AN5047
应用笔记

STM8S001J3微控制器入门

引言

本应用笔记补充了 STM8S001J3 数据手册和 *应用笔记中的信息 STM8S和STM8A入门*（AN2752）。它说明了基于STM8S001J3 8位微控制器构建应用所需的最低硬件和软件环境。

本应用笔记分为以下几章：

- 电源
- 时钟管理
- 复位控制
- 模数转换器（ADC）
- 调试工具支持
- STM8S001J3固件建议
- 引脚排列功能概述
- 外设特定的使用和限制
- STM8软件工具链

本文还含有一些STM8S001J3器件特定的硬件和软件建议。

目录

图片索引

1 硬件要求汇总

为了围绕STM8S001J3构建应用，应用板至少需要提供以下特性：

- 电源
- 时钟管理
- 复位管理
- 调试工具支持，单线接口模块(SWIM)连接器。

2 电源

STM8S001J3器件可以通过一个外部的3.0 V到5.5 V电源供电。

一个片上电源管理系统负责提供内核逻辑单元所需的1.8 V数字电源，具有正常和低功耗两种工作模式。电源管理系统也能够检测主要的外部电源（3.0 V至5.5 V）和内部电源（1.8 V）上的电压跌落。

2.1 主电源对

STM8S001J3包含一对供电引脚，VDD/VSS（3.0 V至5.5 V）专用于核心调节器供电、I/O引脚电源和模拟功能的供电。

电源对VDD/VSS应通过滤波陶瓷电容（100 nF）并联一个1至2 µF的电容解耦。陶瓷电容应尽可能靠近供电引脚。

2.2 VCAP引脚上的电容

主调压器的稳定性是通过将外部电容连接到VCAP引脚实现的。请参考STM8S001J3 器件数据手册以获取C_{EXT}特性的更多信息。

在任何情况下，CEXT容量必须处于指定范围，考虑工作温度、电容生产变化以及由于电压偏置引起的相对容量降低，这些都能在电容的规范中找到。

在设备上电复位（POR）期间外部电容充电，外部电源必须在V_{DD}电压未降到POR电平的情况下提供峰值电流（建议对于更弱的电源，在VDD/VSS上使用10 µF的电容）。

2.3 开机/关机复位（POR/PDR）

输入供电由上电/掉电复位电路监控。监控电压范围为0.7 V到2.7 V。

在电源开启时，POR/PDR保持器件处于复位，直到电源电压达到它们指定的工作区域。复位释放的上限在产品数据手册的电气特性一节中定义。迟滞（POR > PDR）用以确保准确检测电压上升和下降。

当电源电压下降到$V_{POR/PDR}$门限值以下时，POR/PDR也会产生一个复位。

3 时钟管理

3.1 时钟管理概述

STM8S001J3器件提供了一个灵活的选择内核和外设（ADC、存储器、数字外设）时钟的方式。这些器件具有内部和外部时钟源输入以及一个输出时钟(CCO)。

3.2 内部时钟

嵌入式RC振荡器具有一个内部电容（C）和一个内部阶梯电阻（R）。

STM8S001J3有两类内部时钟：

- 高速内部时钟（HSI），运行于16 MHz
- 低速内部时钟（LSI），运行于128 kHz。

复位之后，CPU在内部RC（HSI时钟信号）8分频的频率下启动，即2 MHz。

3.3 外部时钟

STM8S001J3 器件仅支持连至外部振荡器。无法连至外部晶体/谐振器。

若需使用外部振荡器，选项位EXTCLK必须置位（工厂默认）。

注： *当没有使用外部时钟时，OSCIN引脚可以用作通用I/O引脚。*

4 复位管理概述

STM8S001J3微控制器没有外部复位信号（NRST引脚）。初始复位来自上电复位（POR）。

内部复位源为：

- 上电复位（POR）/欠压复位（BOR）
  在上电期间，POR保持器件处于复位，直到电源电压（VDD）达到BOR开始工作的电压水平。
- 独立看门狗复位（IWDG）
- 窗口看门狗复位（WWDG）
- 软件复位
  应用软件可以触发复位。
- SWIM复位
  一个连接到SWIM接口的外部器件可以请求SWIM模块生成微控制器复位。
- 非法操作代码复位
  如果执行的代码不与任何操作代码或字节前值相符，就会生成一个复位。
- 电磁敏感性（EMS）复位
  如果关键寄存器被损坏或者错误加载，就会生成复位。

5 模数转换器 (ADC)

5.1 模拟电源

ADC单元使用通用模拟供电和数字参考电压电源V_{DD}。

5.2 模拟输入

STM8S001J3具有三个外部模拟输入通道（AIN2、AIN4/AIN5、AIN6）和一个内部模拟输入通道（AIN7 - 连至内部参考电压），由ADC一次转换一个。每个外部模拟输入都由I/O复用（模拟输入AIN4和AIN5共享相同的引脚）。

若需激活模拟输入AIN2，必须置位复用功能位AFR2。

更多详细信息，请参考STM8S001J3器件数据手册和参考手册。

6 调试支持

6.1 SWIM（单线接口模块）概述

在线调试和编程模式通过一个单线硬件接口管理，该接口基于开漏线，具有超快存储编程特性。除了与在线调试模块耦合，SWIM也可以进行RAM和外设的非侵入式读/写操作。这使得在线调试器非常强大，接近全功能仿真器的性能。

SWIM引脚可以用作标准I/O，如果用户想使用它调试，会存在一些限制。

请参考用户手册*STM8 SWIM通信协议和调试模块*（UM0470）以获取SWIM协议的详细信息。

6.2 可靠SWIM连接建议

6.2.1 程序存储器应总是含有有效的程序循环

若需通过SWIM接口提供可靠的STM8S001J3器件连接，建议不要完全擦除程序存储器（例如通过批量擦除：置位并移除RDP保护）。

另一建议是新加载的执行代码必须总是一个有效的程序循环，例如while (1)代码。STM8S001J3器件工厂默认的程序存储器内容是while (1) 代码（请参见 *图 1*）。

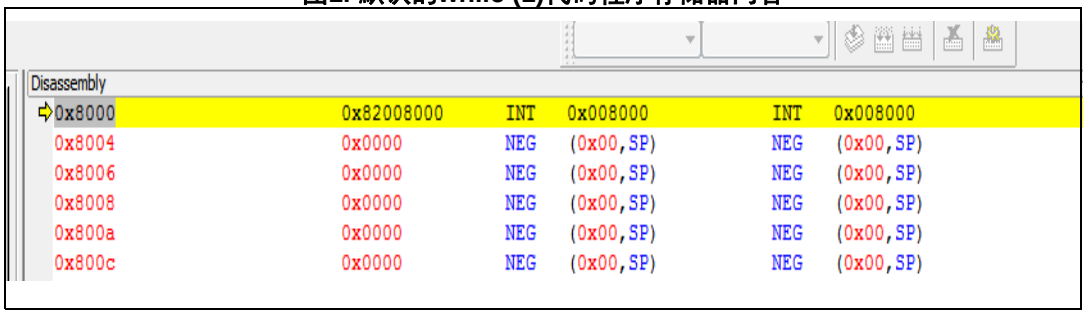
说明：

若程序存储器为空（0x00内容），则设备行为如下：

1. 上电后，执行空代码（0x0000操作码 = 指令NEG (0x00, SP)），直到PC计数器达到8K字节程序存储器末端（末端地址0x9FFF）。在HSI时钟为2 MHz的情况下，它会耗时大约4 ms达到8K字节存储器空间末端。
2. 达到8K字节程序存储器末端后，程序继续，从并不存在的存储器获取并执行代码。读取到的并不存在的存储器是一个随机内容，会导致执行非法指令。执行非法指令会产生软件复位，程序重启。在最差的情况下，会每隔4 ms复位一次。
3. 仅通过“动态连接”方法才能通过SWIM接口编程设备。无法使用“复位状态下连接”方法，因为在此设备上无法使用复位引脚（NRST）。
4. 当设备执行代码时，可使用“动态连接”模式，但若在SWIM连接期间有设备复位（内部复位），则终止此连接，必须用调试工具才能再执行。请注意，软件复位可能每隔4 ms发生，导致难以成功连接到调试工具（通常每10次尝试才能成功一次）。
5. 一旦连接成功，则可成功使用有效固件编程设备。

*图 1*提供了加载到设备中的while (1)代码样例（工厂默认存储器内容） - 复位向量指向复位向量。

图1. 默认的while (1)代码程序存储器内容



6.2.2 具有SWIM功能的引脚配置之前的时延

若应用使用SWIM引脚8上的另一功能（例如GPIO输出），在更改引脚功能之前，建议在固件中加入约5 s的时延。此行为能让用户在设备上电后将设备置为SWIM模式（在此时延期间），并重新编程设备。

因为STM8S001J3设备上没有复位引脚（NRST），所以无法使用“复位状态下连接”方法。若设备复位后SWIM引脚8立即置为I/O模式，则设备无法通过SWIM接口连接，并被永久锁定。

在最终/锁定的代码中可移除此初始时延（通常足够快的I/O引脚功能是需要的）。

7 STM8S001J3固件建议

STM8S001J3器件的启动代码必须根据如下建议配置。

7.1 将未绑定的GPIOS设置为安全状态

GPIOPA2、PB0、PB1、PB2、PB3、PB6、PB7、PC1、PC2、PC7、PD0、PD2、PD4、PD7、PE5和PF4仅内部存在于设备：它们不绑定到引脚。

设备复位后，这些GPIO将被硬件配置为输入浮空模式。建议将其设置为输出模式/低输出电平状态。EMC抗性增强（GPIO接地），设备功耗更低（浮动输入上没有I/O电平跳变）。

7.2 SWIM引脚重配置之前的时延

若需通过SWIM接口调试设备，建议在固件启动代码中增加约5 s的时延。在最终/锁定的代码中可移除此时延（参见*第 7.4 节*中的示例时延代码）。

在此时延期间会保持SWIM引脚功能。用户可通过SWIM接口连接设备并重新编程设备。请参见*第 6.2.2 节：具有SWIM功能的引脚配置之前的时延第 9页*。

若固件未实现此时延，并立即将引脚8上的一些GPIO重配置为输出模式，则无法通过SWIM接口连接设备（因为此引脚现在是一个输出）。设备会永久锁住（因为无法更改固件）。

若应用无法接受此初始时延，有一个选项是固件在特定条件下重新使能SWIM引脚功能。

下面列出了如何管理重新使能SWIM引脚的一些例子：

- 若在启动期间某个特定引脚接地（或一些引脚的组合处于特定状态），则固件在启动期间保持SWIM引脚功能。
- 若应用中使用了一些通信接口，则可实现一个特定指令，在此之后SWIM引脚功能启用，调试器可连接设备。
- 应用可实现一个简单的引导加载程序，通过通信接口（而不是通过SWIM接口）更新固件。

7.3 适当的GPIO模式编程

STM8S001J3 器件有多个引脚，提供至多个GPIO的连接。因此任何一个GPIO所选用的模式会影响连接到同一引脚的其他相关GPIO。适当设置GPIO模式，防止绑定到同一引脚的GPIO产生冲突非常重要（包括其复用功能）。

举例：PD1、PC6、PD3和PD5 GPIO都在引脚8上。PD1上的上拉启用也会在PC6、PD3和PD5上看到。

7.4 建议的启动代码举例

```
/* MAIN.C文件 */

#include "stm8s.h"
#include "stm8s_gpio.h"
#include "stm8s_clk.h"

#ifdef _COSMIC_
  #define ASM  _asm
#endif
#ifdef _IAR_
  #define ASM  asm
#endif
/* 应在复位后增加此时延，以便能访问SWIM引脚并能在上电后对设备重新编程（否则设备将被锁住）
*/
#define STARTUP_SWIM_DELAY_5S      \
  {                          \
  ASM("      PUSHW X       \n"   \
      "      PUSH  A       \n"   \
      "      LDW X, #0xFFFF \n"   \
      "loop1: LD  A, #50     \n"   \
                          \
      "loop2: DEC A         \n"   \
      "      JRNE loop2     \n"   \
                          \
      "      DECW X         \n"   \
      "      JRNE loop1     \n"   \
                          \
      "      POP  A         \n"   \
      "      POPW X          " ); \
  }
/* 未连接的引脚作为输出低状态（最佳的EMC抗性）
(PA2, PB0, PB1, PB2, PB3, PB6, PB7, PC1, PC2, PC7, PD0, PD2, PD4, PD7, PE5, PF4) */
#define CONFIG_UNUSED_PINS_STM8S001 \
{ \
  GPIOA->DDR |= GPIO_PIN_2; \
  GPIOB->DDR |= GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3 | GPIO_PIN_6 |
GPIO_PIN_7; \
  GPIOC->DDR |= GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_7; \
  GPIOD->DDR |= GPIO_PIN_0 | GPIO_PIN_2 | GPIO_PIN_4 | GPIO_PIN_7; \
  GPIOE->DDR |= GPIO_PIN_5; \
  GPIOF->DDR |= GPIO_PIN_4; \
}
/* 用于测试的引脚 */
#define TEST_PORT GPIOA
```

```
#define TEST_PIN  GPIO_PIN_3
//#define HSE_TEST

/* STM8S001固件举例：建议的引脚启动 + 测试功能 */
main()
{
  /* -------------STM8S001启动---------- */
  /* 配置未绑定的引脚 */
  CONFIG_UNUSED_PINS_STM8S001;
  /* SWIM连接延迟：~5秒 */
  STARTUP_SWIM_DELAY_5S;
  /* ---------------------------------------- */

  /* 将所有STM8S001引脚配置为具有上拉的输入 */
  GPIO_Init(GPIOA, GPIO_PIN_1, GPIO_MODE_IN_PU_NO_IT);    // 引脚 1
  GPIO_Init(GPIOA, GPIO_PIN_3, GPIO_MODE_IN_PU_NO_IT);    // 引脚 3
GPIO_Init(GPIOB,GPIO_PIN_4,GPIO_MODE_OUT_PP_LOW_FAST);//引脚6（PB4没有上拉-将它配置为输出低）
  GPIO_Init(GPIOC, GPIO_PIN_3, GPIO_MODE_IN_PU_NO_IT);    // 引脚 7
  GPIO_Init(GPIOC, GPIO_PIN_6, GPIO_MODE_IN_PU_NO_IT);    // 引脚 8

  /* 禁用外设时钟以降低功耗 */
  CLK->PCKENR1 = 0x00;
  CLK->PCKENR2 = 0x00;

  /* 用于测试外部HSE时钟 */
  /* 确保选项位EXTCLK=1 */
  #ifdef HSE_TEST
  /* 测试HSE（外部时钟） - 在PA1（引脚1）上应用输入时钟 */
 GPIO_Init(GPIOC, GPIO_PIN_4, GPIO_MODE_OUT_PP_LOW_FAST); // CCO位于PC4（引脚7）上
  CLK_CCOConfig(CLK_OUTPUT_CPU); // 时钟输出位于PC4/CCO（引脚7）
  CLK_ClockSwitchConfig(CLK_SWITCHMODE_AUTO, CLK_SOURCE_HSE, DISABLE,
                        CLK_CURRENTCLOCKSTATE_DISABLE); // 将HSE设置为时钟
  #endif //HSE_TEST

  /* 初始化测试引脚 */
  GPIO_Init(TEST_PORT, TEST_PIN, GPIO_MODE_OUT_PP_LOW_FAST);
  while (1)
  {
    /* 与测试引脚切换 */
    GPIO_WriteReverse(TEST_PORT, TEST_PIN);
  }
}
```

8 引脚排列功能概述

*图 2*和 *表 1* 显示了 STM8S001J3 器件引脚和每个引脚的外设功能概述（SO8N封装）。

一个引脚通常被多个GPIO共用（多个GPIO内部连至相同引脚）。对于一个给定引脚，在同一时刻仅应编程一个功能。请参见器件数据手册以获取更多信息。

图2. STM8S001J3引脚排列

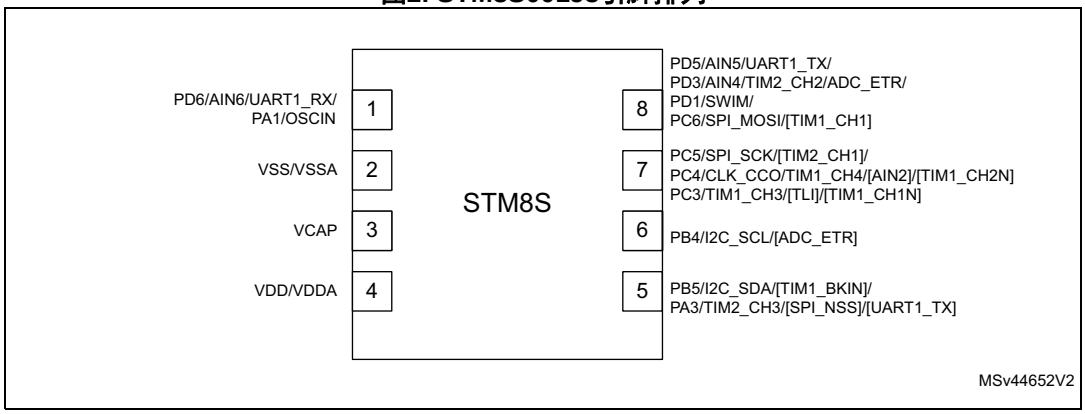


表1. STM8S001J3复用功能概述

外设	引脚号							
	1	2	3	4	5	6	7	8
UART	RxD(1)	-	-	-	TxD(1)	-	-	TxD
I2C：	-	-	-	-	SDA	SCL	-	-
ADC	AIN6	-	-	-	-	ETR	AIN2	AIN5/ETR
TIM1	-	-	-	-	BKIN	-	CH1N/CH2N/ CH3/CH4	CH1
TIM2	-	-	-	-	CH3		CH1	CH2
OSC	OSCIN	-	-	-	-	-	CCO	-
电源	-	VSS	VCAP	VDD	-	-	-	-
调试	-	-	-	-	-	-	-	SWIM
SPI	-	-	-	-	-	-	SCK	MOSI

1. 1.通过重新映射UART1_TX（AFR0=1及AFR1=1）到引脚5，引脚1上的UART1_RX复用功能变为不可用。UART1仅能用于单线半双工模式或智能卡读卡器模拟模式。

9 外设特定使用

STM8S001J3 器件为8引脚SO8N封装。因为多个GPIO连至相同的引脚，所以有多种特定的外设使用，如下所述。

9.1 UART1

RxD在引脚1上。TxD在引脚8上（能够重映射至引脚5）。

若需将TxD重映射至引脚5，则两个复用功能位AFR0和AFR1必须置为1。

通过将TxD重映射至引脚5，（引脚1上的）RxD功能变为不可用，UART1仅能用于单线半双工模式或智能卡读卡器模拟模式）。

9.2 I2C：

SCL在引脚6上。SDA在引脚5上。

SDA引脚不是一个真的开漏引脚，它只是一个伪开漏，因为引脚7与另一个GPIO共用（在它上面实现了保护二极管至V_{DD}）。

9.3 SPI

SCK在引脚7上。MOSI在引脚8上。

因为缺少MISO信号，所以SPI必须仅用于如下工作模式之一：

- 单向传输（主模式 = 仅发送/接收，从模式 = 仅接收）
- 双线上的单工主同步传输，可能有双向数据。

9.4 振荡器（OSCIN输入）

外部振荡器可连至引脚1上（OSCIN），用于精确的HSE时钟输入。

HSE特定使用：

- 无法连接外部晶振。
- 选项位EXTCLK必须置为1，用于外部振荡器输入（对于STM8S001J3设备为工厂默认）。

9.5 ADC

ADC外部输入通道（AIN6、AIN2和AIN4/AIN5）在引脚1、7和8上。外部ADC触发（ADC_ETR）在引脚6上。

ADC有一个内部通道AIN7，其内部连至内部带隙参考电压。通过测量内部参考电压，可以确定另一外部ADC通道上的绝对电平。

ADC特定使用：

- 可以使用扫描模式，但对于缺失的ADC通道，则执行虚拟转换。
- 若需启用AIN2通道输入，复用功能重映射位AFR2必须置为1。

9.6 定时器（TIM1、TIM2）

定时器通道输入/输出位于引脚5、7和8上。定时器刹车输入功能（TIM1_BKIN）在引脚5上。

必须小心编程定时器通道以避免和另一定时器通道输出碰撞（在一个给定引脚上仅有一个输出定时器通道）。

9.7 SWIM

SWIM在引脚8上。在设备复位后，PD1/SWIM引脚的内部上拉激活。

引脚8可被另一输出功能使用，通过调试器将设备的SWIM连接禁用（请参见*图 2*和*表 1*）。在代码开发期间，建议固件在设备复位（上电）后实现约5 s的时延，将引脚8设为需要的输出功能。若引脚8上使用了其它功能，则必须通过软件禁用SWIM功能（CFG_GCR寄存器中的SWD位）（请参见*第 7.2节：SWIM引脚重配置之前的时延第 10页*以获取更详细信息）。

10 STM8软件工具链

为了在STM8S001J3器件上编写、编译和运行第一个软件，需要下列软件工具链的组件：

- 集成开发环境
- 编译器
- 固件库（可选，用于方便启动）。

若需更详细信息，请参考应用笔记 *STM8S和STM8A入门*（AN2752）。

10.1 集成开发环境

10.1.1 ST工具集：STVD、STVP

集成开发环境ST Visual Develop (STVD) 为全程控制应用开发（从构建和调试应用代码到微控制器编程）提供了简便易用且高效的环境。STVD是免费ST工具套件的一部分，该套件还包括ST Visual Programmer(STVP)编程接口和ST Assembler Linker。

为了构建应用程序，STVD无缝集成了C和汇编语言工具链，包括Cosmic和Raisonance C语言编译器与ST Assembler Linker。在调试时，STVD提供了一个集成的仿真器（软件），同时支持包括低成本ST-LINK在线调试/编程器等全套硬件工具。

对于基于STM8S001J3器件的应用，STVD还提供了读、写和验证微控制器存储器的接口。此接口基于STVP，支持STM8S001J3器件和编程工具。

免费的用于STM8的ST工具套件可以从意法半导体主页上获取*www.st.com*。

10.1.2 STM8的IAR embedded workbench

IAR embedded workbench IAR-EWSTM8是一个软件开发工具，为STM8AF、STM8AL、STM8L和STM8S系列器件提供了完整支持。

此产品由第三方提供，不属于意法半导体。若需规范和购买部件包的完整和最新信息，请参考IAR系统网站。

10.1.3 RIDE-STM8软件开发环境

RIDE-STM8为STM8集成了Raisonance C编译器（RKit-STM8安装）。

此产品由第三方提供，不属于意法半导体。若需规范和购买部件包的完整和最新信息，请参考Raisonance网站。

10.2 编译器

STM8S001J3器件可以由一个包括在ST工具套件中的免费汇编工具链编程。由于内核是为了支持高级语言而优化设计，因此建议使用C编译器。

STM8的C编译器由第三方公司IAR Systems、Cosmic和Raisonance提供。

这些C编译器的免费版具有有限的生成代码，可从第三方网站获取。

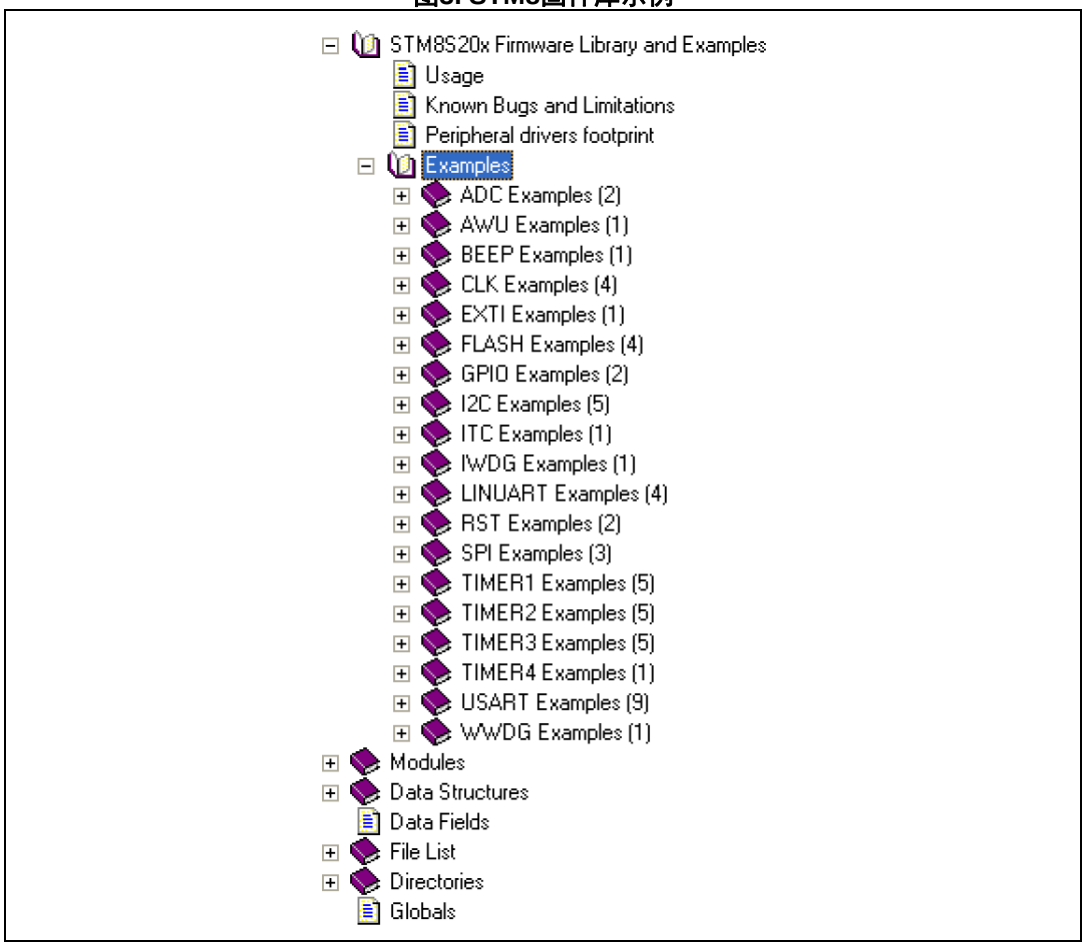
10.3 固件库

STM8固件库对每种STM8外设有一组完整的源代码样例，以严格的ANSI-C语法编写，完全兼容MISRA C 2004（请参见*图 3*）。

所有的示例都配有用于STVD和Cosmic C编译器的工作区和项目定义文件，这使用户可以轻松载入和编辑到开发环境中。运行在意法半导体STM8评估板上的示例可以轻松地根据其它类型硬件进行移植。

若需STM8固件库的额外信息，请参考意法半导体主页*www.st.com.*

图3. STM8固件库示例



11 文档和在线帮助

与STM8S001J3器件相关的文档资源如下：

应用

- STM8S001J3数据手册
- *STM8S001J3*、*STM8S003xx*、*STM8S103xx和STM8S903xx器件限制*（ES0102）
- *参考手册STM8S系列和STM8AF系列8位微控制器*（RM0016）
- 编程手册*如何对STM8S和STM8A的Flash编程存储器和数据EEPROM编程*（PM0051）
- *STM8 CPU编程手册*（PM0044）

工具

- STM8固件库和版本说明（作为帮助文件包含详细的固件库描述）
- ST visual develop指导（在ST-工具链中作为帮助文件）
- 用户手册*ST visual develop（STVD）*（UM0036）
- *STM8 SWIM通信协议和调试模块*用户手册（UM0470）
- Cosmic C编译器用户手册
- IAR embedded workbench开发指南。

开发者们可在微控制器论坛community.st.com上交流意见。这是寻找不同应用思路的最好的地方。除此之外，网站还有一个关于微控制器的FAQ技术资料，它提供了许多问题和解决方案。

12 版本历史

表2. 文档版本历史

日期	版本	变更
2017年6月29日	1	初始版本。
2018年6月28日	2	更新了： – *第 2节：电源* – *第 2.1节：主电源对* – *第 2.2节：VCAP引脚上的电容* – *第 5.2节：模拟输入* – *第 7.2节：SWIM引脚重配置之前的时延* – *第 9.1节：UART1* – *第 9.5节：ADC* – *表 1：STM8S001J3复用功能概述*

表3. 中文文档版本历史

日期	版本	变更
2019年4月4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。