life.augmented

AN4073
应用笔记

使用 STM32F2xx 和 STM32F4xx 微控制器时如何提高 ADC 测量精度

前言

本应用笔记旨在展示对于使用 STM32F2xx 和 STM32F4xx 微控制器的应用，怎样提高其 A/D 转换的精度。

本文还讲解了可应用于降低 ADC 误差的固件方法，给出了为得到更好的 ADC 精度，编写固件的一些通用技巧。

请注意，本应用笔记提供的数据仅供参考。除非特别指出，测量条件皆为实验室典型条件而非生产环境。

表 1 列出了本应用笔记涉及的微控制器。

表 1. 适用产品

类型	部件编号
微控制器	STM32F2xx (STM32F20x， STM32F21x) STM32F4xx (STM32F405， STM32F407， STM32F415， STM32F417， STM32F42x， STM32F43x)

www.st.com

目录

1 会影响 ADC 精度的参数概述 6

2 提高转换精度的固件技巧 7

2.1 平均 7

2.1.1 平均 N 个 ADC 采样 7

2.1.2 平均 N-X 个 ADC 采样 8

2.2 其他建议 10

3 STM32F2 和 STM32F4 实际测量 11

3.1 测量条件 11

3.1.1 硬件设置 11

3.1.2 固件设置 11

3.2 结果 11

3.2.1 当 ART 为 ON 时的 ADC 测量 12

3.2.2 当 ART 为 OFF 时的 ADC 测量 13

3.2.3 当 （数据 + 指令）缓存 ON+ 预取 OFF 时的 ADC 测量 15

3.3 时间考虑 17

3.4 测量结论 17

4 STM32F4 ADC 精度选项 18

4.1 ADC 精度的配置选项 18

4.1.1 选项 1 18

4.1.2 选项 2 18

4.2 实际测量 19

4.2.1 硬件设置 19

4.2.2 通用固件设置 19

4.2.3 结果 19

4.3 测量结论 24

附录 A 平均 N 个 ADC 采样：源代码. 25

附录 B 平均 N-X 个 ADC 采样：源代码. 26

附录 C 激活选项 1 和选项 2 的固件顺序 28

C.1 选项 1 28

C.2 选项 2 29

修订历史 30

表格索引

表 1. 适用产品 1
表 2. 当 ART 为 ON 时的 ADC 编码分布 （单位为 LSB） 12
表 3. 当 ART 为 OFF 时的 ADC 编码分布 （单位为 LSB） 13
表 4. 当 （数据 + 指令）缓存 ON + 预取 OFF 时的 ADC 编码分布 （单位为 LSB）15
表 5. 计算平均所需时间 17
表 6. ADCxDC2 的使用与 ADC 模式 19
表 7. 当 ART 为 ON 时的 ADC 编码分布 20
Table 8. 当 ART 为 OFF 时的 ADC 编码分布 21
表 9. 当 （数据 + 指令）缓存 ON + 预取 OFF 时的 ADC 编码分布 22
表 10. 文档修订历史 30

图片索引

图 1. 平均技巧的图形表示 7
图 2. N 个采样平均算法 8
图 3. 平均 N-X 个 ADC 采样算法 9
图 4. 柱状图图形表示与编码的离散度 12
图 5. 当 ART 为 ON， V_{IN} = 0.3 V 时的 ADC 编码分布 13
图 6. 当 ART 为 OFF， V_{IN} = 1.65 V 时的 ADC 编码分布 14
图 7. 当 （D+I）缓存 ON + 预取 OFF V_{IN} = 0.3 V 时的 ADC 编码分布 15
图 8. 当 （D+I）缓存 ON + 预取 OFF V_{IN} = 1.65 V 时的 ADC 编码分布 16
图 9. 当 （D+I）缓存 ON + 预取 OFF V_{IN} = 3 V 时的 ADC 编码分布 16
图 10. 当 ART 为 ON， V_{IN} = 0.3 V 时的 ADC 编码分布 20
图 11. 当 ART 为 OFF， V_{IN} = 1.65 V 时的 ADC 编码分布 21
图 12. 当 （D+I）缓存 ON + 预取 OFF V_{IN} = 0.3 V 时的 ADC 编码分布 22
图 13. 当 （D+I）缓存 ON + 预取 OFF V_{IN} = 1.65V 时的 ADC 编码分布 23
图 14. 当 （D+I）缓存 ON + 预取 OFF V_{IN} = 3 V 时的 ADC 编码分布 23

1 会影响 ADC 精度的参数概述

模数转换的精度会影响系统的质量和效率。为提高精度，您需要理解与 ADC 有关的误差，以及对它们有影响的参数。

ADC 本身无法确保结果的精度。它取决于您的系统设计。因此，在您开始开发之前，需要仔细的做些准备。

具体影响 ADC 精度的参数与应用密切相关。比较常见的因素有：PCB 布局、参考电压源、I/O 切换所造成的影响和模拟输入源的输出阻抗。

若需 ADC 误差的更详细信息，请参考应用笔记 AN2834：*如何在 STM32F10xxx 器件中得到最佳的 ADC 精度*及 AN3137：*STM8L 器件上的 A/D 转换器*。

2 提高转换精度的固件技巧

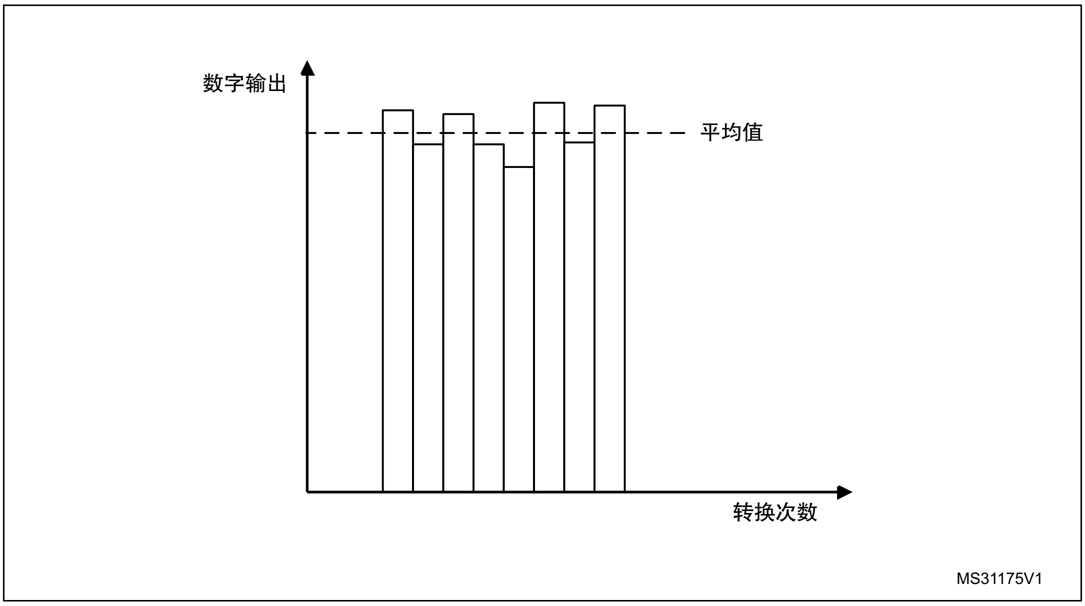
2.1 平均

平均是一个简单的技巧，即您对模拟输入多次采样，取所有转换结果的均值。此技巧有助于消除模拟输入上的噪声或错误转换的影响。

2.1.1 平均 N 个 ADC 采样

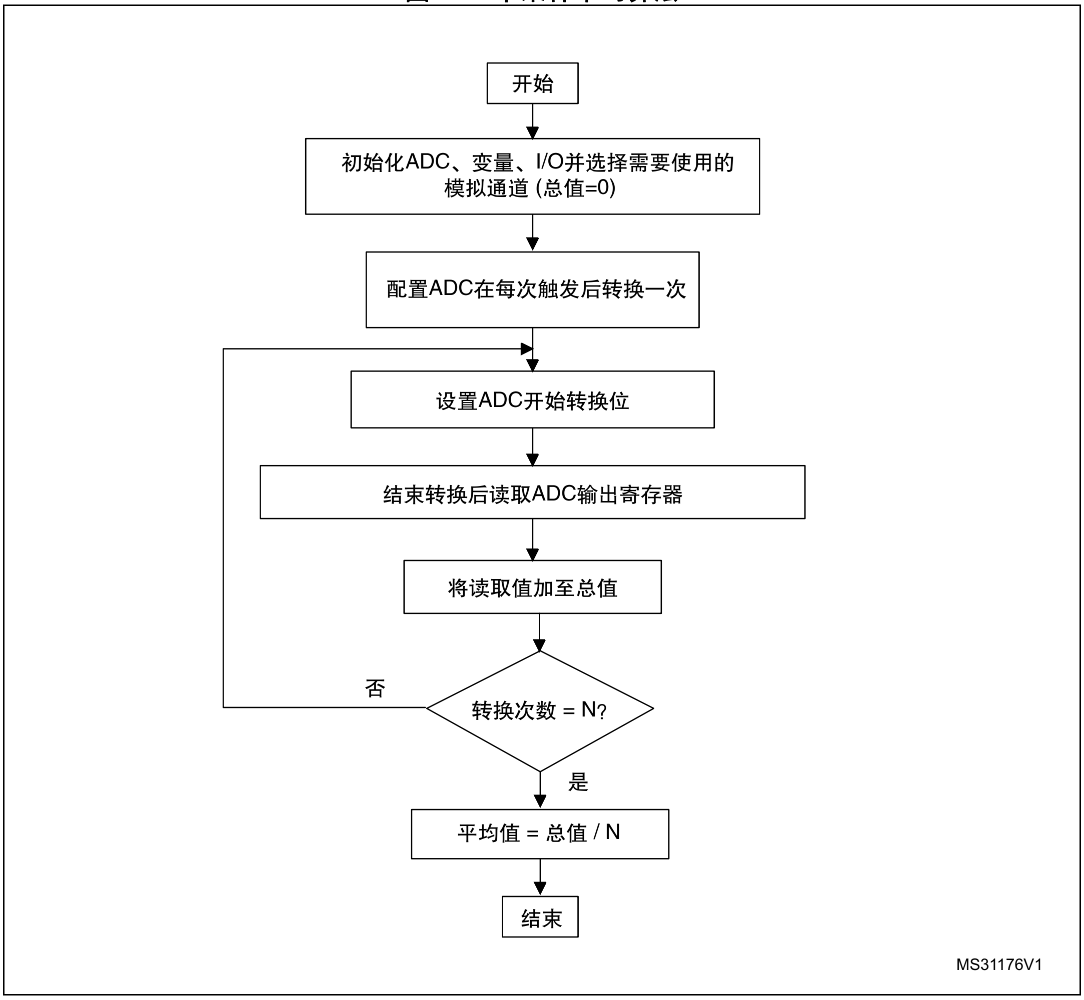
当使用此方法时，采样数最好为 2 的倍数 （N 应为 2 的倍数）。这会使平均计算更有效率，因为对转换值之和求平均时的除法可通过向右移位做到。这节省了 CPU 时间和执行除法算法所需的代码存储器 （在 Cortex-Mx 内核中，右移操作仅花费 1 个 CPU 周期）。

图 1. 平均技巧的图形表示



此平均技巧用于测量一个模拟输入引脚上的电压。总共 N 个转换计算平均。这在固件中用循环做到。

图 2. N 个采样平均算法



总转换时间 = （采样数 * ADC 转换时间）+ 计算时间。

计算时间 = 读取结果、相加、将总数除以采样数的平均计算所花费的时间。

总转换时间和用于平均的采样数之间存在折中，取决于模拟信号的变化范围及计算可用的时间。

注： 若需所用代码源的更详细信息，请参考附录 A。

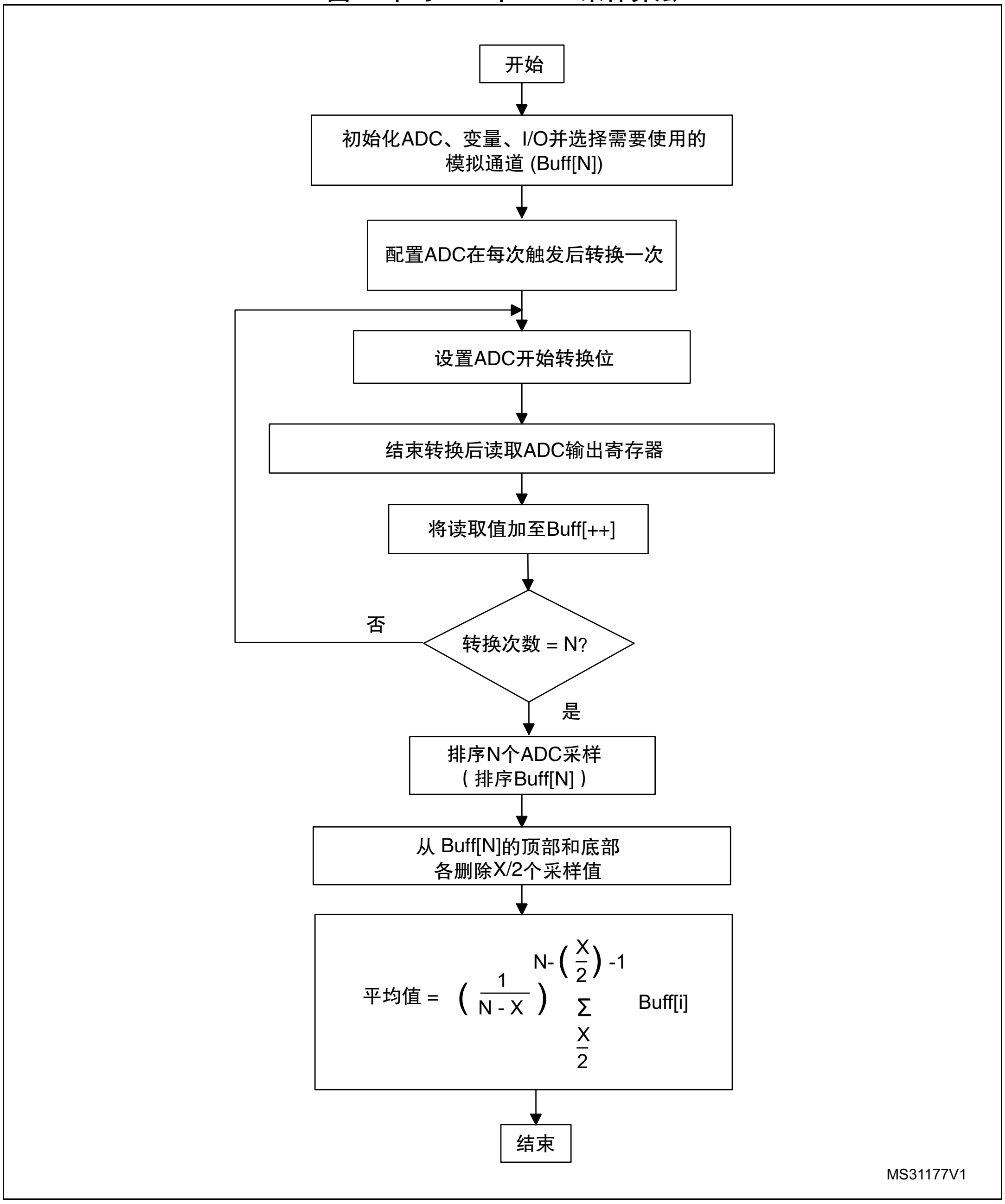
2.1.2 平均 N-X 个 ADC 采样

此方法基于先取 N 个 ADC 采样，将它们从最大值到最小值 （或相反顺序）排序，然后删除两端的 X 个采样。

建议选择 N 和 X 为 2 的倍数。

此平均方法比之前一个更有效，因为它删除了会影响平均的两端值，而且它在执行时间和转换精度之间给出了一个较好的折中。

图 3. 平均 N-X 个 ADC 采样算法



总转换时间 = （采样数 * ADC 转换时间）+ 计算时间。

计算时间 = 读取结果、对 N 个 ADC 采样排序、删除 X 个两端的采样、将剩余的 ADC 采样加在一起、将总数除以 N-X 计算平均所花费的时间。

注： *若需所用代码源的更详细信息，请参考附录 B。*

2.2 其他建议

ADC 转换结果为输入电压与参考电压的比值。如果参考电压不准确 （噪声影响），转换结果也会不准确。因此，为取得更高的转换精度，除了固件设计外，硬件设计时也需要尽可能的降低噪声。

微控制器在执行指令时会在内部电源网络上产生一些不容忽视的噪声。若要过滤此噪声，可使用微控制器封装上的 V_{DDA} （或 V_{REF}）和 VSSA 模拟供电引脚；您可连接一个电容滤波器至这些电源引脚以过滤高频噪声。

这里是一些通用的固件设计技巧，可用来降低系统噪声，达到更好的 ADC 转换精度：

1. 在即将开始 ADC 转换时，避免启动任何通信外设的数据传输。 I/O 的切换可能会在供电电压中产生一些噪声。
2. 避免切换高灌电流 I/O，因为会在电源中产生噪声波纹。
3. 避免在同一 I/O 端口上切换数字输出，因为 A/D 输入正在被转换。这会将切换噪声引入模拟输入中。
4. 建议将 STM32F2/F4 ART 配置为启用数据缓存 + 指令缓存，禁用预取。这可避免 CPU 对 Flash 额外访问而导致额外的噪声，在一些应用中这会极大降低 ADC 精度。

3 STM32F2 和 STM32F4 实际测量

本章仅适用于所有 STM32F2xx、 STM32F405、 STM32F415、 STM32F407、 STM32F417 微控制器。

本章给出了一些 ADC 精度测量，将之前讲解的方法付诸实践。

3.1 测量条件

3.1.1 硬件设置

- STM32F407ZGT6 焊接于测试板上 （仅需最少数量的其它硬件元件）
- 环境温度：25°C
- MCU 上有 3 个电源：V_{DD}/V_{SS}、 V_{DDA}/V_{SSA}、 V_{REF+}/V_{REF-}
- 电源范围：$V_{DD}=V_{DDA}=V_{REF+} = 3.3$ V， $f_{ADC} = 36$ MHz
- 时钟源：外部时钟 （8 MHz）由发生器提供， PLL 启用， $f_{CPU} = 144$ MHz
- 测试三个固定的模拟输入电压：0.3 V、 1.65 V、 3 V

3.1.2 固件设置

- ADC 通道 2 用于单次转换模式
- 对每个固定模拟输入电压采集 50000 次
- 使用五个固件方法检查 ADC 转换数据：
  - 原始数据 （不平均）
  - 平均 4 个 ADC 采样
  - 平均 6 个 ADC 采样
  - 平均 8 个 ADC 采样
  - 平均 8 个 ADC 采样，删除 4 个极值

注： *所有测试都使用 $f_{ADC} = 36$ MHz，采样时间 = 3 个 ADC 周期， ADC 分辨率 = 12 位以达到最快的 ADC 转换 （2.4 Msps）。*

3.2 结果

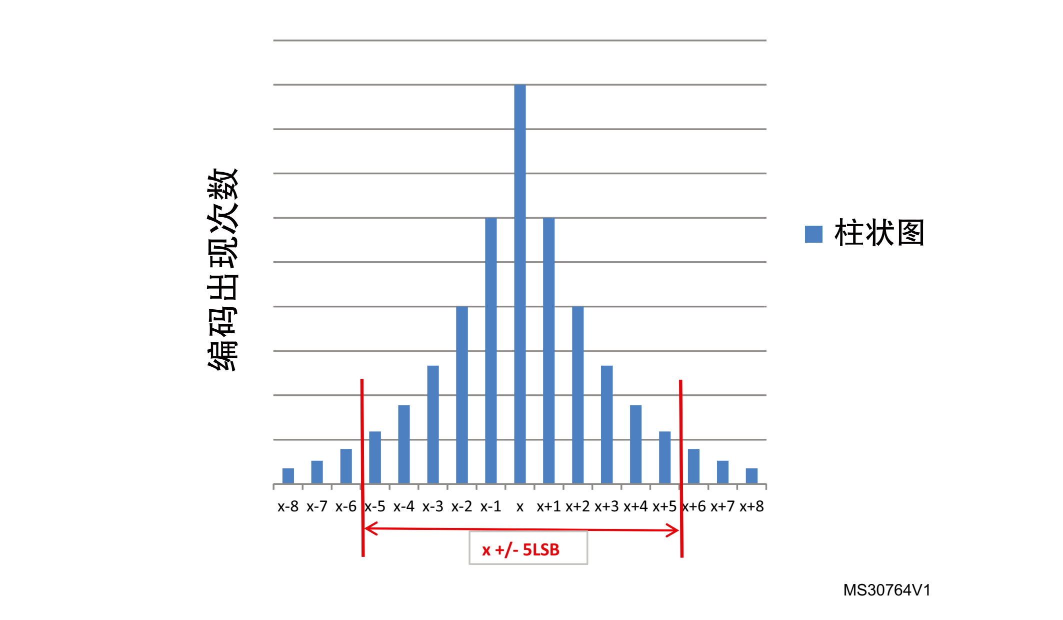
这些测量已基于 ADC 特性执行，被后续研究证实 （约 100 次测试），其中评估了很多参数以分析每个参数的影响。

当评估三个不同 ART 配置对 ADC 精度的影响时，考虑了如下测量：

- ART ON：（数据缓存 + 指令缓存 + 预取） ON
- ART OFF：（数据缓存 + 指令缓存 + 预取） OFF
- （数据缓存 + 指令缓存） ON + 预取 OFF

注： *所有结果都与柱状图中出现最多的值周围 +-5 LSB 范围进行比较评估 （参见图 4）。*

图 4. 柱状图图形表示与编码的离散度



3.2.1 当 ART 为 ON 时的 ADC 测量

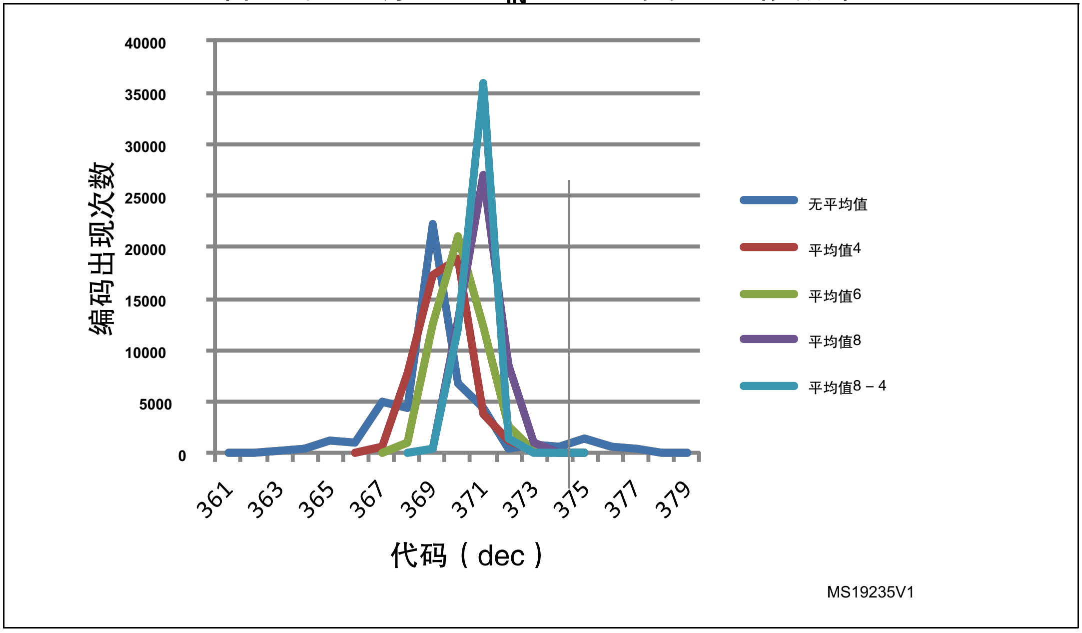
表 2 和*图 5* 显示了当 ART 为 ON 时的 ADC 编码分布。

表 2. 当 ART 为 ON 时的 ADC 编码分布 （单位为 LSB）

V_{IN}	原始数据		4 个值平均		6 个值平均		8 个值平均		8 个值平均，删除 4 个	
	离散度（LSB）	采样超过 +-5 LSB	离散度（LSB）	采样超过 +-5 LSB	离散度（LSB）	采样超过 +-5 LSB	离散度（LSB）	采样超过 +-5 LSB	离散度（LSB）	采样超过 +-5 LSB
0.3 V	19	6.37 %	10	0 %	9	0 %	7	0 %	8	0 %
1.65 V	21	7.90 %	13	0.05 %	10	0 %	10	0 %	9	0 %
3 V	21	21.53 % (1)	15	0.38 %	13	0.13 %	12	0.006 %	12	0.04 %

1. 总共超过 100 次测试中发现的最坏情况

图 5. 当 ART 为 ON， V_{IN} = 0.3 V 时的 ADC 编码分布



3.2.2 当 ART 为 OFF 时的 ADC 测量

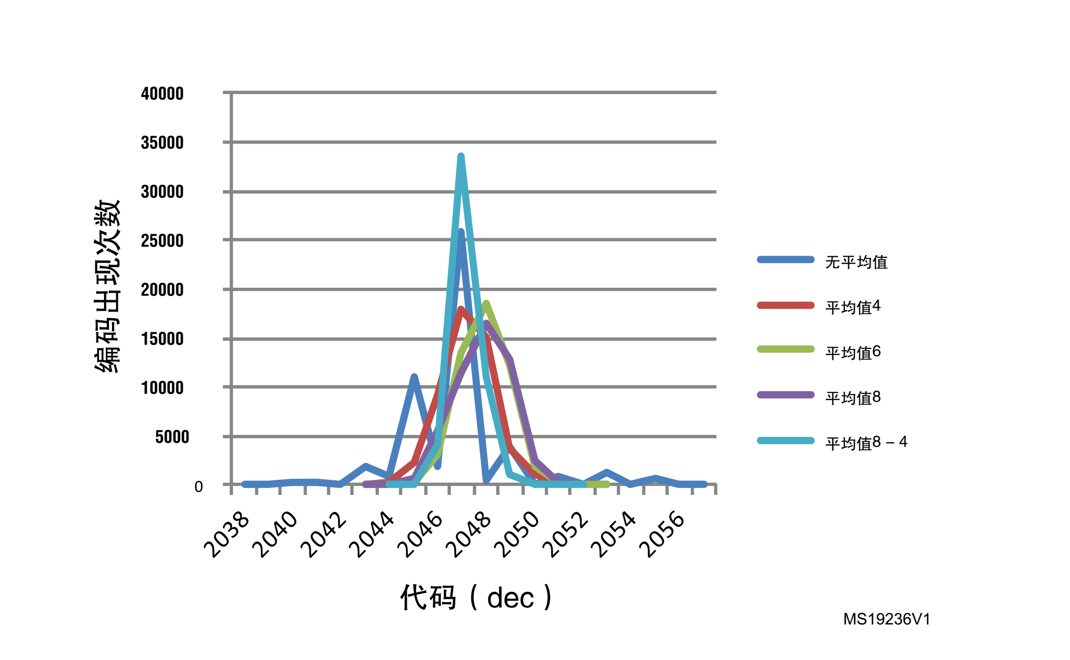
表 3 和 *图 6* 显示了当 ART 为 OFF 时的 ADC 编码分布。

表 3. 当 ART 为 OFF 时的 ADC 编码分布 （单位为 LSB）

V_{IN}	原始数据		4 个值平均		6 个值平均		8 个值平均		8 个值平均，删除 4 个	
	离散度（LSB）	采样超过 +-5 LSB	离散度（LSB）	采样超过 +-5 LSB	离散度（LSB）	采样超过 +-5 LSB	离散度（LSB）	采样超过 +-5 LSB	离散度（LSB）	采样超过 +-5 LSB
0.3 V	18	4.07 %	11	0.004	9	0 %	7	0 %	8	0 %
1.65 V	20	5.99 %	11	0.01 %	11	0.002 %	10	0 %	9	0 %
3 V	24	16.28 % (1)	18	6.92 %	13	0.044 %	11	0.008 %	12	0.028 %

1. 总共超过 100 次测试中发现的最坏情况

图 6. 当 ART 为 OFF， V_{IN} = 1.65 V 时的 ADC 编码分布



3.2.3 当（数据 + 指令）缓存 ON+ 预取 OFF 时的 ADC 测量

表 4 及*图 7* 至*图 9* 显示了当（数据 + 指令）缓存为 ON、预取为 OFF 时的 ADC 编码分布。

表 4. 当（数据 + 指令）缓存 ON + 预取 OFF 时的 ADC 编码分布（单位为 LSB）

V_{IN}	原始数据		4 个值平均		6 个值平均		8 个值平均		8 个值平均，删除 4 个	
	离散度（LSB）	采样超过 +-5 LSB	离散度（LSB）	采样超过 +-5 LSB	离散度（LSB）	采样超过 +-5 LSB	离散度（LSB）	采样超过 +-5 LSB	离散度（LSB）	采样超过 +-5 LSB
0.3 V	16	0.06 %	7	0 %	5	0 %	4	0 %	4	0 %
1.65 V	18	0.064 %	8	0 %	6	0 %	5	0 %	4	0 %
3 V	17	0.068 %	8	0 %	6	0 %	4	0 %	4	0 %

图 7. 当（D+I）缓存 ON + 预取 OFF V_{IN} = 0.3 V 时的 ADC 编码分布

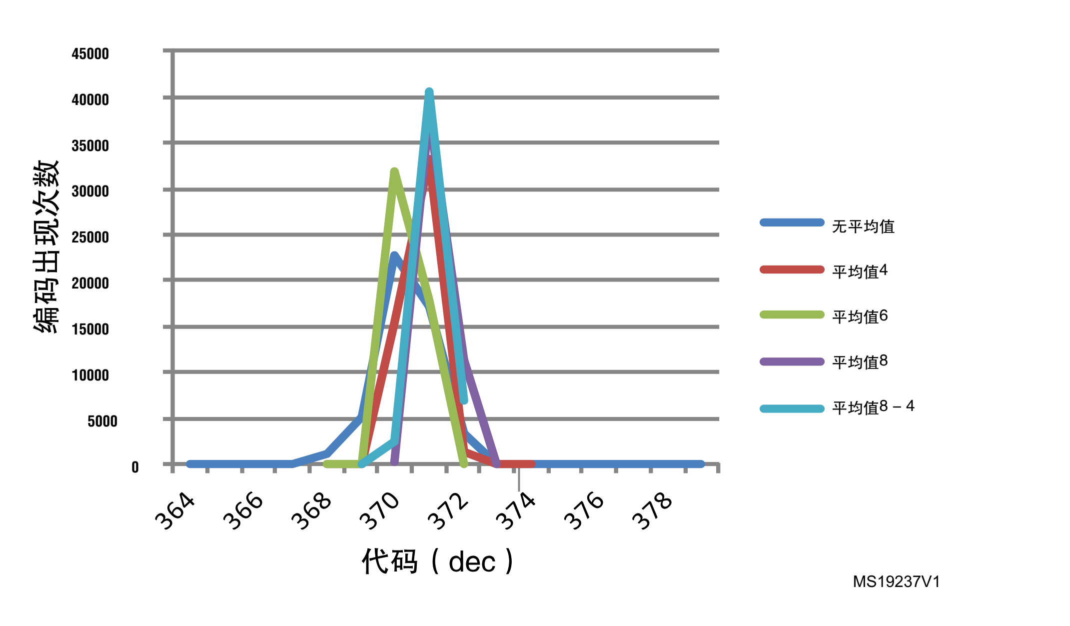


图 8. 当 （D+I）缓存 ON + 预取 OFF V_{IN} = 1.65 V 时的 ADC 编码分布

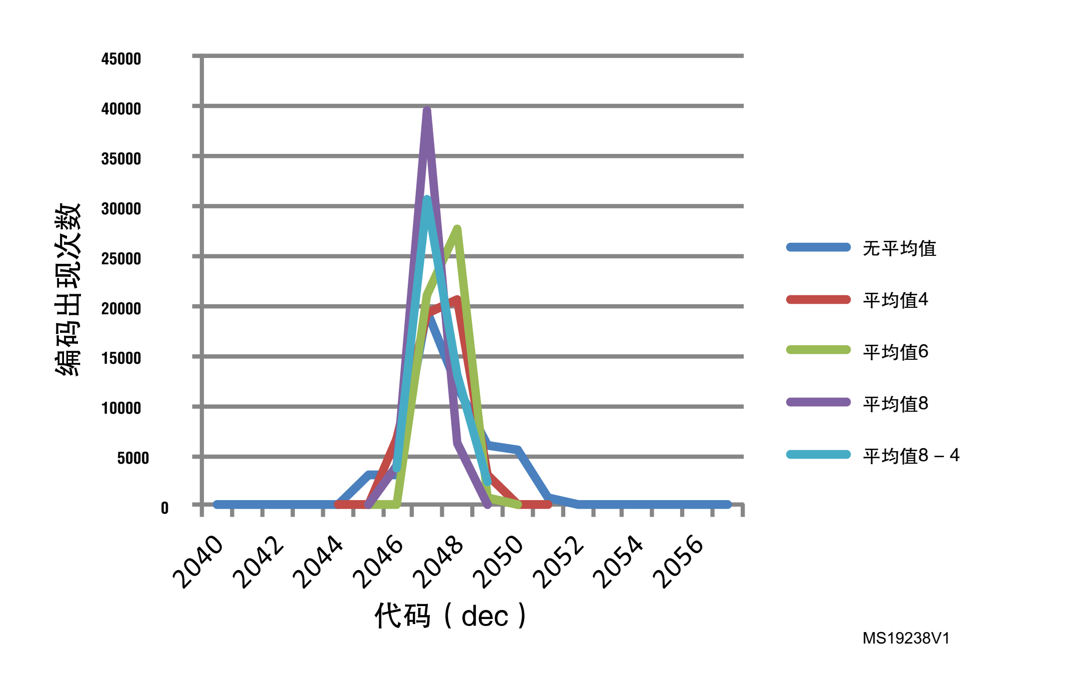
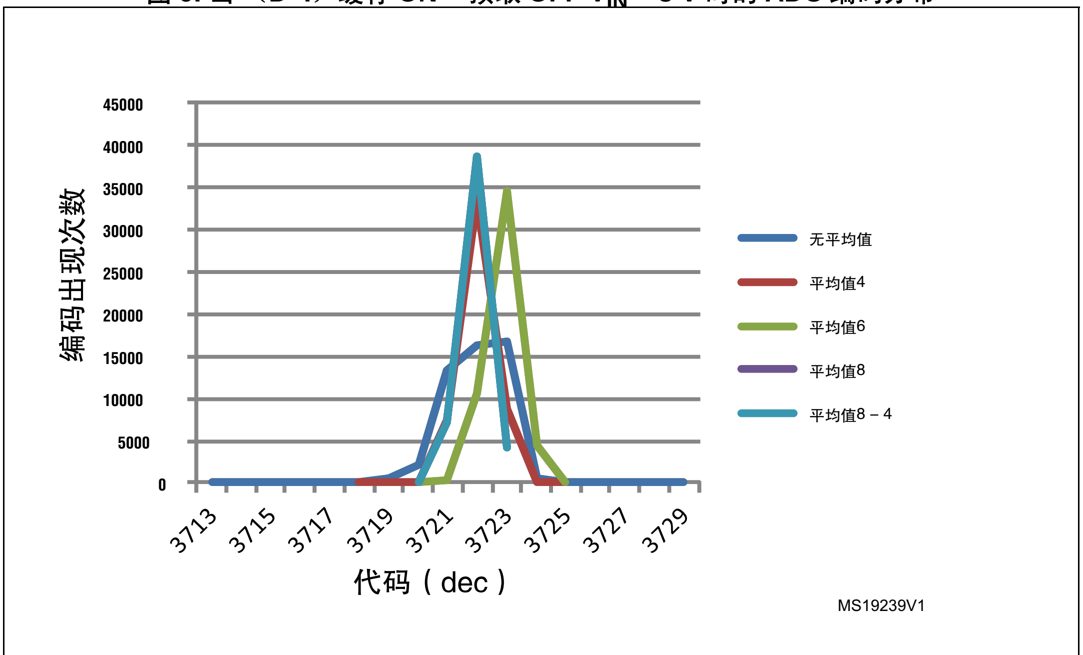


图 9. 当 （D+I）缓存 ON + 预取 OFF V_{IN} = 3 V 时的 ADC 编码分布



3.3 时间考虑

下表给出了每种平均算法的执行时间，以 CPU 周期表示。

注： *这些时间测量的考虑都基于最优的 ART 配置：（数据 + 指令）缓存 ON，预取 OFF，这会导致最佳的 ADC 抗噪声能力。*

在其它 ART 配置中，这些测量可能会导致若干周期的差别。

表 5. 计算平均所需时间

计算时间	4 个值平均 (1)	6 个值平均 (1)	8 个值平均 (1)	8 个值平均，删除 4 个 (2)
CPU 周期(3)	18	26	36	517

1. 计算 = N 个采样相加然后除以 N 所花费的时间。
2. 计算 = 排序 8 个 ADC 采样，删除 4 个两端的值，其它 4 个采样相加然后除以 4 所花费的时间。此结果取决于排序算法 （在此例中，排序算法基于连续置换，没有优化速度）。
3. CPU 频率为 144 MHz，如*第 3.1.1 章节*中所述。

注： *请注意，得到 N 个采样所需的时间 = N * （$t_{SAMPLING}$ + $t_{CONVERSION}$）*

3.4 测量结论

由前面的结果我们可以得出结论，当使用 STM32F2/F405/F415/F407/F417 ADC 时，为得到最精确的转换结果，必须相应地仔细配置 ART Flash 加速器。此配置必须为 （数据 + 指令）缓存 ON 及预取 OFF，以限制 Flash 产生的内部噪声，达到最佳精度。

本节显示了您如何才能在 A/D 转换精度和需要的 A/D 转换速度之间选择最佳的折中。

例如，上面的测量结果表明，设置 ADC 为 12 位分辨率且 ART Flash 加速器配置为 （数据缓存 + 指令缓存） ON、预取 OFF 时，若期望转换值偏差不超过 +/-5 LSB，采用求 4 个转换值平均数的算法就可以得到要求 （此时采样速率最高位 0.6Msps）。

4 STM32F4 ADC 精度选项

本章仅适用于 STM32F42x/F43x 微控制器。

在说明 STM32F42x 和 STM32F43x ADC 评估结果之前，先详细说明这些产品上可增强 ADC 精度的嵌入式芯片可配选项。

4.1 ADC 精度的配置选项

默认情况下，在这些产品上，在微控制器内部， ADC 模块和其它模块之间都激活了噪声过滤技术。这些技术能降低电源噪声、信号干扰和 EMI 引入的噪声。若需此情况下实际 ADC 结果的更详细信息，请参考 4.2.3 节。

为进一步提高 ADC 精度，可使用两个可配置选项。本文全文使用如下命名：

- 默认：默认的 STM32 配置，复位后一直激活。
- 选项 1：第一个可配置选项，可由固件激活。
- 选项 2：第二个可配置选项，可由固件激活。

4.1.1 选项 1

此选项可由固件激活或不激活。

默认情况下，它是不激活的。当由固件激活时，它会不断屏蔽预取机制产生的额外 flash 访问，防止其产生随机噪声影响 ADC 精度。

可置位 PWR_CR 寄存器中的 ADCDC1 （位 13）对其激活。

警告： **仅在如下条件下才能置位此位：**
- 预取必须为 OFF。
- VDD 电压范围为 2.4 V 至 3.6 V。
- 当 SYSCFG_PMC 寄存器中的 ADCxDC2 位置位时，不可置位此位。

若需启用选项 1 的准确固件顺序，请参考*第 C.1 章节：选项 1*。

激活此选项时的 ADC 精度结果将在下面章节详细说明。

4.1.2 选项 2

此选项可由固件激活或不激活。

默认情况下，它是不激活的。当由固件激活时，它会在最后的 ADC 采样周期期间屏蔽内部 flash 噪声。这将提高 ADC 抗内部意外 flash 噪声的能力。

可置位 SYSCFG_PMC 寄存器中的 ADCxDC2 （位 16 - 位 18）对其激活。

警告： **仅在如下条件下才能置位这些位：**
- ADC 时钟最低为 30MHz。
- 若需要同时使用多个 ADC 且各个 ADC 不同时启动或采样时间不

同，则必须仅能选择一个 ADCxDC2 位。
- ADC 分辨率应为 12 位。
- 当 PWR_CR 寄存器中的 ADCDC1 位置位时，不可置位此位。

在此选项中，每个 ADC 都在 SYSCFG_PMC 寄存器中有一位。这就能更灵活地为每个 ADC 单独启用或禁用此选项。

如前面的警告中所述，在某些情况下这三位不应同时激活。

下面是 ADC 模式与 ADCxDC2 位的同时使用：

表 6. ADCxDC2 的使用与 ADC 模式

	单独模式 (1)	多重 ADC 模式					
		注入同时	规则同时	交织	交替触发	注入同时 + 规则同时	规则同时 + 交替触发
一个 ADCxDC2	是	是	是	是	是	是	是
一些或所有 ADCxDC2	否	(2)	(2)	否	(3)	(2)	(2)

1. 当两个或三个 ADC 单独激活，每个都配置于单独模式时，仅应使用一个 ADCxDC2 位。
2. 在所有 ADC 都使用相同采样时间的情况下，可使用一些或所有 ADCxDC2 位。
3. 若转换触发的开始受控，使得每个 ADC 采样结束之间的时间间隔大于等于 15 个 ADC 周期，则可使用一些或所有 ADCxDC2 位。

若需启用选项 2 的准确固件顺序，请参考*第 C.2 章节：选项 2*。

激活此选项时的 ADC 精度将在下面章节详细说明。

4.2 实际测量

在本节中做了实际 ADC 测量，以证明这些 ADC 精度选项的影响。

4.2.1 硬件设置

用于 ADC 测试的硬件环境与*第 3.1.1 章节：硬件设置*中所述相同，使用 STM32F43x。

4.2.2 通用固件设置

- ADC1 通道 1 用于单转换模式，
- 采样时间固定为 3 个 ADC 时钟周期，
- 对每个固定模拟输入电压采集 50000 次，

所有测试都使用 FADC = 36 MHz，采样时间 = 3 个 ADC 周期， ADC 分辨率 = 12 位以达到最快的 ADC 转换 （2.4 MSPS）。

4.2.3 结果

为评估这些产品上内置的 ADC 精度选项的影响，进行了如下测量。使用的三种不同 ART 配置为：

- ART ON：（数据缓存 + 指令缓存 + 预取） ON，
- ART OFF：（数据缓存 + 指令缓存 + 预取） OFF，
- （数据缓存 + 指令缓存） ON + 预取 OFF

1. 当 ART 为 ON 时的 ADC 测量

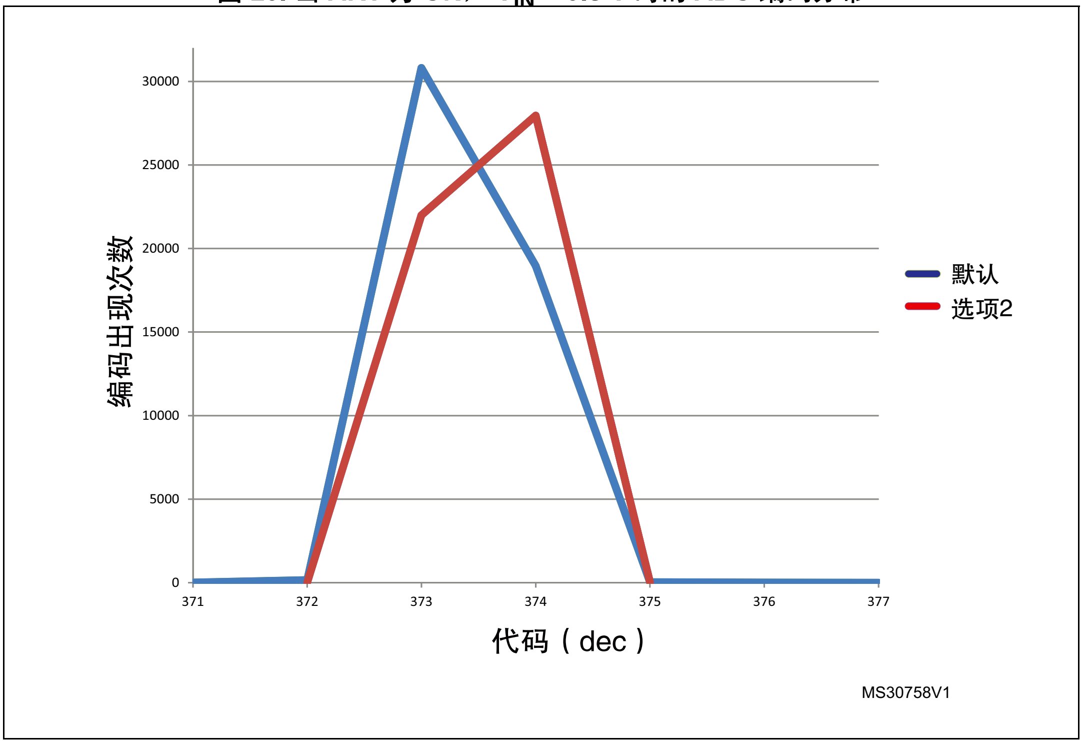
下面的图表显示了当 ART 为 ON 时的 ADC 编码分布，分两种不同情况：默认的 STM32 配置和选项 2 激活时的配置：

表 7. 当 ART 为 ON 时的 ADC 编码分布

V_{IN}	默认		选项 1		选项 2	
	离散度（LSB）	超过 +-5 LSB 的采样	离散度（LSB）	超过 +-5 LSB 的采样	离散度（LSB）	超过 +-5 LSB 的采样
0.3 V	7	0%	(1)		4	0%
1.65 V	9	0%			6	0%
3 V	7	0%			6	0%

1. 在此情况下，因为预取为 ON，所以选项 1 不可用。

图 10. 当 ART 为 ON， V_{IN} = 0.3 V 时的 ADC 编码分布



在相同条件下，当 ART 为 ON 时，对 STM32F2xx 或 STM32F405/F415/F407/F417 使用默认的 STM32 配置会产生的最大离散度为 21 LSB （请参考*表 2*），而在此情况下，最大离散度为 9 LSB。

这证实了这些产品上内置的默认 STM32 配置可达到的离散编码约为 STM32F2xx/F405/F415/F407/F417 的 50 %，约为选项 2 激活时的 75 %。

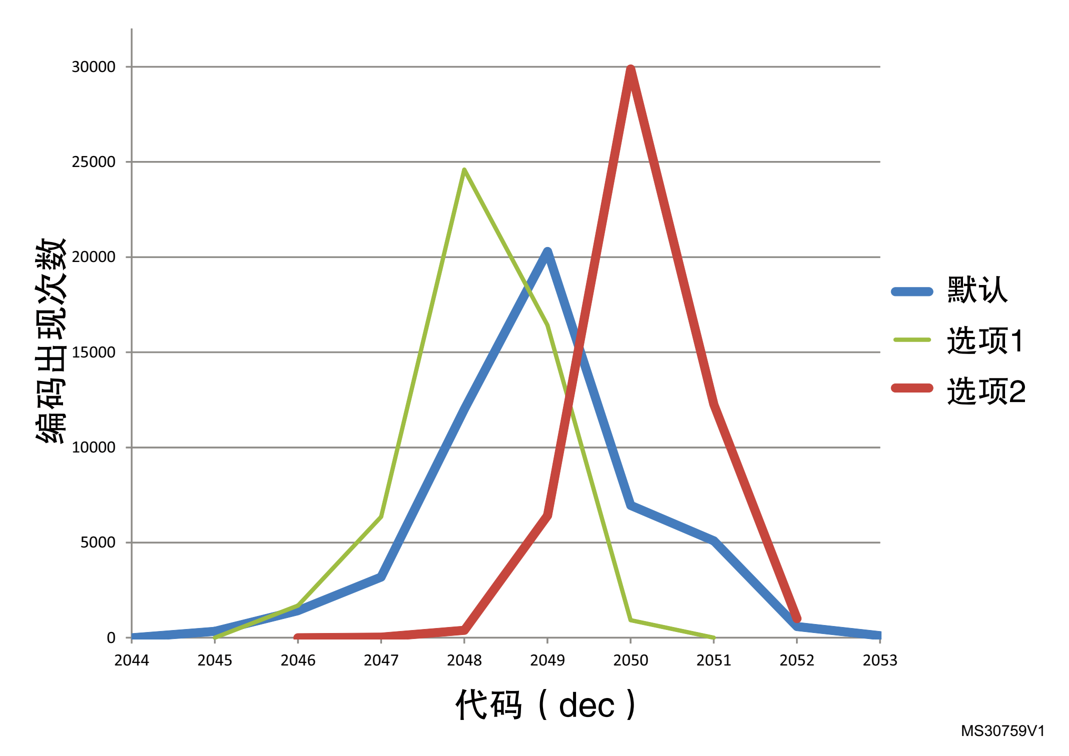
2. 当 ART 为 OFF 时的 ADC 测量

下面的图表显示了当 ART 为 OFF 时的 ADC 编码分布，分三种不同情况：默认的 STM32 配置、选项 1 和选项 2 激活时的配置。

Table 8. 当 ART 为 OFF 时的 ADC 编码分布

V_{IN}	默认		选项 1		选项 2	
	离散度（LSB）	超过 +-5 LSB 的采样	离散度（LSB）	超过 +5 LSB 的采样	离散度（LSB）	超过 +-5 LSB 的采样
0.3 V	9	0%	5	0%	5	0%
1.65 V	10	0%	7	0%	7	0%
3 V	8	0%	6	0%	6	0%

图 11. 当 ART 为 OFF， V_{IN} = 1.65 V 时的 ADC 编码分布



在相同条件下，当 ART 为 OFF 时，对 STM32F2xx 或 STM32F405/F415/F407/F417 使用默认的 STM32 配置会产生的最大离散度为 24 LSB （请参考*表 3*），而在此情况下，最大离散度为 10 LSB。

这证实了这些产品上内置的默认 STM32 配置可达到的离散编码超过 STM32F2xx/F405/F415/F407/F417 的 50 %，约为二选项之一激活时的 75 %。

3. 当 （数据 + 指令）缓存 ON+ 预取 OFF 时的 ADC 测量

下面的图表显示了当 （数据 + 指令）缓存 ON、预取 OFF 时的 ADC 编码分布，分三种不同情况：默认的 STM32 配置、选项 1 和选项 2 激活时的配置。

表 9. 当 （数据 + 指令）缓存 ON + 预取 OFF 时的 ADC 编码分布

V_{IN}	默认		选项 1		选项 2	
	离散度（LSB）	超过 +-5 LSB 的采样	离散度（LSB）	超过 +-5 LSB 的采样	离散度（LSB）	超过 +-5 LSB 的采样
0.3 V	7	0%	4	0%	4	0%
1.65 V	8	0%	6	0%	6	0%
3 V	7	0%	6	0%	7	0%

在相同条件下，当 （数据 + 指令）缓存 ON、预取 OFF 时，对 STM32F2xx 或 STM32F405/F415/F407/F417 使用默认的 STM32 配置会产生的最大离散度为 18 LSB （请参考*表 4*），而在此情况下，最大离散度为 8 LSB。

这证实了这些产品上内置的默认 STM32 配置可达到的离散编码约为 STM32F2xx/F405/F415/F407/F417 的 50 %，约为二选项之一激活时的 75 %。

图 12. 当 （D+I）缓存 ON + 预取 OFF V_{IN} = 0.3 V 时的 ADC 编码分布

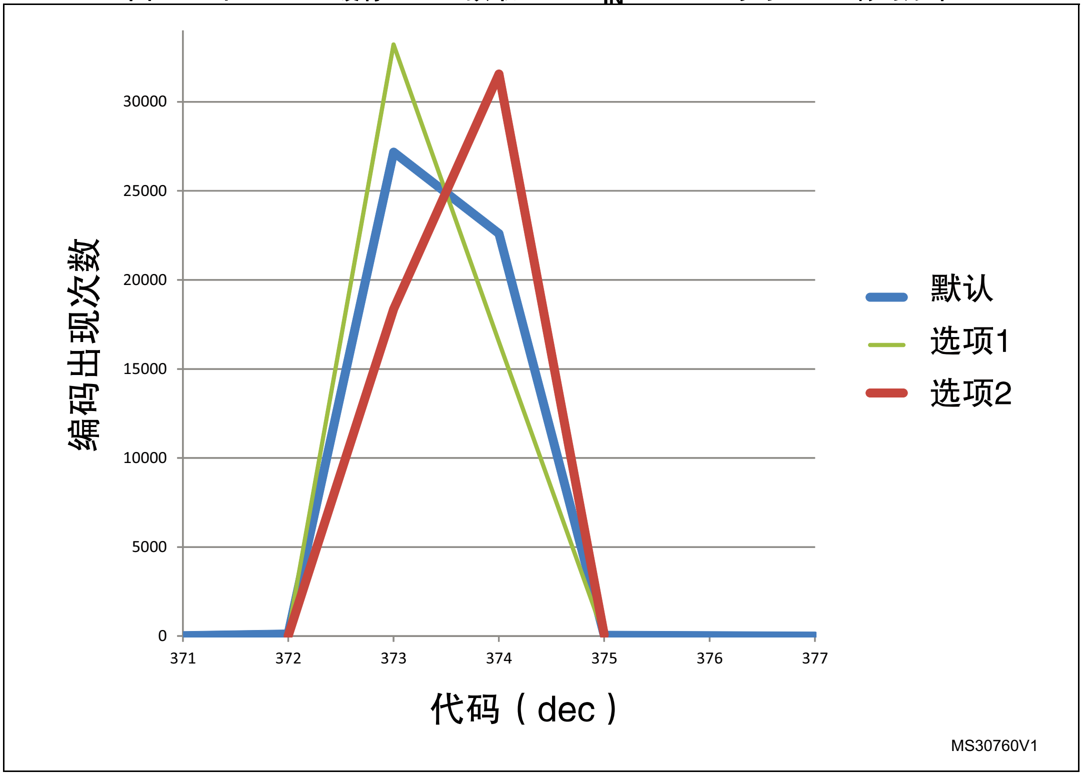


图 13. 当 （D+I）缓存 ON + 预取 OFF V_{IN} = 1.65V 时的 ADC 编码分布

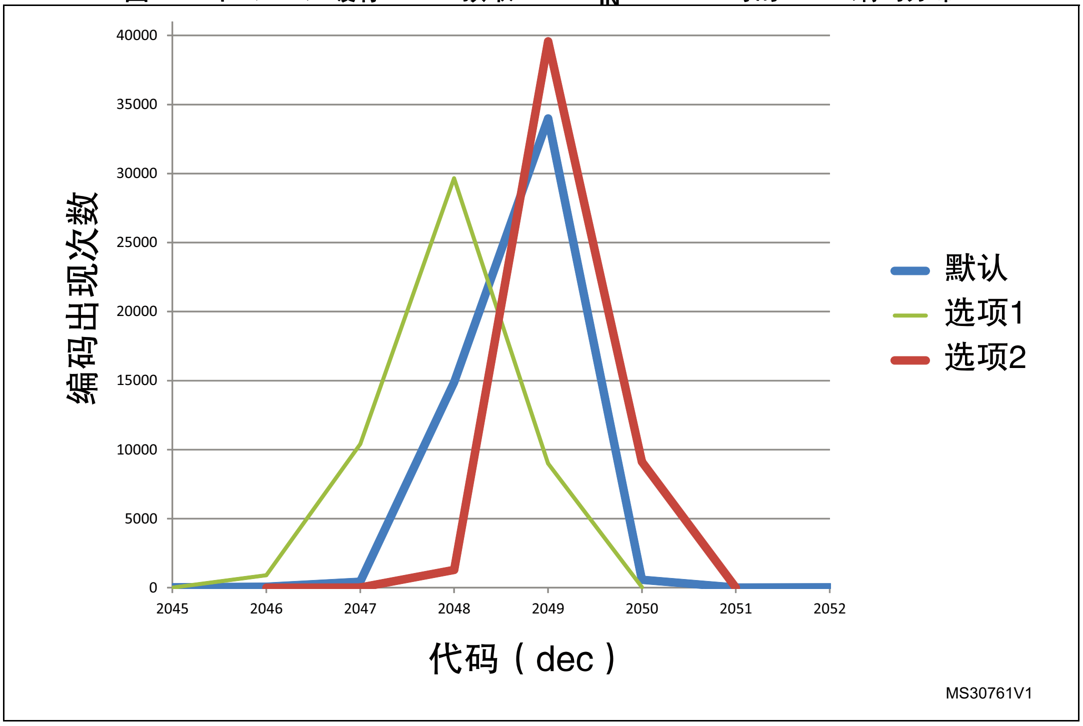
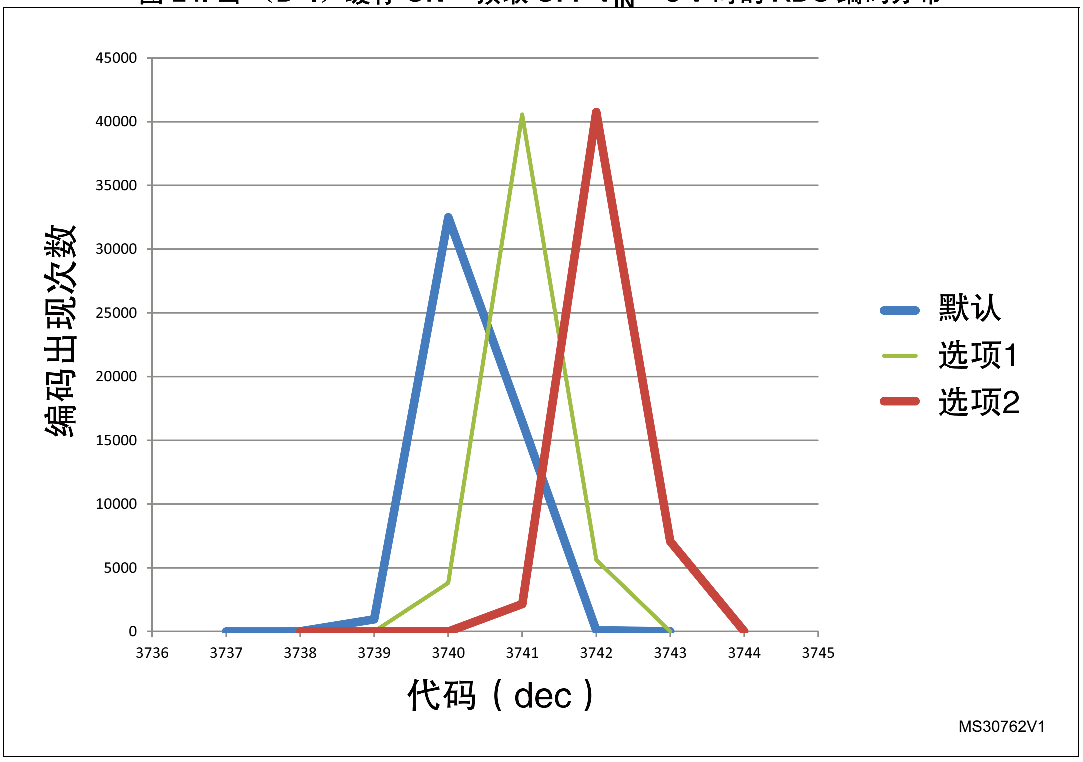


图 14. 当 （D+I）缓存 ON + 预取 OFF V_{IN} = 3 V 时的 ADC 编码分布



4.3 测量结论

基于这些测量及比较 STM32F2xx 和 STM32F405/F415/F407/ F417 的测量，我们可得出结论：这些产品上内置的 ADC 精度选项可将 ADC 代码离散度改善高达 75 %。

对于适用的产品，您可根据应用对 ADC 精度的需要，选择使用默认配置工作或启用二选项之一。

基于这些典型测量可知，这些产品若需达到最佳的 ADC 精度，您需要了解：

- 对于默认配置：预取 OFF + 数据缓存 ON + 指令缓存 ON 可给出最大为 8 的编码离散度。
- 当激活选项 1 时：预取 OFF + 数据缓存 ON + 指令缓存 ON 可给出最大为 6 的编码离散度。
- 当激活选项 2 时：预取 ON + 数据缓存 ON + 指令缓存 ON 可给出最大为 6 的编码离散度。

然而，如前面所述，仅在特定条件下才能应用选项 1 和选项 2 （请参考*第 4.1 章节*），否则会导致 MCU 故障。

附录 A 平均 N 个 ADC 采样：源代码

```
/**
  * @简介：得到 N 个 ADC 采样的均值
  * @输入参数：要做平均的 ADC 采样数
  * @返回值：均值
  */
uint16_t ADC_GetSampleAvgN(uint8_t N)
{
  uint32_t avg_sample =0x00;
  uint16_t adc_sample[8]={0,0,0,0,0,0,0,0};
  uint8_t index=0x00;

  /* 得到 N 个 ADC 采样 */
  for (index=0x00; index<N; index++)
  {
      /* ADC 开始转换 */
   ADC_SoftwareStartConv(ADC1);
    /* 等待转换结束 */
    while(ADC_GetFlagStatus(ADC1,ADC_FLAG_EOC) == RESET);
    /* 储存 ADC 采样 */
    adc_sample[index] = ADC_GetConversionValue(ADC1);
  }

  /* 累加 N 个 ADC 采样 */
  for (index=0; index<N; index++)
  {
   avg_sample += adc_sample[index];
  }

  /* 计算 N 个 ADC 采样的均值 */
  avg_sample /= N;

    /* 返回均值 */
  return avg_sample;
 }
```

附录 B 平均 N-X 个 ADC 采样：源代码

```
/**
  * @简介：得到 N-X 个 ADC 采样的均值
  * @输入参数：要做平均的 ADC 采样数
  * @输入参数：要做平均的 ADC 采样数
  * @返回值：均值
  */
uint16_t ADC_GetSampleAvgNDeleteX(uint8_t N , uint8_t X)
{
  uint32_t avg_sample =0x00;
  uint16_t adc_sample[8]={0,0,0,0,0,0,0,0};
  uint8_t index=0x00;


  for (index=0x00; index<N; index++)
  {
      /* ADC 开始转换 */
   ADC_SoftwareStartConv(ADC1);
    /* 等待转换结束 */
    while(ADC_GetFlagStatus(ADC1,ADC_FLAG_EOC) == RESET);
    /* 储存 ADC 采样 */
    adc_sample[index] = ADC_GetConversionValue(ADC1);
  }

  /* 排序 N-X 个 ADC 采样 */
    Sort_tab(adc_sample,N);

  /* 累加 N 个 ADC 采样 */
    for (index=X/2; index<N-X/2; index++)
  {
   avg_sample += adc_sample[index];
  }

  /* 计算 N-X 个 ADC 采样的均值 */
  avg_sample /= N-X;

    /* 返回均值 */
  return avg_sample;
}



/**
```

```
  * @简介：排序 N 个 ADC 采样
  * @输入参数：要做排序的 ADC 采样
  * @输入参数：要做排序的 ADC 采样数
  * @返回值：无
  */
void Sort_tab(uint16_t tab[], uint8_t lenght)
{
   uint8_t l=0x00, exchange =0x01;
   uint16_t  tmp=0x00;

 /* 排序 tab */
    while(exchange==1)
    {
    exchange=0;
    for(l=0; l<lenght-1; l++)
    {
      if( tab[l] > tab[l+1] )
      {
      tmp = tab[l];
      tab[l] = tab[l+1];
      tab[l+1] = tmp;
      exchange=1;
      }
    }
    }
}
```

附录 C 激活选项 1 和选项 2 的固件顺序

C.1 选项 1

```
/**
* @简介：启用或禁用 ADC Option_1 配置。
* @输入参数 NewState：ADCDC1 位的新状态。
*            此参数可为：ENABLE 或 DISABLE。
* @返回值：无
  */
void SET_ADCOption1 (FunctionalState NewState)
{

  /* ENABLE PWR 时钟 */
  RCC_APB1PeriphClockCmd(RCC_APB1Periph_PWR,ENABLE);

  if (NewState != DISABLE)
  {
    /* 设置 ADCDC1 位 */
    PWR->CR |= ((uint32_t)PWR_CR_ADCDC1);
  }
  else
  {
    /* 复位 ADCDC1 位 */
  PWR->CR &= (uint32_t)(~PWR_CR_ADCDC1);
  }
}
```

C.2 选项 2

```
/**
  * @简介：启用或禁用 ADCx Option_2 配置。
* @输入参数 ADCxDC2：要使用的 ADCxDC2 位。
*           此参数可取如下值：
  *             @arg SYSCFG_PMC_ADCxDC2：所有 ADCxDC2 位
  *             @arg SYSCFG_PMC_ADC1DC2：ADC1DC2 位
  *             @arg SYSCFG_PMC_ADC2DC2：ADC2DC2 位
  *             @arg SYSCFG_PMC_ADC3DC2：ADC3DC2 位
  * @输入参数 NewState：ADCxDC2 位的新状态。
*           此参数可为：ENABLE 或 DISABLE。
* @返回值：无
  */
void SET_ADCOption2 (uint32_t ADCxDC2, FunctionalState NewState)
{

  /* 启用 SYSCFG 时钟 */
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_SYSCFG,ENABLE);

  if (NewState != DISABLE)
  {
    /* 置位 ADCxDC2 */
    SYSCFG->PMC |= (uint32_t)ADCxDC2;
  }
  else
  {
    /* 复位 ADCxDC2 */
    SYSCFG->PMC &=(uint32_t)(~ADCxDC2);
  }
}
```

修订历史

表 10. 文档修订历史

日期	修订	变更
2012 年 3 月 19 日	1	初始版本。
2012 年 3 月 21 日	2	在第 9 页插入了缺少的文本。
2012 年 8 月 10 日	3	在*前言*中，将 "STM32F20x、 STM32F21x、 STM32F40x、 STM32F41x 微控制器 " 替换为："STM32F20x / STM32F21x （revZ、 revY、 revX）和 STM32F40x/STM32F41x （revZ）微控制器 "。 增加了*附录 A: 平均 N 个 ADC 采样：源代码*和*附录 B: 平均 N-X 个 ADC 采样：源代码*，以及*第 2.1.1 章节*和*第 2.1.2 章节*中的相关注释。
2013 年 2 月 25 日	4	应用于 STM32F2xx （STM32F20x、 STM32F21x）和 STM32F4xx （STM32F40x、 STM32F41x、 STM32F42x、 STM32F43x）微控制器。 *第 3 章节*现在专用于 STM32F2xx、 STM32F405、 STM32F415、 STM32F407 和 STM32F417 产品，新的*第 4 章节*专用于 STM32F42x/F43x 产品。 增加*附录 C: 激活选项 1 和选项 2 的固件顺序*。
2013 年 7 月 02 日	5	更改了*第 4.1.1 章节*中告警的 VDD 范围

请仔细阅读：

中文翻译仅为方便阅读之目的。该翻译也许不是对本文档最新版本的翻译，如有任何不同，以最新版本的英文原版文档为准。

本文档中信息的提供仅与 ST 产品有关。意法半导体公司及其子公司 （“ST”）保留随时对本文档及本文所述产品与服务进行变更、更正、修改或改进的权利，恕不另行通知。

所有 ST 产品均根据 ST 的销售条款出售。

买方自行负责对本文所述 ST 产品和服务的选择和使用，ST 概不承担与选择或使用本文所述 ST 产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为 ST 授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在 ST 的销售条款中另有说明，否则，ST 对 ST 产品的使用和 / 或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途 （及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

意法半导体的产品不得应用于武器。此外，意法半导体产品也不是为下列用途而设计并不得应用于下列用途：（A）对安全性有特别要求的应用，例如，生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）汽车应用或汽车环境，且 / 或（D）航天应用或航天环境。如果意法半导体产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向意法半导体发出了书面通知，采购商仍将独自承担因此而导致的任何风险，意法半导体的产品设计规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML 或 JAN 正式认证产品适用于航天应用。

经销的 ST 产品如有不同于本文档中提出的声明和 / 或技术特点的规定，将立即导致 ST 针对本文所述 ST 产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大 ST 的任何责任。

ST 和 ST 徽标是 ST 在各个国家或地区的商标或注册商标。

本文档中的信息取代之前提供的所有信息。

ST 徽标是意法半导体公司的注册商标。其他所有名称是其各自所有者的财产。

© 2015 STMicroelectronics 保留所有权利

意法半导体集团公司

澳大利亚 - 比利时 - 巴西 - 加拿大 - 中国 - 捷克共和国 - 芬兰 - 法国 - 德国 - 中国香港 - 印度 - 以色列 - 意大利 - 日本 - 马来西亚 - 马耳他 - 摩洛哥 - 菲律宾 - 新加坡 - 西班牙 - 瑞典 - 瑞士 - 英国 - 美国

www.st.com