
使用指南

N32H7xx系列DSMU使用指南

简介

此文档的目的在于让使用者能够快速熟悉 N32H7xx 系列微控制器（MCU）的 DSMU 外设基本功能及使用方法，用于指导用户更容易的了解和使用本产品。

目录

目录I

1. 概述	2
2. DSMU 外设介绍	3
2.1. DSMU 简介	3
2.2. DSMU 功能框图	3
3. DSMU 初始化	4
3.1. DSMU 通道初始化	4
3.1.1. 时钟输出配置: OutputClock	5
3.1.2. 输入信号配置: Input	5
3.1.3. 外部串行接口配置: SerialInterface	6
3.1.4. 模拟看门狗滤波器配置: Awd	6
3.1.5. 数据后处理配置: Offset、RightBitShift	6
3.2. DSMU 数据滤波器初始化	7
3.2.1. 规则转换配置: RegularParam	7
3.2.2. 注入转换配置: InjectedParam	8
3.2.3. 滤波参数配置: FilterParam	8
4. DSMU 数据滤波器附加功能	9
4.1. 模拟看门狗	9
4.2. 极值检测	9
4.3. 短路与时钟缺失检测	9
5. DSMU 规则与注入转换	10
5.1. 规则转换操作流程	10
5.2. 注入转换操作流程	10
5.3. 触发方式说明	10
6. 事件、中断、DMA	11
7. 注意事项	12
7.1. DSMU 时钟源	12
7.2. 时钟频率要求	12
7.3. 采样速率	12
7.4. 数据有效范围	13
8. 版本历史	14
9. 声明	15

1. 概述

本文介绍了 N32H7xx 系列微控制器（MCU）的 DSMU 外设的基本功能和使用方法，用于指导用户更容易的了解和使用本产品，详细功能说明、参数、注意事项等等请参照用户手册 EN_UM_N32H76x_78x Series User Manual。

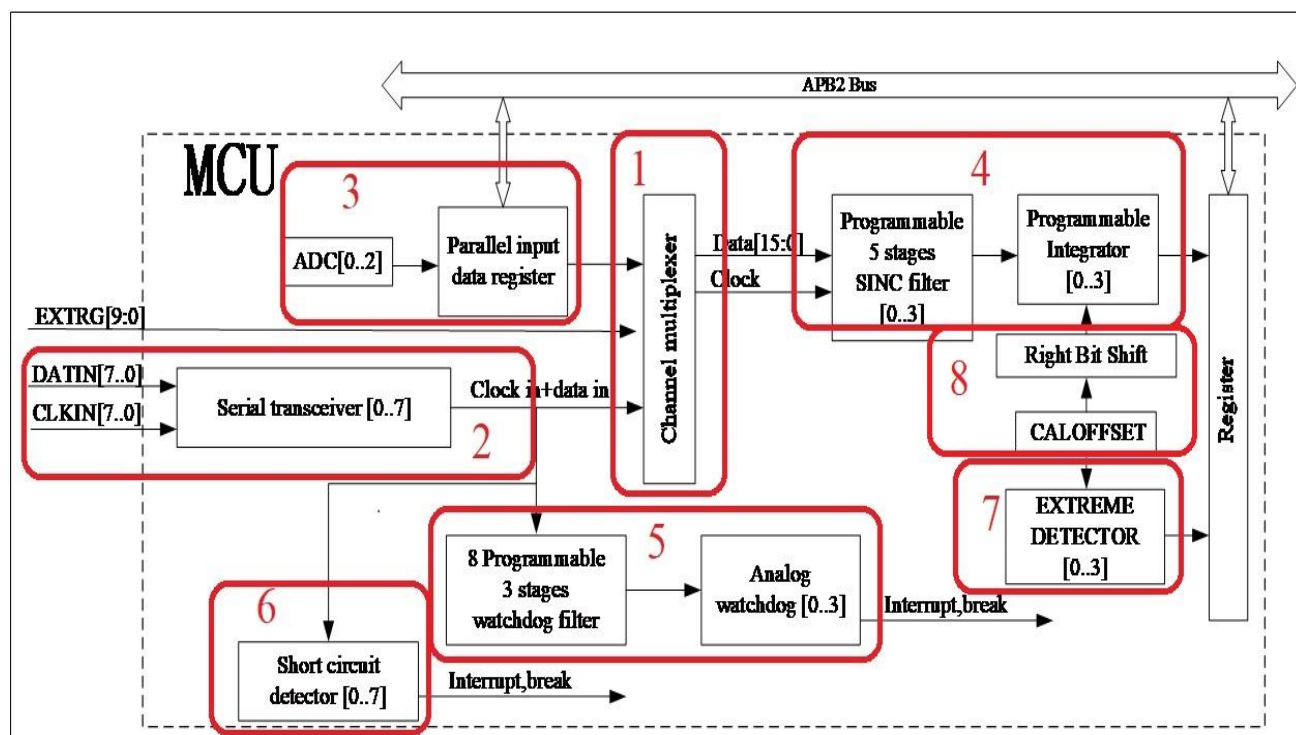
2. DSMU 外设介绍

2.1. DSMU 简介

DSMU 为 Delta Sigma Module Unit 的简称，主要用于连接外部 Sigma-Delta ADC 器件，通过 Sinc 滤波器，对输入串行数据样本进行“滑动平均”计算，获取预期的滤波数据。同时支持对 MCU 内部并行数据（内部 ADC 转换结果、输入数据寄存器）进行滤波处理，支持 DMA。

2.2. DSMU 功能框图

DSMU 模块结构框图如下图所示，大致可分为 8 个部分：



各部分功能如下：

- 1) Channel multiplexer: 通道多路复用器，支持 8 个输入通道，每个通道可独立配置为外部（串行）或内部（并行）通道。
- 2) Serial transceiver: 外部通道串行数据接口，将串行数据转换为并行数据。
- 3) Parallel input data register: 16 位数据输入寄存器，可通过 CPU 或 DMA 直接写入，也可通过固定通道连接到 MCU 内置 ADC：ADC1 连接到通道 0、ADC2 连接到通道 1、ADC3 连接到通道 2。
- 4) SINC filter+Integrator: 共 4 个数据滤波器（sinc 滤波器+积分器），可独立配置连接到任意输入通道。
- 5) Analog watchdog: 共 4 个模拟看门狗，可监控外部通道输入的串行数据或数据滤波器输出结果。当用于监控外部通道输入时，每个通道均提供额外的看门狗 Sinc 滤波器。
- 6) Short circuit detector: 共 8 路短路检测，对外部串行通道输入的 1bit 数据流进行监控。
- 7) Extreme detector: 共 4 路极值检测，对数据滤波器输出数据进行简单分析，并分别记录最小值与最大值。
- 8) Right Bit Shift + CALOFFSET: 共 4 个数据后处理单元（右移+校正），针对每个输入通道可分别配置，对数据滤波器输出数据右移（0~31 位）后，再叠加一个可配置的偏移量并输出最终转换结果。

3. DSMU 初始化

DSMU 初始化可分为通道初始化及数据滤波器初始化，分别介绍如下。

3.1. DSMU 通道初始化

DSMU 支持 8 个输入通道，每个通道均可独立配置为外部串行数据通道或内部并行数据通道。

当配置为外部串行通道，用于连接外部器件时，可配置为 SPI 或 Manchester 接口，支持短路检测、时钟缺失检测、以及模拟看门狗功能。硬件上支持以下 2 种连接方式：

- 当外部器件需要 DSMU 提供时钟（如数字麦克风）时，DSMU 通过 CKOUT、DATINy 引脚与之连接。
- 当外部器件不需要 DSMU 提供时钟（如 Sigma-Delta ADC）时，通过 CKINy、DATINy 引脚与之连接。

每个串行输入通道均可独立配置：

- 模拟看门狗滤波器：当某个串行输入通道连接到模拟看门狗时，输入数据经模拟看门狗滤波器处理后输出到看门狗，过采样率和滤波阶数可配置。
- 短路检测：当串行数据接口检测到的连续数据‘1’或‘0’个数达到阈值（可配置）时，触发短路事件，还可配置输出刹车信号。初始化中不包括短路检测功能，需要单独配置。
- 时钟缺失检测：以 CKOUT 信号（信号源必须配置为系统时钟）为基准检测时钟输入引脚 CKINy，当超过 8 个 CKOUT 周期未检测到 SPI 接口时钟电平变化、或超过 2 个 CKOUT 周期未检测到 Manchester 接口信号电平变化，产生时钟缺失事件。初始化中不包括时钟缺失检测功能，需要单独配置。

另外，每个通道还可独立配置数据后处理参数：

- 数据后处理（右移+校正）作用于数据滤波器输出，当某个输入通道（串行或并行）连接到数据滤波器时，滤波后数据进一步进行右移、校正后输出最终结果。

以下基于通道初始化结构体“DSMU_Channel_InitType”介绍。

```
typedef struct
{
    DSMU_Channel_OutputClockType    OutputClock;
    DSMU_Channel_InputType          Input;
    DSMU_Channel_SerialInterfaceType SerialInterface;
    DSMU_Channel_AwdType            Awd;
    int32_t                          Offset;
    uint32_t                         RightBitShift;
} DSMU_Channel_InitType;
```

3.1.1. 时钟输出配置: OutputClock

```
typedef struct
{
    FunctionalState    Activation;
    uint32_t           Selection;
    uint32_t           Divider;
}DSMU_Channel_OutputClockType;
```

其中:

- **Activation:** 时钟输出使能控制, 部分外接器件(如数字麦克风)需要 DSMU 提供时钟, 必须为 ENABLE。
- **Selection:** 输出时钟源可选
 - 系统时钟: DSMU 模块时钟, 参照函数 `RCC_ConfigDSMUKerClk`。
 - Audio 时钟: 仅用于时钟输出, 参照函数 `RCC_ConfigDSMUKerAClk`。
- **Divider:** 输出时钟分频系数 (1~255), 禁用时配置为 0。
- 通过 CKOUT 引脚输出给外部器件。
- 时钟输出只能在第 1 个输入通道的 DSMU_CH0CFG1 寄存器中配置, 对所有通道有效。

3.1.2. 输入信号配置: Input

```
typedef struct
{
    uint32_t Multiplexer;
    uint32_t DataPacking;
    uint32_t Pins;
}DSMU_Channel_InputType;
```

其中:

- **Multiplexer:** 输入信号源选择, 外部引脚、内置 ADC、输入数据寄存器。
- **DataPacking:** 输入数据寄存器 DSMU_CHyDATIN 数据格式。
 - **Standard:** 1 个当前通道数据, 存放在低 16bit INDATAT0, 高 16bit INDATAT1 无效。
 - **Interleaved:** 2 个当前通道数据, 依次存放在 INDATAT0、INDATAT1。
 - **Dual:** 1 个当前通道数据存放在 INDATAT0, 1 个下一个通道数据存放在 INDATAT1。
- **Pins:** 外部通道数据来源, 当前通道或下一个通道。
 - 需要 DSMU 提供时钟时, 通过 CKOUT 引脚连接外部器件串行时钟引脚, CKINx 未使用。
 - 不需要 DSMU 提供时钟时, 通过 CKINx 引脚连接外部器件串行时钟引脚, CKOUT 未使用。

3.1.3. 外部串行接口配置：SerialInterface

```
typedef struct
{
    uint32_t Type;
    uint32_t SpiClock;
} DSMU_Channel_SerialInterfaceType;
```

其中：

- Type：外部串行数据接口及时钟沿配置。
 - 接口：SPI 或 Manchester。
 - 采样时钟边沿：上升沿、或下降沿。
- SpiClock：SPI 接口时钟配置。
 - 采样时钟源：CKIN、或 CKOUT。
 - CKOUT 采样时钟边沿：第 1 个上升/下降沿、或第 2 个上升/下降沿。

3.1.4. 模拟看门狗滤波器配置：Awd

```
typedef struct
{
    uint32_t FilterOrder;
    uint32_t Oversampling;
} DSMU_Channel_AwdType;
```

其中：

- FilterOrder：模拟看门狗滤波器阶数，可配置 FastSinc、Sinc¹、Sinc²、Sinc³。
- Oversampling：模拟看门狗滤波器过采样率，1~32 可配置。
- 模拟看门狗滤波器在每个输入通道独立配置，在当前通道连接到模拟看门狗时有效。
 - 模拟看门狗只有 4 个，每个模拟看门狗均可独立配置连接到任意输入通道。
 - 当输入通道连接到看门狗时，输入数据经当前通道看门狗滤波器处理后输出到看门狗。

3.1.5. 数据后处理配置：Offset、RightBitShift

其中：

- RightBitShift：数据滤波器输出数据右移位数，0~31 位。
- Offset：数据滤波器输出数据校正量，24 位有符号整数。
- 数据右移与校正每个输入通道独立配置，作用于数据滤波器输出。

- 只有 4 个数据滤波器，每个滤波器可独立配置连接到任意输入通道。
- 滤波器输出数据先右移再校正，然后输出最终结果。
- 当某个输入通道连接到数据滤波器时，滤波器的后处理参数在所连接的输入通道配置。

3.2. DSMU 数据滤波器初始化

DSMU 支持 4 个数据滤波器，每个数据滤波器由 1 个 SINC 滤波器和 1 个积分器组成，均可独立配置以及连接到任意输入通道，其中：

- SINC 滤波计算相当于滑动平均（过采样率可配置），并且可进行多次滑动平均（滤波阶数）。
- 积分器相当于对 SINC 滤波结果进行累加（过采样率可配置）。

数据滤波器以规则转换或注入转换的方式进行数据处理：

- 当配置为规则转换时，每个滤波器只能连接到一个输入通道。
- 当配置为注入转换时，每个滤波器可连接到多个输入通道。

数据滤波器输出数据后，还需进行右移及校正处理才能输出最终结果。右移及校正参数由连接到滤波器的通道独立配置。

以下基于通道初始化结构体“DSMU_Filter_InitType”介绍。

```
typedef struct
{
    DSMU_Filter_RegularParamType    RegularParam;
    DSMU_Filter_InjectedParamType    InjectedParam;
    DSMU_Filter_FilterParamType      FilterParam;
} DSMU_Filter_InitType;
```

3.2.1. 规则转换配置：RegularParam

```
typedef struct
{
    uint32_t      Trigger;
    FunctionalState FastMode;
    FunctionalState DmaMode;
} DSMU_Filter_RegularParamType;
```

其中：

- Trigger：规则转换触发类型，软件触发或同步触发。初始化时仅作为判断条件，不写入寄存器。
- FastMode：数据滤波器快速转换模式，仅适用于规则转换。
- DmaMode：规则转换结果 DMA 读取控制。

3.2.2. 注入转换配置：InjectedParam

```
typedef struct
{
    uint32_t      Trigger;
    FunctionalState ScanMode;
    FunctionalState DmaMode;
    uint32_t      ExtTrigger;
} DSMU_Filter_InjectedParamType;
```

其中：

- **Trigger**：注入转换触发类型，软件触发、同步触发、或外部触发。初始化时仅作为判断条件，不写入寄存器。
- **ScanMode**：数据滤波器扫描模式控制，仅适用于注入转换。
- **DmaMode**：注入转换结果 DMA 读取控制。
- **ExtTrigger**：外部触发源选择。

3.2.3. 滤波参数配置：FilterParam

```
typedef struct
{
    uint32_t      SincOrder;
    uint32_t      Oversampling;
    uint32_t      IntOversampling;
} DSMU_Filter_FilterParamType;
```

其中：

- **SincOrder**：SINC 滤波器滤波阶数，可配置 FastSinc、Sinc¹、Sinc²、Sinc³、Sinc⁴、Sinc⁵。
- **Oversampling**：SINC 滤波器过采样率, 1~1024 可配置。
- **IntOversampling**：积分器过采样率, 1~256 可配置。

4. DSMU 数据滤波器附加功能

4.1. 模拟看门狗

每个数据滤波器提供 1 个模拟看门狗，用于实时监测串行输入数据或最终转换结果是否超出指定范围。

每个看门狗高、低阈值可分别独立配置，超出高、低阈值将分别产生独立的事件，并可配置输出刹车信号。

- 当模拟看门狗用于监测输入数据时，串行数据经模拟看门狗滤波器处理后输出到看门狗。
- 当模拟看门狗用于监测最终结果时，数据滤波器输出数据经右移、校正后输出到看门狗。

4.2. 极值检测

极值检测用于记录当前数据滤波器最终结果的最大值与最小值，可能是一个或多个通道的极值。用户可随时读取极值寄存器（最大值或最小值），读取后极值寄存器复位为默认值并重新开始统计。

4.3. 短路与时钟缺失检测

短路与时钟缺失检测作用于输入串行通道，在每个输入通道独立配置。但状态标志只能在第 1 个数据滤波器的状态寄存器 DSMU_FLT0STS 中查询。

5. DSMU 规则与注入转换

5.1. 规则转换操作流程

规则转换只适用于单个输入通道，使用时可按照以下步骤进行：

- 1) 初始化 DSMU 通道与数据滤波器。
- 2) 配置规则通道与工作模式，参照函数 `DSMU_FilterConfigRegChannel`。支持 2 种工作模式：
 - 单次模式：每次触发只进行一次转换。
 - 连续模式：触发后连续转换。
- 3) 根据需要配置附加功能：短路检测、时钟缺失检测、模拟看门狗、极值检测。
- 4) 开始规则转换，参照函数 `DSMU_RegConvStart`，支持 2 种触发方式。
 - 软件触发
 - 同步触发

5.2. 注入转换操作流程

注入转换适用于多个通道（注入转换组），使用时可按照以下步骤进行：

- 1) 初始化 DSMU 通道与数据滤波器。
- 2) 配置注入通道组，参照函数 `DSMU_FilterConfigInjChannel`。
- 3) 根据需要配置附加功能：短路检测、时钟缺失检测、模拟看门狗、极值检测。
- 4) 开始注入转换，参照函数 `DSMU_InjConvStart`，支持 3 种触发方式：
 - 软件触发
 - 同步触发
 - 外部触发
- 5) 注入转换支持 2 种工作模式，在初始化时配置：
 - 单次模式：每次触发只转换一个通道。
 - 扫描模式：触发后转换注入组中的所有通道，根据通道编号从低到高依次转换。

5.3. 触发方式说明

DSMU 数据转换共支持 3 种触发方式：

- 软件触发：通过在数据滤波器寄存器对应控制位写 1 触发规则或注入转换。
- 同步触发：仅适用于第 2、3、4 数据滤波器。当第 1 个滤波器 `DSMU_FLT0` 开始转换时，当前数据滤波器开始同步进行规则或注入转换。
- 外部触发：通过外部事件触发，仅适用于注入转换。

6. 事件、中断、DMA

- DSMU 共支持以下事件，所有事件均可配置产生中断：
 - 时钟缺失事件
 - 短路事件
 - 模拟看门狗事件
 - 规则转换过载事件
 - 注入转换过载事件
 - 规则转换完成事件
 - 注入转换完成事件
- 其中，部分事件可配置输出刹车信号，连接到 MCU 定时器 ATIM、GTIM、SHRTIM，详见用户手册：
 - 短路事件
 - 模拟看门狗事件
- 内部并行通道，即输入数据寄存器可通过 DMA 写入。
- 规则或注入转换结果可分别配置通过 DMA 读取。

7. 注意事项

7.1. DSMU 时钟源

DSMU 涉及 3 个时钟源：

- APB2 总线时钟 PCLK2：寄存器接口时钟。
- DSMU Kernel clock：模块工作时钟 DSMU_CLK，时钟源可配置：
 - APB2 总线时钟 PCLK2。
 - System Bus 时钟 1/2/4/8/16 分频。
- DSMU Kernel A clock：Audio 时钟 ACLK，仅用于时钟输出，时钟源可配置：
 - APB2 总线时钟 PCLK2。
 - PLL1B/PLL2B/PLL3A。
 - Peripherals BUS clock。
 - I2S CKIN 引脚输入时钟。

7.2. 时钟频率要求

应用中，DSMU 及外部接口时钟频率必须满足以下要求：

- $DSMU_CLK \geq PCLK2$ 。
- 当某个输入通道配置为并行数据输入时：
 - 标准输入模式： $DSMU_CLK > (2 * \text{输入数据速率})$ 。
 - 交错输入模式： $DSMU_CLK > (4 * \text{输入数据速率})$ 。
- 当某个输入通道配置为串行 SPI 接口，且由外部器件输入采样时钟时：
采样时钟最大为 20M，且小于 $(DSMU_CLK/4)$ 。
- 当某个输入通道配置为串行 Manchester 接口时：
 - Manchester 恢复时钟最大 10M，且小于 $(DSMU_CLK/6)$ 。
 - 输出时钟分频系数 CLKOUTDIV 需要满足以下要求：
 $DSMU_CLK/(2 * CLKOUTDIV) < \text{恢复时钟} < DSMU_CLK/(CLKOUTDIV+1)$

7.3. 采样速率

DSMU 数据滤波需要对输入数据进行过采样，可大大提高数据精度，同时也会降低输出采样速率：

- 输出采样速率=输入数据采样率/综合过采样率。
- 综合过采样率由以下 3 个参数决定：
 - SINC 滤波器过采样率 FOSR：FOSR 位域值+1
 - 滤波器阶数 FORD：FastSinc=4，Sinc¹、Sinc²、Sinc³、Sinc⁴、Sinc⁵=1~5。
 - 积分器过采样率 IOSR：IOSR 位域值+1。
- 当进行规则转换并启用快速转换模式时：
 - 第 1 次转换综合过采样率： $FOSR * (FORD + IOSR - 1) + 1$
 - 后续综合过采样率： $FOSR * IOSR$
- 当进行注入转换或不启用快速转换模式的规则转换时：
 - 综合过采样率： $FOSR * (FORD + IOSR - 1) + 1$

7.4. 数据有效范围

DSMU 输入输出均为有符号整型数据，有效范围如下：

- 输入最高分辨率为 16bits: $-2^{15} \sim (2^{15}-1)$ 。
- 输出最高分辨率为 24bits: $-2^{23} \sim (2^{23}-1)$ 。
- 中间计算结果最高分辨率为 32bits: $-2^{31} \sim (2^{31}-1)$ 。

因此，在配置数据滤波器参数时，需要注意避免数据溢出：

- 输出结果有效位不能大于 24bits
- 过采样后的结果（进行右移操作前）有效位不能大于 32bits
- 可按照下式评估数据滤波器参数：
 - 当采用 FastSinc 滤波时: $2 * FOSR^2 * IOSR \leq 2^{31}$
 - 当采用 Sinc^{1~5} 滤波时: $FOSR^{FORD} * IOSR \leq 2^{31}$

8. 版本历史

版本	日期	备注
V1.0.0	2025-6-24	创建文档

9. 声明

国民技术股份有限公司（下称“国民技术”）对此文档拥有专属产权。依据中华人民共和国的法律、条约以及世界其他法域相适用的管辖，此文档及其中描述的国民技术产品（下称“产品”）为公司所有。

国民技术在此并未授予专利权、著作权、商标权或其他任何知识产权许可。所提到或引用的第三方名称或品牌（如有）仅用作区别之目的。

国民技术保留随时变更、订正、增强、修改和改良此文档的权利，恕不另行通知。请使用人在下单购买前联系国民技术获取此文档的最新版本。

国民技术竭力提供准确可信的资讯，但即便如此，并不推定国民技术对此文档准确性和可靠性承担责任。

使用此文档信息以及生成产品时，使用者应当进行合理的设计、编程并测试其功能性和安全性，国民技术不对任何因使用此文档或本产品而产生的任何直接、间接、意外、特殊、惩罚性或衍生性损害结果承担责任。

国民技术对于产品在系统或设备中的应用效果没有任何故意或保证，如有任何应用在其发生操作不当或故障情况下，有可能致使人员伤亡、人身伤害或严重财产损失，则此类应用被视为“不安全使用”。

不安全使用包括但不限于：外科手术设备、原子能控制仪器、飞机或宇宙飞船仪器、所有类型的安全装置以及其他旨在支持或维持生命的应用。

所有不安全使用的风险应由使用人承担，同时使用人应使国民技术免于因为这类不安全使用而导致被诉、支付费用、发生损害或承担责任时的赔偿。

对于此文档和产品的任何明示、默示之保证，包括但不限于适销性、特定用途适用性和不侵权的保证，国民技术可在法律允许范围内进行免责。

未经明确许可，任何人不得以任何理由对此文档的全部或部分进行使用、复制、修改、抄录和传播。