life.augmented

AN4230
应用笔记

使用NIST统计测试集验证STM32微控制器随机数生成

引言

很多标准都规定了构造要求和参考、随机数发生器（RNG）的验证和使用，以便检验其生成的输出是否是真正的随机数。

本应用笔记中包含的一些指南用于检验下表列出的所选STM32微控制器（MCU）中嵌入的RNG外设生成的数字的随机性。该检验基于NIST（国家技术标准委员会）的统计测试集（STS）SP 800-22rev1a（2010年4月）或SP 800-90b（2018年1月）进行。

本文档结构如下：

- STM32微控制器随机数发生器概述（参见*第 1 节*）
- NIST SP800-22b测试集（请参见*第 2 节*）
- 运行NIST SP800-22b测试和分析需要执行的步骤（请参见*第 3 节*）
- NIST SP800-90b测试集（请参见*第 4 节*）
- 运行NIST SP800-90b测试和分析需要执行的步骤（请参见*第 5 节*）

表1. 适用产品

类型	产品	
	使用SP800-22rev1a检查	**使用 SP800-90b检查**
微控制器	STM32F2系列、STM32F4系列、STM32F7系列 STM32H742、STM32H743/753、STM32H745/755、STM32H747/757系列、STM32H750超值系列 STM32L0系列、STM32L4系列、STM32L4+系列	STM32H7A3/7B3系列、STM32H7B0超值系列、STM32L5系列

目录

图片目录

1 STM32 MCU RNG

1.1 引言

为加密应用程序使用的随机数发生器（RNG）通常会生成由随机的0或1位组成的序列。

随机数发生器基本上分为两类，分别是：

- 确定性RNG或伪RNG（**PRNG**）

  确定性RNG包含的算法会通过名为种子的初始值生成位序列。为确保向前不可预测性，获取种子时必须多加留意。如果已知种子和生成算法，PRNG生成的数值是完全可预测的。由于很多情况下生成算法是公开可用的，因此种子必须保密，并通过TRNG来生成。
- 非确定性RNG或真RNG（**TRNG**）

  非确定性RNG生成的随机性取决于一些不受人为控制的不可预测物理源（熵源）。

在一些STM32微控制器上实施的RNG硬件外设属于真随机数发生器。

1.2 STM32 MCU实施说明

下表列出了嵌入了RNG外设的基于STM32 Arm®(a)内核的微控制器。

表2. 嵌入RNG硬件外设的STM32系列

系列	STM32系列
STM32F2系列	STM32F2x5, STM32F2x7
STM32F4系列	STM32F405/415、STM32F407/417、STM32F410、STM32F427/437、STM32F429/439、STM32F469/479
STM32F7系列	STM32F7x5, STM32F7x6
STM32L0系列	STM32L05x、STM32L06x、STM32L072/073
STM32L4系列	STM32L4x6
STM32L4+系列	所有系列
STM32H7系列	STM32H742、STM32H743/753、STM32H745/755、STM32H747/757、STM32H750超值系列、STM32H7A3/7B3、STM32H7B0超值系列
STM32L5系列	STM32L5x2

arm

a. Arm是Arm Limited（或其子公司）在美国和/或其他地区的注册商标。

STM32 MCU中实施的真RNG基于模拟电路。该电路生成的连续模拟噪声用于RNG处理，以生成32位随机数。

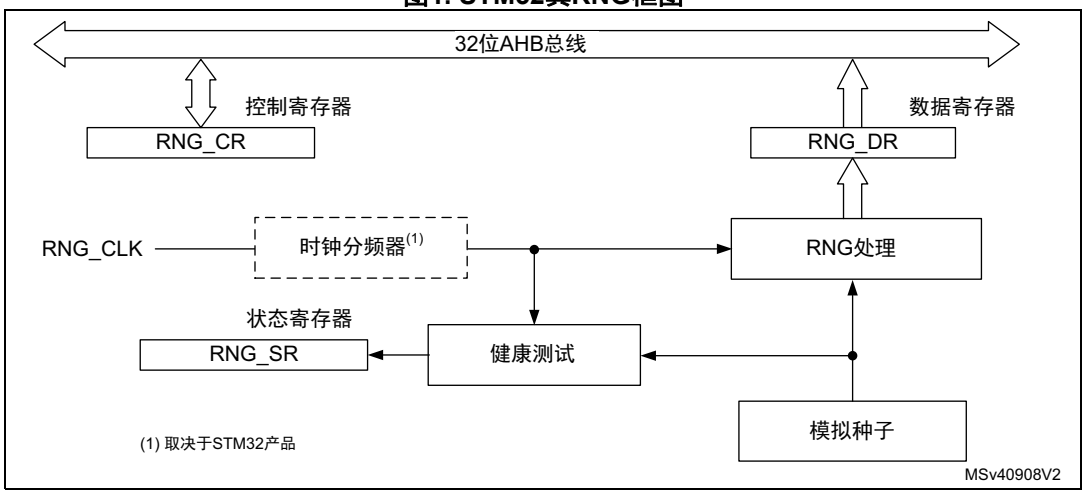
该模拟电路由几个环形振荡器组成，振荡器的输出进行异或运算。

RNG处理由采用恒定频率的专用时钟计时，对于微控制器的子设备，还可以使用RNG外设内部的分频器简化RNG专用时钟。

有关RNG外设的详细信息，参见STM32参考手册。

下图为STM32微控制器中真RNG的简化图。

图1. STM32真RNG框图



2 NIST SP800-22b测试集

2.1 引言

NISTSP800-22b统计测试集用于检验用于加密应用的随机数发生器的质量。NIST一篇标题为“*A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications*”的文章对该测试集进行了全面介绍。

2.2 NIST SP800-22b测试集说明

NIST SP800-22b统计测试集“sts-2.1.1”是由NSIT开发的软件包，可在NIST网站上下载（请在csrc.nist.gov上搜索*download the NIST Statistical Test Suite*）。

源代码使用ANSI C编写。NIST统计测试集由15种测试组成，用于验证二进制序列的随机性。这些测试主要针对序列中可能存在的各类非随机性问题。

这些测试可以如下分类：

- **频率测试**
  - 频率（单比特）测试

    测量0和1在序列中的分布情况，并检查结果是否与真随机数序列的预期结果相似。
  - 块中的频率测试

    检查M位块中1的频率是否近似为通过预期随机性原理得到的M/2。
  - 运行测试

    评估不同长度的1和0的预期运行总数是否是随机序列的预期结果。
  - 测试块中运行最长的“1”

    检查序列中的长运行“1”：
- **线性测试**
  - 二进制矩阵秩测试

    评估32x32二进制矩阵秩的分布。
  - 线性复杂度测试

    确定有限序列的线性复杂度。
- **相关性测试**（通过傅里叶变换）
  - 离散傅里叶变换（频谱）测试

    通过基于离散傅里叶变换的频谱测试评估位串的谱频率。此测试易受序列中的周期性影响。
- **查找特殊字符串测试**
  - 非重叠模板匹配测试

    评估m位非周期性组合的频率。
  - 重叠模板匹配测试

    评估m位周期性模板的频率。

- **熵测试**
  - Maurer“通用统计”测试
    评估L位块二进制序列的压缩率。
  - 连续测试
    评估所有2^m m位块的分布。

注： *对于m = 1的情况，连续测试相当于第 2.2 节的频率测试。*

  - 近似熵测试
    评估位串的熵，将所有m位组合的频率与所有(m+1)位组合的频率进行对比。

- **随机游走测试**
  - 累积和测试
    评估部分序列的和是否过大或过小；用于指示过多的 0 或 1。
  - 随机偏移测试
    评估随机游走周期内的状态分布。
  - 随机偏移变化测试
    检测与达到不同随机游走状态的预期次数的偏差。

上述测试中，每项测试都基于计算出的测试统计值，而测试统计值是测试序列的函数。

统计测试用于计算**Pvalue**，该值是完美随机数发生器生成的序列随机性小于被测序列的概率。

更多关于NIST统计测试集的详细信息，请参见以下NIST文章*A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications” Special Publication 800-22*
Revision 1a，该文章可从NIST网站中获取。

3 NIST SP800-22b测试集运行和分析

3.1 固件说明

如前一节所述，要运行NIST统计测试集，需要使用两个固件，一个固件位于STM32微控制器端，另一个位于NIST SP800-22b测试集端。

3.1.1 STM32 MCU侧

根据要求提供固件包。有关更多详细信息，请联系当地意法半导体销售代表。

该程序允许使用STM32RNG外设生成随机数。该程序还会在工作站上检索这些数，以便使用NIST统计测试集对其进行测试。

每个固件程序用于生成10个64 KB的随机数块。输出文件包含5,120,000个随机位，这些随机位将采用NIST统计测试进行测试。

根据NIST统计测试集的建议，输出文件格式可以为以下之一：

- 如果私有定义*FILE_ASCII_FORMAT* 在*main.c*文件中未被注释掉，则为ASCII 0和1组成的序列
- 如果私有定义*FILE_BINARY_FORMAT* 在*main.c*文件中未被注释掉，则为随机字节二进制文件。

更多关于程序说明和设置的详细信息，请参见固件包中的*自述*文件。

注： *可通过main.c文件中的SendToWorkstation()函数更改USART配置。*

可以通过如下修改main.c文件中的"Private define"（私有定义）更改输出值：
#define NUMBER_OF_RANDOM_BITS_TO_GENERATE 512000
#define BLOCK_NUMBER 10

3.1.2 在NIST SP800-22b测试集端

下载到工作站上后，NIST统计测试集包sts-2.1.1会检验STM32 RNG外设的输出文件的随机性。

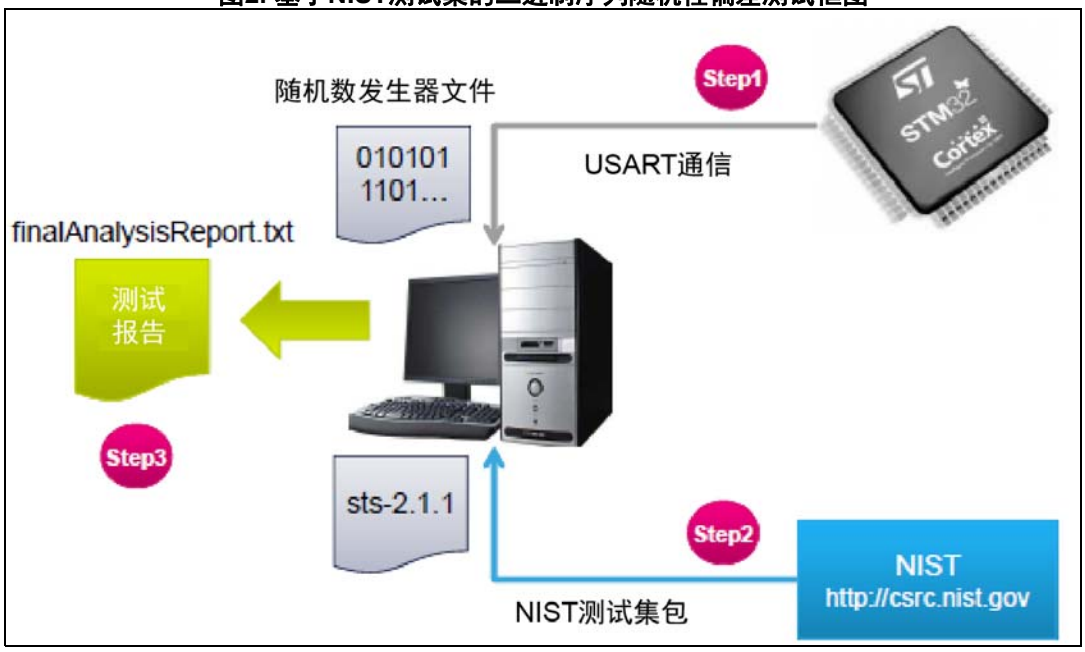
待分析的发生器文件必须存储在*data*文件夹(sts-2.1.1\data)下。

更多关于NIST统计测试工作原理的详细信息，请参见NIST文章 *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*中的入门知识章节。

3.2 NIST SP800-22b测试集步骤

下图介绍了使用NIST统计测试集包sts-2.1.1验证STM32 MCU生成的输出数随机性所需执行的步骤。

图2. 基于NIST测试集的二进制序列随机性偏差测试框图



3.2.1 第一步：随机数发生器

将STM32板件连接至工作站。根据板件类型建立如下连接：

- 通过零调制解调器母头/母头RS232线缆
- 通过A转mini-B型USB线缆

STM32RNG通过UART固件运行，以生成*第 3.1.1 节：STM32MCU侧*中所述的随机数。使用PuTTY（免费开源终端仿真器，串行控制台和网络文件传输应用程序）等终端仿真应用程序将数据存储在工作站上。

3.2.2 第二步：NIST统计测试

按照NIST统计测试集文件中所述使用visual C++编译器编译sts-2.1.1程序包，以生成可执行程序。

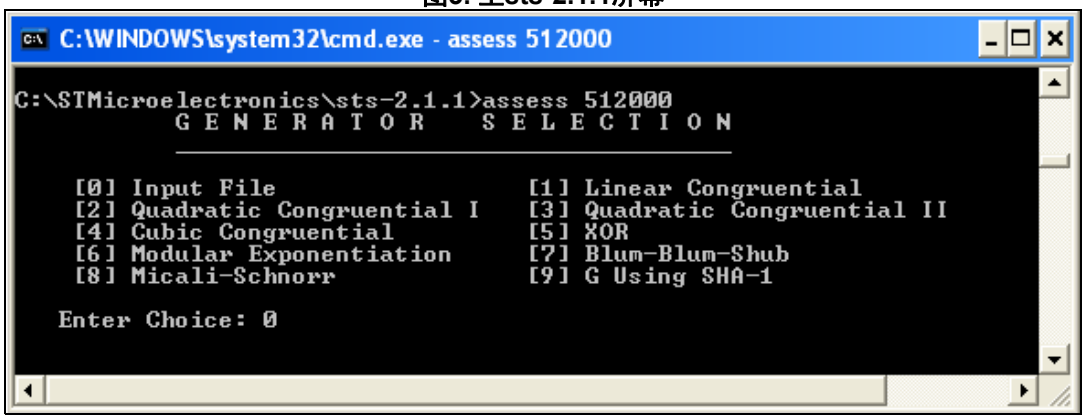
运行NIST统计测试集程序之后，会显示一系列菜单提示，供用户选择要分析的数据以及要应用的统计测试。

在该应用笔记中，NIST统计测试集会在名称*assess.exe*下进行编译，并会保存到*NIST_Test_Suite_OutputExample*文件夹下。如前文所述，随机数定义为*每个块512000*位。

各个步骤的详细说明如下：

1. 显示的第一个屏幕如下所示。

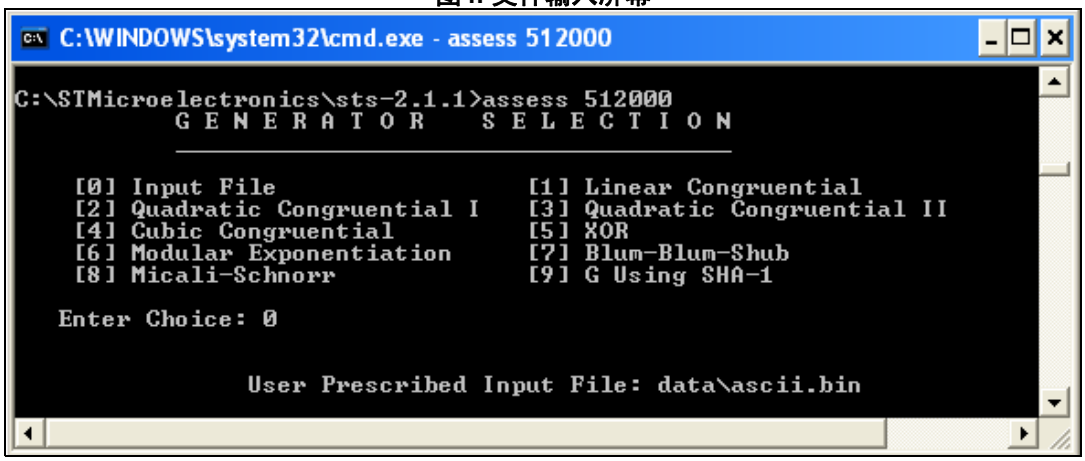
图3. 主sts-2.1.1屏幕



如果输入数值0，程序会要求输入待测试随机数的文件名和路径。

2. 第二个屏幕如下所示。

图4. 文件输入屏幕

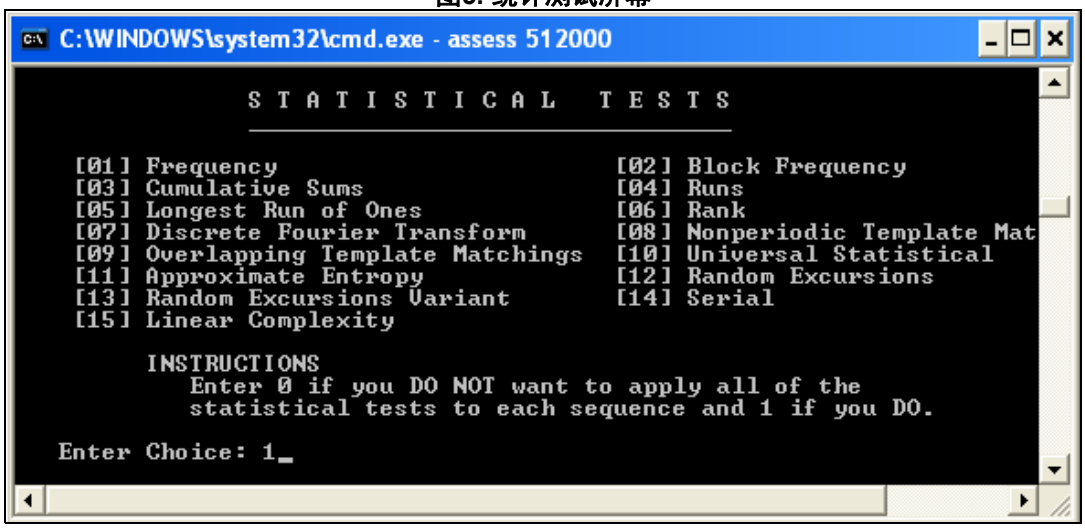


该应用笔记提供的每个系列的示例由两个文件组成，文件是使用NIST建议的以下文件格式通过STM32 RNG生成的：

– *ascii.bin*：由ASCII 0和1组成的序列
– *binary.bin*：数据文件中的每个字节包含8个数据位

3. NIST统计测试集会显示可通过以下所示屏幕运行的15个测试。

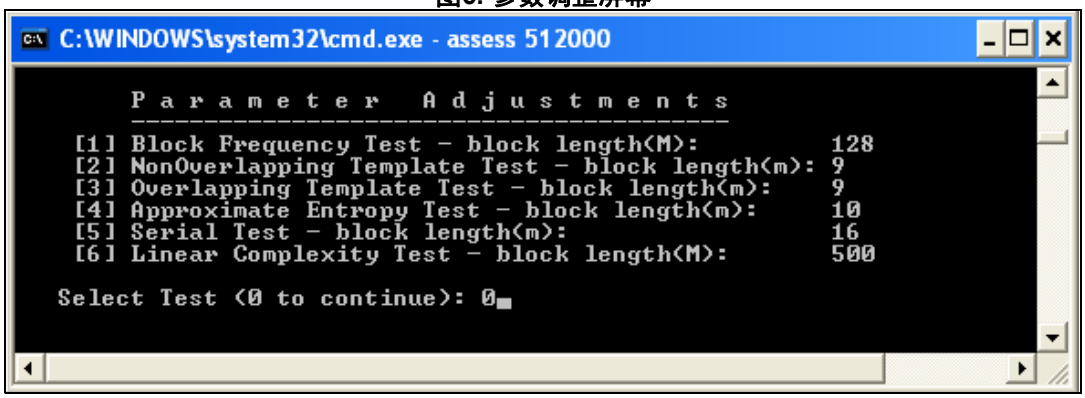
图5. 统计测试屏幕



在这种情况下，选择了“1”应用到所有统计测试。

4. 可以在以下所示屏幕中进行参数调整。

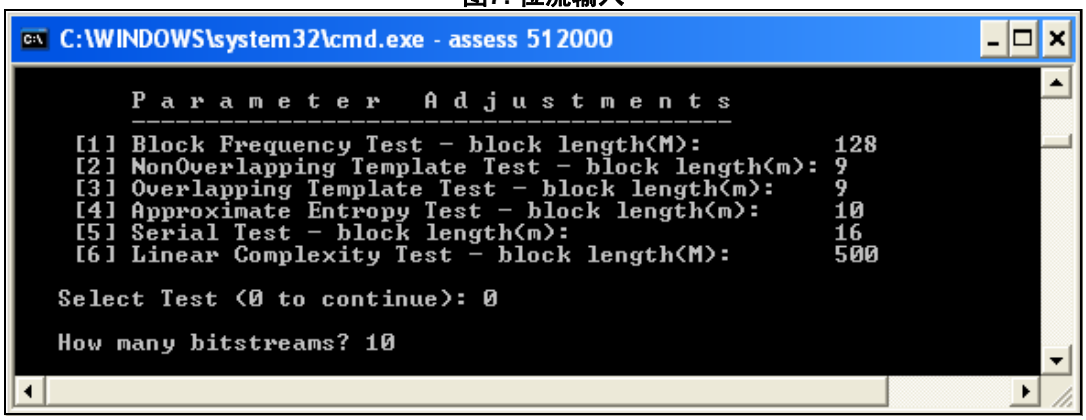
图6. 参数调整屏幕



本例中，会保持默认设置，并会选择“0”值进入下一步。

5. 用户需要提供位流数。

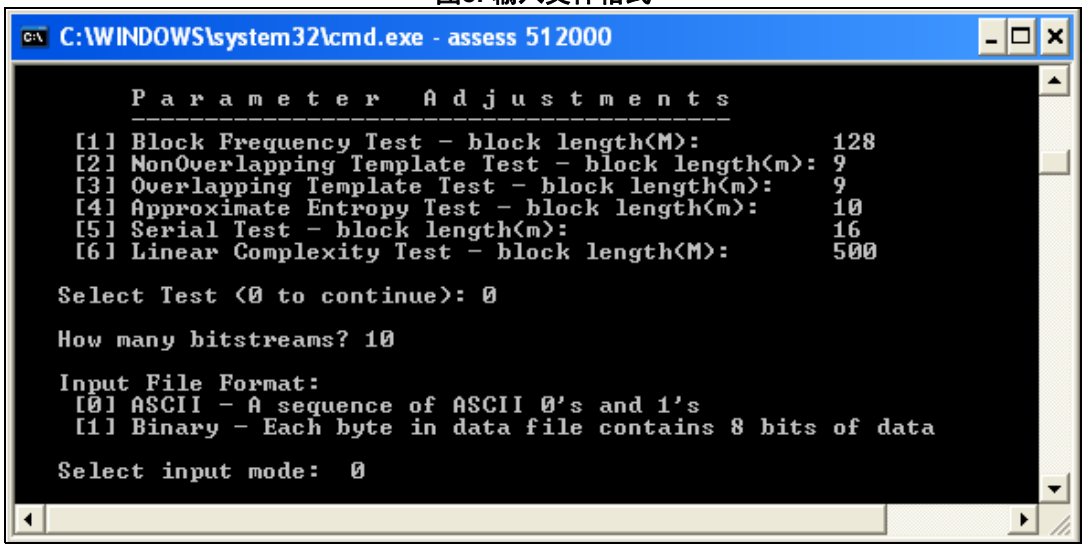
图7. 位流输入



NIST统计测试集要求输入比特流数：在该示例中输入了10，表示选择10个512 KB（5,12 MB）的块。

6. 随后用户必须通过以下屏幕指定文件包含的是以ASCII格式存储的位还是以二进制格式存储的十六进制字符串。

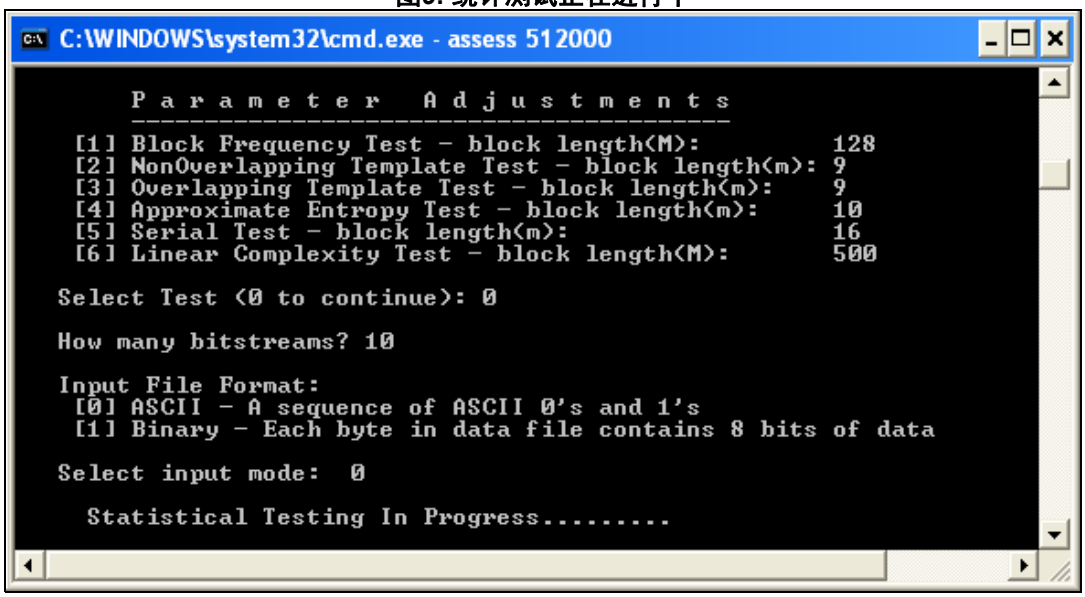
图8. 输入文件格式



选择了“0”值是因为文件采用的是ASCII格式。

7. 输入所有必填输入后，NIST统计测试集会开始对输入文件进行分析。

图9. 统计测试正在进行中



8. 完成测试流程后，将显示以下屏幕。

图10. 统计测试完成

统计测试结果可以在sts-2.1.1\experiments\AlgorithmTesting中找到。

3.2.3 第三步：测试报告

NIST统计测试提供的分析例程有助于对分析结果进行解读。统计测试完成后，会生成一个名为*finalAnalysisReport*的文件，文件保存位置为*sts2.1.1\experiments\AlgorithmTesting*。

报告包含15项测试的实验结果摘要（参见*附录A*）。

NIST统计测试还针对每项测试提供详细报告，报告保存位置为*sts-2.1.1\experiments\AlgorithmTesting\<测试集名称>*。

以下两个示例的完整NIST统计测试集输出报告位于“NIST_Test_Suite_OutputExample下：

- *Ascii_File_Format*示例，带两个文件夹：
  - *Input_File*：包含采用ascii格式保存的随机数发生器。
  - *Final_Analysis_Report*：包含基于该输入文件的完整NIST统计测试集输出报告、实验结果摘要以及每个测试对应的报告。
- *Binary_File_Format*示例，带两个文件夹：
  - *Input_File*：包含采用二进制格式保存的随机数发生器。
  - *Final_Analysis_Report*：包含基于该输入文件的完整NIST统计测试集输出报告、实验结果摘要以及每个测试对应的报告。

4 NIST SP800-90b测试集

4.1 引言

密码随机位发生器（RBG）又称为随机数发生器（RNG），需要使用噪声源产生具有某种程度的不可预测性的输出，这种不可预测性表示为最小熵。

NIST SP800-90b统计测试集的特殊性在于通过其估计熵源质量的标准化方法，检验用于加密应用的随机生成器的质量。

NIST一篇标题为“*Recommendation for the Entropy Sources Used for Random Bit Generation*”的文章全面介绍了该套件。

4.2 NIST SP800-90b测试集说明

IST SP800-90b统计测试集可以从GitHub网站（https://github.com/usnistgov/SP800-90B_EntropyAssessment）中下载。

SP800-90B_EntropyAssessment C++程序包实现了特殊出版物800-90B中包含的最小熵评估方法。

项目由两个单独的部分组成：

- IID测试，确认数据集为IID（独立且分布均匀）
- 非IID测试，为所提供的任何数据提供对最小熵的估计

使用非IID测试对STM32可认证的TRNG噪声源进行测试。

4.2.1 非IID跟踪：非IID数据的熵估计

并非所有的噪声源都能产生IID输出。具有相关值的序列会导致对熵估计过高。但是，不同的估计数量减少了对源熵估计过高的概率。

对于非IID数据，必须计算出噪声源的输出和任何调节组件的输出的各个估计量（以下详细说明），但以下情况除外：

- 已针对键控调节组件进行审查的三种键控算法：HMAC、CMAC和CBC-MAC
- 已针对非键控调节组件进行审查的三个非键控函数：FIPS 180或FIPS 202中指定的所有经批准的哈希函数，Hash_df和Block_Cipher_df

注： *STM32可认证的TRNG使用经批准的CMAC调节组件（NIST CAVP编号C1327）。*

将所有估计中的最小值作为该建议的熵源的熵评估。

估计器如下：

- **最常用值估计**：为输入数据集中最常用值的比例p构建出置信区间，然后根据该置信区间的上限估计每个样本的最小熵。
- **碰撞估计**：用于测量数据集中第一个重复值的平均样本数。该方法根据碰撞次数估计最可能输出值的概率。
- **马尔可夫估计**：因为样本值始终取决于前一个样本的值，因此马尔可夫模型可以作为模板用于测试具有相关性的源。在测量了来自输入数据集的连续值之间的相关性之后，该模型提供基于输出的任何子序列中存在的熵的最小熵估计值，而不是每个输出的最小熵的估计值。
- **压缩估计**：基于数据集的压缩量，通过压缩估计值来计算熵率。该估计是通过生成值的字典，然后基于该字典计算产生输出所需的平均样本数，进行计算。即使在使用该统计测试具有相关性的序列时已经达到了压缩率，也仍然可以获得熵率。
- **T元组估计**：检查出现在输入数据集中的t元组（如二元组或三元组）的频率，并基于该频率估计每个样本的熵。
- **最长重复子串（LRS）估计**：该方法基于输入数据集中重复子串（元组）的数量，估计源的碰撞熵。这是一种补充估计，因为其处理对象是对t元组估计值而言过大的元组大小
- **窗口中多重最常用（multiMCW）预测估计**：基于最后n个输出，多重最常见窗口中每个子预测器旨在猜测下一个输出。multiMCW预测器记录每个子预测器正确预测在n个先前输出的窗口中最常出现的值的次数。要预测下一个值，我们使用预测最正确的子预测器。
- **滞后预测估计**：基于指定的滞后，滞后预测器的每个子预测器预测出下一个输出。记分板由滞后预测器保存，用于记录每个子预测器正确的预测次数，并且预测最正确的子预测器用于预测下一个值。
- **计数型多重马尔可夫模型（MultiMMC）预测估计**：由多个MMC子预测器组成。由每个MMC预测器记录从一个输出到后续输出的转换频率，并基于从当前输出中最常观察到的转换进行预测。n个MMC子预测器中的每1至n个深度并行运行，并且预测数最正确的一个用于预测下一个值。
- **LZ78Y预测估计**：目前为止已添加到字典中的字符串将保存在预测器字典中。该字典会不断添加新的字符串，直到达到其最大容量。每次处理样本时，最近n个样本中的每个子字符串都会更新或添加到字典中。

5 NIST SP800-90b测试集运行和分析

5.1 固件说明

要按照上一节中的说明运行NIST统计测试集，需要使用两个固件，一个固件位于STM32微控制器端，另一个位于NIST SP800-90b测试集端。

5.1.1 STM32 MCU端

根据要求提供固件包。有关更多详细信息，请联系当地意法半导体销售代表。

该程序允许使用STM32RNG外设生成随机数。该程序还会将这些数推送到工作站，以便使用NIST统计测试集对其进行测试。

每个固件程序用于生成两个64 KB的随机数块。输出文件包含1,024,000个随机位，这些随机位将采用NIST统计测试进行测试。

注： *可通过main.c文件中的SendToWorkstation()函数更改USART配置。*

可以通过如下修改main.c文件中的"Private define"（私有定义）更改输出值：
#define NUMBER_OF_RANDOM_BITS_TO_GENERATE 512000
#define BLOCK_NUMBER 2

5.1.2 NIST SP800-90b测试集端

下载到工作站上后，NIST统计测试集包会检验STM32 RNG外设的输出文件的随机性。

待分析的发生器文件必须存储在*bin*文件夹(...\bin\data)下。

有关NIST统计测试的工作原理的更多详细信息，请参见GitHub网站（https://github.com/usnistgov/SP800-90B_EntropyAssessment）上提供的*自述*文件。

5.2 NIST SP800-90B测试集步骤

5.2.1 第一步：随机数发生器

通过A转mini-B型USB线缆将STM32板连接到工作站

STM32RNG通过UART固件运行，以生成*第 5.1.1 节：STM32MCU端*中所述的随机数。使用PuTTY（免费开源终端仿真器，串行控制台和网络文件传输应用程序）等终端仿真应用程序将数据存储在工作站上。

5.2.2 第二步：NIST统计测试

Makefile用于编译程序，如*第 5.1.2 节*中所述*readme*文件中所述。

对于非IID测试，用户必须遵循以下详细步骤：

1. 使用Makefile编译程序：make non_iid
2. 使用以下操作运行程序：

./ea_non_iid [-i|-c] [-a|-t] [-v] [-l <index>,<samples>] <file_name> [bits_per_symbol]

其中

- **-i** 表示数据是无条件的并返回初始熵估计（默认）。
- **-c** 表示数据是有条件的。
- **-a** 估计二进制文件中所有数据的熵（默认）。
- **-t**将创建的数据的位串表示截断为前1 MB。
- **-l** 按*字节索引到文件中后（最多）读取数据样本。
- **-v**：用于更多输出的详细级别标志（可选，可多次使用）
- **bits_per_symbol**： 每个符号的位数。每个符号应适合单个字节。

示例： ./ea_non_iid ../bin/l5.bin 1 -i -t -v

5.2.3 第三步：测试报告

NIST统计测试提供的分析例程有助于对分析结果进行解读：

- 对于非IID测试，将提供每个IID测试的结果，并在最后提供最终最小熵。

6 结论

该应用笔记介绍了使用NIST统计测试集SP800-22rev1a（2010年4月版）或SP800-90B（2018年1月版）检验由STM32微控制器随机数发生器外设生成的数字随机性的主要规则和步骤。

附录A NIST SP800-22b统计测试集

结果以表格形式表示，表格中包含p行、q列，其中：

- p，行数，对应于应用的统计测试数
- q，列数（q = 13），分布如下：
  - 第1-10列对应于10个P-value的频率
  - 第11列是通过应用chi-square test11得出的Pvalue
  - 第12列是通过的二进制序列所占的比例
  - 第13列是对应的统计测试

以下示例为测试结果的第一部分和最后一部分。更多详细信息，请参见*sts-2.1.1\experiments\AlgorithmTesting*下的*finalAnalysisReport*文件。

第1部分

P-VALUES一致性与通过序列比例对应的结果

发生器为<data/ascii.bin>

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
0	1	2	1	2	1	1	1	0	1	0.911413	10/10	Frequency
1	1	0	1	3	0	2	1	1	0	0.534146	10/10	BlockFrequency
0	1	3	3	0	1	0	2	0	0	0.122325	10/10	CumulativeSums
1	1	3	1	0	1	1	1	0	1	0.739918	10/10	CumulativeSums
2	0	2	2	1	1	0	1	0	1	0.739918	10/10	Runs
1	0	1	1	0	3	1	1	0	2	0.534146	9/10	LongestRun
1	2	1	0	2	1	1	0	0	2	0.739918	10/10	Rank
3	0	1	2	1	1	0	1	0	1	0.534146	9/10	FFT
1	1	1	0	0	2	1	2	0	2	0.739918	10/10	NonOverlappingTemplate
1	1	0	0	1	1	1	3	0	2	0.534146	10/10	NonOverlappingTemplate
0	2	1	0	4	0	2	0	0	1	0.066882	10/10	NonOverlappingTemplate
0	0	0	1	1	3	0	2	1	2	0.350485	10/10	NonOverlappingTemplate
0	1	2	2	1	1	1	2	0	0	0.739918	10/10	NonOverlappingTemplate
2	2	1	0	2	0	1	1	1	0	0.739918	10/10	NonOverlappingTemplate
1	0	2	2	1	1	1	0	1	1	0.911413	10/10	NonOverlappingTemplate
0	0	1	1	0	0	2	3	1	2	0.350485	10/10	NonOverlappingTemplate

第2部分

2	0	1	0	1	2	1	0	2	1	0.739918	10/10	OverlappingTemplate
1	0	2	1	0	2	2	1	1	0	0.739918	10/10	Universal
1	1	0	0	2	0	2	3	1	0	0.350485	10/10	ApproximateEntropy
0	1	1	1	1	0	0	0	1	0	----	5/5	RandomExcursions
1	1	0	0	2	0	0	0	0	1	----	5/5	RandomExcursions
0	1	1	1	0	0	0	0	1	1	----	5/5	RandomExcursions
0	0	0	0	0	1	1	0	2	1	----	5/5	RandomExcursions
1	0	0	0	3	0	0	0	1	0	----	5/5	RandomExcursions
0	0	0	1	1	0	0	1	1	1	----	5/5	RandomExcursions
1	0	1	1	0	2	0	0	0	0	----	5/5	RandomExcursions
1	0	0	0	1	1	1	0	1	0	----	5/5	RandomExcursions
2	1	0	1	1	0	0	0	0	0	----	5/5	RandomExcursionsVariant
2	1	0	0	1	1	0	0	0	0	----	5/5	RandomExcursionsVariant
1	1	0	2	1	0	0	0	0	0	----	5/5	RandomExcursionsVariant
1	2	0	1	1	0	0	0	0	0	----	5/5	RandomExcursionsVariant
1	1	1	1	0	0	0	1	0	0	----	5/5	RandomExcursionsVariant
1	1	0	1	1	0	0	0	0	1	----	5/5	RandomExcursionsVariant
0	1	0	2	1	0	0	0	0	1	----	5/5	RandomExcursionsVariant
0	0	0	1	0	1	0	3	0	0	----	5/5	RandomExcursionsVariant
0	0	0	0	0	0	2	1	1	1	----	5/5	RandomExcursionsVariant
0	0	1	0	0	0	1	1	1	1	----	5/5	RandomExcursionsVariant
0	0	0	1	0	0	2	0	2	0	----	5/5	RandomExcursionsVariant
0	1	0	0	1	1	1	1	0	0	----	5/5	RandomExcursionsVariant
1	0	0	2	0	1	1	0	0	0	----	5/5	RandomExcursionsVariant
1	0	0	0	2	1	0	0	0	1	----	5/5	RandomExcursionsVariant
0	0	0	1	1	0	1	1	1	0	----	5/5	RandomExcursionsVariant
0	0	0	0	2	0	2	0	0	1	----	5/5	RandomExcursionsVariant
0	0	1	0	1	2	1	0	0	0	----	5/5	RandomExcursionsVariant
0	0	1	0	0	2	2	0	0	0	----	5/5	RandomExcursionsVariant
1	1	0	0	0	2	3	0	2	1	0.350485	10/10	Serial
0	2	1	0	3	1	0	1	1	1	0.534146	10/10	Serial
2	1	1	1	0	1	1	3	0	0	0.534146	10/10	LinearComplexity

- -

对于大小为10个二进制序列的样本，每次统计测试（随机偏移（变化）测试）的最低通过率大概是8。

对于大小为5个二进制序列的样本，随机偏移（变化）测试的最低通过率大概是4。

要获得更具体的指南，请使用文档附录部分中提供的MAPLE程序构建可能性表。

附录B NIST SP800-90b统计测试集

以下是执行非IID测试的示例。

$ ./ea_non_iid ../bin/l5.bin 1 -i -t -v

打开文件：'../bin/l5.bin'

二进制符号数：1024000

符号字母由2个唯一符号组成

正在运行非IID测试...

运行最常用值估计...

MCV估计：模式= 530185，p-hat = 0.51775878906249995，p_u = 0.51903071907139675

最常用值估计（位串）= 0.946108/1位

正在运行熵统计估计（仅位串）...

碰撞估计：X-bar = 2.4954732991667945，sigma-hat = 0.4999801177822241 2，p = 0.55717151750062044

碰撞测试估计（位串）= 0.843807/1位

马尔可夫估计：P_0=0.4822412109375，P_1=0.51775878906249995，P_0,0=0.48618305677846313，P_0,1=0.51381694322153693，P_1,0=0.47857068759018079，P_1,1=0.52142931240981927，p_max = 6.2798397734367098e-37

马尔可夫测试估计（位串）= 0.939536/1位

压缩估计：X-bar = 5.2113069751685934，sigma-hat = 1.0187463366852749，p = 0.035431813237513987

压缩测试估计（位串）= 0.803135/1位

正在运行元组估计...

t元组估计：t = 16，p-hat_max = 0.53187518804173173，p_u = 0.53314533218239224

LRS估计：u = 17，v = 38，P_{max,W} = 0.50423097749594759，p_u = 0.50550366496585808

T元组测试估计（位串）= 0.907399/1位

LRS测试估计（位串）= 0.984207/1位

正在运行预测器估计...

MultiMCW预测估计：N = 1023937，Pglobal' = 0.51634782805743162 (C = 527405) Plocal = 0.42674646958653317 (r = 21)

窗口中多重最常见（MultiMCW）预测测试估计（位串）= 0.953585/1位

滞后预测估计：N = 1023999，Pglobal' = 0.50526439621655594 (C = 516087) Plocal = 0.40830971576974662 (r = 20)

滞后预测测试估计（位串）= 0.984890/1位

MultiMMC预测估计：N = 1023998，Pglobal' = 0.51899462537798435 (C = 530147) Plocal = 0.42674521449187308 (r = 21)

计数型多重卡尔可夫模型（MultiMMC）预测测试估计（位串）= 0.946208/1位

LZ78Y预测估计：N = 1023983，Pglobal' = 0.51899440603936897 (C = 530139) Plocal = 0.42674552311446512 (r = 21)

LZ78Y预测测试估计（位串）= 0.946209/1位

H_original：1.000000

H_bitstring：0.803135

min（H_original，1 X H_bitstring）：0.803135

版本历史

表3. 文档版本历史

日期	版本	变更
2013年5月13日	1	初始版本。
2016年6月22日	2	更新了： – *简介*。 – *第1节：STM32微控制器随机数发生器*。 – *图1：框图*。 – *第3.1节：固件描述*。 – *第3.2.1节：第一步：随机数发生器*。 – *第3.2.2节：第二步：NIST统计测试*。 – *第4节：总结*。 增加了*图2：嵌入RNG硬件外设的STM32系列*。
2019年10月1日	3	更新了： – *简介*和*表1：适用产品* – *表2：嵌入RNG硬件外设的STM32系列* – *图1：STM32真RNG框图* – *第6节：总结* – *附录A：NIST SP800-22b统计测试集* 增加了： – *第4节：NIST SP800-90b测试集* – *第5节：NIST SP800-90b测试集运行和分析* – *附录B：NIST SP800-90b统计测试集*
2019年10月10日	4	更新了*表2：嵌入RNG硬件外设的STM32系列*。
2020年1月8日	5	更新了： – *表 1：适用产品* – *表 2：嵌入RNG硬件外设的STM32系列* – *第 5.1.1节： STM32 MCU端*

表4. 中文文档版本历史

日期	版本	变更
2021年8月26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。若需 ST 商标的更多信息，请参考 *www.st.com/trademarks*。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。