binho

应用笔记

使用 Binho Supernova 和 Pulsar 设置 I2C SCL 占空比和精确的总线频率

AN0015

文档信息

关键词	AN0015, Supernova, Pulsar, I2C, SCL, 占空比, tHIGH, tLOW, 时序裕量, 总线频率, pyCosmicSDK
摘要	一些 I2C 目标器件以及一些认证计划关心的不仅是每秒有多少个 SCL 周期，还包括每个周期中高电平与低电平的分配。 主机适配器固件 4.6.0 和 pyCosmicSDK 1.6.0 使 Binho Supernova 或 Pulsar 能够直接设置这一分配， 高电平占比为 20 至 80%，步进为 10；设置后总线频率也会变得精确，而不是取近似值。本应用笔记介绍该调用， 提供一个应用比例并打印适配器报告值的脚本，并给出在 100 kHz 和 400 kHz 下用逻辑分析仪在总线上测得的比例。

Binho Inc.
42 Dore Street, Unit A, San Francisco, CA 94103

修订版 1.0
6 NOVEMBER 2026

1 简介

I2C 控制器为每个 SCL 周期选择高电平时间和低电平时间。I2C 规范对两者都规定了最小值（tHIGH 和 tLOW），大多数控制器会自行选择分配方式。这本身没有问题，直到某个目标器件对此敏感，或者某个测试计划要求特定的分配：例如用 30:70 的时钟检查目标器件在高电平阶段较短时能否正确采样，或用 40:60 的时钟匹配目标器件将来所在系统的时序。

此前，Binho 适配器自行选择这一分配。在 100 kHz 时默认约 46% 为高电平；在 400 kHz 时约 40%。无法请求其他分配。主机适配器固件 4.6.0 和 pyCosmicSDK 1.6.0 新增了一个设置它的调用。

1.1 范围

本文档涵盖：

- `I2cController.set_duty_cycle` 的作用，以及它不改变的内容
- 固件接受的比例，以及对拒绝的比例返回什么
- 为什么设置比例也会使频率精确
- 一个应用比例、产生可供测量的通信并打印适配器报告值的脚本
- 在 100 kHz 和 400 kHz 下于总线上测得的分配

本文档不涉及 20 至 80% 以外的比例、Supernova I3C 端口上的占空比控制，也不涉及 SPI。SPI 时钟时序是另一项固件更改。

1.2 适用器件

表 1. 适用器件

适配器	标识	依据
Binho Supernova	硬件版本 C，固件 4.6.0 预发布版本 `80cbb484`	已测量，本文档
Binho Pulsar	硬件版本 C，固件 4.6.0 预发布版本 `ea77618b`	在固件验证期间测量，与 Supernova 相差在 1% 以内；本文档未重新测量
任一适配器	固件 4.5.1 或更早版本	以 `FW_UNSUPPORTED_COMMAND` 拒绝该调用

> **说明：** 本版本基于预发布版本编写。固件将其版本报告为带构建后缀的 `4.5.1`，因为当时版本字符串尚未改为 4.6.0。该调用及其行为与 4.6.0 正式版相同。

1.3 相关文档

表 2. 相关文档

文档	标题
UM10204	I2C-bus specification and user manual, Rev. 7, NXP Semiconductors
AN0014	在 Binho Supernova 上为 NVMe-MI 实现单端口 I2C/SMBus/I3C 操作

2 系统概述

`set_duty_cycle(high_pct)` 设置每个 SCL 周期中时钟保持高电平的占比。固件的实现方式决定了以下三点：

- **频率保持不变。** 总线保持传给 `bring_up` 或 `configure` 的频率。只有分配会改变。
- **频率变得精确。** 未设置比例时，适配器选择其可实现的最接近的分频值，并且可能向上取整超过请求值：550 kHz 实际运行在 600.6 kHz。设置比例后，固件将 I2C 外设切换到更精细的时钟，因此频率不会高于请求值，且最多低 0.29%。同样的 550 kHz 实际运行在 549.8 kHz。
- **比例会保持。** 之后的 `configure` 会保留该比例，并在新频率下应用。`set_duty_cycle(0)` 或重新执行 `initialize` 会恢复默认时序。

该调用返回当前生效的设置，类型为 `I2cDutyCycle`，含两个字段：

表 3. I2cDutyCycle 的字段

字段	含义
`frequency_hz`	设置比例时，为总线上的频率。`high_pct` 为 0 时，为请求的频率，默认时序可能对其取近似
`high_pct`	生效的比例，单位为百分比；0 表示默认时序

固件不会将比例限制在 I2C 规范范围内。100 kHz 下 30% 的时钟其 tHIGH 为 3.0 us，低于 Standard-mode 的 4.0 us 最小值。这是有意为之：裕量测试正需要超出规范的时序。在产品中使用某个比例之前，请对照规范检查。

3 硬件设置

3.1 所需设备

表 4. 所需设备

项目
Binho Supernova 或 Pulsar，固件 4.6.0 或更高版本
USB-C 线缆
一个 I2C 目标器件；如果只测量 SCL 则无需
SCL 上的示波器或逻辑分析仪，用于测量结果

3.2 连接

表 5. 与 I2C 端口的连接

适配器	目标器件或仪器	是否必需
SCL	目标器件 SCL、仪器探头	必需
SDA	目标器件 SDA	有目标器件时必需
GND	目标器件和仪器的地	必需
3.3 V	目标器件电源	可选

3.3 总线注意事项

适配器从看到 SCL 为高电平的时刻开始计量 tHIGH。因此缓慢的上升沿会延长周期：使用 2.2 kOhm 上拉电阻时， 1 MHz 的时钟实际约为 905 kHz。在 1 MHz 下请使用 470 Ohm 或更强的上拉电阻。在 100 kHz 和 400 kHz 下，1 kOhm 至 2.2 kOhm 即可。

4 软件设置

安装 pyCosmicSDK（最新版本）：

```
pip install --upgrade pycosmicsdk
```

脚本 `i2c_duty_cycle.py` 位于资源压缩包中。它不需要其他任何依赖。

5 占空比调用

单独列出的调用如下：

```
import pycosmicsdk as p

dev = p.Device.open(model=p.DeviceModel.SUPERNOVA)
bus = dev.i2c()
bus.set_voltage(voltage_mv=3300)
bus.bring_up(frequency_hz=400_000, pull_up=p.I2cPullUp.KOHM_1_0)
print(bus.set_duty_cycle(high_pct=40))
# I2cDutyCycle(frequency_hz=400000, high_pct=40)
```

表 6. set_duty_cycle 接受的值

`high_pct`	结果
0	默认时序，与固件 4.5.1 相同
20, 30, 40, 50, 60, 70, 80	每个周期中该占比为高电平；频率精确
其他任何值	`FirmwareError`，状态 `FW_INVALID_PARAMETER`；保持当前生效的比例

在固件 4.5.1 及更早版本上，该调用引发 `FirmwareError`，状态为 `FW_UNSUPPORTED_COMMAND`，总线继续正常工作。

6 操作步骤

1. 连接适配器以及（如使用）SCL 上的仪器。
2. 询问适配器在所关注的频率下对每个比例报告的值。每个频率通过 `configure` 应用，然后依次应用每个比例，最后恢复默认时序：

```
$ python i2c_duty_cycle.py --model supernova --pull-up KOHM_1_0 table --khz 100 400
1000
SUPERNOVA FW-4.5.1-80cbb484  I2C 3300 mV  pull-up KOHM_1_0
request   100 kHz  ->    100000 Hz  default
request   100 kHz  ->    100000 Hz  20 % high
request   100 kHz  ->    100000 Hz  30 % high
request   100 kHz  ->    100000 Hz  40 % high
request   100 kHz  ->    100000 Hz  50 % high
request   100 kHz  ->    100000 Hz  60 % high
request   100 kHz  ->    100000 Hz  70 % high
request   100 kHz  ->    100000 Hz  80 % high
request   400 kHz  ->    400000 Hz  default
request   400 kHz  ->    400000 Hz  20 % high
request   400 kHz  ->    400000 Hz  30 % high
request   400 kHz  ->    400000 Hz  40 % high
request   400 kHz  ->    400000 Hz  50 % high
request   400 kHz  ->    400000 Hz  60 % high
request   400 kHz  ->    400000 Hz  70 % high
request   400 kHz  ->    400000 Hz  80 % high
request  1000 kHz  ->   1000000 Hz  default
request  1000 kHz  ->    999024 Hz  20 % high
request  1000 kHz  ->    999024 Hz  30 % high
request  1000 kHz  ->    999024 Hz  40 % high
request  1000 kHz  ->    999024 Hz  50 % high
request  1000 kHz  ->    999024 Hz  60 % high
request  1000 kHz  ->    999024 Hz  70 % high
request  1000 kHz  ->    999024 Hz  80 % high
```

在 1 MHz 下，精确频率为 999024 Hz，比请求值低 0.1%。`default` 行报告的是请求值而非总线实际值；第 8.2 节给出了两者的差异。

3. 应用一个比例。使用 `--traffic` 时，脚本会让 SCL 持续运行该秒数，以便仪器进行采集：

```
$ python i2c_duty_cycle.py --model supernova --pull-up KOHM_1_0 set --khz 100 --high
30
SUPERNOVA FW-4.5.1-80cbb484  I2C 3300 mV  pull-up KOHM_1_0
request   100 kHz  ->    100000 Hz  30 % high
```

4. 不在接受范围内的比例会被拒绝，当前生效的比例保持不变：

```
$ python i2c_duty_cycle.py --model supernova --pull-up KOHM_1_0 set --khz 100 --high
35
error: i2c_set_duty_cycle: FW_INVALID_PARAMETER: At least one of the parameters of
the command is not valid.
SUPERNOVA FW-4.5.1-80cbb484  I2C 3300 mV  pull-up KOHM_1_0
```

5. 在默认时序会取近似的频率下，将某个比例与默认时序进行比较：

```
$ python i2c_duty_cycle.py --model supernova --pull-up KOHM_1_0 set --khz 550 --high
50
SUPERNOVA FW-4.5.1-80cbb484  I2C 3300 mV  pull-up KOHM_1_0
request   550 kHz  ->    549798 Hz  50 % high
$ python i2c_duty_cycle.py --model supernova --pull-up KOHM_1_0 set --khz 550 --high
0
SUPERNOVA FW-4.5.1-80cbb484  I2C 3300 mV  pull-up KOHM_1_0
request   550 kHz  ->    550000 Hz  default
```

第二行报告 550000 Hz，因为它回显的是请求值。在总线上，默认时序实际运行在 600.6 kHz（第 8.2 节）。

7 参考

表 7. i2c_duty_cycle.py 选项

选项	默认值	含义
`--model`	任意	`supernova` 或 `pulsar`，在两者都连接时使用
`--serial`	任意	适配器序列号
`--voltage-mv`	3300	总线电压
`--pull-up`	`KOHM_2_2`	板载上拉电阻，任意 `I2cPullUp` 名称

表 8. i2c_duty_cycle.py 命令

命令	选项	操作
`set`	`--khz`、`--high`、`--traffic`、`--address`	应用一个比例；使用 `--traffic` 时，在该秒数内从 `--address` 读取
`table`	`--khz`（一个或多个）	打印 0 以及 20 至 80% 各比例下报告的频率和比例，然后恢复默认时序

通信默认从地址 0x50 读取。若该地址没有目标器件，每次读取为一个地址字节加一个 NACK，仍会产生九个 SCL 周期。

8 测量结果

8.1 测试配置

表 9. 测试配置

项目	值
适配器	Binho Supernova，硬件版本 C，固件 `FW-4.5.1-80cbb484`（4.6.0 预发布版本）
主机软件	pyCosmicSDK 1.6.0 预发布版本，`integration/1.6.0`，提交 `baacfdba`
总线	I2C 端口，3.3 V，1 kOhm 板载上拉电阻，无目标器件
通信	`set --traffic 10`，每次采集约 4800 次传输
仪器	SCL 上的 Acute TravelBus 逻辑分析仪，800 MS/s，阈值 1.6 V
采集	每个测量点 3 次，断电重启后每个测量点再采集 1 次；每次约 15000 个 SCL 周期

8.2 结果

中位值。比例列为一次采集中 tHIGH 与周期之比的中位数；p1 至 p99 列为其分布范围。每个测量点的四次采集结果相差在 0.1% 以内。

表 10. 在总线上测得的 SCL 时序

请求	比例	频率	高电平	tHIGH	tLOW	高电平，p1 至 p99
100 kHz	默认	100.0 kHz	46.4 %	4645 ns	5360 ns	46.1 至 46.9 %
100 kHz	30	100.0 kHz	29.7 %	2975 ns	7030 ns	29.4 至 30.1 %
100 kHz	40	100.0 kHz	39.9 %	3990 ns	6015 ns	39.2 至 40.1 %
100 kHz	50	100.0 kHz	49.6 %	4965 ns	5035 ns	49.2 至 50.2 %
400 kHz	默认	400.0 kHz	39.6 %	990 ns	1510 ns	38.6 至 41.1 %
400 kHz	30	400.0 kHz	28.0 %	700 ns	1800 ns	27.2 至 30.7 %
400 kHz	40	400.0 kHz	39.5 %	990 ns	1515 ns	38.8 至 40.7 %
400 kHz	50	400.0 kHz	48.8 %	1220 ns	1280 ns	48.2 至 50.7 %

在 400 kHz 下，测得的高电平占比比设定比例低 0.5 至 2 个百分点。这是上升沿造成的：分析仪只有在 SCL 越过 1.6 V 后才将其视为高电平，而在 400 kHz 下这一延迟占周期的比例更大。

另外两个测量点是在固件验证期间、在同一更改的早期版本上测得的，并非使用上述配置：

- 1 MHz，470 Ohm 上拉电阻，Supernova：比例 30、40 和 50 时为 1000.0 kHz，高电平占比分别为 29.5、37.0 和 50.5%。

- 550 kHz：在两种适配器以及 4.5.1 正式版上，默认时序均运行在 600.6 kHz，高电平占比 50.2%。比例 50 时运行在 551.0 kHz。

20、60、70 和 80% 比例通过适配器的报告值（第 6 节第 2 步）进行了检查，未在总线上测量。

9 故障排除

表 11. 故障排除

现象	可能原因及措施
`FW_UNSUPPORTED_COMMAND`	固件 4.5.1 或更早版本。将适配器更新到 4.6.0 或更高版本。
`FW_INVALID_PARAMETER`	比例不是 0，也不是 20 至 80 之间步进为 10 的值。之前的比例仍然生效。
`FW_INTERFACE_NOT_INITIALIZED`	I2C 总线未作为控制器启动。请先调用 `bring_up`。
在 1 MHz 下测得的频率低于请求值	上升沿过慢。使用 470 Ohm 或更强的上拉电阻（第 3.3 节）。
脚本重新启动后比例恢复为默认	`initialize` 会恢复默认时序。每次执行会初始化总线的 `bring_up` 之后都要调用 `set_duty_cycle`。
`table` 引发 `CapabilityError`	`configure` 需要硬件版本 C 的 Supernova。请改为每个频率使用一次 `set`。

10 缩略语

表 12. 缩略语

术语	含义
I2C	Inter-Integrated Circuit（集成电路间总线）
SCL	串行时钟线
SDA	串行数据线
tHIGH	SCL 时钟的高电平时间
tLOW	SCL 时钟的低电平时间
NACK	非应答

11 关于文档中源代码的说明

本文档中展示以及随附压缩包中提供的示例代码仅供评估和参考。这些代码以 MIT 许可证发布，许可证文本包含在压缩包中。

压缩包 `AN0015-assets.zip` 包含 `i2c_duty_cycle.py` 和 `LICENSE`。

12 修订历史

表 13. 修订历史

版本	日期	说明
0.1	25 September 2026	供内部评审的草稿，在固件 4.6.0 和 pyCosmicSDK 1.6.0 预发布版本上测量。
1.0	6 November 2026	正式发布。

法律信息（英文）

Definitions

Draft

A document marked draft is under internal review and subject to formal approval. Binho Inc. makes no representation or warranty as to its accuracy or completeness and accepts no liability arising from its use.

Disclaimers

Limited warranty and liability

Information in this document is believed to be accurate and reliable. Binho Inc. makes no representation or warranty, express or implied, as to its accuracy or completeness, accepts no liability for the consequences of its use, and takes no responsibility for content originating outside Binho Inc.

In no event shall Binho Inc. be liable for any indirect, incidental, punitive, special or consequential damages, including without limitation lost profits, lost savings, business interruption, or costs of removing or replacing any product, whether or not based on tort, warranty, breach of contract or any other legal theory. Notwithstanding any damages a customer might incur, the aggregate and cumulative liability of Binho Inc. is limited in accordance with its terms and conditions of commercial sale.

Right to make changes

Binho Inc. reserves the right to change the information in this document, including specifications and product descriptions, at any time and without notice. This document supersedes all information supplied prior to its publication.

Suitability for use

Binho Inc. products are not designed, authorized or warranted for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure of a Binho Inc. product can reasonably be expected to result in personal injury, death, or severe property or environmental damage. Any such inclusion or use is at the customer's own risk and Binho Inc. accepts no liability for it.

Applications

Applications described here are illustrative only. Binho Inc. makes no warranty that they are suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their own applications and products, for appropriate design and operating safeguards, and for determining whether a Binho Inc. product is suitable and fit for their purpose. Binho Inc. accepts no liability for assistance with applications or customer product design.

Third-party products and specifications

This document refers to products, boards, specifications and documentation supplied by third parties, which remain the responsibility of their respective owners. Binho Inc. makes no warranty regarding them and their behavior may change without notice. Register maps, protocol details and device identifiers reproduced here were observed on the specific hardware and firmware versions stated in this document and must be confirmed against the vendor's own documentation for the reader's part.

Example code

Source code supplied with this document is provided as an example, for evaluation and reference. It is not qualified for production use, has not been developed under a functional safety or security process, and is licensed under the terms accompanying it in the distribution archive.

Terms and conditions of commercial sale

Binho Inc. products are sold subject to the general terms and conditions of commercial sale published by Binho Inc., unless otherwise agreed in a valid written individual agreement, in which case only the terms of that agreement apply.

Export control

This document and the items it describes may be subject to export control regulations. Export may require prior authorization from the competent authorities.

Security

All products may be subject to unidentified vulnerabilities or may support security standards with known limitations. The customer is responsible for the design and operation of its applications and products throughout their lifecycle to reduce the effect of such vulnerabilities, and for compliance with all legal, regulatory and security requirements concerning its products.

Trademarks

All referenced brands, product names, service names and trademarks are the property of their respective owners. **Binho** and the Binho logo are trademarks of Binho Inc. **Acute** 和 **TravelBus** 是 Acute Technology Inc. 的商标或注册商标。