

GXCAS TECHNOLOGY

1. 概述

GX21M15 是一款温度-数字转换器，使用片上带隙温度传感器和 Σ - Δ A/D 转换技术并带有过热检测输出。GX21M15 包含一系列数据寄存器：储存诸如器件工作模式、OS 工作模式、OS 极性和 OS 故障队列等器件设置的配置寄存器(Conf)；具体描述见第 7 节“功能描述”。储存数字温度数据的温度寄存器(Temp)以及储存可编程过热关断门限和滞后极限的设置点寄存器(Tos 和 Thyst)，可通过 2 线串行 I²C 总线接口与控制器进行通信。该器件还提供开漏输出(OS)，可在温度超过编程门限限制值时激活。该温度传感器共有 3 个逻辑地址引脚，因此可在同一总线上最多连接 8 个器件而不会发生地址冲突。

GX21M15 可配置为不同的工作模式。可将其设为正常模式以定期监视环境温度，或设为关断模式以使功耗最小化。OS 输出可在两种可选模式中的任意一种模式下工作：OS 比较器模式或 OS 中断模式。其输出有效状态可配置为高电平或低电平。出现连续故障数以激活 OS 输出的故障队列以及设置点极限阈值均为可编程。

温度寄存器始终存储 11 位二进制补码数据，温度分辨率为 0.125°C。对于精确测量热漂移或热逃逸的应用，这一较高的温度分辨率尤其有用。当访问 GX21M15 时，不会中断当前正在进行的温度转换(即，I²C 总线部分完全独立于 Σ - Δ 转换器部分)且连续访问 GX21M15 而无需等待通信时间，即使是一个温度转换时间，但这不会妨碍器件使用新的温度转换结果去更新温度寄存器。温度寄存器更新后，可立即使用新的转换结果。

GX21M15 上电时处于正常工作模式，即 OS 处于比较器模式、温度阈值为 80°C 且滞后值为 75°C，因此可用作带有那些预定义温度设置点的独立恒温器。

2. 特征

- 引脚完全兼容行业标准 LM75 和 LM75A 的管脚，并提供 0.125°C 的温度分辨率以及 2.8 V 至 5.5 V 的电源供电范围
- I²C 总线接口，同一总线上最多带 8 个器件
- 电源电压范围：2.8 V 至 5.5 V
- 温度范围为 -55°C 至 +125°C
- 频率范围为 20 Hz 至 400 kHz，具有总线故障超时功能，可防止总线挂起
- 提供 0.125 °C 温度分辨率的 11 位 ADC
- 温度精度：
 - ◆ 0°C 至 +50 °C：±0.5°C
 - ◆ -25°C 至 +100°C：±1°C
 - ◆ -55°C 至 +125 °C：±1.5°C
- 可编程的报警温度阈值和回滞值

- 关断模式下的供电电流为 0.1μA，以实现节能
- 上电时可单独作为恒温器使用
- ESD 保护超过 JESD22-A114 4500 V HBM 和 JESD22-C101 1000 V CDM
- 根据 JEDEC 标准 JESD78 完成超过 100 mA 的闩锁测试
- 小型 8 引脚封装类型：SOP8, MSOP8/TSSOP8

3. 应用

- 系统热管理
- 个人计算机
- 电子设备
- 工业控制器

4. 订货信息

表 1. 订货信息

产品型号	顶面标识	封装		
		名称	描述	版本
GX21M15	GX21M15	SOP8	塑料小尺寸外形封装：8 引线	V1.0
GX21M15U	GX21M15U	MSOP8	塑料薄收缩小尺寸外形封装：8 引线	V1.0

5. 结构图

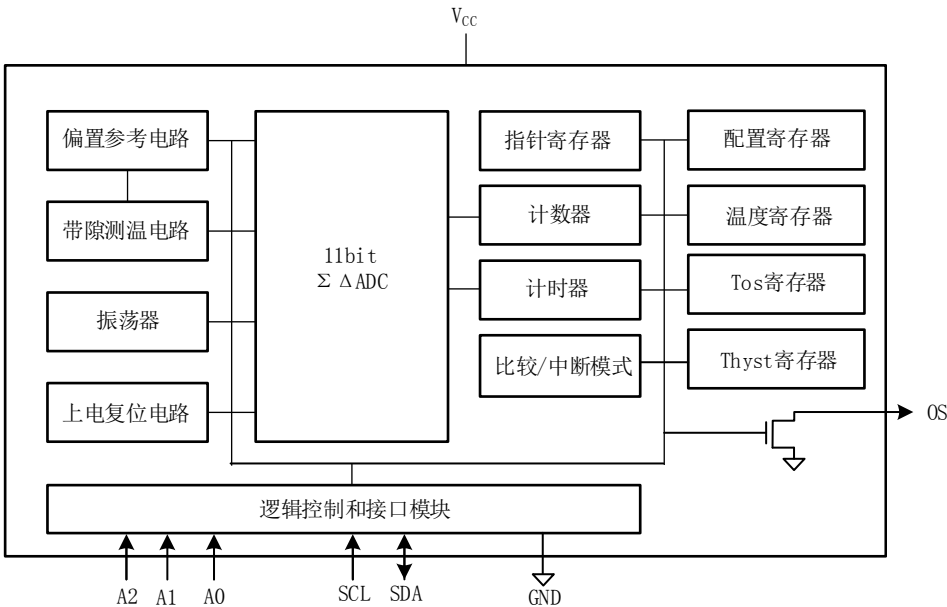
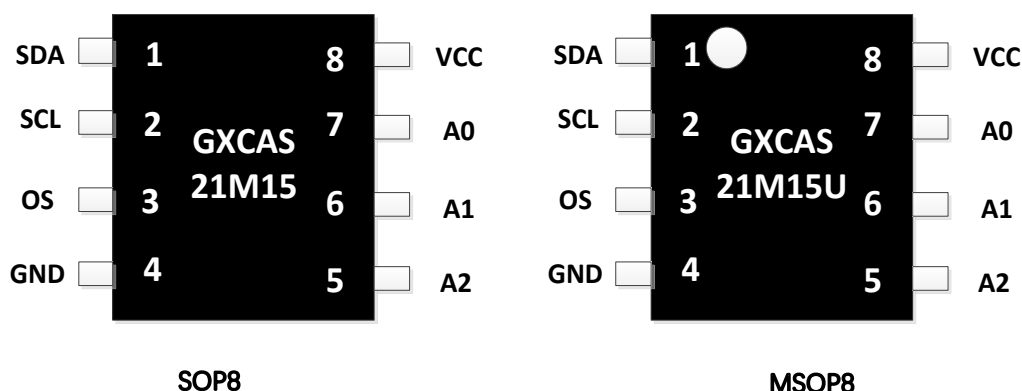


图 1. 芯片结构图

6. 引脚信息

6.1 引脚分布



6.2 引脚描述

表 2. 引脚描述

引脚名称	引脚	描述
SDA	1	数字 I/O, I ² C 总线双向数据线; 开漏输出。
SCL	2	数字输入, I ² C 总线时钟输入。
OS	3	过温保护/中断输出; 开漏。
GND	4	地, 连接至系统地
A2	5	数字输入, 用户配置地址位 2。
A1	6	数字输入, 用户配置地址位 1。
A0	7	数字输入, 用户配置地址位 0。
V _{CC}	8	供电端口

7. 功能描述

7.1 工作描述

GX21M15 使用片上带隙温度传感器以 0.125°C 的分辨率测量设备温度, 并将 11 位 A-D 转换产生的 11 位 2 进制数存储到设备温度寄存器中。该温度寄存器可由 I²C 总线上的控制器随时读取。读取温度数据不会影响读取操作期间正在进行的温度转换。

设备可以设置为在正常或睡眠模式下运行。在正常工作模式下, 温度每 100ms 执行转换一次, 并且在每次转换结束时更新温度寄存器。在大约 100ms 的转换周期 (T_{conv}) 中, 设备只需工作 10ms 完成温度到数据的转换, 称为“温度转换时间” ($t_{conv(T)}$), 剩余时间设备为空闲模式。该功能可显著降低设备功耗。在睡眠模式下, 设备为空闲, 数据转换被禁用, 温度寄存器保存最新结果; 但设备 I²C 总线接口仍然有效, 可以执行寄存器写入/读取操作。通过对配置寄存器的 B0 位进行编程, 可以控制设备的操作模式。当设备上电或从睡眠模式恢复到正常模式时, 温度转换启动。

此外, 在正常模式下的每次转换结束时, 温度寄存器中的温度数据将自动与过温关断寄存器 Tos 中的数据 (或 $T_{th(ots)}$) 以及存储在滞后寄存器 Thyst 中的数据 (T_{hyst}) 进行比较, 以便相应地确定设备的输出状态。设备的 Tos 和 Thyst 寄存器具有读/写能力, 它们

都使用 9 位 2 进制数字数据进行操作。为了与这个 9 位运算相匹配，温度寄存器只使用其 11 位数据中的 9 位 MSB 位进行比较。

OS 端口的输出响应的比较操作方式由配置寄存器的 B1 位决定，同时用户定义的故障队列由配置位 B3 和 B4 决定。

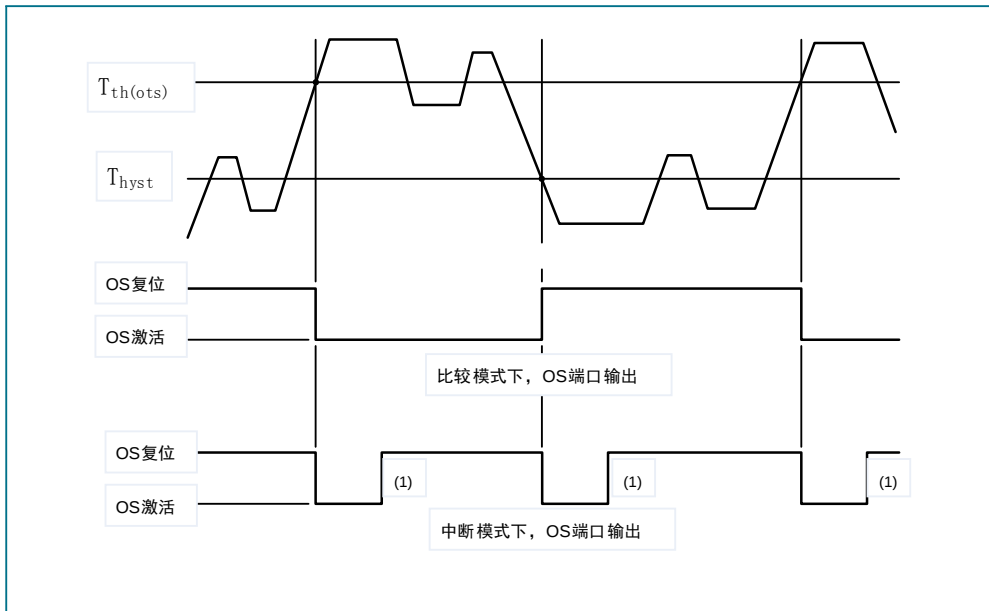
在比较器模式下，OS 端口的输出行为类似于恒温器。当温度超过 $T_{th(ots)}$ 时激活，当温度降至 T_{hyst} 以下时复位。读取寄存器或将芯片置于睡眠模式不会更改芯片的输出状态。在该模式下，操作系统的输出可以用来控制冷却风扇或热开关。

在中断模式下，OS 端口的输出行为用于热中断。当设备通电时，只有当温度超过 $T_{th(ots)}$ 时才首先激活；然后它将无限期地保持该状态，直到通过读取任意寄存器来重置。一旦通过交叉 $T_{th(ots)}$ 激活 OS 输出，然后复位，只有当温度降至 T_{hyst} 以下时，它才能再次激活；然后，它将无限期地保持激活状态，直到通过读取任意寄存器进行重置。操作系统中断操作将按以下顺序继续： $T_{th(ots)}$ trip、Reset、 T_{hyst} trip、Reset、 $T_{th(ots)}$ trip、Reset、 T_{hyst} trip、Reset 等。通过设置配置寄存器的位 0 使设备进入睡眠模式，也会重置操作 OS 输出。

在这两种情况下，即比较器模式和中断模式，只有在满足由故障队列定义的多个连续故障时，OS 输出才会被激活。故障队列可编程并存储在配置寄存器的两个位 B3 和 B4 中。此外，通过相应地设置配置寄存器位 B2，OS 输出有效状态可选择为高或低。

芯片上电时，设备进入正常模式， $T_{th(ots)}$ 为 80°C ， T_{hyst} 为 75°C ，OS 激活状态为低，故障队列等于 1。在上电后大约 100ms 内完成第一次转换，此次转换完成之前温度读数数据不可用。

OS 端口对温度的响应如图 6 所示



(1) OS 通过读取寄存器或将芯片配置成睡眠模式来重置。假设故障队列在每个 $T_{th(ots)}$ 和 T_{hyst} 交叉点处满足。

图 6. OS 端口输出图

7.2 I²C 总线接口

GX21M15 可以连接到兼容的 2 线串行接口 I²C 总线上，作为控制器或主设备控制下的从设备，使用两个设备终端 SCL 和 SDA 通信。必须通过终端提供从终端 SDA/SDA 向控制器写入数据的信号。注意，如果 I²C 总线上的上拉电阻器没有按照 I²C 总线的要求安装，那么这两个端口中的每一个都需要一个大约 10K 的外部上拉电阻。总线通信协议见第 7.10 节。

7.2.1 总线错误超时

如果 SDA 保持低电平的时间超过 t_{to} （最小 75ms/13.3Hz；保证在最小 50ms/20Hz），GX21M15 将重置为空闲状态（SDA 释放），并等待新的启动条件。这确保 GX21M15 不会在传输发生冲突时挂断总线。

7.3 从机地址

I²C 总线上的 GX21M15 从机地址由引脚 A2、A1 和 A0 的逻辑决定。它们中的每一个管脚通常连接到 GND（逻辑 0），或 V_{CC} （逻辑 1）。这些引脚代表设备 7 位地址的 3 个 LSB 位。地址数据的其他 4 个 MSB 位通过 GX21M15 内部的引线强制预设为“1001”。表 4 显示设备的完整地址，并指出最多可以将 8 个设备连接到同一总线上，而不存在地址冲突。因为输入引脚 SCL、SDA 和 A2 到 A0 没有内部偏置，所以它们在任何应用中都不应该保持浮动。

表 4. 地址列表

1 = V_{CC} ; 0 = GND.

MSB				LSB		
1	0	0	1	A2	A1	A0

7.4 寄存器列表

GX21M15 在指针寄存器旁边包含 4 个数据寄存器，如表 5 所示。表 5 还显示了寄存器通电时的指针值、读/写能力和默认内容

表 5. 寄存器列表

寄存器名称	指针值	读/写	上电默认	描述
Conf	01h	读/写	00h	配置寄存器：包含 8 位数据字节；用于设置芯片工作条件；默认值为 0。
Temp	00h	只读	n/a	温度寄存器：包含 2 个 8 位数据字节；用于存储测量的温度数据。
Tos	03h	读/写	5000h	过温关断寄存器：包含 2 个 8 位数据字节，存储 $T_{th}(ots)$ 值；默认值=80°C。
Thyst	02h	读/写	4B00h	回滞寄存器：包含 2 个 8 位数据字节；用于存储 T_{hyst} 值；默认值=75°C。

7.4.1 指针寄存器

指针寄存器包含 1 个 8 位数据，其中 2 个 LSB 位表示其他 4 个寄存器的指针值，其他 6 个 MSB 位等于 0，如表 6 和表 7 所示。用户无法访问指针寄存器，但它通过在总线命令中包含指针数据字节来选择数据寄存器进行写/读操作。

表 6. 指针寄存器

B7	B6	B5	B4	B3	B2	B[1:0]
0	0	0	0	0	0	指针值

表 7. 指针值

B1	B0	寄存器
0	0	温度寄存器 (Temp)
0	1	配置寄存器 (Conf)
1	0	回滞寄存器 (Thyst)
1	1	过温关断寄存器 (Tos)

因为执行总线命令（包括指针字节）时，指针值被锁存到指针寄存器中，所以从 GX21M15 读取的指针字节可能包括也可能不包括在语句中。要再次读取最近一次指针已预置的寄存器时，指针字节不必包括在内。要读取与前一次不同的寄存器时，必须包括指针字节。但是，对 GX21M15 的写入操作时，必须始终在语句中包含指针字节。总线通信协议见第 7.10 节。

上电时，指针值等于 00，并选择温度寄存器；然后，用户可以在不指定指针字节的情况下读取温度数据。

7.4.2 配置寄存器

配置寄存器 (Conf) 是 1 个写/读寄存器，包含 1 个 8 位非补码数据字节，这些字节用于配置芯片适应不同的工作条件。。表 8 显示了该寄存器的位分配。

表 8. 配置寄存器

Legend: * = default value.

位	符号	权限	值	描述
B[7:5]	reserved	读/写	000*	保留给厂家使用；正常操作时应保持为零
B[4:3]	OS_F_QUE[1:0]	读/写		OS 故障队列编程
			00*	队列值= 1
			01	队列值= 2
			10	队列值= 4
			11	队列值= 6
B2	OS_POL	读/写		OS 极性选择
			0*	OS 激活输出低电平
			1	OS 激活输出高电平
B1	OS_COMP_INT	读/写		OS 工作模式选择
			0*	OS 比较模式
			1	OS 中断模式
B0	SHUTDOWN	读/写		芯片工作模式选择
			0*	正常
			1	睡眠

7.4.3 温度寄存器

温度寄存器 (Temp) 在每次模数转换结束时保存温度测量或监视器的结果。此寄存器为只读，包含 2 个 8 位数据字节，由 1 个最高有效字节 (MSByte) 和 1 个最低有效字节 (LSByte) 组成。然而，这两个字节中只有 11 位用于以 2 的补码格式存储温度数据，分辨率为 0.125C°。表 9 显示了数据字节中温度数据的位排列。

Table 9. Temp register

MSByte								LSByte							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	X	X	X	X	X

当读取温度寄存器时，2 个数据字节 (MSByte 和 LSByte) 的所有 16 位都提供给总线，并且必须全部由控制器采集以完成总线操作。但是，只应使用 11 个最高有效位，LSByte 的 5 个最低有效位为零，应忽略。从 11 位温度数据计算温度值的方法之一是：

1. 如果温度数据 MSByte 位 D10=0，则温度为正且
温度值为 (C°) =+ (温度数据) ×0.125C° ；
2. 如果温度数据 MSByte 位 D10=1，则温度为负且
温度值 (C°) = (两个温度数据的补补码) ×0.125C°。

表 10 显示了温度数据和温度值的示例。

表 10. 温度寄存器值

11bit 二进制数	十六进制数	十进制数	温度值
011 1111 1000	3F8	1 016	+127.000℃
011 1111 0111	3F7	1 015	+126.875℃
011 1111 0001	3F1	1 009	+126.125 ℃
011 1110 1000	3E8	1 000	+125.000℃
000 1100 1000	0C8	200	+25.000 ℃
000 0000 0001	001	1	0.125℃
000 0000 0000	000	0	0.000℃
111 1111 1111	7FF	-1	-0.125 ℃
111 0011 1000	738	-200	-25.000℃
110 0100 1001	649	-439	-54.875℃
110 0100 1000	648	-440	-55.000℃

对于替代工业标准 XX75 的 9 位温度数据的应用，只需使用两个字节的 9 个 MSB 位，而忽略低位的 7 个 LSB。GX21M15 以分辨率为 0.5C° 的 9 位温度数据的定义方式与标准 XX75 的定义方式完全相同，并且与 Tos 和 Thyst 寄存器类似。

也可以使用单字节读取命令读取温度的 MSB。那么温度分辨率将改为 1.00C°。

7.4.4 过温关断阈值 (Tos) 和回滞 (Thyst)寄存器

这两个寄存器是写/读寄存器，也称为设定点寄存器。它们用于存储用户定义的温度限值，称为过温关断阈值 (Tth (ots)) 和滞后温度 (Thyst) ，用于设备看门狗操作。每次转换结束时，温度数据将与存储在这两个寄存器中的数据进行比较，以设置 OS 端输出的状态；见第 7.1 节。

每个设定点寄存器包含两个 8 位数据，由一个 MSByte 和一个 LSByte 组成，与温度寄存器相同。然而，两个字节中只有 9 位被用来以 2 进制形式存储设定值数据，分辨率为 0.5C°。表 11 和表 12 显示了 Tos 数据和 Thyst 数据在数据字节中的位排列。

注意，因为在设定值寄存器中只使用 9 位数据，所以设备只使用温度数据的 9 MSB 进行数据比较。

表 11. 过温关断阈值寄存器

MSByte								LSByte							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
D8	D7	D6	D5	D4	D3	D2	D1	D0	X	X	X	X	X	X	X

表 12. 回滞寄存器

MSByte								LSByte							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
D8	D7	D6	D5	D4	D3	D2	D1	D0	X	X	X	X	X	X	X

当读取设定值寄存器时，所有 16 位被提供给总线，并且必须由控制器收集以完成总线操作。但是，只应使用 9 个最高有效位，并且 LSByte 的 7 个低位等于零，应该被忽略。

表 13 显示了设定数据和值的示例。

表 13. Tos 和 Thyst 设定数据及数值

二进制数	十六进制数	十进制数	温度值
0 1111 1010	0FA	250	+125C°
0 0011 0010	32	50	+25C°
0 0000 0001	001	1	+0.5C°
0 0000 0000	000	0	0C°
1 1111 1111	1FF	-1	-0.5C°
1 1100 1110	1CE	-50	-25.0C°
1 1001 0010	192	-110	-55.0C°

7.5 OS 输出和极性

OS 端为漏极开路输出，其状态表示第 7.1 节所述的设备看门狗操作的结果。为了观察这种输出状态，需要一个外部上拉电阻器。电阻器应尽可能大，可以高达 200 kΩ 去最大限度地减小由高 OS 陷波电流产生内部加热而引起的温度读数误差。

通过对寄存器 Conf 的位 B2 (OS_POL) 进行编程，可以选择 OS 输出活动状态为高或低：将位 OS_POL 设置为逻辑 1，选择 OS 激活输出高电平，将位 B2 设置为逻辑 0，设置 OS 激活输出低电平。通电时，位 OS_POL 等于逻辑 0，OS 激活输出为低电平。

7.6 OS 比较和中断模式

如第 7.1 节所述，OS 输出响应为温度寄存器数据与 Tos 和 Thyst 寄存器中数据的比较结果，具体取决于所选的操作系统模式：OS 比较或中断模式。通过配置 Conf 寄存器的位 B1 (OS_COMP_INT) 来选择 OS 模式：将位 OS_COMP_INT 设置为逻辑 1，选择 OS 中断模式，设置为逻辑 0，选择 OS 比较模式。通电时，位 OS_COMP_INT 默认为逻辑 0，选择 OS 比较模式。

辑 0，并选择 OS 比较器。

这两种模式的主要区别在于，在比较模式下，当温度超过 $T_{th(ots)}$ 时，OS 激活，当温度降至 T_{hyst} 以下时复位，读取寄存器或将芯片睡眠模式不会改变 OS 输出的状态；在中断模式下，一旦通过超过 $T_{th(ots)}$ 或低于 T_{hyst} 被激活，OS 输出将无限期地保持激活状态，直到读取寄存器，然后 OS 输出被重置。

温度阈值 $T_{th(ots)}$ 和 T_{hyst} 必须满足 $T_{th(ots)} > T_{hyst}$ 。否则 OS 输出状态将未定义。

7.7 OS 故障队列

故障队列定义为激活 OS 输出必须连续发生的故障数。为避免因噪音而误开启。因为故障是在数据转换结束时确定的，所以故障队列也被定义为返回温度跳闸的连续转换次数。

通过编程寄存器 Conf 中的两个位 B4 和 B3 (OS_F_QUE[1:0])，可以选择故障队列的值。请注意，编程数据和故障队列值不一样。表 14 显示了它们之间的一对一关系。上电时，故障队列数据=0，故障队列值=1。

表14. 故障队列列表

故障队列数据		故障队列数值
OS_F_QUE[1]	OS_F_QUE[0]	十进制
0	0	1
0	1	2
1	0	4
1	1	6

7.8 睡眠模式

通过编程寄存器 Conf 的位 B0 (睡眠) 来选择工作模式。将 B0 设置为逻辑 1 将使设备进入睡眠模式。将 B0 位重置为逻辑 0 将使其返回正常模式。

在睡眠模式下，会消耗大约 0.1uA 的电流；温度转换停止，但 I2C 总线接口保持活动，可以执行寄存器写入/读取操作。进入睡眠模式后，芯片输出将在比较器模式下保持不变，并在中断模式下重置。

7.9 上电默认和上电复位

GX21M15 上电默认状态:

- 正常工作模式
- OS 比较模式
- $T_{th(ots)} = 80\text{ }^{\circ}\text{C}$
- $T_{hys} = 75^{\circ}\text{C}$
- OS 输出激活为低电平
- 指针寄存器值 00 (温度寄存器)

当电源电压下降到低于约 1.0V (POR) 并维持 2us 以上，然后再次上升时，芯片将重置为上电默认状态。

7.10 寄存器读写协议

主机与 GX21M15 之间的通信必须严格遵循 I2C 总线管理定义的规则。GX21M15 寄存器读/写操作的协议遵循以下定义，如图 7 至图 12 所示：

1. 在通信之前，I2C 总线必须空闲或不忙。这意味着 SCL 和 SDA 线路必须由总线上的所有设备释放，并且它们通过总线上拉电阻器拉高。
2. 主机必须提供通信所需的 SCL 时钟脉冲。数据以每 8 位数据的 9 个 SCL 时钟脉冲序列传输，随后是 1 位是 ACK。
3. 在数据传输过程中，除了启动和停止信号外，当 SCL 信号较高时，SDA 信号必须稳定。这意味着 SDA 信号只能在 SCL 线路的低持续时间内改变。
4. S：启动信号，由主机发出以启动通信，当 SCL 高时，SDA 从高变低。
5. RS：重新启动信号，与启动信号相同，用于启动写命令之后的读命令。
6. P：停止信号，由主机发出以停止通信，SDA 从低变高，而 SCL 为高。总线释放。
7. W：写入位，当写入命令中的写入/读位为 0 时。
8. R：读取位，当读取命令中的写入/读位为 1 时。
9. A：从机 ACK 位，由 GX21M15 返回。如果正常工作，则为 0；如果不正常，则为 1。主机必须在此期间释放 SDA 线路，以便让总线上的设备控制 SDA 线路。
10. A'：主机 ACK 位，不由从机返回，而是由主机或主机在读取 2 字节数据时发出。在这个时钟周期内，主机必须将 SDA 线路设置为低，以便通知设备第一个字节已被读取，以便设备将第二个字节提供到总线上。
11. NA：No ACK 位。在这个时钟周期内，设备和主机都在数据传输结束时释放 SDA 线路，然后使主机能够生成停止信号。
12. 在写入协议中，数据从主机发送到设备，主机控制 SDA 线路，但设备向总线发送设备确认信号的时钟周期除外。
13. 在读协议中，数据由设备发送到总线，主机必须在设备向总线提供数据并控制 SDA 线路的时间段释放 SDA 线路，主机向总线发送主机确认信号的时钟周期除外。

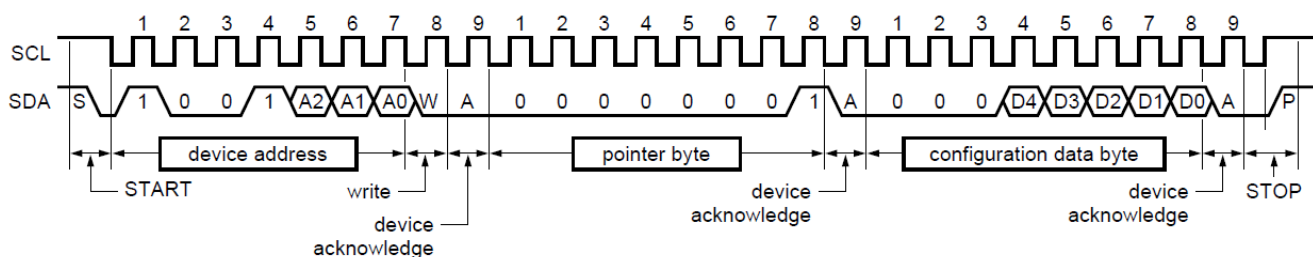


图 7.写配置寄存器 (1-byte data)

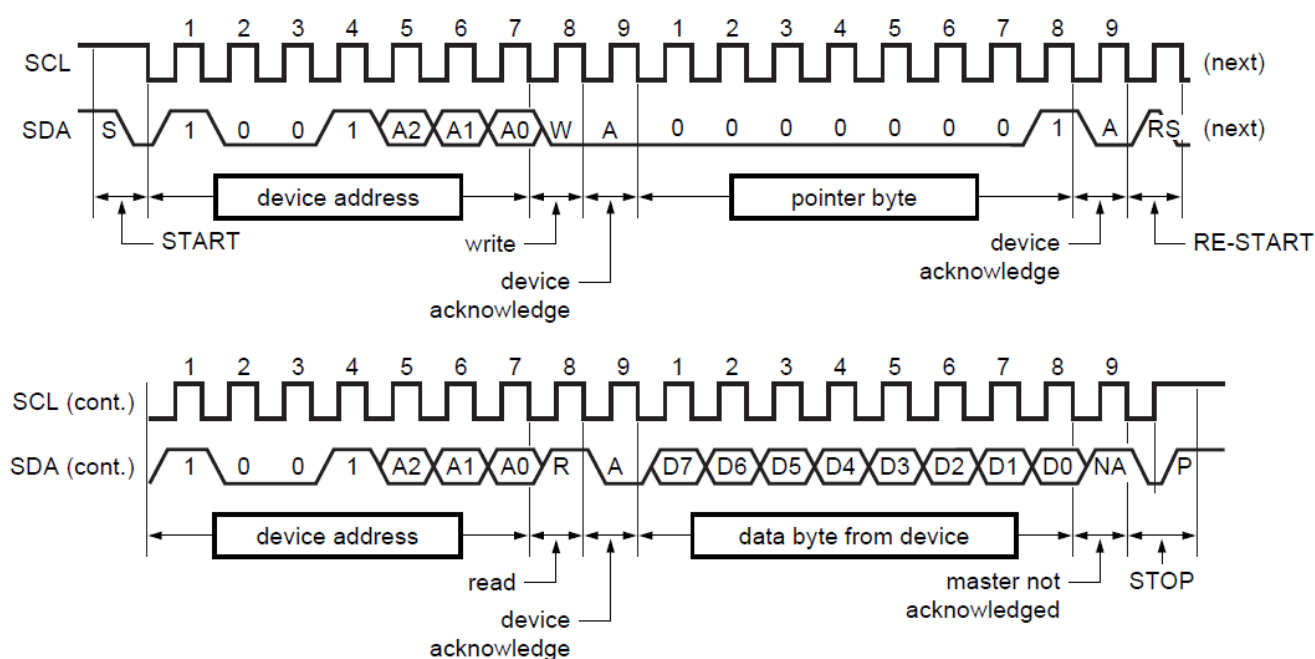


图 8.读取配置寄存器 (1-byte data)

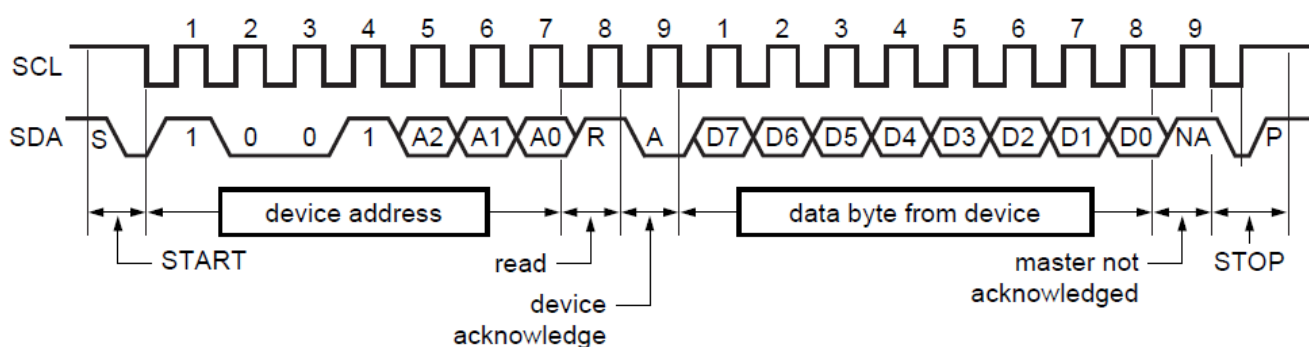


图 9.使用预设指针读取配置或温度寄存器(1-byte data)

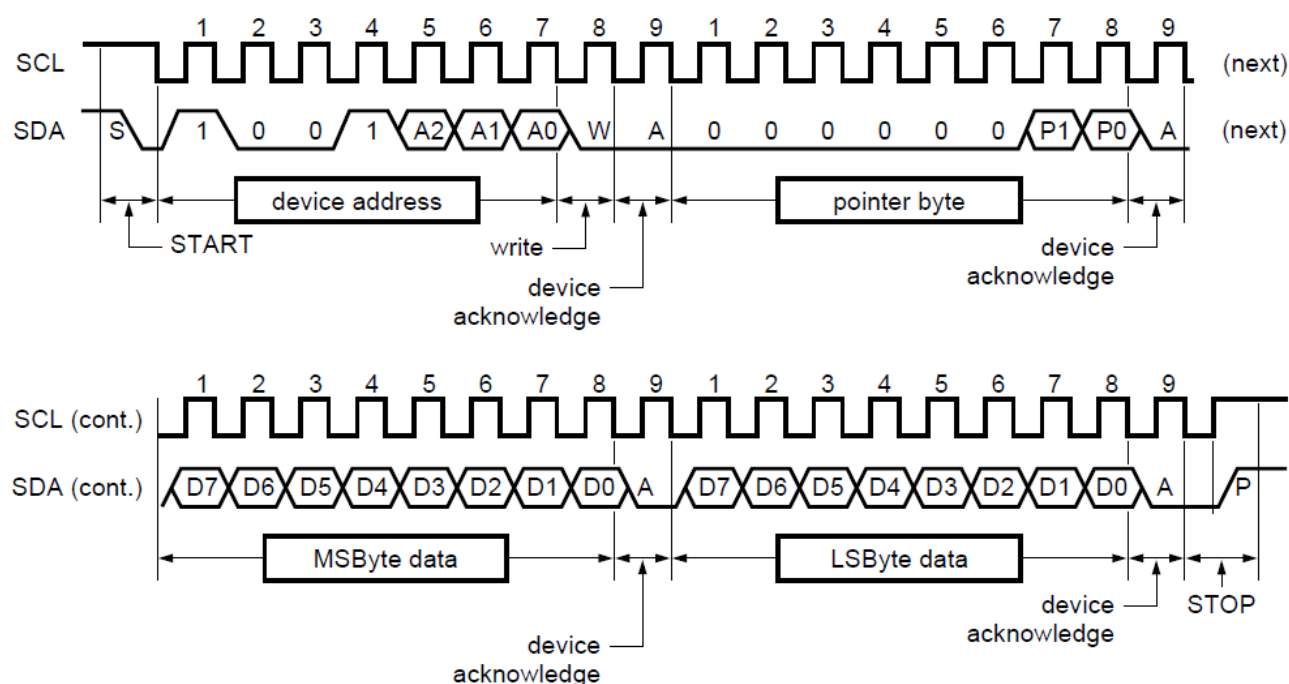


图 10. 写 Tos 或 Thyst 寄存器 (2-byte data)

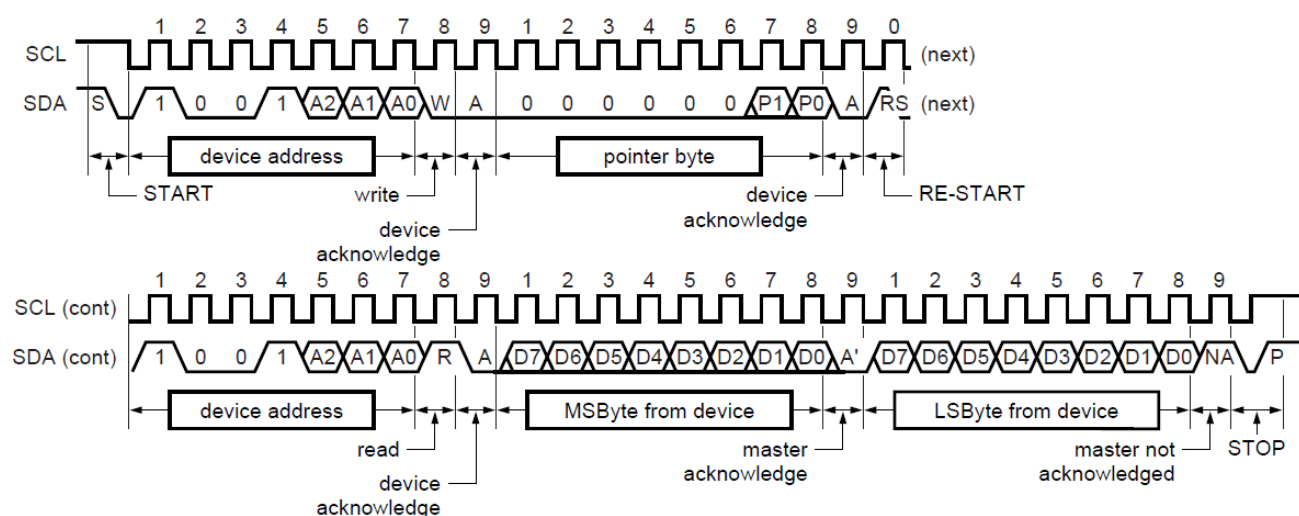


图 11. 读取温度，Tos 或 Thyst 寄存器 (2-byte data)

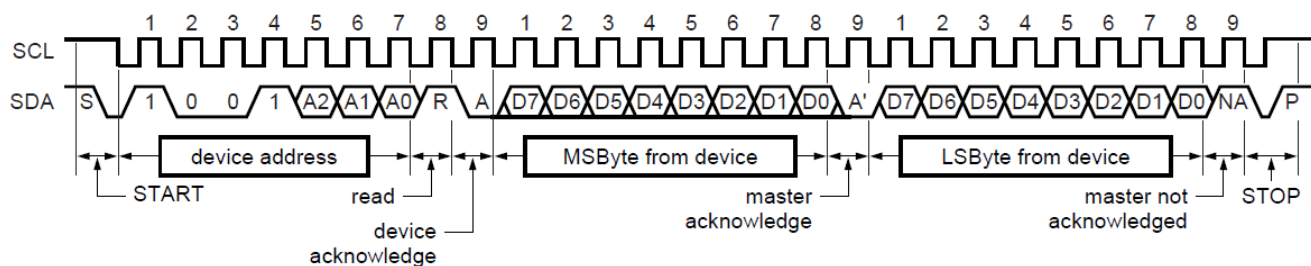


图 12. 使用预设指针读取温度，Tos 或 Thyst 寄存器 (2-byte data)

8. 应用信息

8.1 典型应用

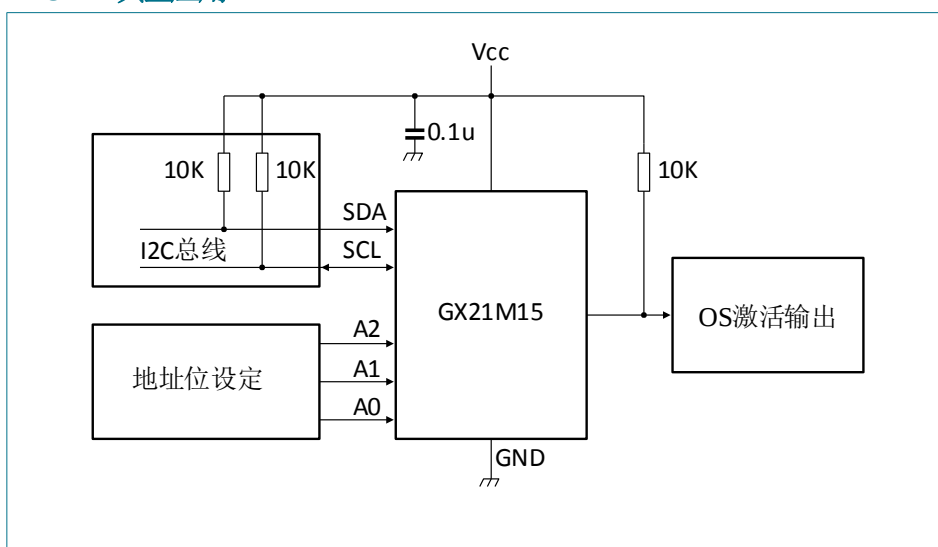


图 13. 典型应用

8.2 XX75A 、NXP LM75B 和 GX21M15 对比

表15.XX75A 和 GX21M15 对比

XX75A 、NXP LM75B 和 GX21M15 对比			
型号	LM75A	NXP LM75B	GX21M15
同一温度下，2.8-5.5V 变电压下测温误差	误差约 0.5℃	误差约 0.75℃	无误差
中断模式下，OS 端口在 shutdown 模式下自动复位	no	yes	yes
在总线未被锁定时，支持单比特温度寄存器读出	no	yes	yes
总线故障超时功能(75 ms, 200 ms)	no	yes	yes
最小数据保持时间(tHD;DAT)	10 ns	0 ns	0 ns
转换时间/ 转换周期 (typical)	100 ms / 100 ms	10 ms / 100 ms	10 ms / 100 ms
关断电流(typical value)	3.5 uA	0.2uA	0.2 uA
HBM ESD 保护等级(minimum)	>2000 V	>4000 V	>6000 V
MM ESD 保护等级(minimum)	>200 V	>450 V	>450 V
CDM ESD 保护等级(minimum)	>1000 V	>2000 V	>2000 V

8.3 温度精度

由于温度传感器的局部通道测量其自身从其本体传递的模具温度，因此必须稳定和饱和器件主体的温度，以提供稳定的读数。由于 GX21M15 工作在低功率水平，器件封装的热梯度对测量的影响很小。测量的准确性更多地取决于环境温度的定义，环境温度的定义受到不同因素的影响：安装设备的印刷电路板；与设备本体接触的气流（如果环境空气温度和印刷电路板温度相差很大，然后，由于模具和环境之间的热路径不同，测量可能不稳定）。当设备完全浸入热槽中时，热槽的恒温液体将提供最佳的温度环境。安装在温度恒定的空气中的密封端金属管内的热探针也提供了一种良好的温度测量方法。

如果要计算自加热的效果，请使用下面的方程式 1：

方程式 1 是计算自加热效果的公式：

$$\Delta T = R_{th(j-a)} \times [(V_{DD} \times I_{DD(AV)}) + (V_{OL(SDA)} \times I_{OL(sink)(SDA)}) + (V_{OL(EVENT)} \times I_{OL(sink)EVENT})] \quad (1)$$

其中：

$$\Delta T = T_j - T_{amb}$$

T_j = 结温

T_{amb} = 室温

$R_{th(j-a)}$ = 封装热阻

V_{DD} = 供电电压

$I_{DD(AV)}$ = 平均电流

$V_{OL(SDA)}$ = SDA 低输出电压

$V_{OL(EVENT)}$ = $\overline{EVENT}$ 低输出电压

$I_{OL(sink)(SDA)}$ = SDA 低输出电流

$I_{OL(sink)EVENT}$ = $\overline{EVENT}$ 低输出电流

计算举例：

T_{amb} (笔记本电脑内的典型温度) = 50 °C

$I_{DD(AV)} = 400 \mu A$

$V_{DD} = 3.6 V$

Maximum $V_{OL(SDA)} = 0.4 V$

$I_{OL(sink)(SDA)} = 1 mA$

$V_{OL(EVENT)} = 0.4 V$

$I_{OL(sink)EVENT} = 3 mA$

$R_{th(j-a)} = 56 ^\circ C/W$

由于功耗而产生的自发热：

$$\Delta T = 56 \times [(3.6 \times 0.4) + (0.4 \times 3) + (0.4 \times 1)] = 56 ^\circ C/W \times 3.04 mW = 0.17 ^\circ C \quad (2)$$

8.4 噪声影响

GX21M15 器件的设计包括实现良好的抗噪性能：

- 在 SCL 和 SDA 引脚都有低通滤波器；
- SCL 和 SDA 端口输入电压阈值回滞电压最小值约为 500mV；
- 所有引脚都有 ESD 保护电路，以防止在电涌损坏。地址、OS、SCL 和 SDA 上的 ESD 保护将其接地。在任何电源电压下，address/OS 基于门锁的器件击穿电压通常为 11V，SCL/SDA 通常为 9.5V，但会随工艺和温度的变化而变化。由于从 SCL 或 SDA 到 VCC 没有保护二极管，因此当 V_{CC} 未供电时，GX21M15 将不会保持 I2C 线路的低电平，因此，如果 GX21M15 断电，则允许 I2C 总线继续运行。

但是，当设备在非常嘈杂的环境中使用，建议采用良好的布局做法和额外的噪声滤波器：

- 在 V_{CC} 引脚使用去耦电容。
- 保持数字路径远离开关电源。
- 长距离通信使用适当的终端。
- 在 SCL 和 SDA 线路上增加电容，以提高总线上低通滤波器的特性。

9. 极限值

表 16. 极限值

按照绝对最高评级制度(IEC 60134).

符号	参数	状态	最小	最大	单位
V_{CC}	供电电压		-0.3	+6.0	V
V_I	输入电压	在输入端口	-0.3	+6.0	V
I_I	输入电流	在输入端口	-5.0	+5.0	mA
$I_{O(sink)}$	输出漏电流	OS 端口	-	10.0	mA
V_O	输出电压	OS 端口	-0.3	+6.0	V
T_{stg}	保存温度		-65	+150	°C
T_j	结温		-	150	°C

10. 推荐工作条件

表 17. 参考工作特性

符号	参数	状态	最小	典型	最大	单位
V_{CC}	供电电压		2.8	-	5.5	V
T_{amb}	室温		-55	-	+125	°C

11. 静态特性

表 18. 静态特性

$V_{CC} = 2.8 \text{ V to } 5.5 \text{ V}$; $T_{amb} = -55 \text{ }^{\circ}\text{C to } +125 \text{ }^{\circ}\text{C}$; 除非特殊说明。

符号	参数	状态	最小	典型 ^[1]	最大	单位
T_{acc}	温度精度	$T_{amb} = -25 \text{ }^{\circ}\text{C to } +100 \text{ }^{\circ}\text{C}$	-1	-	+1	$^{\circ}\text{C}$
		$T_{amb} = -55 \text{ }^{\circ}\text{C to } +125 \text{ }^{\circ}\text{C}$	-2	-	+2	$^{\circ}\text{C}$
T_{res}	温度分辨率	11-bit 数字数据	-	0.125	-	$^{\circ}\text{C}$
$t_{conv(T)}$	温度装换时间	正常模式	-	10	-	ms
T_{conv}	转换周期	正常模式	-	100	-	ms
$I_{DD(AV)}$	average supply current	正常模式: I ² C 总线未激活	-	80	200	μA
		正常模式: I ² C 总线激活; $f_{SCL} = 400 \text{ kHz}$	-	-	300	μA
		睡眠模式	-	0.2	1.0	μA
V_{IH}	输入高电平	数字引脚 (SCL, SDA, A2 to A0)	$0.7 \times V_{CC}$	-	$V_{CC} + 0.3$	V
V_{IL}	输入低电平	数字引脚	-0.3	-	$0.3 \times V_{CC}$	V
$V_{I(hys)}$	输入回滞电压	SCL 和 SDA 引脚	-	300	-	mV
		A2, A1, A0 引脚	-	150	-	mV
I_{IH}	高电压输入电流	数字引脚; $V_I = V_{CC}$	-1.0	-	+1.0	μA
I_{IL}	低电压输入电流	数字引脚; $V_I = 0 \text{ V}$	-1.0	-	+1.0	μA
V_{OL}	输出低电压	SDA 和 OS 引脚; $I_{OL} = 3 \text{ mA}$	-	-	0.4	V
		$I_{OL} = 4 \text{ mA}$	-	-	0.8	V
I_{LO}	输出漏电流	SDA 和 OS pins; $V_{OH} = V_{CC}$	-	-	10	μA
N_{fault}	故障数	可编程; 故障数转换	1	-	6	
$T_{th(ots)}$	过温关断阈值温度	默认值	-	80	-	$^{\circ}\text{C}$
T_{hys}	回滞温度点	默认值	-	75	-	$^{\circ}\text{C}$
C_i	输入电容	数字引脚	-	20	-	pF

[1] 典型值是在 $V_{CC} = 3.3 \text{ V}$ 和 $T_{amb} = 25 \text{ }^{\circ}\text{C}$ 的条件下得到。

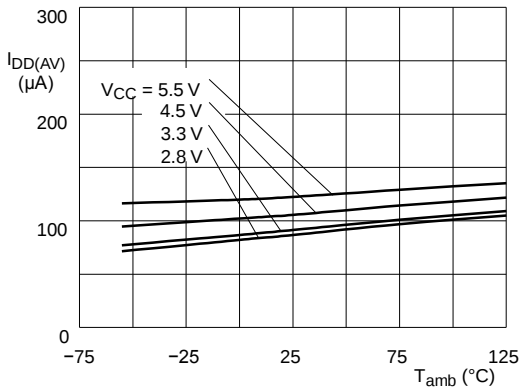


表 14. I²C 未激活时，平均电流与温度关系曲线

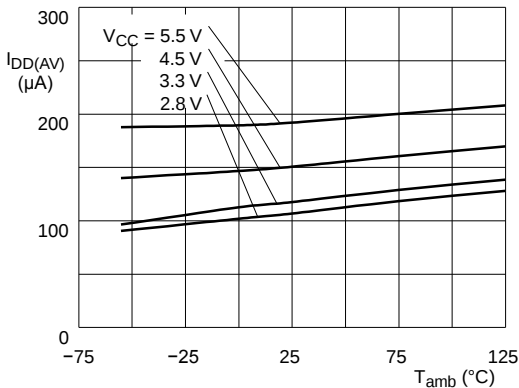


表 15. I²C 激活时，平均电流与温度关系曲线

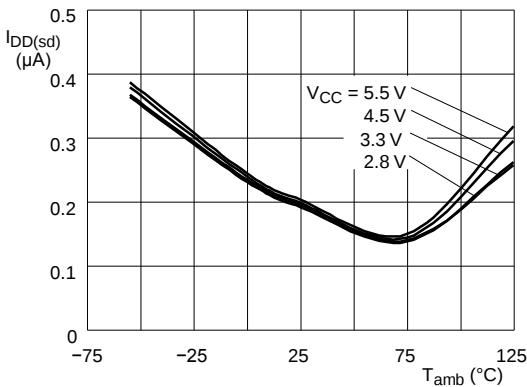


表 16. 睡眠模式下供电电流和温度关系

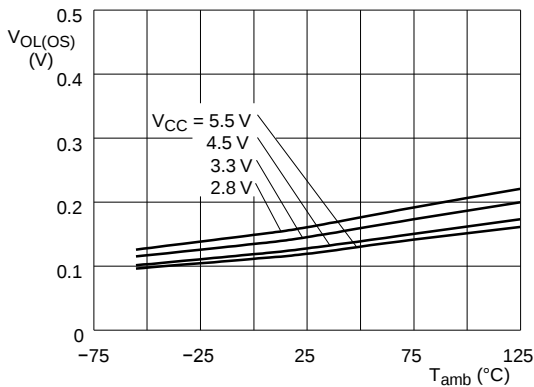


表 17. I_{OL} = 4 mA 时，OS 输出电压和温度关系

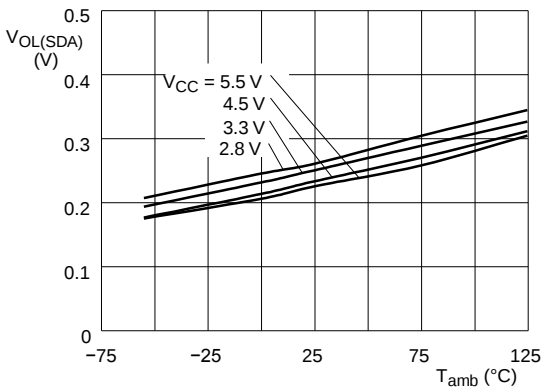


表 18. I_{OL} = 4 mA 时 SDA 电压和温度关系

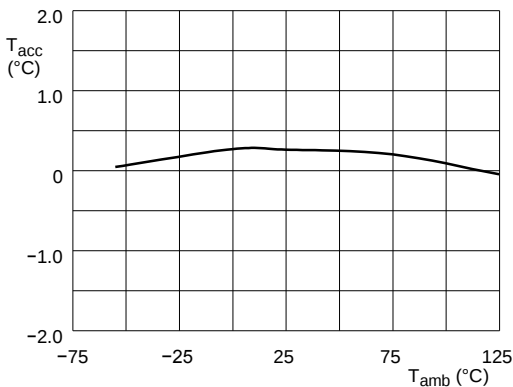


Fig 19. V_{CC} = 3.3 V 时典型测温精度

12. 动态特性

表19. I2C 总线接口动态特性^[1]

$V_{CC} = 2.8\text{ V to }5.5\text{ V}$; $T_{amb} = -55\text{ }^{\circ}\text{C to }+125\text{ }^{\circ}\text{C}$; 除非特别说明.

符号	参数	状态	最小	典型	最大	单位
f_{SCL}	SCL 时钟频率	见 图 20	0.02	-	400	kHz
t_{HIGH}	SCL 时钟高电平周期		0.6	-	-	μs
t_{LOW}	SCL 时钟低电平周期		1.3	-	-	μs
$t_{HD;STA}$	启动信号保持（重复）时间		100	-	-	ns
$t_{SU;DAT}$	数据建立时间		100	-	-	ns
$t_{HD;DAT}$	数据保持时间		0	-	-	ns
$t_{SU;STO}$	停止信号建立时间		100	-	-	ns
t_f	下降时间	SDA 和 OS 输出端口; $C_L = 400\text{ pF}$; $I_{OL} = 3\text{ mA}$	-	250	-	ns
t_{to}	超时时间		^{[2][3]} 75	-	200	ms

[1] 这些规格由设计保证，不是在生产中测试得到。

[2] 这是 SDA 串行接口复位的时间下限。

[3] 将 SDA 线路保持在低位超过 t_{to} 的时间将导致 GX21M15 将 SDA 重置为串行总线通信的空闲状态(SDA 为 1)。

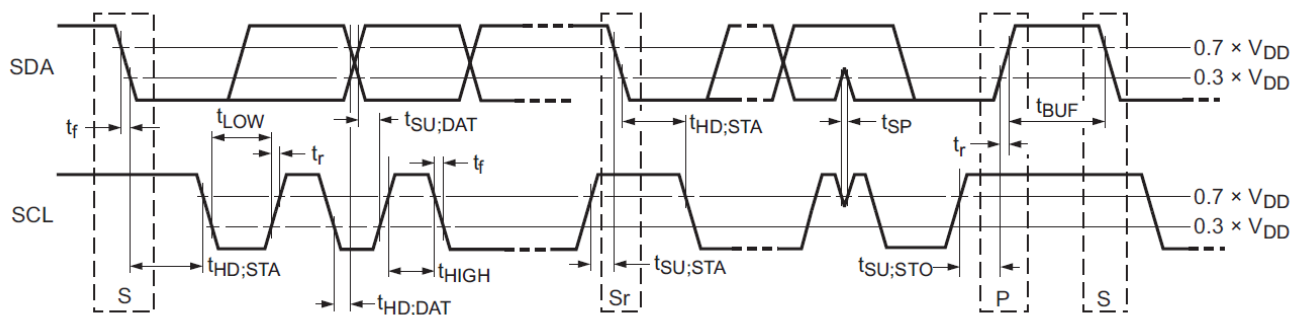
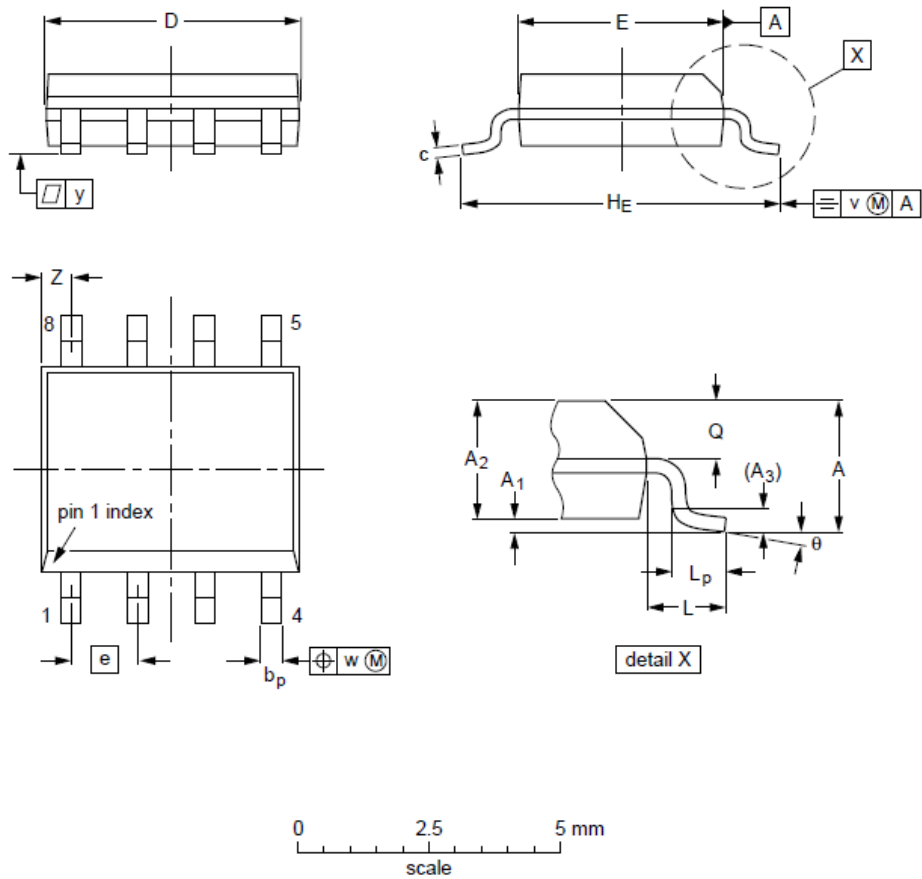


图 20. 时序图

13. 封装信息

SOP8: 塑料小尺寸外形封装：8 引线塑封体宽 3.9 mm



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

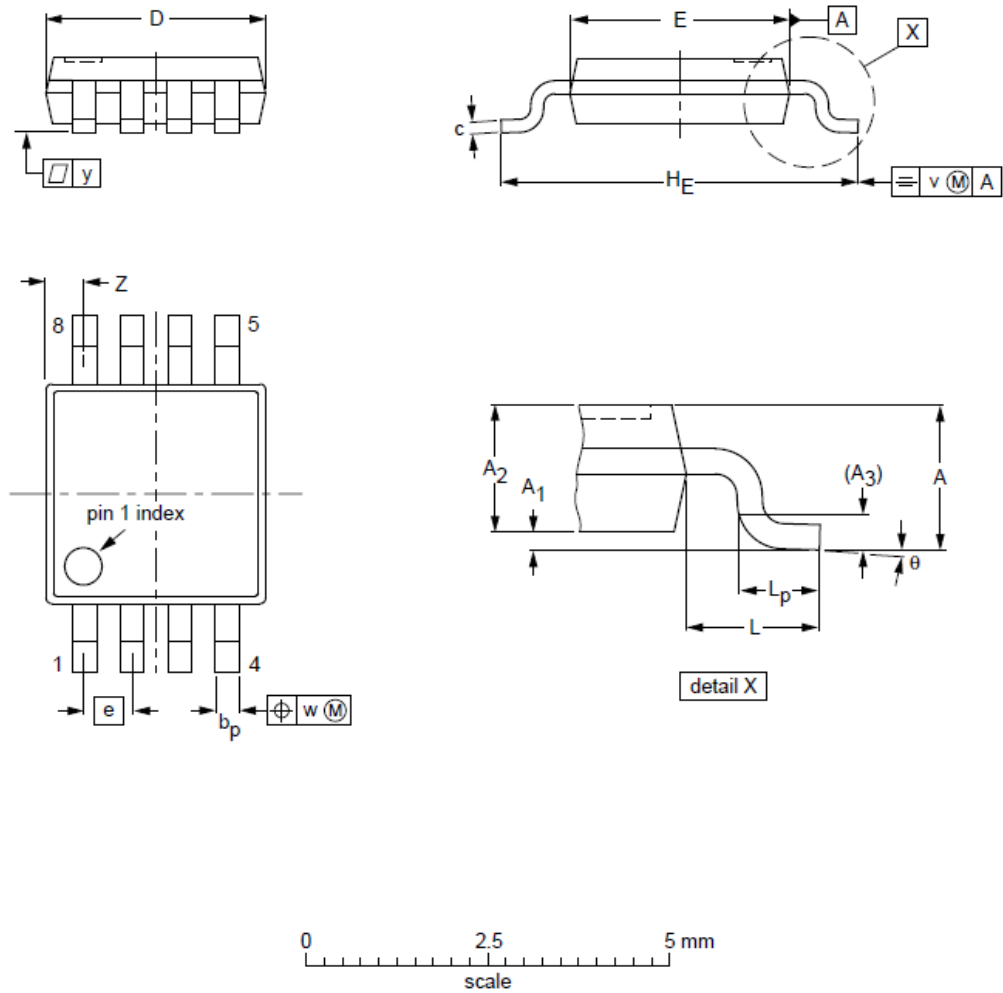
UNIT	A max.	A ₁	A ₂	A ₃	b _p	c	D ⁽¹⁾	E ⁽²⁾	e	H _E	L	L _p	Q	v	w	y	Z ⁽¹⁾	θ
mm	1.75	0.25 0.10	1.45 1.25	0.25	0.49 0.36	0.25 0.19	5.0 4.8	4.0 3.8	1.27	6.2 5.8	1.05	1.0 0.4	0.7 0.6	0.25	0.25	0.1	0.7 0.3	8° 0°
inches	0.069	0.010 0.004	0.057 0.049	0.01	0.019 0.014	0.0100 0.0075	0.20 0.19	0.16 0.15	0.05	0.244 0.228	0.041	0.039 0.016	0.028 0.024	0.01	0.01	0.004	0.028 0.012	

Notes

1. Plastic or metal protrusions of 0.15 mm (0.006 inch) maximum per side are not included.
2. Plastic or metal protrusions of 0.25 mm (0.01 inch) maximum per side are not included.

图 21. SOP8 封装尺寸图

MSOP8/TSSOP8: 塑料薄收缩小尺寸外形封装：8 引线，塑封体宽 3 mm



DIMENSIONS (mm are the original dimensions)

UNIT	A _{max.}	A ₁	A ₂	A ₃	b _p	c	D ⁽¹⁾	E ⁽²⁾	e	H _E	L	L _p	v	w	y	Z ⁽¹⁾	θ
mm	1.1	0.15 0.05	0.95 0.80	0.25	0.45 0.25	0.28 0.15	3.1 2.9	3.1 2.9	0.65	5.1 4.7	0.94	0.7 0.4	0.1	0.1	0.1	0.70 0.35	6° 0°

- Notes
- 1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.
 - 2. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

图 22. MSOP8/TSSOP8 封装尺寸图

14. 焊接: PCB 引脚尺寸图

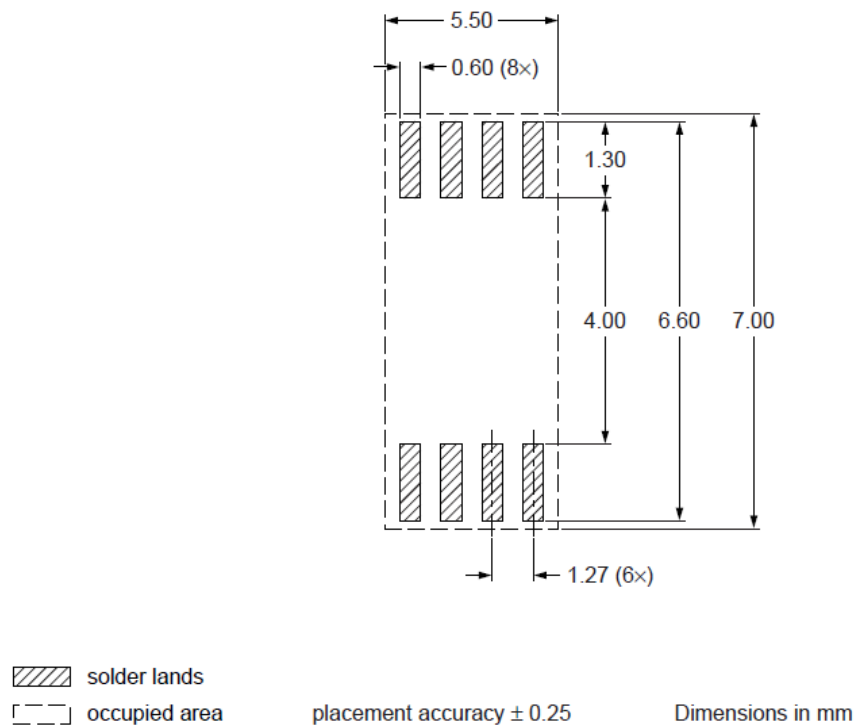


图 23. SOP8 PCB 引脚尺寸图

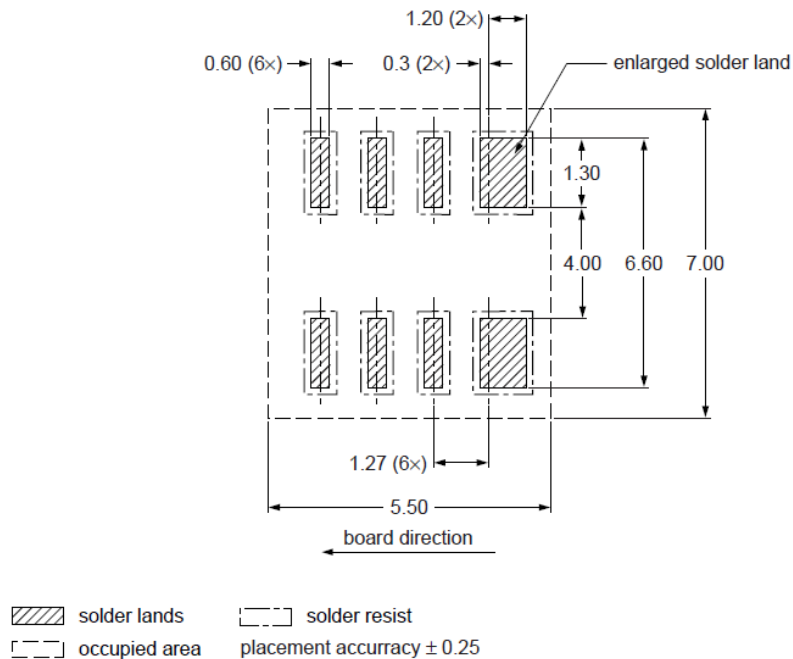


图 24. MSOP8/TSSOP8 PCB 引脚尺寸

Revision History

REVISION	REVISION	DESCRIPTION	PAGES
V1.0	04/11/2020	first release	all