钮开始采集
- Step 3.** 再次按下按钮停止采集
- Step 4.** 将 SD 卡连接到 PC, 并打开任意.csv (逗号分隔值) 文件来验证记录的数据。

图 12. 用文本编辑器 (左) 或 Excel (右) 打开的日志文件

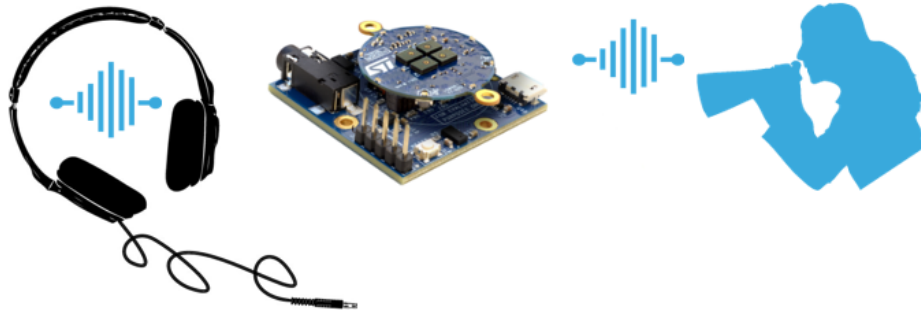


### 3.2.2 AudioLoop

AudioLoop 应用程序只能在 CoinStation 上运行, 因为小型 Cradle 板上没有音频 DAC 和 3.5 mm 插孔。

下载后, 应用程序将配置麦克风和音频 DAC。麦克风采集的声音随后会在连接的耳机或扬声器上播放。

图 13. Audioloop 应用程序框图



### 3.2.3

#### BLE\_SampleApp

BLE\_SampleApp 在两个硬件配置上都可以运行。

下载后，BlueCoin LED 开始闪烁：器件正在等待通过蓝牙进行连接。

在 Android 或 iOS 智能手机上打开 STBLESensor 应用程序，并使用它来连接 BlueCoin。

图 14. 通过 BlueMS 应用程序进行 BLE\_SampleApp BLE 连接



### 3.2.4

#### GestureDetect

GestureDetect 应用程序只能在 CoinStation 上运行，因为小型 Cradle 板上没有 Time-of-Flight 传感器。



版本历史

表 1. 文档版本历史

日期	版本	变更
2017 年 7 月 12 日	1	初始版本。
2019 年 3 月 5 日	2	更新了前言、第 2.1 节 概述和图 2. STSW-BCNKT01 软件架构。 增加了第 2.5.1 节 Audio_SD。



## 目录

<b>1</b>	什么是 STM32Cube? .....	2
<b>1.1</b>	STM32Cube 架构 .....	2
<b>2</b>	STM32Cube 的 STSW-BCNKT01 软件扩展 .....	4
<b>2.1</b>	概述 .....	4
<b>2.2</b>	架构 .....	4
<b>2.3</b>	文件夹结构 .....	5
<b>2.4</b>	APIs .....	6
<b>2.5</b>	STSW-BCNKT01 软件扩展应用程序 .....	7
<b>2.5.1</b>	Audio_SD .....	7
<b>2.5.2</b>	DataLog .....	7
<b>2.5.3</b>	AudioLoop .....	8
<b>2.5.4</b>	BLE_SampleApp .....	9
<b>2.5.5</b>	手势识别 .....	10
<b>3</b>	系统设置指南 .....	12
<b>3.1</b>	硬件配置和主板设置 .....	12
<b>3.2</b>	软件和硬件兼容性 .....	13
<b>3.2.1</b>	Datalog .....	13
<b>3.2.2</b>	AudioLoop .....	14
<b>3.2.3</b>	BLE_SampleApp .....	15
<b>3.2.4</b>	GestureDetect .....	15
	版本历史 .....	16



表一览

表 1. 文档版本历史 ..... 16



## 图一览

图 1.	固件架构 .....	2
图 2.	STSW-BCNKT01 软件架构 .....	5
图 3.	STSW-BCNKT01 软件包文件夹结构 .....	5
图 4.	DataLog 应用程序框图 .....	8
图 5.	AudioLoop 应用程序框图 .....	9
图 6.	BLE_SampleApp 应用程序框图 .....	10
图 7.	手势识别应用程序框图 .....	11
图 8.	BlueCoin 和 CoinStation .....	12
图 9.	BlueCoin 和 Cradle 扩展 .....	12
图 10.	STM32 Nucleo 板, CoinStation SWD 连接器 .....	13
图 11.	串口配置和接收数据示例 .....	14
图 12.	用文本编辑器（左）或 Excel（右）打开的日志文件 .....	14
图 13.	Audioloop 应用程序框图 .....	15
图 14.	通过 BlueMS 应用程序进行 BLE_SampleApp BLE 连接 .....	15



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